

Diffusion Priors for Physics-Informed CT and PET Reconstruction

3rd International Workshop on Machine Learning and Quantum Computing Applications in Medicine and Physics, Warsaw, Poland.

Clémentine Phung-Ngoc, Antoine De Paepe, Thore Dassow, Corentin Vazia, Alexandre Bousse

9th of September, 2026

Contents

1. Introduction
2. Diffusion Models for Image Reconstruction
3. Motion-compensated Head CBCT
4. PCCT Material Decomposition
5. Joint reconstruction of activity and attenuation in CT-less PET
6. Conclusion
7. Appendix 1: diffusion models
8. Appendix 2: additional results & quantitative analysis

Introduction

Image acquisition and reconstruction

- **Forward problem**: acquisition



- **Inverse problem**: image reconstruction

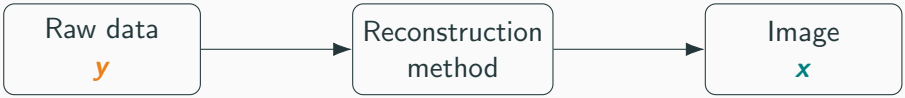
- **Challenge**: to reconstruct a high-quality image x from noisy/undersampled y.

Image acquisition and reconstruction

- **Forward problem**: acquisition



- **Inverse problem**: image reconstruction



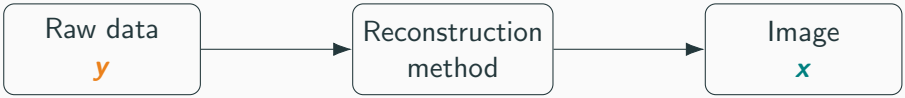
- **Challenge**: to reconstruct a high-quality image x from noisy/undersampled y.

Image acquisition and reconstruction

- **Forward problem**: acquisition

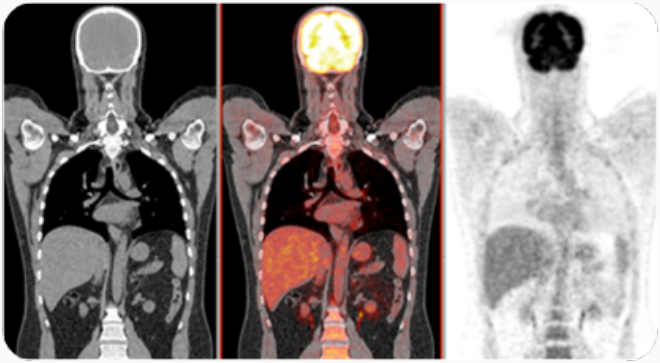


- **Inverse problem**: image reconstruction



- **Challenge**: to reconstruct a high-quality image $\boldsymbol{x}$ from noisy/undersampled $\boldsymbol{y}$.

Computed tomography (CT) and positron emission tomography (PET)



CT | PET/CT fusion | PET

- **Computed tomography (CT)**: X-ray-based imaging that shows detailed body structure.
- **Positron emission tomography (PET)**: radiotracer-based imaging that shows metabolic activity, for example to detect tumours and metastases.

CT Model

- **Image to reconstruct**: $\boldsymbol{x} = \boldsymbol{\mu}$ the X-ray attenuation map $\boldsymbol{\mu}$.
- $\boldsymbol{\mu}(E)$ depends on the the energy E of the photons.
- Measurement: $\boldsymbol{y} = [y_1, \dots, y_n]^\top$ detected photons along each X-ray beam i.
- **Expected detections**: given by the Beer–Lambert law:

$$\mathbb{E}[y_i \mid \boldsymbol{\mu}] = \int h(E) \cdot \mathrm{e}^{-[P\boldsymbol{\mu}(E)]_i} \, \mathrm{d}E + \underbrace{s_i}_{\text{scatter}}$$

  where $[\boldsymbol{P}\boldsymbol{x}]_i = \int_{L_i} x(\boldsymbol{r}) \, \mathrm{d}\boldsymbol{r}$ and $h(E)$ is the X-ray source spectrum.
- To reduce the ill-posedness, the model is assumed **monochromatic**:

$$\mathbb{E}[y_i \mid \bar{\boldsymbol{\mu}}] \approx I \cdot \mathrm{e}^{-[P\bar{\boldsymbol{\mu}}]_i} + s_i$$

  where $\bar{\boldsymbol{\mu}} \approx \int \boldsymbol{\mu}(E) h(E) \mathrm{d}E$ and $I = \int E h(E) \mathrm{d}E$.
- **Approximate noise model** for energy integrating detectors (EIDs):

$$(y_i \mid \bar{\boldsymbol{\mu}}) \sim \text{Poisson}(\mathcal{A}_i(\bar{\boldsymbol{\mu}})) + \underbrace{\mathcal{N}\left(0, \sigma^2\right)}_{\text{electronic noise}}, \quad \mathcal{A}_i(\bar{\boldsymbol{\mu}}) \triangleq \mathbb{E}[y_i \mid \bar{\boldsymbol{\mu}}]$$



CT Model

- **Image to reconstruct**: $\boldsymbol{x} = \boldsymbol{\mu}$ the X-ray attenuation map $\boldsymbol{\mu}$.
- $\boldsymbol{\mu}(E)$ depends on the the energy E of the photons.
- Measurement: $\boldsymbol{y} = [y_1, \dots, y_n]^\top$ detected photons along each X-ray beam i.
- **Expected detections**: given by the Beer–Lambert law:

$$\mathbb{E}[y_i \mid \boldsymbol{\mu}] = \int h(E) \cdot \mathrm{e}^{-[\boldsymbol{P}\boldsymbol{\mu}(E)]_i} \, \mathrm{d}E + \underbrace{s_i}_{\text{scatter}}$$

where $[\boldsymbol{P}\boldsymbol{x}]_i = \int_{L_i} x(\boldsymbol{r}) \, \mathrm{d}\boldsymbol{r}$ and $h(E)$ is the X-ray source spectrum.

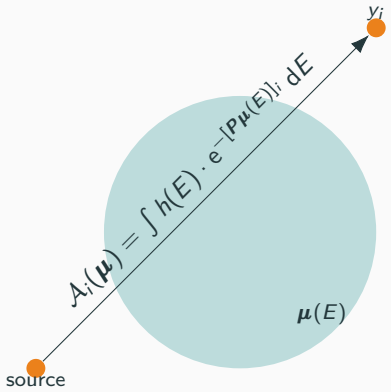
- To reduce the ill-posedness, the model is assumed **monochromatic**:

$$\mathbb{E}[y_i \mid \bar{\boldsymbol{\mu}}] \approx I \cdot \mathrm{e}^{-[P\bar{\mu}]_i} + s_i$$

where $\bar{\boldsymbol{\mu}} \approx \int \boldsymbol{\mu}(E) h(E) \mathrm{d}E$ and $I = \int E h(E) \mathrm{d}E$.

- **Approximate noise model** for energy integrating detectors (EIDs):

$$(y_i \mid \bar{\boldsymbol{\mu}}) \sim \text{Poisson}(\mathcal{A}_i(\bar{\boldsymbol{\mu}})) + \underbrace{\mathcal{N}\left(0, \sigma^2\right)}_{\text{electronic noise}}, \quad \mathcal{A}_i(\bar{\boldsymbol{\mu}}) \triangleq \mathbb{E}[y_i \mid \bar{\boldsymbol{\mu}}]$$



CT Model

- **Image to reconstruct**: $\boldsymbol{x} = \boldsymbol{\mu}$ the X-ray attenuation map $\boldsymbol{\mu}$.
- $\boldsymbol{\mu}(E)$ depends on the the energy E of the photons.
- Measurement: $\boldsymbol{y} = [y_1, \dots, y_n]^\top$ detected photons along each X-ray beam i.
- **Expected detections**: given by the Beer–Lambert law:

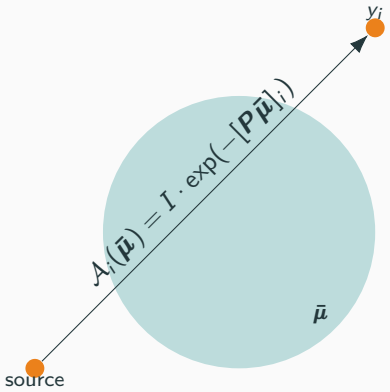
$$\mathbb{E}[y_i \mid \boldsymbol{\mu}] = \int h(E) \cdot \mathrm{e}^{-[\boldsymbol{P}\boldsymbol{\mu}(E)]_i} \, \mathrm{d}E + \underbrace{s_i}_{\text{scatter}}$$

  where $[\boldsymbol{P}\boldsymbol{x}]_i = \int_{L_i} x(\boldsymbol{r}) \, \mathrm{d}\boldsymbol{r}$ and $h(E)$ is the X-ray source spectrum.
- To reduce the ill-posedness, the model is assumed **monochromatic**:

$$\mathbb{E}[y_i \mid \bar{\boldsymbol{\mu}}] \approx I \cdot \mathrm{e}^{-[\boldsymbol{P}\bar{\boldsymbol{\mu}}]_i} + s_i$$

  where $\bar{\boldsymbol{\mu}} \approx \int \boldsymbol{\mu}(E) h(E) \mathrm{d}E$ and $I = \int E h(E) \mathrm{d}E$.
- **Approximate noise model** for energy integrating detectors (EIDs):

$$(y_i \mid \bar{\boldsymbol{\mu}}) \sim \text{Poisson}(\mathcal{A}_i(\bar{\boldsymbol{\mu}})) + \underbrace{\mathcal{N}\left(0, \sigma^2\right)}_{\text{electronic noise}}, \quad \mathcal{A}_i(\bar{\boldsymbol{\mu}}) \triangleq \mathbb{E}[y_i \mid \bar{\boldsymbol{\mu}}]$$



Photon-counting computed tomography (PCCT)

- **Photon-counting detectors (PCDs)**: capable of counting the photons and of measuring their energy.
- The detections are regrouped into energy bins, $[E_{k-1}, E_k[,\ k = 1, \dots, K$.
- **Expected counts at beam i, energy bin k**:

$$\mathbb{E}[y_{i,k} \mid \boldsymbol{\mu}] = \int h_k(E) \cdot \mathrm{e}^{-[\boldsymbol{P}\boldsymbol{\mu}(E)]_i} \, \mathrm{d}E + s_{i,k}$$

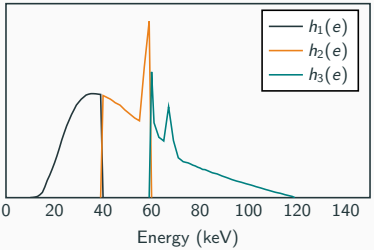
  Can be approximated to reconstructed multiple attenuation images $\bar{\boldsymbol{\mu}}_k$.

- **Exact noise model**:

$$(y_{i,k} \mid \boldsymbol{\mu}) \sim \text{Poisson}(\mathcal{A}_{i,k}(\boldsymbol{\mu})) \quad , \ \mathcal{A}_{i,k}(\boldsymbol{\mu}) \triangleq \mathbb{E}[y_{i,k}|\boldsymbol{\mu}]$$

  → **less noise in reconstructed images**

- PCDs smaller than EIDs → **higher resolution**.



- **Material decomposition**: material density images

$$\boldsymbol{x} = \{\boldsymbol{x}_1, \dots, \boldsymbol{x}_{n_\mathrm{m}}\}$$

- Example: $\boldsymbol{x}_1$: bones; $\boldsymbol{x}_2$: soft tissues; $\boldsymbol{x}_3$: iodine.
- Integration in the forward model through $\boldsymbol{\mu}$:

$$\boldsymbol{\mu}(E) = \sum_{m=1}^{n_\mathrm{m}} \mu_m^{\mathrm{mat}}(E)\boldsymbol{x}_m \triangleq \boldsymbol{\mathcal{F}}(\boldsymbol{x}, E)$$

Photon-counting computed tomography (PCCT)

- **Photon-counting detectors (PCDs)**: capable of counting the photons and of measuring their energy.
- The detections are regrouped into energy bins, $[E_{k-1}, E_k[,\ k = 1, \dots, K$.
- **Expected counts at beam i, energy bin k**:

$$\mathbb{E}[y_{i,k} \mid \boldsymbol{\mu}] = \int h_k(E) \cdot \mathrm{e}^{-[\boldsymbol{P}\boldsymbol{\mu}(E)]_i} \, \mathrm{d}E + s_{i,k}$$

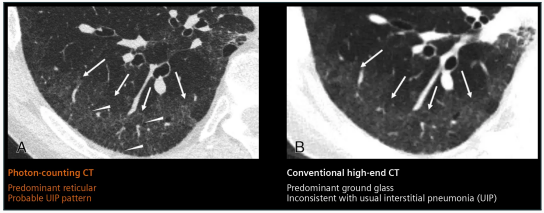
Can be approximated to reconstructed multiple attenuation images $\bar{\boldsymbol{\mu}}_k$.

- **Exact noise model**:

$$(y_{i,k} \mid \boldsymbol{\mu}) \sim \text{Poisson}(\mathcal{A}_{i,k}(\boldsymbol{\mu})) \quad , \ \mathcal{A}_{i,k}(\boldsymbol{\mu}) \triangleq \mathbb{E}[y_{i,k}|\boldsymbol{\mu}]$$

➔ **less noise in reconstructed images**

- PCDs smaller than EIDs ➔ **higher resolution**.



- **Material decomposition**: material density images

$$\boldsymbol{x} = \{\boldsymbol{x}_1, \dots, \boldsymbol{x}_{n_\mathrm{m}}\}$$

- Example: $\boldsymbol{x}_1$: bones; $\boldsymbol{x}_2$; soft tissues; $\boldsymbol{x}_3$: iodine.
- Integration in the forward model through $\boldsymbol{\mu}$:

$$\boldsymbol{\mu}(E) = \sum_{m=1}^{n_\mathrm{m}} \mu_m^{\mathrm{mat}}(E)\boldsymbol{x}_m \triangleq \boldsymbol{\mathcal{F}}(\boldsymbol{x}, E)$$

Photon-counting computed tomography (PCCT)

- **Photon-counting detectors (PCDs)**: capable of counting the photons and of measuring their energy.
- The detections are regrouped into energy bins, $[E_{k-1}, E_k[$, $k = 1, \dots, K$.
- **Expected counts at beam i, energy bin k**:

$$\mathbb{E}[y_{i,k} \mid \boldsymbol{\mu}] = \int h_k(E) \cdot \mathrm{e}^{-[\boldsymbol{P}\boldsymbol{\mu}(E)]_i} \, \mathrm{d}E + s_{i,k}$$

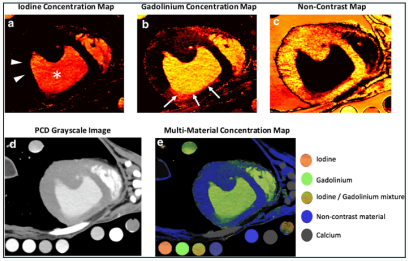
Can be approximated to reconstructed multiple attenuation images $\bar{\boldsymbol{\mu}}_k$.

- **Exact noise model**:

$$(y_{i,k} \mid \boldsymbol{\mu}) \sim \text{Poisson}(\mathcal{A}_{i,k}(\boldsymbol{\mu})) \quad , \; \mathcal{A}_{i,k}(\boldsymbol{\mu}) \triangleq \mathbb{E}[y_{i,k}|\boldsymbol{\mu}]$$

→ **less noise in reconstructed images**

- PCDs smaller than EIDs → **higher resolution**.



- **Material decomposition**: material density images

$$\boldsymbol{x} = \{\boldsymbol{x}_1, \dots, \boldsymbol{x}_{n_\mathrm{m}}\}$$

- Example: $\boldsymbol{x}_1$: bones; $\boldsymbol{x}_2$; soft tissues; $\boldsymbol{x}_3$: iodine.
- Integration in the forward model through $\boldsymbol{\mu}$:

$$\boldsymbol{\mu}(E) = \sum_{m=1}^{n_\mathrm{m}} \mu_m^{\mathrm{mat}}(E)\boldsymbol{x}_m \triangleq \boldsymbol{\mathcal{F}}(\boldsymbol{x}, E)$$

PET Model

- **Object to reconstruct**: $\boldsymbol{x} = \boldsymbol{\lambda}$ the radiactivity distribution.
- **Measurement**: $\boldsymbol{y} = [y_1, \ldots, y_{n_d}]$ detected γ-ray photon pairs along each line of response (LOR) i.
- **Expected detections**:

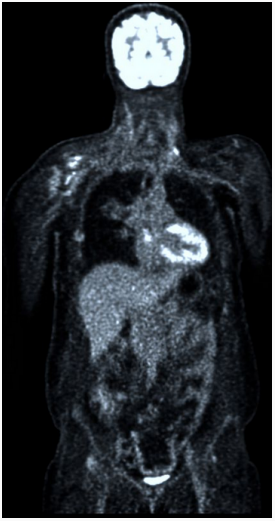
$$\mathbb{E}[y_i \mid \boldsymbol{\lambda}, \boldsymbol{\mu}] = \underbrace{[\boldsymbol{P}\,\boldsymbol{\lambda}]_i}_{\text{emission}} \cdot \underbrace{\mathrm{e}^{-[\boldsymbol{P}\,\boldsymbol{\mu}]_i}}_{\text{attenuation}} + \underbrace{\boldsymbol{r} + \boldsymbol{s}}_{\substack{\text{scatter} \\ \& \text{ randoms}}}$$

where is the γ-ray attenuation.

- **Exact noise model**:

$$y_i \sim \text{Poisson}(\mathcal{A}_i(\boldsymbol{\lambda}, \boldsymbol{\mu})) \quad , \quad \mathcal{A}_i(\boldsymbol{\lambda}, \boldsymbol{\mu}) \triangleq \mathbb{E}[y_i \mid \boldsymbol{\lambda}, \boldsymbol{\mu}]$$

- $\boldsymbol{\mu}$ derived from CT or MR.
- If not, $\boldsymbol{x} = (\boldsymbol{\lambda}, \boldsymbol{\mu})$ can be reconstructed from $\boldsymbol{y}$ only [Rezaei et al. 2012], severely ill-posed in absence of time-of-flight (TOF) information.



PET Model

- **Object to reconstruct**: $\boldsymbol{x} = \boldsymbol{\lambda}$ the radiactivity distribution.
- **Measurement**: $\boldsymbol{y} = [y_1, \ldots, y_{n_d}]$ detected γ-ray photon pairs along each line of response (LOR) i.
- **Expected detections**:

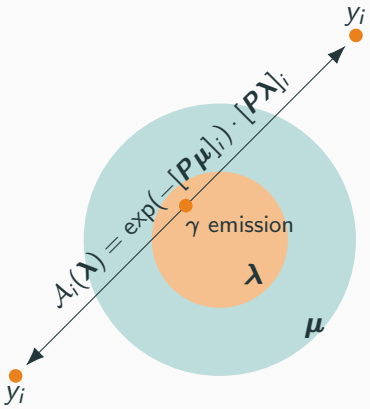
$$\mathbb{E}[y_i \mid \boldsymbol{\lambda}, \boldsymbol{\mu}] = \underbrace{[\boldsymbol{P}\boldsymbol{\lambda}]_i}_{\text{emission}} \cdot \underbrace{\mathrm{e}^{-[\boldsymbol{P}\boldsymbol{\mu}]_i}}_{\text{attenuation}} + \underbrace{\boldsymbol{r} + \boldsymbol{s}}_{\substack{\text{scatter} \\ \& \text{ randoms}}}$$

where is the γ-ray attenuation.

- **Exact noise model**:

$$y_i \sim \text{Poisson}(\mathcal{A}_i(\boldsymbol{\lambda}, \boldsymbol{\mu})) \quad , \quad \mathcal{A}_i(\boldsymbol{\lambda}, \boldsymbol{\mu}) \triangleq \mathbb{E}[y_i \mid \boldsymbol{\lambda}, \boldsymbol{\mu}]$$

- $\boldsymbol{\mu}$ derived from CT or MR.
- If not, $\boldsymbol{x} = (\boldsymbol{\lambda}, \boldsymbol{\mu})$ can be reconstructed from $\boldsymbol{y}$ only [Rezaei et al. 2012], severely ill-posed in absence of time-of-flight (TOF) information.



PET Model

- **Object to reconstruct**: $\boldsymbol{x} = \boldsymbol{\lambda}$ the radiactivity distribution.
- **Measurement**: $\boldsymbol{y} = [y_1, \ldots, y_{n_d}]$ detected γ-ray photon pairs along each line of response (LOR) i.
- **Expected detections**:

$$\mathbb{E}[y_i \mid \boldsymbol{\lambda}, \boldsymbol{\mu}] = \underbrace{[\boldsymbol{P}\boldsymbol{\lambda}]_i}_{\text{emission}} \cdot \underbrace{\mathrm{e}^{-[\boldsymbol{P}\boldsymbol{\mu}]_i}}_{\text{attenuation}} + \underbrace{\boldsymbol{r} + \boldsymbol{s}}_{\substack{\text{scatter} \\ \& \text{ randoms}}}$$

where is the γ-ray attenuation.

- **Exact noise model**:

$$y_i \sim \text{Poisson}(\mathcal{A}_i(\boldsymbol{\lambda}, \boldsymbol{\mu})) \quad , \quad \mathcal{A}_i(\boldsymbol{\lambda}, \boldsymbol{\mu}) \triangleq \mathbb{E}[y_i \mid \boldsymbol{\lambda}, \boldsymbol{\mu}]$$

- $\boldsymbol{\mu}$ derived from CT or MR.
- If not, $\boldsymbol{x} = (\boldsymbol{\lambda}, \boldsymbol{\mu})$ can be reconstructed from $\boldsymbol{y}$ only [Rezaei et al. 2012], severely ill-posed in absence of time-of-flight (TOF) information.

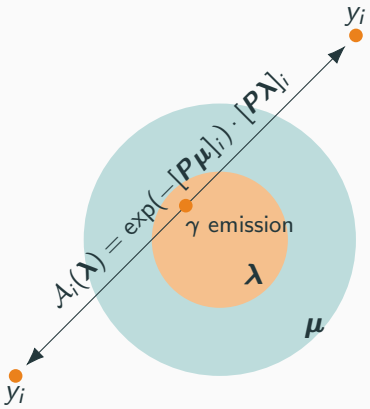


Image reconstruction: notations

- $\boldsymbol{x} = [x_1, \dots, x_n]^\top$: image(s) to reconstruct, with prior PDF $p(\boldsymbol{x})$.
- $\boldsymbol{y} = [y_1, \dots, y_m]^\top$: measurement vector with conditional PDF $p(\boldsymbol{y} \mid \boldsymbol{x})$.
- $\boldsymbol{\mathcal{A}} = [\mathcal{A}_1, \dots, \mathcal{A}_m]^\top$: forward model, such that

$$\boldsymbol{\mathcal{A}}(\boldsymbol{x}) = \mathbb{E}[\boldsymbol{y} \mid \boldsymbol{x}]$$

- **Standard image reconstruction:** find $\boldsymbol{x}$ s.t.

$$\boldsymbol{y} \approx \boldsymbol{\mathcal{A}}(\boldsymbol{x}) \quad \text{or} \quad \min_{\boldsymbol{x}} \ell\big(\boldsymbol{y}, \boldsymbol{\mathcal{A}}(\boldsymbol{x})\big)$$

where ℓ is a loss function, usually defined such that

$$\ell\big(\boldsymbol{y}, \boldsymbol{\mathcal{A}}(\boldsymbol{x})\big) \approx -\log p(\boldsymbol{y} \mid \boldsymbol{x})$$

E.g., monochromatic CT: $\ell\big(\boldsymbol{y}, \boldsymbol{R}\boldsymbol{x}\big) \approx \frac{1}{2}\|\boldsymbol{P}\boldsymbol{x} - \boldsymbol{b}\|_{\boldsymbol{W}}^2$, $\boldsymbol{b} = -\log \boldsymbol{y}/\boldsymbol{I}$ [Elbakri and Fessler 2002].

- **Regularisation**:

$$\min_{\boldsymbol{x}} \ell\big(\boldsymbol{y}, \boldsymbol{\mathcal{A}}(\boldsymbol{x})\big) + R(\boldsymbol{x})$$

- R designed to "mimic" the prior:

$$R(\boldsymbol{x}) \propto -\log p(\boldsymbol{x})$$

- Typically handcrafted, e.g.,
  - $R(\boldsymbol{x}) = \beta\|\boldsymbol{T}\boldsymbol{x}\|_1$ (TV prior)
  - $R(\boldsymbol{x}) = \beta\frac{1}{2}\|\boldsymbol{T}\boldsymbol{x}\|_2^2$ (quadratic prior)
  - Huber prior, etc.
- Standard algorithm for differentiable prior:

$$\boldsymbol{x}^{(q+1)} = \boldsymbol{x}^{(q)} - \alpha\left(\nabla_{\boldsymbol{x}} L\left(\boldsymbol{x}^{(q)}\right) + \nabla R\left(\boldsymbol{x}^{(q)}\right)\right)$$

with $L(\boldsymbol{x}) = \ell(\boldsymbol{y}, \boldsymbol{\mathcal{A}}(\boldsymbol{x}))$.

Image reconstruction: notations

- $\boldsymbol{x} = [x_1, \dots, x_n]^\top$: image(s) to reconstruct, with prior PDF $p(\boldsymbol{x})$.
- $\boldsymbol{y} = [y_1, \dots, y_m]^\top$: measurement vector with conditional PDF $p(\boldsymbol{y} \mid \boldsymbol{x})$.
- $\boldsymbol{\mathcal{A}} = [\mathcal{A}_1, \dots, \mathcal{A}_m]^\top$: forward model, such that

$$\boldsymbol{\mathcal{A}}(\boldsymbol{x}) = \mathbb{E}[\boldsymbol{y} \mid \boldsymbol{x}]$$

- **Standard image reconstruction:** find $\boldsymbol{x}$ s.t.

$$\boldsymbol{y} \approx \boldsymbol{\mathcal{A}}(\boldsymbol{x}) \quad \text{or} \quad \min_{\boldsymbol{x}} \ell\big(\boldsymbol{y}, \boldsymbol{\mathcal{A}}(\boldsymbol{x})\big)$$

where ℓ is a loss function, usually defined such that

$$\ell\big(\boldsymbol{y}, \boldsymbol{\mathcal{A}}(\boldsymbol{x})\big) \approx -\log p(\boldsymbol{y} \mid \boldsymbol{x})$$

E.g., monochromatic CT: $\ell\big(\boldsymbol{y}, \boldsymbol{R}\boldsymbol{x}\big) \approx \frac{1}{2}\|\boldsymbol{P}\boldsymbol{x} - \boldsymbol{b}\|^2_{\boldsymbol{W}}$, $\boldsymbol{b} = -\log \boldsymbol{y}/I$ [Elbakri and Fessler 2002].

- **Regularisation**:

$$\min_{\boldsymbol{x}} \ell\big(\boldsymbol{y}, \boldsymbol{\mathcal{A}}(\boldsymbol{x})\big) + R(\boldsymbol{x})$$

- R designed to "mimic" the prior:

$$R(\boldsymbol{x}) \propto -\log p(\boldsymbol{x})$$

- Typically handcrafted, e.g.,
  - $R(\boldsymbol{x}) = \beta\|\boldsymbol{T}\boldsymbol{x}\|_1$ (TV prior)
  - $R(\boldsymbol{x}) = \beta\frac{1}{2}\|\boldsymbol{T}\boldsymbol{x}\|_2^2$ (quadratic prior)
  - Huber prior, etc.
- Standard algorithm for differentiable prior:

$$\boldsymbol{x}^{(q+1)} = \boldsymbol{x}^{(q)} - \alpha\left(\nabla_{\boldsymbol{x}} L\left(\boldsymbol{x}^{(q)}\right) + \nabla R\left(\boldsymbol{x}^{(q)}\right)\right)$$

with $L(\boldsymbol{x}) = \ell(\boldsymbol{y}, \boldsymbol{\mathcal{A}}(\boldsymbol{x}))$.

Normal dose reconstruction

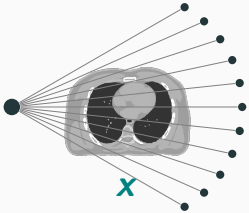
standard
acquisition



x

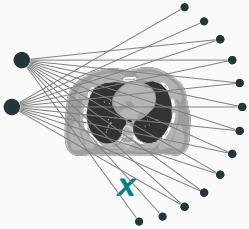
Normal dose reconstruction

standard
acquisition



Normal dose reconstruction

standard
acquisition



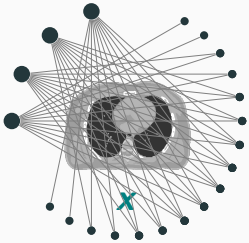
Normal dose reconstruction

standard
acquisition



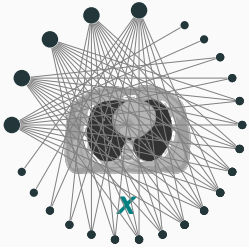
Normal dose reconstruction

standard
acquisition



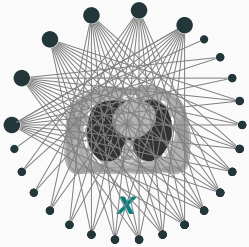
Normal dose reconstruction

standard
acquisition



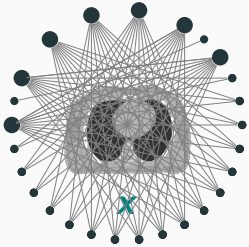
Normal dose reconstruction

standard acquisition



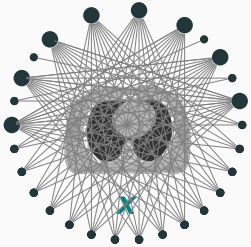
Normal dose reconstruction

standard
acquisition



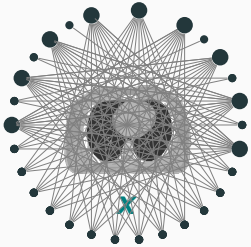
Normal dose reconstruction

standard
acquisition



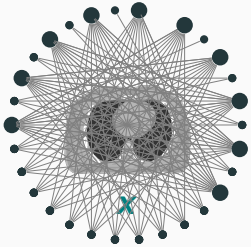
Normal dose reconstruction

standard
acquisition



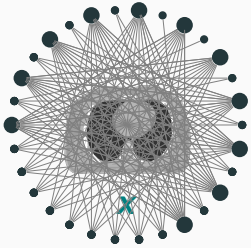
Normal dose reconstruction

standard
acquisition



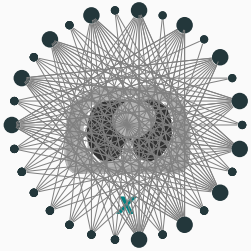
Normal dose reconstruction

standard
acquisition



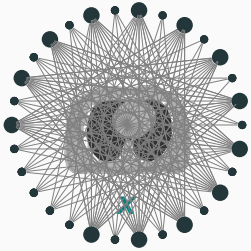
Normal dose reconstruction

standard
acquisition



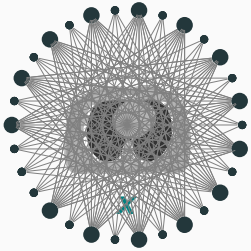
Normal dose reconstruction

standard
acquisition



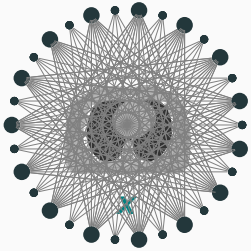
Normal dose reconstruction

standard
acquisition



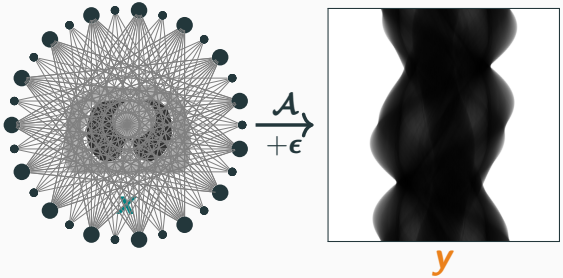
Normal dose reconstruction

standard
acquisition



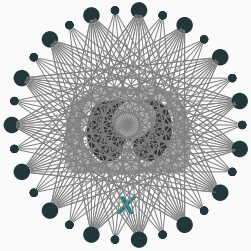
Normal dose reconstruction

standard acquisition



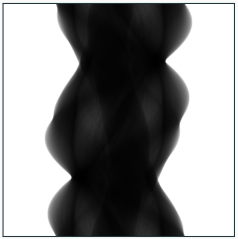
Normal dose reconstruction

standard acquisition



x

$\mathcal{A}$, $+\epsilon$



y

$\min_x \ell\big(y, \mathcal{A}(x)\big)$



$\widehat{x}$

Low-dose reconstruction

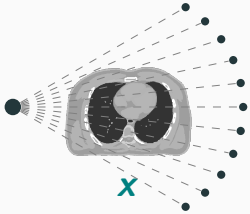
low-dose
acquisition



$\boldsymbol{x}$

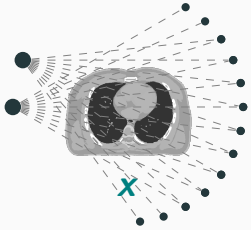
Low-dose reconstruction

low-dose
acquisition



Low-dose reconstruction

low-dose
acquisition



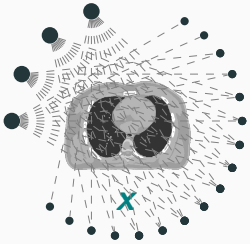
Low-dose reconstruction

low-dose
acquisition



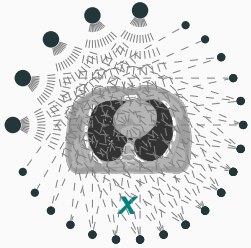
Low-dose reconstruction

low-dose
acquisition



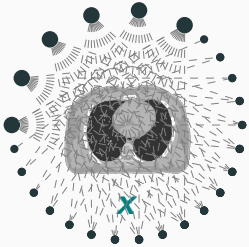
Low-dose reconstruction

low-dose
acquisition



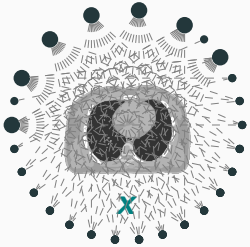
Low-dose reconstruction

low-dose
acquisition



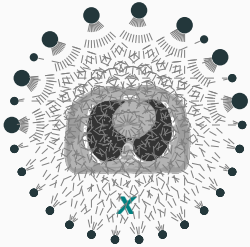
Low-dose reconstruction

low-dose
acquisition



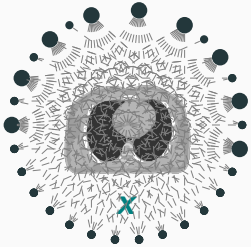
Low-dose reconstruction

low-dose
acquisition



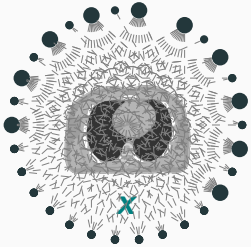
Low-dose reconstruction

low-dose
acquisition



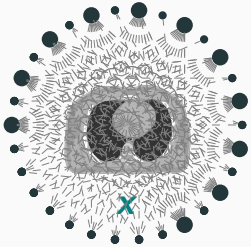
Low-dose reconstruction

low-dose
acquisition



Low-dose reconstruction

low-dose
acquisition



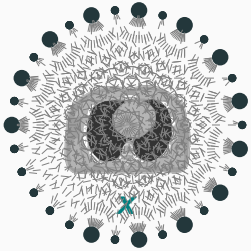
Low-dose reconstruction

low-dose
acquisition



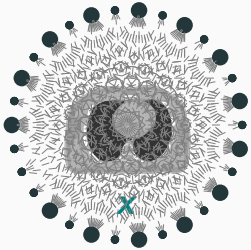
Low-dose reconstruction

low-dose
acquisition



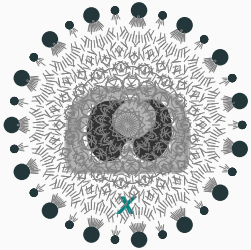
Low-dose reconstruction

low-dose
acquisition



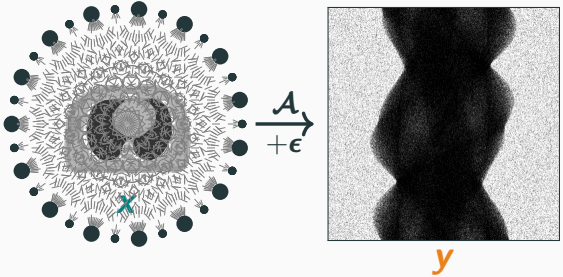
Low-dose reconstruction

low-dose
acquisition



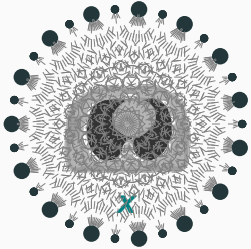
Low-dose reconstruction

low-dose
acquisition



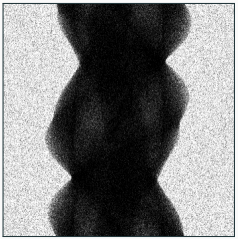
Low-dose reconstruction

low-dose acquisition



x

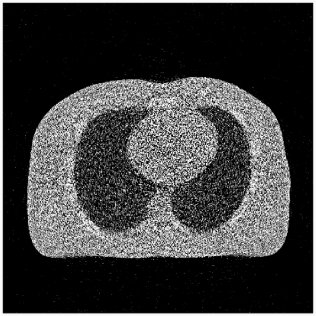
$\mathcal{A}$
$+\epsilon$



y

$\min_x \ell\big(y, \mathcal{A}(x)\big)$

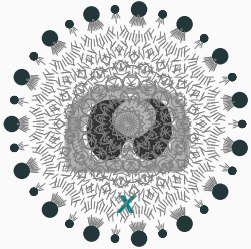
noise



$\widehat{x}$

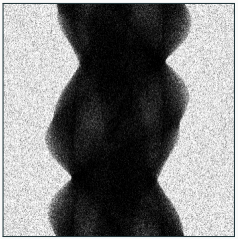
Low-dose reconstruction

low-dose
acquisition



x

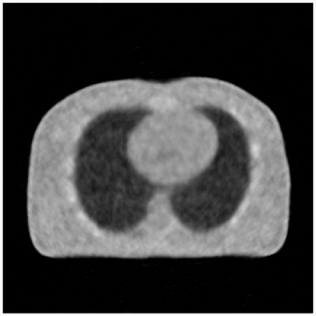
$\mathcal{A}$
$+\epsilon$



y

$\min_x \ell(y, \mathcal{A}(x))$
$+R(x)$

resolution ↘
noise ↘



$\widehat{x}$

Sparse-view reconstruction

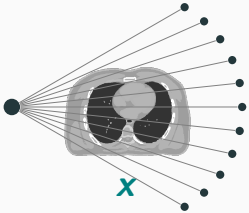
sparse-view



$\boldsymbol{x}$

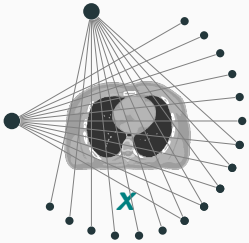
Sparse-view reconstruction

sparse-view



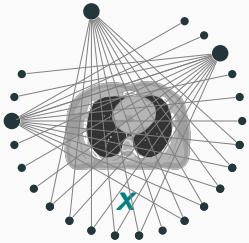
Sparse-view reconstruction

sparse-view



Sparse-view reconstruction

sparse-view



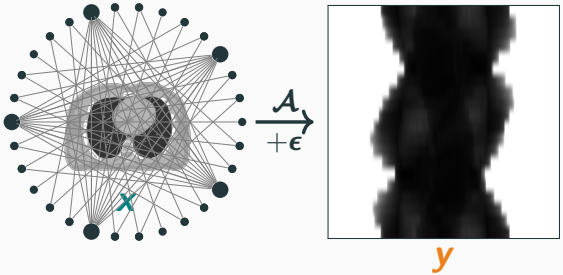
Sparse-view reconstruction

sparse-view



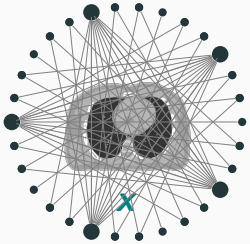
Sparse-view reconstruction

sparse-view

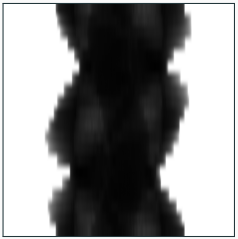


Sparse-view reconstruction

sparse-view



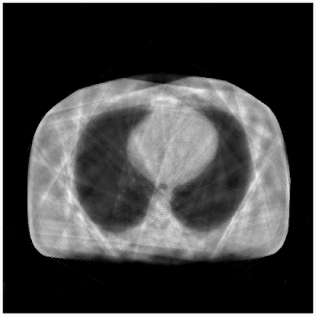
$\mathcal{A}$
$+\epsilon$



y

$\min_x \ell(y, \mathcal{A}(x))$

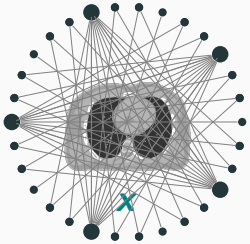
artefacts



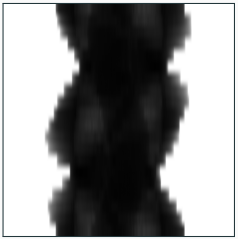
$\widehat{x}$

Sparse-view reconstruction

sparse-view



$\mathcal{A}$
$+\epsilon$



y

$\min_x \ell(y, \mathcal{A}(x))$
$+\beta\|Tx\|_1$

resolution ↘
artefacts ↘



$\widehat{x}$

AI: new paradigm

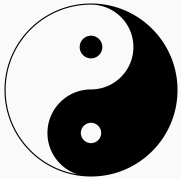


AI: new paradigm

dose ↘

noise ↘
resolution ↗

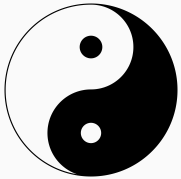
dose ↘



noise ↘
resolution ↗

conventional
priors

dose ↘



noise ↘
resolution ↗

conventional
priors

AI → better priors

Diffusion Models for Image Reconstruction

AI: End-to-end Supervised Models vs Generative Models

End-to-end supervised methods

What is learned: a mapping to invert the measurement process

$$(\boldsymbol{y}^{\text{degraded}} \text{ or } \boldsymbol{x}^{\text{degraded}}) \mapsto \boldsymbol{x}$$

- **Pros**
  - Simple training objective and inference
  - Great performances
- **Cons**
  - Requires paired labels (noisy–clean, raw–reference)
  - Can be sensitive to protocol and scanner differences

Examples: FBPConvNet [Jin et al. 2017], algorithm unrolling [Monga et al. 2021];

generative methods

What is learned: an image prior distribution

$$\boldsymbol{x} \sim p(\boldsymbol{x})$$

- **Pros**
  - Requires clean images only
  - Naturally combines with physics for data consistency
- **Cons**
  - Training and sampling are more computationally demanding

Focus here: diffusion priors + physics-informed sampling.

AI: End-to-end Supervised Models vs Generative Models

End-to-end supervised methods

What is learned: a mapping to invert the measurement process

$$(\boldsymbol{y}^{\text{degraded}} \text{ or } \boldsymbol{x}^{\text{degraded}}) \mapsto \boldsymbol{x}$$

- **Pros**
  - Simple training objective and inference
  - Great performances
- **Cons**
  - Requires paired labels (noisy–clean, raw–reference)
  - Can be sensitive to protocol and scanner differences

Examples: FBPConvNet [Jin et al. 2017], algorithm unrolling [Monga et al. 2021];

generative methods

What is learned: an image prior distribution

$$\boldsymbol{x} \sim p(\boldsymbol{x})$$

- **Pros**
  - Requires clean images only
  - Naturally combines with physics for data consistency
- **Cons**
  - Training and sampling are more computationally demanding

Focus here: diffusion priors + physics-informed sampling.

Generative models: idea [Duff et al. 2021]

Image space $\mathcal{X}$



Generative models: idea [Duff et al. 2021]

Image space $\mathcal{X}$





Generative models: idea [Duff et al. 2021]

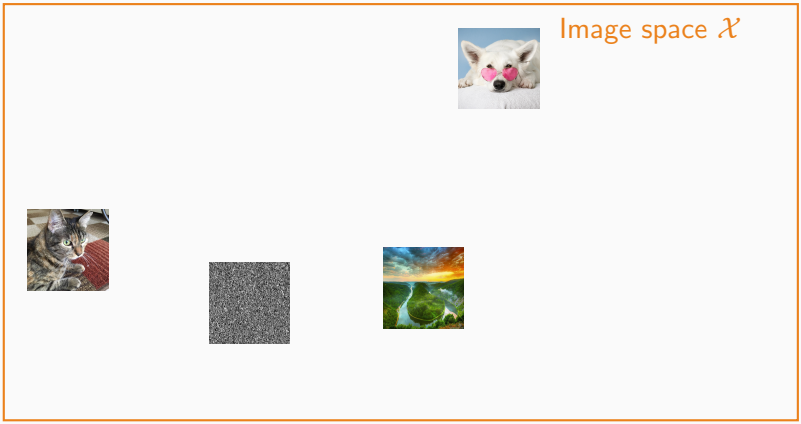
Image space $\mathcal{X}$



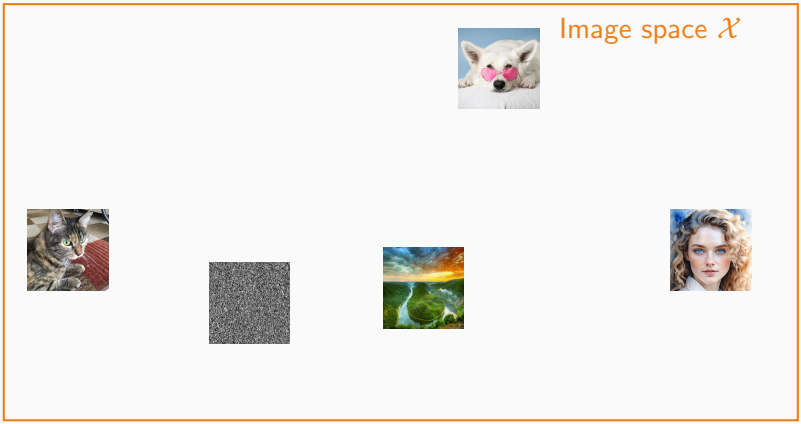




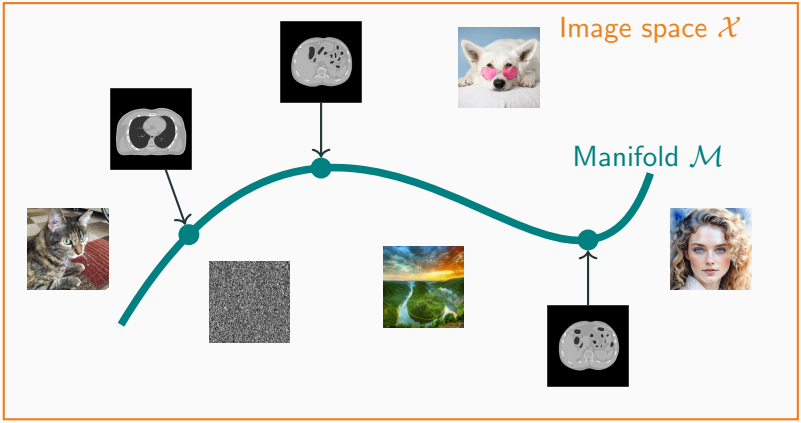
Generative models: idea [Duff et al. 2021]



Generative models: idea [Duff et al. 2021]

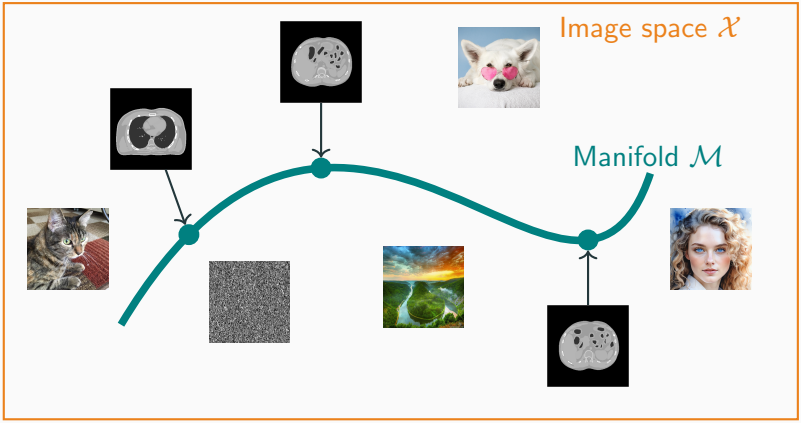


Generative models: idea [Duff et al. 2021]

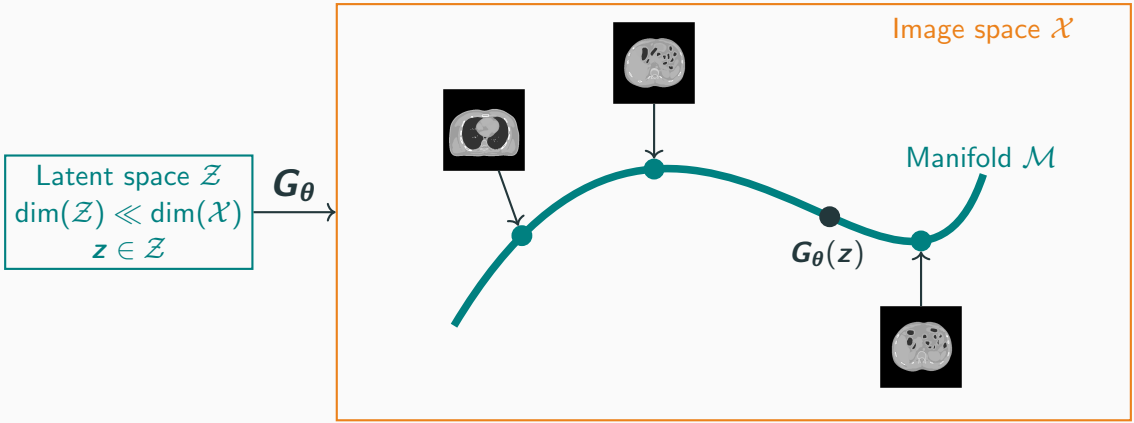


Generative models: idea [Duff et al. 2021]

Latent space $\mathcal{Z}$
$\dim(\mathcal{Z}) \ll \dim(\mathcal{X})$
$\boldsymbol{z} \in \mathcal{Z}$



Generative models: idea [Duff et al. 2021]



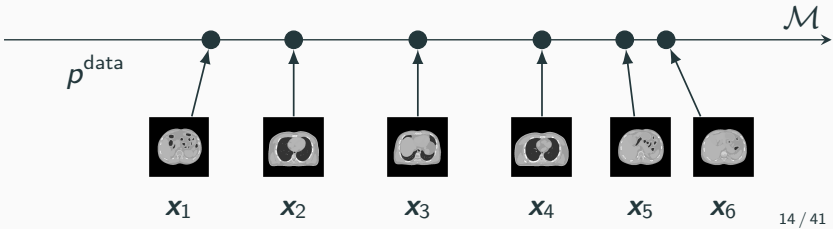
Generative models: desired properties

- Data manifold $\mathcal{M}$, training data $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_L\}$
- Training data distribution: $p^{\text{data}} = \frac{1}{L}\sum_{\ell=1}^{L} \delta_{\boldsymbol{x}_\ell}$
- $\boldsymbol{z} \sim p_0$
- Deep generator: $\boldsymbol{G}_\theta \colon \mathcal{Z} \to \mathcal{X}$
- $\boldsymbol{G}_\theta(\boldsymbol{z}) \sim p_\theta$: trained as continuous extension of p^{data} to $\mathcal{M}$



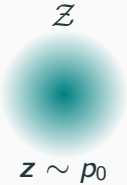
Generative models: desired properties

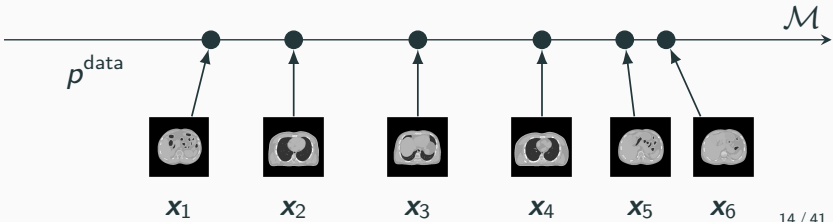
- Data manifold $\mathcal{M}$, training data $\{\boldsymbol{x}_1, \dots, \boldsymbol{x}_L\}$
- Training data distribution: $p^{\text{data}} = \frac{1}{L} \sum_{\ell=1}^{L} \delta_{\boldsymbol{x}_\ell}$
- $\boldsymbol{z} \sim p_0$
- Deep generator: $\boldsymbol{G}_\theta \colon \mathcal{Z} \to \mathcal{X}$
- $\boldsymbol{G}_\theta(\boldsymbol{z}) \sim p_\theta$: trained as continuous extension of p^{data} to $\mathcal{M}$



Generative models: desired properties

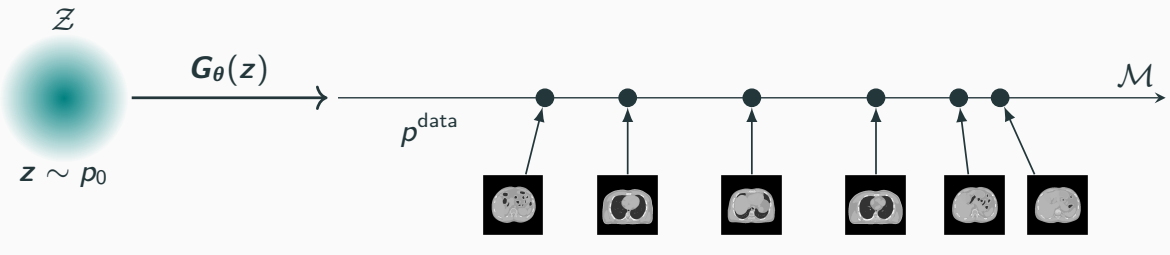
- Data manifold $\mathcal{M}$, training data $\{x_1, \dots, x_L\}$
- Training data distribution: $p^{\text{data}} = \frac{1}{L} \sum_{\ell=1}^{L} \delta_{x_\ell}$
- $z \sim p_0$
- Deep generator: $G_\theta : \mathcal{Z} \to \mathcal{X}$
- $G_\theta(z) \sim p_\theta$: trained as continuous extension of p^{data} to $\mathcal{M}$





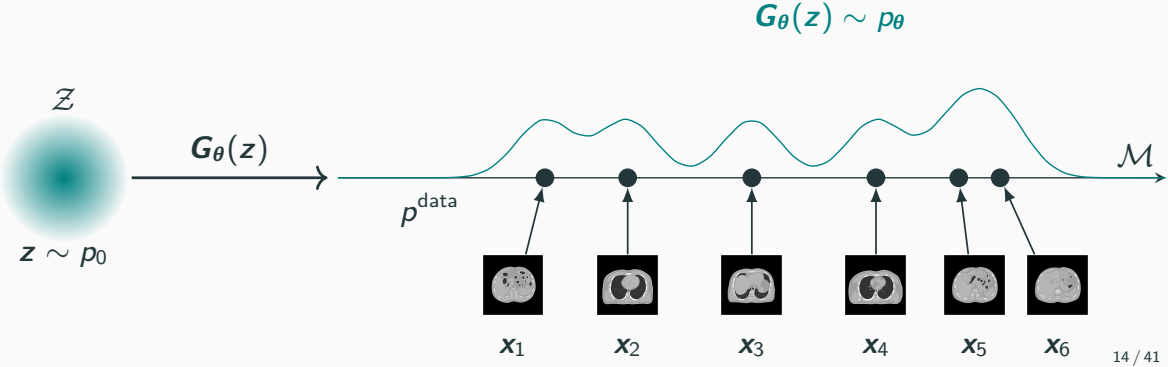
Generative models: desired properties

- Data manifold $\mathcal{M}$, training data $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_L\}$
- Training data distribution: $p^{\text{data}} = \frac{1}{L} \sum_{\ell=1}^{L} \delta_{\boldsymbol{x}_\ell}$
- $\boldsymbol{z} \sim p_0$
- Deep generator: $\boldsymbol{G}_{\boldsymbol{\theta}} \colon \mathcal{Z} \to \mathcal{X}$
- $\boldsymbol{G}_{\boldsymbol{\theta}}(\boldsymbol{z}) \sim p_{\boldsymbol{\theta}}$: trained as continuous extension of p^{data} to $\mathcal{M}$



Generative models: desired properties

- Data manifold $\mathcal{M}$, training data $\{x_1, \dots, x_L\}$
- Training data distribution: $p^{\text{data}} = \frac{1}{L} \sum_{\ell=1}^{L} \delta_{x_\ell}$
- $z \sim p_0$
- Deep generator: $G_\theta \colon \mathcal{Z} \to \mathcal{X}$
- $G_\theta(z) \sim p_\theta$: trained as continuous extension of p^{data} to $\mathcal{M}$



Methods that use a generator G_θ

Variational Autoencoders (VAEs)

- **Pros:**
  - Stable training
  - Clear probabilistic interpretation
- **Cons:**
  - Limited sharpness
  - Slightly blurred images

Generative Adversarial Networks (GANs)

- **Pros:**
  - High visual quality
  - Sharp details
- **Cons:**
  - Unstable training
  - Mode collapse

Diffusion Models

- **Pros:**
  - High visual quality
  - Stable training
- **Cons:**
  - High computational cost
  - Slow sampling

Methods that use a generator G_θ

Variational Autoencoders (VAEs)

- **Pros:**
  - Stable training
  - Clear probabilistic interpretation
- **Cons:**
  - Limited sharpness
  - Slightly blurred images

Generative Adversarial Networks (GANs)

- **Pros:**
  - High visual quality
  - Sharp details
- **Cons:**
  - Unstable training
  - Mode collapse

Diffusion Models

- **Pros:**
  - High visual quality
  - Stable training
- **Cons:**
  - High computational cost
  - Slow sampling

Methods that use a generator G_θ

Variational Autoencoders (VAEs)

- **Pros:**
  - Stable training
  - Clear probabilistic interpretation
- **Cons:**
  - Limited sharpness
  - Slightly blurred images

Generative Adversarial Networks (GANs)

- **Pros:**
  - High visual quality
  - Sharp details
- **Cons:**
  - Unstable training
  - Mode collapse

Diffusion Models

- **Pros:**
  - High visual quality
  - Stable training
- **Cons:**
  - High computational cost
  - Slow sampling

Diffusion Denoising Probabilistic Model (DDPM) [Ho et al. 2020]

$p^{\text{data}} \sim$ 

- **Encoding**: starting from $\boldsymbol{x}_0 \sim p^{\text{data}}$, then sampling $\boldsymbol{x}_0 \to \boldsymbol{x}_1 \to \boldsymbol{x}_2 \to \cdots \to \boldsymbol{x}_T$ as

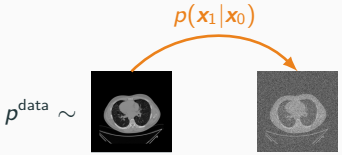
$$\boldsymbol{x}_t = \sqrt{\alpha_t}\boldsymbol{x}_{t-1} + \sqrt{1-\alpha_t}\boldsymbol{\epsilon}_t \quad , \quad \boldsymbol{\epsilon}_t \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$$

  with $\alpha_t \searrow 0$ so that $\boldsymbol{x}_T \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$.
- **Decoding**: starting from $\boldsymbol{x}_T \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$, then sampling $\boldsymbol{x}_T \to \boldsymbol{x}_{T-1} \to \boldsymbol{x}_{T-2} \to \cdots \to \boldsymbol{x}_0$ as

$$\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t \sim q(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t)$$

- Can also synergistically sample multichannel images (e.g., RGB images, PET/CT, PET/MR, spectral CT, etc.)

Diffusion Denoising Probabilistic Model (DDPM) [Ho et al. 2020]



- **Encoding**: starting from $x_0 \sim p^{\text{data}}$, then sampling $x_0 \to x_1 \to x_2 \to \cdots \to x_T$ as

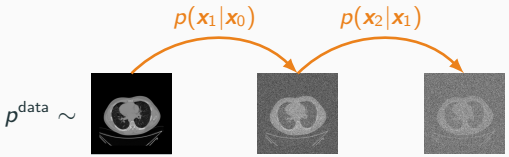
$$x_t = \sqrt{\alpha_t} x_{t-1} + \sqrt{1-\alpha_t}\epsilon_t \quad , \quad \epsilon_t \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$$

  with $\alpha_t \searrow 0$ so that $x_T \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$.
- **Decoding**: starting from $x_T \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$, then sampling $x_T \to x_{T-1} \to x_{T-2} \to \cdots \to x_0$ as

$$x_{t-1} \mid x_t \sim q(x_{t-1} \mid x_t)$$

- Can also synergistically sample multichannel images (e.g., RGB images, PET/CT, PET/MR, spectral CT, etc.)

Diffusion Denoising Probabilistic Model (DDPM) [Ho et al. 2020]



- **Encoding**: starting from $x_0 \sim p^{\text{data}}$, then sampling $x_0 \to x_1 \to x_2 \to \cdots \to x_T$ as

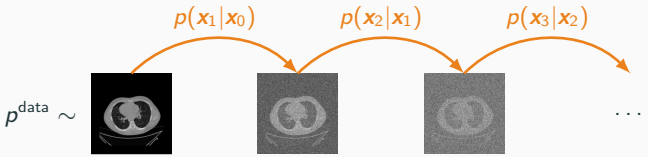
$$x_t = \sqrt{\alpha_t} x_{t-1} + \sqrt{1-\alpha_t}\epsilon_t \quad , \quad \epsilon_t \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$$

  with $\alpha_t \searrow 0$ so that $x_T \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$.
- **Decoding**: starting from $x_T \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$, then sampling $x_T \to x_{T-1} \to x_{T-2} \to \cdots \to x_0$ as

$$x_{t-1} \mid x_t \sim q(x_{t-1} \mid x_t)$$

- Can also synergistically sample multichannel images (e.g., RGB images, PET/CT, PET/MR, spectral CT, etc.)

Diffusion Denoising Probabilistic Model (DDPM) [Ho et al. 2020]



- **Encoding**: starting from $x_0 \sim p^{\text{data}}$, then sampling $x_0 \to x_1 \to x_2 \to \cdots \to x_T$ as

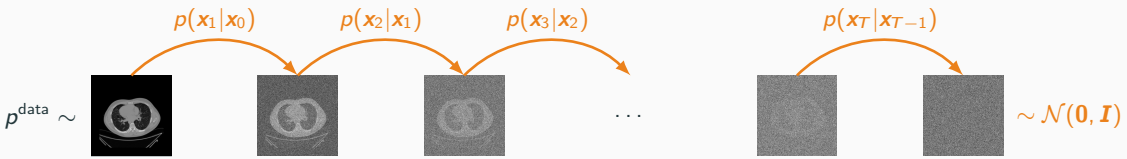
$$x_t = \sqrt{\alpha_t} x_{t-1} + \sqrt{1-\alpha_t}\epsilon_t \quad , \quad \epsilon_t \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$$

  with $\alpha_t \searrow 0$ so that $x_T \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$.
- **Decoding**: starting from $x_T \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$, then sampling $x_T \to x_{T-1} \to x_{T-2} \to \cdots \to x_0$ as

$$x_{t-1} \mid x_t \sim q(x_{t-1} \mid x_t)$$

- Can also synergistically sample multichannel images (e.g., RGB images, PET/CT, PET/MR, spectral CT, etc.)

Diffusion Denoising Probabilistic Model (DDPM) [Ho et al. 2020]



- **Encoding**: starting from $\boldsymbol{x}_0 \sim p^{\text{data}}$, then sampling $\boldsymbol{x}_0 \to \boldsymbol{x}_1 \to \boldsymbol{x}_2 \to \cdots \to \boldsymbol{x}_T$ as

$$\boldsymbol{x}_t = \sqrt{\alpha_t}\boldsymbol{x}_{t-1} + \sqrt{1-\alpha_t}\boldsymbol{\epsilon}_t \quad , \quad \boldsymbol{\epsilon}_t \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$$

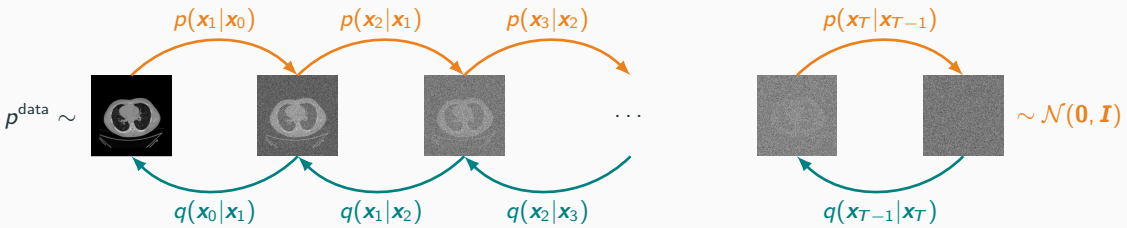
  with $\alpha_t \searrow 0$ so that $\boldsymbol{x}_T \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$.

- **Decoding**: starting from $\boldsymbol{x}_T \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$, then sampling $\boldsymbol{x}_T \to \boldsymbol{x}_{T-1} \to \boldsymbol{x}_{T-2} \to \cdots \to \boldsymbol{x}_0$ as

$$\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t \sim q(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t)$$

- Can also synergistically sample multichannel images (e.g., RGB images, PET/CT, PET/MR, spectral CT, etc.)

Diffusion Denoising Probabilistic Model (DDPM) [Ho et al. 2020]



- **Encoding**: starting from $\boldsymbol{x}_0 \sim p^{\text{data}}$, then sampling $\boldsymbol{x}_0 \to \boldsymbol{x}_1 \to \boldsymbol{x}_2 \to \cdots \to \boldsymbol{x}_T$ as

$$\boldsymbol{x}_t = \sqrt{\alpha_t}\boldsymbol{x}_{t-1} + \sqrt{1-\alpha_t}\boldsymbol{\epsilon}_t \quad , \quad \boldsymbol{\epsilon}_t \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$$

  with $\alpha_t \searrow 0$ so that $\boldsymbol{x}_T \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$.
- **Decoding**: starting from $\boldsymbol{x}_T \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$, then sampling $\boldsymbol{x}_T \to \boldsymbol{x}_{T-1} \to \boldsymbol{x}_{T-2} \to \cdots \to \boldsymbol{x}_0$ as

$$\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t \sim q(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t)$$

- Can also synergistically sample multichannel images (e.g., RGB images, PET/CT, PET/MR, spectral CT, etc.)

How do we train and sample in diffusion models?

- **Key difficulty:** The true reverse PDF $q(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t)$ is unknown.
- **Key insight:** An approximate reverse transition depends on the score $\nabla \log p_t(\boldsymbol{x}_t)$ through Tweedie's formula, but it is unknown.
- **Training: Score Matching** to approximate the score [Vincent 2011]

$$\min_{\boldsymbol{\theta}} \mathbb{E}_{\boldsymbol{x}_t|\boldsymbol{x}_0, \boldsymbol{x}_0 \sim p^{\text{data}}, t} \left[\left\| \boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t) - \nabla \log p_t(\boldsymbol{x}_t) \right\|_2^2 \right]$$

$$\Leftrightarrow \quad \min_{\boldsymbol{\theta}} \mathbb{E}_{\boldsymbol{x}_t|\boldsymbol{x}_0, \boldsymbol{x}_0 \sim p^{\text{data}}, t} \left[\gamma_t \left\| \hat{\boldsymbol{x}}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t) - \boldsymbol{x}_0 \right\|_2^2 \right]$$

where $\hat{\boldsymbol{x}}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t) \approx \mathbb{E}[\boldsymbol{x}_0 \mid \boldsymbol{x}_t]$ is a denoiser.

- **Result:** Use the learned score $\boldsymbol{s}_{\boldsymbol{\theta}}$ to run the reverse sampling process.

Reverse diffusion: DDPM [Ho et al. 2020]

Require: $T, (\alpha_t)_{t=1}^T, (\sigma_t)_{t=1}^T$
  $\boldsymbol{x}_T \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$
  for $t = T, \ldots, 1$ **do**
    $\hat{\boldsymbol{x}}_{\boldsymbol{\theta}} \leftarrow \frac{\boldsymbol{x}_t + (1-\bar{\alpha}_t)\boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)}{\sqrt{\bar{\alpha}_t}}$ ▷ *Tweedie's estimate*
    $\boldsymbol{\mu}_t = \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})\boldsymbol{x}_t + \sqrt{\bar{\alpha}_{t-1}}(1-\alpha_t)\hat{\boldsymbol{x}}_{\boldsymbol{\theta}}}{1-\bar{\alpha}_t}$
    $\boldsymbol{x}_{t-1} \sim \mathcal{N}(\boldsymbol{\mu}_t, \sigma_t^2 \boldsymbol{I})$
  return $\boldsymbol{x}_0$

Faster alternative (although less stable): Diffusion denoising implicit model (DDIM) [Song et al. 2020]

How do we train and sample in diffusion models?

- **Key difficulty:** The true reverse PDF $q(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t)$ is unknown.
- **Key insight:** An approximate reverse transition depends on the score $\nabla \log p_t(\boldsymbol{x}_t)$ through Tweedie's formula, but it is unknown.
- **Training: Score Matching** to approximate the score [Vincent 2011]

$$\min_{\boldsymbol{\theta}} \mathbb{E}_{\boldsymbol{x}_t|\boldsymbol{x}_0, \boldsymbol{x}_0 \sim p^{\text{data}}, t} \left[\left\| \boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t) - \nabla \log p_t(\boldsymbol{x}_t) \right\|_2^2 \right]$$

$$\Leftrightarrow \quad \min_{\boldsymbol{\theta}} \mathbb{E}_{\boldsymbol{x}_t|\boldsymbol{x}_0, \boldsymbol{x}_0 \sim p^{\text{data}}, t} \left[\gamma_t \left\| \hat{\boldsymbol{x}}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t) - \boldsymbol{x}_0 \right\|_2^2 \right]$$

where $\hat{\boldsymbol{x}}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t) \approx \mathbb{E}[\boldsymbol{x}_0 \mid \boldsymbol{x}_t]$ is a denoiser.

- **Result:** Use the learned score $\boldsymbol{s}_{\boldsymbol{\theta}}$ to run the reverse sampling process.

Reverse diffusion: DDPM [Ho et al. 2020]

Require: $T, (\alpha_t)_{t=1}^T, (\sigma_t)_{t=1}^T$
  $\boldsymbol{x}_T \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$
  for $t = T, \ldots, 1$ **do**
    $\hat{\boldsymbol{x}}_{\boldsymbol{\theta}} \leftarrow \frac{\boldsymbol{x}_t + (1-\bar{\alpha}_t)\boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)}{\sqrt{\bar{\alpha}_t}}$ ▷ *Tweedie's estimate*
    $\boldsymbol{\mu}_t = \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})\boldsymbol{x}_t + \sqrt{\bar{\alpha}_{t-1}}(1-\alpha_t)\hat{\boldsymbol{x}}_{\boldsymbol{\theta}}}{1-\bar{\alpha}_t}$
    $\boldsymbol{x}_{t-1} \sim \mathcal{N}(\boldsymbol{\mu}_t, \sigma_t^2 \boldsymbol{I})$
  return $\boldsymbol{x}_0$

Faster alternative (although less stable): Diffusion denoising implicit model (DDIM) [Song et al. 2020]

How do we train and sample in diffusion models?

- **Key difficulty:** The true reverse PDF $q(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t)$ is unknown.
- **Key insight:** An approximate reverse transition depends on the score $\nabla \log p_t(\boldsymbol{x}_t)$ through Tweedie's formula, but it is unknown.
- **Training: Score Matching** to approximate the score [Vincent 2011]

$$\min_{\boldsymbol{\theta}} \mathbb{E}_{\boldsymbol{x}_t|\boldsymbol{x}_0, \boldsymbol{x}_0 \sim p^{\text{data}}, t} \left[\left\| \boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t) - \nabla \log p_t(\boldsymbol{x}_t) \right\|_2^2 \right]$$

$$\Leftrightarrow \quad \min_{\boldsymbol{\theta}} \mathbb{E}_{\boldsymbol{x}_t|\boldsymbol{x}_0, \boldsymbol{x}_0 \sim p^{\text{data}}, t} \left[\gamma_t \left\| \hat{\boldsymbol{x}}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t) - \boldsymbol{x}_0 \right\|_2^2 \right]$$

where $\hat{\boldsymbol{x}}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t) \approx \mathbb{E}[\boldsymbol{x}_0 \mid \boldsymbol{x}_t]$ is a denoiser.

- **Result:** Use the learned score $\boldsymbol{s}_{\boldsymbol{\theta}}$ to run the reverse sampling process.

Reverse diffusion: DDPM [Ho et al. 2020]

Require: $T, (\alpha_t)_{t=1}^T, (\sigma_t)_{t=1}^T$
  $\boldsymbol{x}_T \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$
  for $t = T, \dots, 1$ **do**
    $\hat{\boldsymbol{x}}_{\boldsymbol{\theta}} \leftarrow \frac{\boldsymbol{x}_t + (1-\bar{\alpha}_t)\boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)}{\sqrt{\bar{\alpha}_t}}$ ▷ *Tweedie's estimate*
    $\boldsymbol{\mu}_t = \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})\boldsymbol{x}_t + \sqrt{\bar{\alpha}_{t-1}}(1-\alpha_t)\hat{\boldsymbol{x}}_{\boldsymbol{\theta}}}{1-\bar{\alpha}_t}$
    $\boldsymbol{x}_{t-1} \sim \mathcal{N}(\boldsymbol{\mu}_t, \sigma_t^2 \boldsymbol{I})$
  return $\boldsymbol{x}_0$

Faster alternative (although less stable): Diffusion denoising implicit model (DDIM) [Song et al. 2020]

Diffusion Posterior Sampling (DPS) [Chung, Kim, et al. 2022]

- We would like to sample $\boldsymbol{x}$ from a measurement $\boldsymbol{y}$ such that

$$\boldsymbol{x} \mid \boldsymbol{y} \sim p(\boldsymbol{x} \mid \boldsymbol{y})$$

- Conditional score given the data $\boldsymbol{y}$:

$$\nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t \mid \boldsymbol{y}) = \underbrace{\nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t)}_{\approx \boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{x}_t)} + \underbrace{\nabla_{\boldsymbol{x}_t} \log p(\boldsymbol{y} \mid \boldsymbol{x}_t)}_{\approx \nabla_{\boldsymbol{x}_t} \log p(\boldsymbol{y} \mid \boldsymbol{x} = \hat{\boldsymbol{x}}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)) \text{ (approx. of log-likelihood)}}$$

DPS [Chung, Kim, et al. 2022]

Require: $\boldsymbol{y}, T, (\alpha)_{t=1}^{T}, (\sigma_t)_{t=1}^{T}, \lambda, \ldots$

$\boldsymbol{x}_T \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$

for $t = T, \ldots, 1$ **do**

$$\hat{\boldsymbol{x}}_0 \leftarrow \frac{\boldsymbol{x}_t + (1 - \bar{\alpha}_t)\boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{x}_t)}{\sqrt{\bar{\alpha}_t}} \quad \triangleright \textit{Tweedie's estimate}$$

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

$$\boldsymbol{x}'_{t-1} \sim \mathcal{N}(\boldsymbol{\mu}_t, \sigma_t^2 \boldsymbol{I}) \quad \triangleright \textit{Sampling}$$

$$\boldsymbol{x}_{t-1} = \boldsymbol{x}'_{t-1} + \lambda \cdot \nabla_{\boldsymbol{x}_t} \log p(\boldsymbol{y} \mid \hat{\boldsymbol{x}}_0(\boldsymbol{x}_t)) \quad \triangleright \textit{Data fid. gradient ascent}$$

return $\boldsymbol{x}_0$

Similar to iterative reconstruction

Diffusion Posterior Sampling (DPS) [Chung, Kim, et al. 2022]

- We would like to sample $\boldsymbol{x}$ from a measurement $\boldsymbol{y}$ such that

$$\boldsymbol{x} \mid \boldsymbol{y} \sim p(\boldsymbol{x} \mid \boldsymbol{y})$$

- Conditional score given the data $\boldsymbol{y}$:

$$\nabla_{x_t} \log p_t(x_t \mid \boldsymbol{y}) = \underbrace{\nabla_{x_t} \log p_t(x_t)}_{\approx s_\theta(x_t)} + \underbrace{\nabla_{x_t} \log p(\boldsymbol{y} \mid x_t)}_{\approx \nabla_{x_t} \log p(\boldsymbol{y} \mid x = \hat{x}_\theta(x_t, t)) \text{ (approx. of log-likelihood)}}$$

DPS [Chung, Kim, et al. 2022]

Require: $\boldsymbol{y}, T, (\alpha)_{t=1}^T, (\sigma_t)_{t=1}^T, \lambda, \ldots$

$x_T \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$

for $t = T, \ldots, 1$ **do**

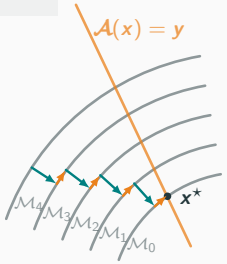
$$\hat{x}_0 \leftarrow \frac{x_t + (1 - \bar{\alpha}_t) s_\theta(x_t)}{\sqrt{\bar{\alpha}_t}} \quad \triangleright \textit{ Tweedie's estimate}$$

$$\boldsymbol{\mu}_t = \frac{\sqrt{\alpha_t}(1 - \bar{\alpha}_{t-1}) x_t + \sqrt{\bar{\alpha}_{t-1}}(1 - \alpha_t)\hat{x}_0}{1 - \bar{\alpha}_t}$$

$$x'_{t-1} \sim \mathcal{N}\left(\boldsymbol{\mu}_t, \sigma_t^2 \boldsymbol{I}\right) \quad \triangleright \textit{ Sampling}$$

$$x_{t-1} = x'_{t-1} + \lambda \cdot \nabla_{x_t} \log p(\boldsymbol{y} \mid \hat{x}_0(x_t)) \quad \triangleright \textit{ Data fid. gradient ascent}$$

return x_0



Similar to iterative reconstruction

DPS with Wavelet Decomposition [Friedrich et al. 2024]

- DDPM and DDIM in the 3-D image space $\mathcal{X} = \mathbb{R}^{n_x \times n_y \times n_z}$ are computationally expensive
- **Solution 1:** use 2-D slices with z-axis consistency [Chung, Ryu, et al. 2023] or multiple perpendicular 2-D slices [Webber et al. 2025].
- **Solution 2:** pretrained Autoencoders $(\boldsymbol{E}_{\psi}, \boldsymbol{D}_{\theta})$ [Rombach et al. 2022; Esser et al. 2024]

$$\mathcal{X} \xrightarrow{\boldsymbol{E}_{\psi}} \mathcal{Z} \xrightarrow{\boldsymbol{D}_{\theta}} \mathcal{X} \qquad \text{with } \dim(\mathcal{Z}) \ll \dim(\mathcal{X})$$

Diffusion in the latent space, i.e., performed with $\boldsymbol{z}_t \in \mathcal{Z}$ and the final output is $\boldsymbol{D}_{\theta}(\boldsymbol{z}_t)$.

- Alternatively, we can deploy a wavelet transform [Friedrich et al. 2024]:

$$\boldsymbol{\mathcal{V}} \colon \mathcal{X} \to \mathcal{Z} = \mathbb{R}^{8 \times \frac{n_x}{2} \times \frac{n_y}{2} \times \frac{n_z}{2}}$$

- Smaller channels → **parallelism and fewer memory**

Wavelet DPS

Require: $T, (\alpha_t)_{t=1}^{T}$
  $\boldsymbol{z}_N \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$
  for $t = T, \dots, 1$ **do**
    $\hat{\boldsymbol{z}}_0 \leftarrow \dfrac{\boldsymbol{z}_t + (1 - \bar{\alpha}_t)\boldsymbol{s}_{\theta}(\boldsymbol{z}_t)}{\sqrt{\bar{\alpha}_t}}$
    $\boldsymbol{\mu}_t = \dfrac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})\boldsymbol{z}_t + \sqrt{\bar{\alpha}_{t-1}}(1-\alpha_t)\hat{\boldsymbol{z}}_0}{1-\bar{\alpha}_t}$
    $\hat{\boldsymbol{x}}_0 \leftarrow \boldsymbol{\mathcal{V}}^{-1}(\hat{\boldsymbol{z}}_0)$
    $\tilde{\boldsymbol{z}}_{t-1} \sim \mathcal{N}\left(\boldsymbol{\mu}_t, \sigma_t^2 \boldsymbol{I}\right)$
    $\boldsymbol{z}_{t-1} = \tilde{\boldsymbol{z}}_{t-1} + \lambda \cdot \nabla_{\boldsymbol{z}_t} \log p\big(\boldsymbol{y} \mid \hat{\boldsymbol{x}}_0(\boldsymbol{z}_t)\big)$
  return $\boldsymbol{\mathcal{V}}^{-1}(\boldsymbol{z}_0)$

DPS with Wavelet Decomposition [Friedrich et al. 2024]

- DDPM and DDIM in the 3-D image space $\mathcal{X} = \mathbb{R}^{n_x \times n_y \times n_z}$ are computationally expensive
- **Solution 1:** use 2-D slices with z-axis consistency [Chung, Ryu, et al. 2023] or multiple perpendicular 2-D slices [Webber et al. 2025].
- **Solution 2:** pretrained Autoencoders $(\boldsymbol{E}_\psi, \boldsymbol{D}_\theta)$ [Rombach et al. 2022; Esser et al. 2024]

$$\mathcal{X} \xrightarrow{\boldsymbol{E}_\psi} \mathcal{Z} \xrightarrow{\boldsymbol{D}_\theta} \mathcal{X} \qquad \text{with } \dim(\mathcal{Z}) \ll \dim(\mathcal{X})$$

Diffusion in the latent space, i.e., performed with $\boldsymbol{z}_t \in \mathcal{Z}$ and the final output is $\boldsymbol{D}_\theta(\boldsymbol{z}_t)$.

- Alternatively, we can deploy a wavelet transform [Friedrich et al. 2024]:

$$\boldsymbol{\mathcal{V}} \colon \mathcal{X} \to \mathcal{Z} = \mathbb{R}^{8 \times \frac{n_x}{2} \times \frac{n_y}{2} \times \frac{n_z}{2}}$$

- Smaller channels → **parallelism and fewer memory**

Wavelet DPS

Require: $T, (\alpha_t)_{t=1}^T$
  $\boldsymbol{z}_N \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$
  for $t = T, \dots, 1$ **do**
    $\hat{\boldsymbol{z}}_0 \leftarrow \dfrac{\boldsymbol{z}_t + (1 - \bar{\alpha}_t)\boldsymbol{s}_\theta(\boldsymbol{z}_t)}{\sqrt{\bar{\alpha}_t}}$
    $\boldsymbol{\mu}_t = \dfrac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})\boldsymbol{z}_t + \sqrt{\bar{\alpha}_{t-1}}(1-\alpha_t)\hat{\boldsymbol{z}}_0}{1-\bar{\alpha}_t}$
    $\hat{\boldsymbol{x}}_0 \leftarrow \boldsymbol{\mathcal{V}}^{-1}(\hat{\boldsymbol{z}}_0)$
    $\tilde{\boldsymbol{z}}_{t-1} \sim \mathcal{N}\left(\boldsymbol{\mu}_t, \sigma_t^2 \boldsymbol{I}\right)$
    $\boldsymbol{z}_{t-1} = \tilde{\boldsymbol{z}}_{t-1} + \lambda \cdot \nabla_{\boldsymbol{z}_t} \log p\big(\boldsymbol{y} \mid \hat{\boldsymbol{x}}_0(\boldsymbol{z}_t)\big)$
  return $\boldsymbol{\mathcal{V}}^{-1}(\boldsymbol{z}_0)$

DPS with Wavelet Decomposition [Friedrich et al. 2024]

- DDPM and DDIM in the 3-D image space $\mathcal{X} = \mathbb{R}^{n_x \times n_y \times n_z}$ are computationally expensive
- **Solution 1:** use 2-D slices with z-axis consistency [Chung, Ryu, et al. 2023] or multiple perpendicular 2-D slices [Webber et al. 2025].
- **Solution 2:** pretrained Autoencoders ($\boldsymbol{E_\psi}, \boldsymbol{D_\theta}$) [Rombach et al. 2022; Esser et al. 2024]

$$\mathcal{X} \xrightarrow{\boldsymbol{E_\psi}} \mathcal{Z} \xrightarrow{\boldsymbol{D_\theta}} \mathcal{X} \qquad \text{with } \dim(\mathcal{Z}) \ll \dim(\mathcal{X})$$

Diffusion in the latent space, i.e., performed with $\boldsymbol{z_t} \in \mathcal{Z}$ and the final output is $\boldsymbol{D_\theta}(\boldsymbol{z}_t)$.

- Alternatively, we can deploy a wavelet transform [Friedrich et al. 2024]:

$$\mathcal{V}\colon \mathcal{X} \to \mathcal{Z} = \mathbb{R}^{8 \times \frac{n_x}{2} \times \frac{n_y}{2} \times \frac{n_z}{2}}$$

- Smaller channels → **parallelism and fewer memory**

Wavelet DPS

Require: $T, (\alpha_t)_{t=1}^T$
$z_N \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$
for $t = T, \dots, 1$ **do**
$\hat{z}_0 \leftarrow \dfrac{z_t + (1-\bar{\alpha}_t) s_\theta(z_t)}{\sqrt{\bar{\alpha}_t}}$
$\boldsymbol{\mu}_t = \dfrac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1}) z_t + \sqrt{\bar{\alpha}_{t-1}}(1-\alpha_t)\hat{z}_0}{1-\bar{\alpha}_t}$
$\hat{x}_0 \leftarrow \boldsymbol{\mathcal{V}}^{-1}(\hat{z}_0)$
$\tilde{z}_{t-1} \sim \mathcal{N}\left(\boldsymbol{\mu}_t, \sigma_t^2 \boldsymbol{I}\right)$
$z_{t-1} = \tilde{z}_{t-1} + \lambda \cdot \nabla_{z_t} \log p\left(\boldsymbol{y} \mid \hat{x}_0(z_t)\right)$
return $\boldsymbol{\mathcal{V}}^{-1}(z_0)$

DPS with Wavelet Decomposition [Friedrich et al. 2024]

- DDPM and DDIM in the 3-D image space $\mathcal{X} = \mathbb{R}^{n_x \times n_y \times n_z}$ are computationally expensive
- **Solution 1:** use 2-D slices with z-axis consistency [Chung, Ryu, et al. 2023] or multiple perpendicular 2-D slices [Webber et al. 2025].
- **Solution 2:** pretrained Autoencoders $(\boldsymbol{E_\psi}, \boldsymbol{D_\theta})$ [Rombach et al. 2022; Esser et al. 2024]

$$\mathcal{X} \xrightarrow{\boldsymbol{E_\psi}} \mathcal{Z} \xrightarrow{\boldsymbol{D_\theta}} \mathcal{X} \qquad \text{with } \dim(\mathcal{Z}) \ll \dim(\mathcal{X})$$

Diffusion in the latent space, i.e., performed with $\boldsymbol{z}_t \in \mathcal{Z}$ and the final output is $\boldsymbol{D_\theta}(\boldsymbol{z}_t)$.

- Alternatively, we can deploy a wavelet transform [Friedrich et al. 2024]:

$$\boldsymbol{\mathcal{V}}\colon \mathcal{X} \to \mathcal{Z} = \mathbb{R}^{8 \times \frac{n_x}{2} \times \frac{n_y}{2} \times \frac{n_z}{2}}$$

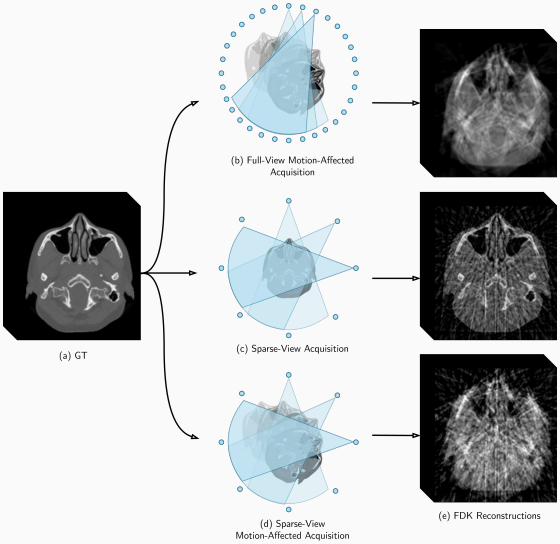
- Smaller channels → **parallelism and fewer memory**

Wavelet DPS

Require: $T, (\alpha_t)_{t=1}^T$
  $\boldsymbol{z}_N \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$
  for $t = T, \dots, 1$ **do**
    $\hat{\boldsymbol{z}}_0 \leftarrow \dfrac{\boldsymbol{z}_t + (1 - \bar{\alpha}_t)\boldsymbol{s_\theta}(\boldsymbol{z}_t)}{\sqrt{\bar{\alpha}_t}}$
    $\boldsymbol{\mu}_t = \dfrac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})\boldsymbol{z}_t + \sqrt{\bar{\alpha}_{t-1}}(1-\alpha_t)\hat{\boldsymbol{z}}_0}{1-\bar{\alpha}_t}$
    $\hat{\boldsymbol{x}}_0 \leftarrow \boldsymbol{\mathcal{V}}^{-1}(\hat{\boldsymbol{z}}_0)$
    $\tilde{\boldsymbol{z}}_{t-1} \sim \mathcal{N}\left(\boldsymbol{\mu}_t, \sigma_t^2 \boldsymbol{I}\right)$
    $\boldsymbol{z}_{t-1} = \tilde{\boldsymbol{z}}_{t-1} + \lambda \cdot \nabla_{\boldsymbol{z}_t} \log p\big(\boldsymbol{y} \mid \hat{\boldsymbol{x}}_0(\boldsymbol{z}_t)\big)$
  return $\boldsymbol{\mathcal{V}}^{-1}(\boldsymbol{z}_0)$

Motion-compensated Head CBCT

Motion in Cone-Beam (CB) CT



- Motion in CBCT causes blur if not accounted for.
- Motion must be estimated and corrected during reconstruction, but it requires a good initialisation.
- **Sparse-view CT makes it even more difficult**.

- $\mathcal{W}_{\boldsymbol{\phi}}: \mathcal{X} \to \mathcal{X}$ is the motion operator of parameter $\boldsymbol{\phi}$ (can be rigid or non-rigid).
- Standard JRM (blind inverse problem):

$$\min_{\boldsymbol{x},\boldsymbol{\phi}} \ell \underbrace{\left(\boldsymbol{y}, \boldsymbol{\mathcal{A}}(\mathcal{W}_{\boldsymbol{\phi}}\boldsymbol{x})\right)}_{L(\boldsymbol{x},\boldsymbol{\phi})} + R(\boldsymbol{x})$$

- Solved by alternating minimisation in $\boldsymbol{x}$ and $\boldsymbol{\phi}$.

$$\boldsymbol{x}^{(q+1)} = \arg\min_{\boldsymbol{x}} L\left(\boldsymbol{x}, \boldsymbol{\phi}^{(q)}\right) + R(\boldsymbol{x})$$

$$\boldsymbol{\phi}^{(q+1)} = \arg\min_{\boldsymbol{\phi}} L\left(\boldsymbol{x}^{(q+1)}, \boldsymbol{\phi}\right)$$

- Requires a good initialisation $\boldsymbol{x}^{\text{init}}$ but this is impossible in sparse-view motion-affected CBCT **without a strong prior**.

JRM-ADM (DDIM-based)

[De Paepe, Bousse, Phung-Ngoc, et al. 2026]

$s \leftarrow S$
$\boldsymbol{v}_s \leftarrow \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$
$\boldsymbol{\phi}_s \leftarrow \mathbf{0}$
while $s > 0$ **do**
  $\hat{\boldsymbol{v}}_0 \leftarrow \boldsymbol{v}_\theta(\boldsymbol{v}_s, s)$ ▷ Tweedie's estimate
  $\hat{\boldsymbol{x}}_0 \leftarrow \mathcal{V}^{-1}(\hat{\boldsymbol{v}}_0)$
  $\tilde{\boldsymbol{x}}_0 \leftarrow \arg\min_{\boldsymbol{x}} \ell\left(\boldsymbol{y}, \boldsymbol{\mathcal{A}}\left(\mathcal{W}_{\boldsymbol{\phi}_s}\boldsymbol{x}\right)\right) + \frac{\gamma_s}{2}\|\boldsymbol{x} - \hat{\boldsymbol{x}}_0\|_2^2$
  $\tilde{\boldsymbol{v}}_0 \leftarrow \mathcal{V}(\tilde{\boldsymbol{x}}_0)$
  $\boldsymbol{v}_{s-\delta s} = \sqrt{\bar{\alpha}_{s-\delta s}}\tilde{\boldsymbol{v}}_0 + \sqrt{1-\bar{\alpha}_{s-\delta s}} \cdot \frac{\boldsymbol{v}_s - \sqrt{\bar{\alpha}_s}\tilde{\boldsymbol{v}}_0}{\sqrt{1-\bar{\alpha}_s}}$
  $\boldsymbol{\phi}_{s-\delta s} = \arg\min_{\boldsymbol{\phi}} \ell\left(\boldsymbol{y}, \boldsymbol{\mathcal{A}}\left(\mathcal{W}_{\boldsymbol{\phi}}\tilde{\boldsymbol{x}}_0\right)\right)$
return $\boldsymbol{x} = \mathcal{V}^{-1}(\boldsymbol{v}_0)$

No scout reconstruction needed.

- $\boldsymbol{\mathcal{W}}_{\boldsymbol{\phi}}: \mathcal{X} \to \mathcal{X}$ is the motion operator of parameter $\boldsymbol{\phi}$ (can be rigid or non-rigid).
- Standard JRM (blind inverse problem):

$$\min_{\boldsymbol{x},\boldsymbol{\phi}} \underbrace{\ell\left(\boldsymbol{y}, \boldsymbol{\mathcal{A}}(\boldsymbol{\mathcal{W}}_{\boldsymbol{\phi}}\boldsymbol{x})\right)}_{L(\boldsymbol{x},\boldsymbol{\phi})} + R(\boldsymbol{x})$$

- Solved by alternating minimisation in $\boldsymbol{x}$ and $\boldsymbol{\phi}$.

$$\boldsymbol{x}^{(q+1)} = \arg\min_{\boldsymbol{x}} L\left(\boldsymbol{x}, \boldsymbol{\phi}^{(q)}\right) + R(\boldsymbol{x})$$

$$\boldsymbol{\phi}^{(q+1)} = \arg\min_{\boldsymbol{\phi}} L\left(\boldsymbol{x}^{(q+1)}, \boldsymbol{\phi}\right)$$

- Requires a good initialisation $\boldsymbol{x}^{\text{init}}$ but this is impossible in sparse-view motion-affected CBCT **without a strong prior**.

JRM-ADM (DDIM-based)

[De Paepe, Bousse, Phung-Ngoc, et al. 2026]

$\boldsymbol{s} \leftarrow \boldsymbol{S}$
$\boldsymbol{v}_s \leftarrow \boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$
$\boldsymbol{\phi}_s \leftarrow \boldsymbol{0}$
while $s > 0$ **do**
  $\hat{\boldsymbol{v}}_0 \leftarrow \boldsymbol{v}_{\boldsymbol{\theta}}(\boldsymbol{v}_s, s)$ ▷ Tweedie's estimate
  $\hat{\boldsymbol{x}}_0 \leftarrow \boldsymbol{\mathcal{V}}^{-1}(\hat{\boldsymbol{v}}_0)$
  $\tilde{\boldsymbol{x}}_0 \leftarrow \arg\min_{\boldsymbol{x}} \ell\left(\boldsymbol{y}, \boldsymbol{\mathcal{A}}\left(\boldsymbol{\mathcal{W}}_{\boldsymbol{\phi}_s}\boldsymbol{x}\right)\right) + \frac{\gamma_s}{2}\|\boldsymbol{x} - \hat{\boldsymbol{x}}_0\|_2^2$
  $\tilde{\boldsymbol{v}}_0 \leftarrow \boldsymbol{\mathcal{V}}(\tilde{\boldsymbol{x}}_0)$
  $\boldsymbol{v}_{s-\delta s} = \sqrt{\bar{\alpha}_{s-\delta s}}\tilde{\boldsymbol{v}}_0 + \sqrt{1-\bar{\alpha}_{s-\delta s}} \cdot \frac{\boldsymbol{v}_s - \sqrt{\bar{\alpha}_s}\tilde{\boldsymbol{v}}_0}{\sqrt{1-\bar{\alpha}_s}}$
  $\boldsymbol{\phi}_{s-\delta s} = \arg\min_{\boldsymbol{\phi}} \ell\left(\boldsymbol{y}, \boldsymbol{\mathcal{A}}\left(\boldsymbol{\mathcal{W}}_{\boldsymbol{\phi}}\tilde{\boldsymbol{x}}_0\right)\right)$
return $\boldsymbol{x} = \boldsymbol{\mathcal{V}}^{-1}(\boldsymbol{v}_0)$

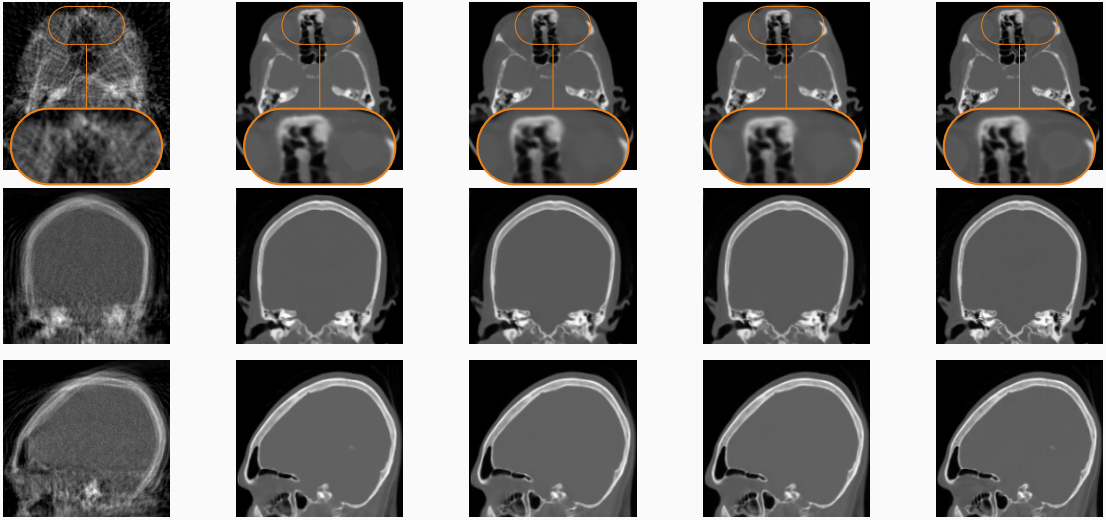
No scout reconstruction needed.

JRM-ADM in CBCT [De Paepe, Bousse, Phung-Ngoc, et al. 2026] (continued)

Experiment

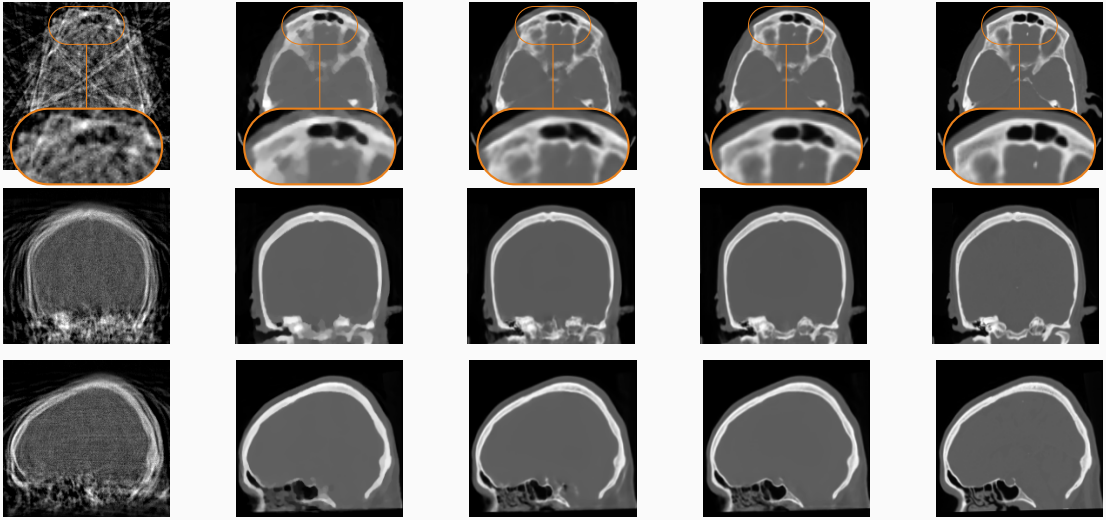
- 296 volumes (192×192×160) from CQ500 dataset [Chilamkurthy et al. 2018] (MDCT).
- The dataset includes fractured skulls.
- **Training:** 263; **validation:** 15; **testing:** 18 (simulated data).
- Simulated 60- and 20-view CBCT projection.

60-view reconstructions



FDK | JRM-TV | JRM-ADM$_{js}$ | JRM-ADM | GT

20-view reconstructions



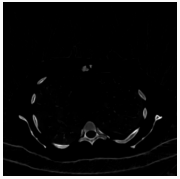
FDK | JRM-TV | JRM-ADM$_{js}$ | JRM-ADM | GT

PCCT Material Decomposition

DPS for PCCT material decomposition [Vazia et al. 2026]

- $\boldsymbol{x} = (\boldsymbol{x}_{\text{bones}}, \boldsymbol{x}_{\text{water}})$.
- Data: energy-binned photon counts $\boldsymbol{y} = \{\boldsymbol{y}_k\}_{k=1}^{n_e}$.
- Forward model:

$$\mathcal{A}_{i,k}(\boldsymbol{x}) = \int h_k(E) \cdot \mathrm{e}^{-[\boldsymbol{A}\boldsymbol{\mathcal{F}}(\boldsymbol{x},E)]_i} \, \mathrm{d}E$$



$\boldsymbol{x}_{\text{bones}}$



$\boldsymbol{x}_{\text{water}}$

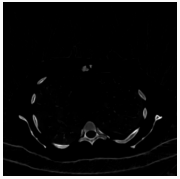
- Two synergistic DPS approaches
  - **Image-based Two-step DPS**: (i) synergistic DPS-based reconstruction of $\bar{\boldsymbol{\mu}} = \{\bar{\boldsymbol{\mu}}_k\}_{k=1}^{n_e}$ from $\boldsymbol{y}$ then $\boldsymbol{x} = \mathcal{F}_{\text{discr}}^{\dagger} \rightarrow$ inaccurate physics.
  - **One-step DPS**: direct synergistic estimation of $\boldsymbol{x} = (\boldsymbol{x}_{\text{bones}}, \boldsymbol{x}_{\text{water}})$ from $\boldsymbol{y}$ by DPS.
- Data simulated from Philips IQon Spectral CT images (512×512) (Poitiers University Hospital) (3 energy bins).
- **Training:** ≈9000; **testing:** ≈1000 (simulated data).

DPS for PCCT material decomposition [Vazia et al. 2026]

- $\boldsymbol{x} = (\boldsymbol{x}_{\text{bones}}, \boldsymbol{x}_{\text{water}})$.
- Data: energy-binned photon counts $\boldsymbol{y} = \{\boldsymbol{y}_k\}_{k=1}^{n_e}$.
- Forward model:

$$\mathcal{A}_{i,k}(\boldsymbol{x}) = \int h_k(E) \cdot \mathrm{e}^{-[\boldsymbol{A}\boldsymbol{\mathcal{F}}(\boldsymbol{x},E)]_i} \, \mathrm{d}E$$

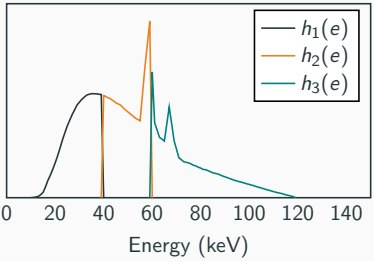
- Two synergistic DPS approaches
  - **Image-based Two-step DPS**: (i) synergistic DPS-based reconstruction of $\bar{\boldsymbol{\mu}} = \{\bar{\boldsymbol{\mu}}_k\}_{k=1}^{n_e}$ from $\boldsymbol{y}$ then $\boldsymbol{x} = \boldsymbol{\mathcal{F}}^{\dagger}_{\text{discr}}$ → inaccurate physics.
  - **One-step DPS**: direct synergistic estimation of $\boldsymbol{x} = (\boldsymbol{x}_{\text{bones}}, \boldsymbol{x}_{\text{water}})$ from $\boldsymbol{y}$ by DPS.
- Data simulated from Philips IQon Spectral CT images (512×512) (Poitiers University Hospital) (3 energy bins).
- **Training:** ≈9000; **testing:** ≈1000 (simulated data).



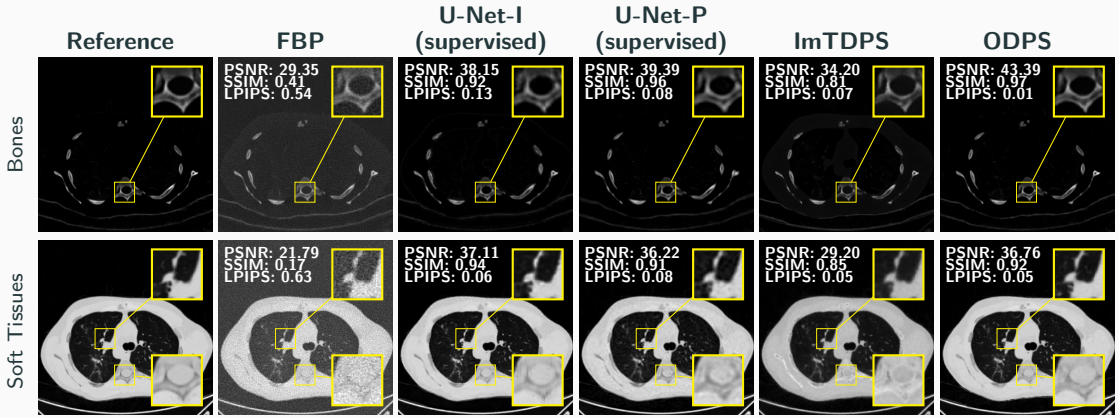
$\boldsymbol{x}_{\text{bones}}$



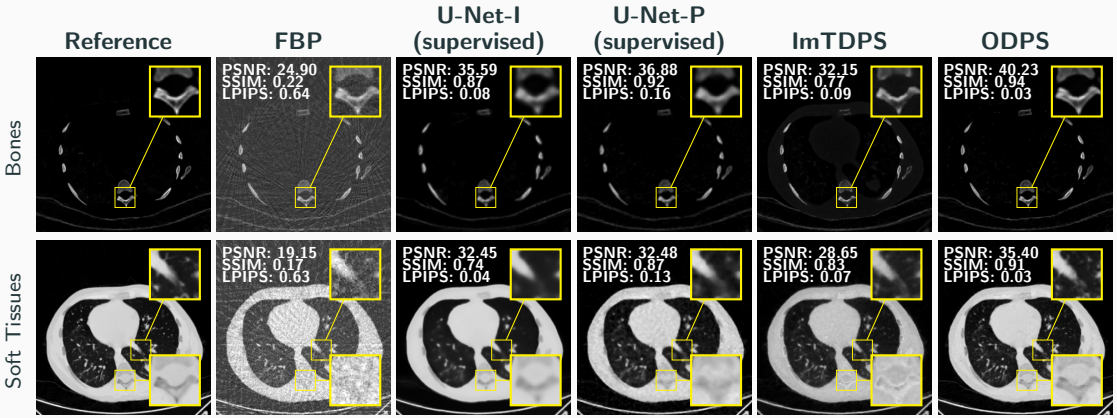
$\boldsymbol{x}_{\text{water}}$



DPS for PCCT material decomposition — low-count [Vazia et al. 2026]



DPS for PCCT material decomposition — sparse-view [Vazia et al. 2026]



Hybrid-DPS for PCCT with untrained material [Vazia et al. 2026]

- $\boldsymbol{x}$: water & bones **trained**.
- $\tilde{\boldsymbol{x}}$: iodine **untrained**.
- **Idea**: assume that $\boldsymbol{x}_t$ and $\tilde{\boldsymbol{x}}_t$ are independent, i.e.,

$$\nabla_{\boldsymbol{x}_t, \tilde{\boldsymbol{x}}_t} \log p_t(\boldsymbol{x}_t, \tilde{\boldsymbol{x}}_t) = \begin{bmatrix} \nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t) \\ \nabla_{\tilde{\boldsymbol{x}}_t} \log p_t(\tilde{\boldsymbol{x}}_t) \end{bmatrix}$$

- Approximation of the untrained score with a pseudo PDF:

$$p_t(\tilde{\boldsymbol{x}}_t) \propto \mathrm{e}^{-R(\tilde{\boldsymbol{x}}_t)}$$

where R is a variational regulariser (e.g., image gradient + Huber).

Hybrid-DPS

$x_N \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$
$\tilde{x}_N = \mathbf{0}$
for $t = T, \ldots, 1$ **do**
  $\hat{x}_0 \leftarrow \dfrac{x_t + (1 - \bar{\alpha}_t) s_\theta(x_t)}{\sqrt{\bar{\alpha}_t}}$
  $\boldsymbol{\mu}_t = \dfrac{\sqrt{\alpha_t}(1 - \bar{\alpha}_{t-1}) x_t + \sqrt{\bar{\alpha}_{t-1}}(1 - \alpha_t)\hat{x}_0}{1 - \bar{\alpha}_t}$
  $x'_{t-1} \sim \mathcal{N}(\boldsymbol{\mu}_t, \sigma_t^2 \boldsymbol{I})$
  $x_{t-1} = x'_{t-1} + \lambda \nabla_{x_t} \log p(\boldsymbol{y} \mid \hat{x}_0(x_t), \tilde{x}_t)$
  $\tilde{x}_{t-1} = \tilde{x}_{t-1} + \nabla_{\tilde{x}_t} \log p(\boldsymbol{y} \mid \hat{x}_0, \tilde{x}_t) - \xi_t \nabla R(\tilde{x}_t)$
return $(x, \tilde{x})$

Hybrid-DPS for PCCT with untrained material [Vazia et al. 2026]

- $\boldsymbol{x}$: water & bones **trained**.
- $\tilde{\boldsymbol{x}}$: iodine **untrained**.
- **Idea**: assume that $\boldsymbol{x}_t$ and $\tilde{\boldsymbol{x}}_t$ are independent, i.e.,

$$\nabla_{\boldsymbol{x}_t, \tilde{\boldsymbol{x}}_t} \log p_t(\boldsymbol{x}_t, \tilde{\boldsymbol{x}}_t) = \begin{bmatrix} \nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t) \\ \nabla_{\tilde{\boldsymbol{x}}_t} \log p_t(\tilde{\boldsymbol{x}}_t) \end{bmatrix}$$

- Approximation of the untrained score with a pseudo PDF:

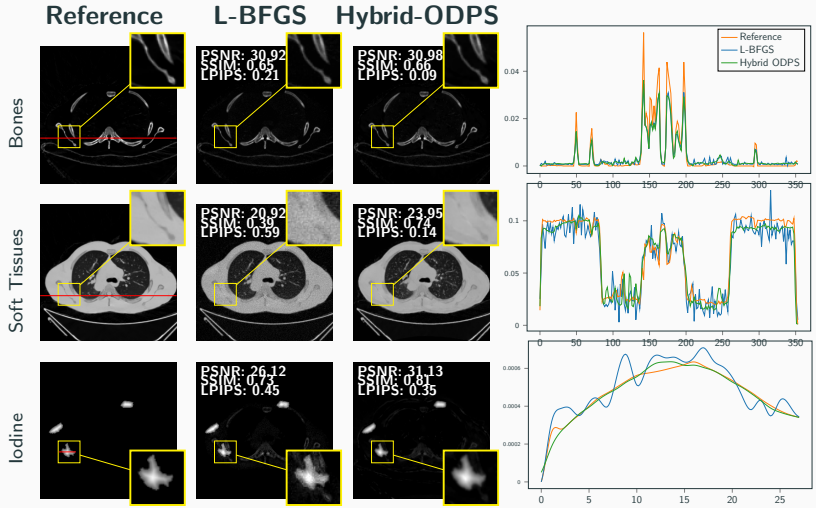
$$p_t(\tilde{\boldsymbol{x}}_t) \propto \mathrm{e}^{-R(\tilde{\boldsymbol{x}}_t)}$$

where R is a variational regulariser (e.g., image gradient + Huber).

Hybrid-DPS

$\boldsymbol{x}_N \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$
$\tilde{\boldsymbol{x}}_N = \boldsymbol{0}$
for $t = T, \ldots, 1$ **do**
  $\hat{\boldsymbol{x}}_0 \leftarrow \dfrac{\boldsymbol{x}_t + (1 - \bar{\alpha}_t)\boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{x}_t)}{\sqrt{\bar{\alpha}_t}}$
  $\boldsymbol{\mu}_t = \dfrac{\sqrt{\alpha_t}(1 - \bar{\alpha}_{t-1})\boldsymbol{x}_t + \sqrt{\bar{\alpha}_{t-1}}(1 - \alpha_t)\hat{\boldsymbol{x}}_0}{1 - \bar{\alpha}_t}$
  $\boldsymbol{x}'_{t-1} \sim \mathcal{N}(\boldsymbol{\mu}_t, \sigma_t^2 \boldsymbol{I})$
  $\boldsymbol{x}_{t-1} = \boldsymbol{x}'_{t-1} + \lambda \nabla_{\boldsymbol{x}_t} \log p(\boldsymbol{y} \mid \hat{\boldsymbol{x}}_0(\boldsymbol{x}_t), \tilde{\boldsymbol{x}}_t)$
  $\tilde{\boldsymbol{x}}_{t-1} = \tilde{\boldsymbol{x}}_{t-1} + \nabla_{\tilde{\boldsymbol{x}}_t} \log p(\boldsymbol{y} \mid \hat{\boldsymbol{x}}_0, \tilde{\boldsymbol{x}}_t) - \xi_t \nabla R(\tilde{\boldsymbol{x}}_t)$
return $(\boldsymbol{x}, \tilde{\boldsymbol{x}})$

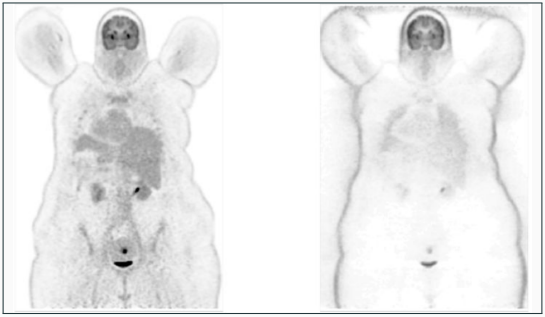
Hybrid-DPS for PCCT with untrained material [Vazia et al. 2026] (continued)



Less noise in soft-tissue and bone images ➔ less noise in the iodine image

Joint reconstruction of activity and attenuation in CT-less PET

PET Attenuation Correction (AC)



- AC is required for quantitative PET.
- AC can be performed with the help of a CT-derived μ-map but at the cost of higher patient dose.

JRAA-DPS [Phung-Ngoc et al. 2026]

- Images to reconstruct: $\boldsymbol{x} = (\boldsymbol{\lambda}, \boldsymbol{\mu})$
- Standard JRAA:

$$\min_{\boldsymbol{\lambda},\boldsymbol{\mu}} \underbrace{\ell\big(\boldsymbol{y}, \boldsymbol{\mathcal{A}}(\boldsymbol{\lambda}, \boldsymbol{\mu})\big)}_{L(\boldsymbol{\lambda},\boldsymbol{\mu})} + R(\boldsymbol{\lambda}, \boldsymbol{\mu})$$

- Solved by alternating minimization in $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$ [Rezaei et al. 2012]:

$$\boldsymbol{\lambda}^{(q+1)} = \arg\min_{\boldsymbol{\lambda}} L\left(\boldsymbol{\lambda}, \boldsymbol{\mu}^{(q)}\right) + R\left(\boldsymbol{\lambda}, \boldsymbol{\mu}^{(q)}\right)$$

$$\boldsymbol{\mu}^{(q+1)} = \arg\min_{\boldsymbol{\mu}} L\left(\boldsymbol{\lambda}^{(q+1)}, \boldsymbol{\mu}\right) + R\left(\boldsymbol{\lambda}^{(q+1)}, \boldsymbol{\mu}\right)$$

- **Severly ill-posed in absence of time-of-flight (TOF) information**.
- Scatter needs to be estimated alongside $(\boldsymbol{\lambda}, \boldsymbol{\mu})$
- Can we derive a DPS approach for JRAA?

JRAA-DPS [Phung-Ngoc et al. 2026]

$z_T \leftarrow (z_T^{\text{act}}, z_T^{\text{attn}}) \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}_{\mathcal{Z}\times\mathcal{Z}})$
$\boldsymbol{\phi}_T \leftarrow \boldsymbol{\phi}_{\text{init}}$ ▷ *scatter initialisation*
for $t = T \dots 1$ **do**
  $\hat{z}_0 \leftarrow \hat{z}_{\boldsymbol{\theta}}(z_t, t)$
  $\boldsymbol{\eta} \sim \mathcal{N}(\mathbf{0}_{\mathcal{Z}\times\mathcal{Z}}, \boldsymbol{I}_{\mathcal{Z}\times\mathcal{Z}})$
  $\tilde{z}_{t-1} \leftarrow \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t} z_t + \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t} \hat{z}_0 + \sigma_t \boldsymbol{\eta}$
  $\hat{x}_0 \leftarrow \boldsymbol{\mathcal{V}}^{-1}(\hat{z}_0)$
  $z_{t-1}^{\text{act}} \leftarrow \tilde{z}_{t-1}^{\text{act}} + \zeta_t \nabla_{z_t^{\text{act}}} \log p(\boldsymbol{y} \mid \boldsymbol{x} = \hat{x}_0(z_t), \boldsymbol{\phi}_t)$
  $z_{t-1}^{\text{attn}} \leftarrow \tilde{z}_{t-1}^{\text{attn}} + \xi_t \nabla_{z_t^{\text{attn}}} \log p(\boldsymbol{y} \mid \boldsymbol{x} = \hat{x}_0(z_t), \boldsymbol{\phi}_t)$
  $\boldsymbol{\phi}_{t-1} \leftarrow \boldsymbol{\phi}_t + \alpha_t \nabla_{\boldsymbol{\phi}_t} \log p(\boldsymbol{y} \mid \boldsymbol{x} = \hat{x}_0(z_t), \boldsymbol{\phi}_t) - \beta_t \nabla H(\boldsymbol{\phi}_t)$ ▷ *scatter estimate*
  $z_{t-1} \leftarrow \left(z_{t-1}^{\text{act}}, z_{t-1}^{\text{attn}}\right)$
return $\boldsymbol{\mathcal{V}}^{-1}(z_0)$

JRAA-DPS [Phung-Ngoc et al. 2026]

- Images to reconstruct: $\boldsymbol{x} = (\boldsymbol{\lambda}, \boldsymbol{\mu})$
- Standard JRAA:

$$\min_{\boldsymbol{\lambda},\boldsymbol{\mu}} \underbrace{\boldsymbol{\ell}\big(\boldsymbol{y}, \boldsymbol{\mathcal{A}}(\boldsymbol{\lambda}, \boldsymbol{\mu})\big)}_{L(\boldsymbol{\lambda},\boldsymbol{\mu})} + R(\boldsymbol{\lambda}, \boldsymbol{\mu})$$

- Solved by alternating minimization in $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$ [Rezaei et al. 2012]:

$$\boldsymbol{\lambda}^{(q+1)} = \arg\min_{\boldsymbol{\lambda}} L\left(\boldsymbol{\lambda}, \boldsymbol{\mu}^{(q)}\right) + R\left(\boldsymbol{\lambda}, \boldsymbol{\mu}^{(q)}\right)$$

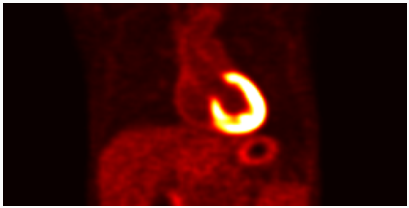
$$\boldsymbol{\mu}^{(q+1)} = \arg\min_{\boldsymbol{\mu}} L\left(\boldsymbol{\lambda}^{(q+1)}, \boldsymbol{\mu}\right) + R\left(\boldsymbol{\lambda}^{(q+1)}, \boldsymbol{\mu}\right)$$

- **Severly ill-posed in absence of time-of-flight (TOF) information**.
- Scatter needs to be estimated alongside $(\boldsymbol{\lambda}, \boldsymbol{\mu})$
- Can we derive a DPS approach for JRAA?

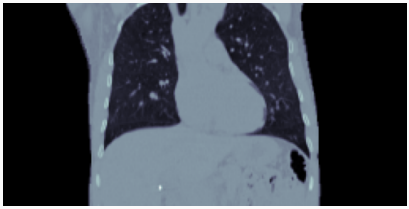
JRAA-DPS [Phung-Ngoc et al. 2026]

$\boldsymbol{z}_T \leftarrow (\boldsymbol{z}_T^{\text{act}}, \boldsymbol{z}_T^{\text{attn}}) \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_{\mathcal{Z}\times\mathcal{Z}})$
$\boldsymbol{\phi}_T \leftarrow \boldsymbol{\phi}_{\text{init}}$ ▷ *scatter initialisation*
for $t = T \dots 1$ **do**
  $\hat{\boldsymbol{z}}_0 \leftarrow \hat{\boldsymbol{z}}_{\boldsymbol{\theta}}(\boldsymbol{z}_t, t)$
  $\boldsymbol{\eta} \sim \mathcal{N}(\boldsymbol{0}_{\mathcal{Z}\times\mathcal{Z}}, \boldsymbol{I}_{\mathcal{Z}\times\mathcal{Z}})$
  $\tilde{\boldsymbol{z}}_{t-1} \leftarrow \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\boldsymbol{z}_t + \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}\hat{\boldsymbol{z}}_0 + \sigma_t\boldsymbol{\eta}$
  $\hat{\boldsymbol{x}}_0 \leftarrow \boldsymbol{\mathcal{V}}^{-1}(\hat{\boldsymbol{z}}_0)$
  $\boldsymbol{z}_{t-1}^{\text{act}} \leftarrow \tilde{\boldsymbol{z}}_{t-1}^{\text{act}} + \zeta_t \nabla_{\boldsymbol{z}_t^{\text{act}}} \log p(\boldsymbol{y} \mid \boldsymbol{x} = \hat{\boldsymbol{x}}_0(\boldsymbol{z}_t), \boldsymbol{\phi}_t)$
  $\boldsymbol{z}_{t-1}^{\text{attn}} \leftarrow \tilde{\boldsymbol{z}}_{t-1}^{\text{attn}} + \xi_t \nabla_{\boldsymbol{z}_t^{\text{attn}}} \log p(\boldsymbol{y} \mid \boldsymbol{x} = \hat{\boldsymbol{x}}_0(\boldsymbol{z}_t), \boldsymbol{\phi}_t)$
  $\boldsymbol{\phi}_{t-1} \leftarrow \boldsymbol{\phi}_t + \alpha_t \nabla_{\boldsymbol{\phi}_t} \log p(\boldsymbol{y} \mid \boldsymbol{x} = \hat{\boldsymbol{x}}_0(\boldsymbol{z}_t), \boldsymbol{\phi}_t) - \beta_t \nabla H(\boldsymbol{\phi}_t)$ ▷ scatter estimate
  $\boldsymbol{z}_{t-1} \leftarrow \left(\boldsymbol{z}_{t-1}^{\text{act}}, \boldsymbol{z}_{t-1}^{\text{attn}}\right)$
return $\boldsymbol{\mathcal{V}}^{-1}(\boldsymbol{z}_0)$

- Diffusion model trained on ($\boldsymbol{\lambda}$, $\boldsymbol{\mu}$) pairs synergistically.
- 1,168 FDG PET/CT-attenuation volume pairs (256×256×128) (Poitiers University Hospital).
- **Training:** 1,026; validation: 112; **testing:** 30 simulations (high-count and low-count) + 1 real raw data.

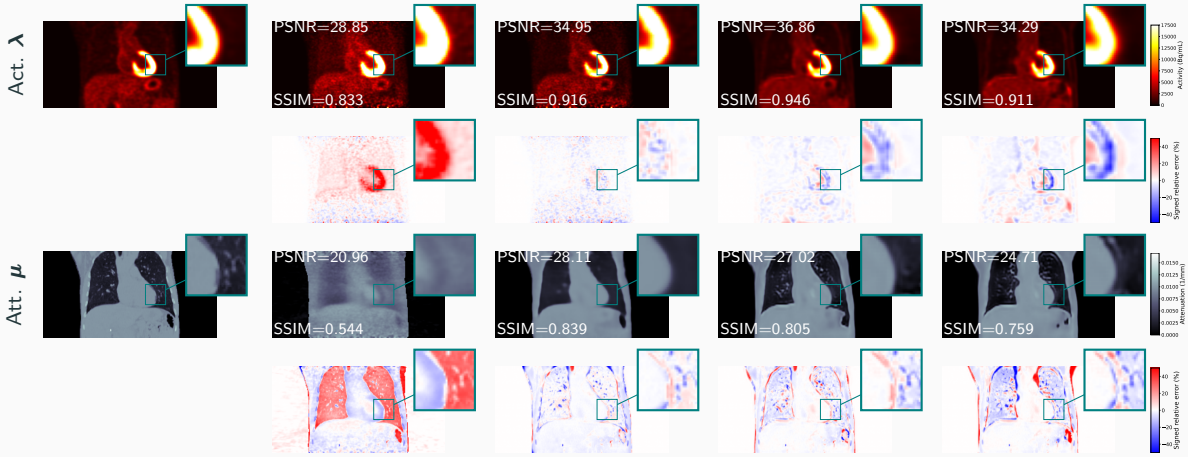


$\boldsymbol{\lambda}$

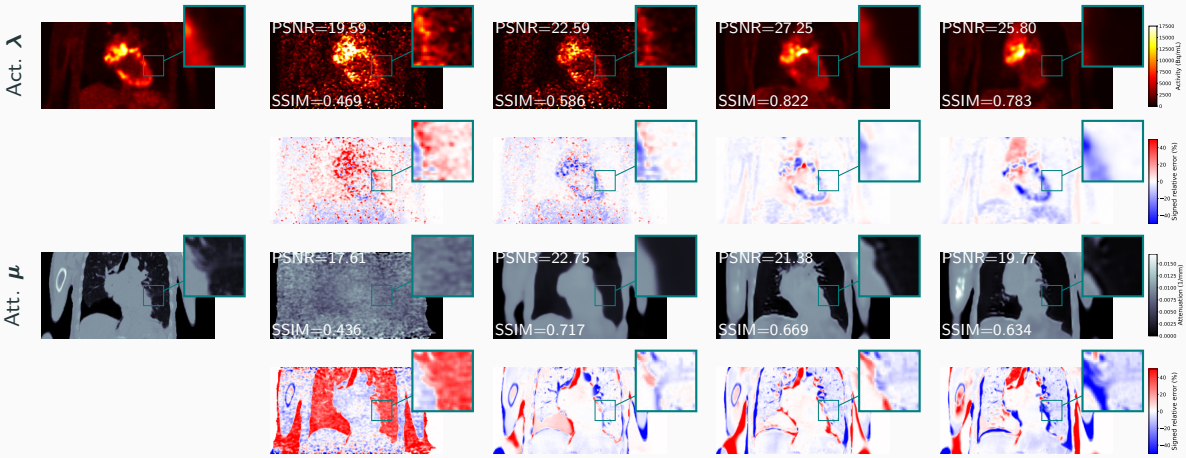


$\boldsymbol{\mu}$

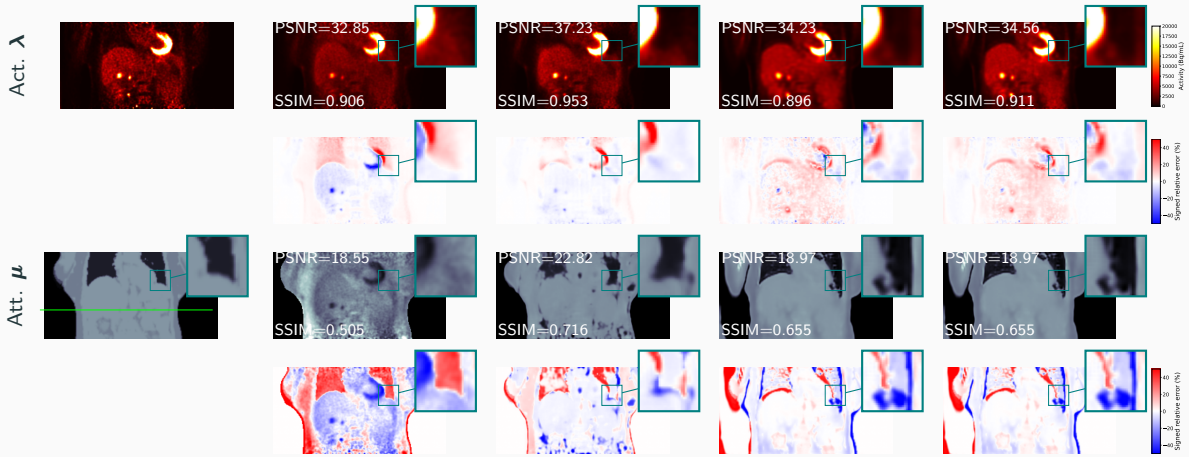
Results on Simulated Data [Phung-Ngoc et al. 2026]



Results on Simulated Data (1% counts) [Phung-Ngoc et al. 2026]



JRAA-DPS Results on Real Data (non-TOF) [Phung-Ngoc et al. 2026]



Conclusion

Conclusion

- Diffusion models enable **high-quality, stable reconstructions** across multiple tasks.
- Only clean images for training.
- Physics-informed sampling allows **data-consistent reconstruction**.
- Demonstrated on:
  - Motion-compensated head CBCT
  - PCCT material decomposition
  - Joint activity–attenuation PET reconstruction
- **Future work:**
  - Faster and cheaper diffusion methods for large volumes (e.g., total body PET) using new encoders [Mellak et al. 2026].
  - Multilevel algorithms for acceleration for score-based denoisers [De Paepe, Bousse, and Visvikis 2026].
  - Unsupervised methods for material decomposition and parametric imaging.

Acknowledgement: PhD students



Clémentine Phung-Ngoc



Antoine De Paepe



Thore Dassow



Corentin Vazia

De Paepe, A., A. Bousse, C. Phung-Ngoc, Y. Mellak, and D. Visvikis (2026). "Adaptive Diffusion Models for Sparse-View Motion-Corrected Head Cone-beam CT". *IEEE Transactions on Radiation and Plasma Medical Sciences* 10.5, 662–672. DOI: 10.1109/TRPMS.2025.3637124. URL: https://arxiv.org/abs/2504.14033.

Phung-Ngoc, C., A. Bousse, A. De Paepe, H.-P. Dang, O. Saut, and D. Visvikis (2026). "Joint Reconstruction of Activity and Attenuation in PET by Diffusion Posterior Sampling in Wavelet Coefficient Space". *IEEE Transactions on Radiation and Plasma Medical Sciences*. DOI: 10.1109/TRPMS.2026.3706239. URL: https://arxiv.org/abs/2505.18782.

Vazia, C., T. Dassow, A. Bousse, J. Froment, B. Vedel, F. Vermet, A. Perelli, J.-P. Tasu, and D. Visvikis (2026). "Material Decomposition in Photon-Counting Computed Tomography with Diffusion Models: Comparative Study and Hybridization with Variational Regularizers". *IEEE Transactions on Radiation and Plasma Medical Sciences*. DOI: 10.1109/TRPMS.2026.3651354. URL: https://arxiv.org/abs/2503.15383.

Acknowledgement: colleagues

Groupe reconstruction, LaTIM

- Thibaut Merlin
- Baptiste Laurent
- Zhihan Wang
- Noel Pinton
- Youness Mellak
- Hong-Phuong Dang
- Dimitris Visvikis

CEA

- Claude Comtat
- Florent Surreau
- Marion Savanier

Créatis

- Bruno Sixou
- Voichita Maxim
- Valentin Gautier

LMBA

- Jacques Froment
- Béatrice Vedel
- Frank Vermet

And also

- Matias Courdurier Bettancourt (PUC)
- Alessandro Perelli (Univ. Glasgow)
- Simon Stute (Centrale Nantes)
- Se Young Chun (SNU)
- Julian Tachella (ENS Lyon)

References i

Chilamkurthy, S., R. Ghosh, S. Tanamala, M. Biviji, N. G. Campeau, V. K. Venugopal, V. Mahajan, P. Rao, and P. Warier (2018). "Deep learning algorithms for detection of critical findings in head CT scans: a retrospective study". *The Lancet* 392.10162, 2388–2396.

Chung, H., J. Kim, M. T. Mccann, M. L. Klasky, and J. C. Ye (2022). "Diffusion posterior sampling for general noisy inverse problems". *arXiv preprint arXiv:2209.14687*.

Chung, H., D. Ryu, M. T. McCann, M. L. Klasky, and J. C. Ye (2023). "Solving 3D inverse problems using pre-trained 2D diffusion models". *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 22542–22551.

De Paepe, A., A. Bousse, C. Phung-Ngoc, Y. Mellak, and D. Visvikis (2026). "Adaptive Diffusion Models for Sparse-View Motion-Corrected Head Cone-beam CT". *IEEE Transactions on Radiation and Plasma Medical Sciences* 10.5, 662–672. DOI: 10.1109/TRPMS.2025.3637124. URL: https://arxiv.org/abs/2504.14033.

De Paepe, A., A. Bousse, and D. Visvikis (2026). "Multilevel Stochastic Plug-and-Play for Sparse-View CT Reconstruction". *arXiv preprint arXiv:2606.24567*. URL: https://arxiv.org/abs/2606.24567.

Duff, M., N. D. Campbell, and M. J. Ehrhardt (2021). "Regularising inverse problems with generative machine learning models". *arXiv preprint arXiv:2107.11191*.

Elbakri, I. A. and J. A. Fessler (2002). "Statistical image reconstruction for polyenergetic X-ray computed tomography". *IEEE transactions on medical imaging* 21.2, 89–99.

Esser, P., S. Kulal, A. Blattmann, R. Entezari, J. Müller, H. Saini, Y. Levi, D. Lorenz, A. Sauer, F. Boesel, et al. (2024). "Scaling rectified flow transformers for high-resolution image synthesis". *Forty-first international conference on machine learning*.

References ii

Friedrich, P., J. Wolleb, F. Bieder, A. Durrer, and P. C. Cattin (2024). "Wdm: 3d wavelet diffusion models for high-resolution medical image synthesis". *MICCAI workshop on deep generative models*. Springer, 11–21.

Ho, J., A. Jain, and P. Abbeel (2020). "Denoising diffusion probabilistic models". *Advances in neural information processing systems* 33, 6840–6851.

Jin, K. H., M. T. McCann, E. Froustey, and M. Unser (2017). "Deep convolutional neural network for inverse problems in imaging". *IEEE transactions on image processing* 26.9, 4509–4522.

Mellak, Y., A. De Paepe, D. Visvikis, and A. Bousse (2026). "Continuous 3-D Latent Diffusion for Medical Generation and Reconstruction". *arXiv preprint arXiv:2607.16491*. URL: https://arxiv.org/abs/2607.16491.

Monga, V., Y. Li, and Y. C. Eldar (2021). "Algorithm unrolling: Interpretable, efficient deep learning for signal and image processing". *IEEE Signal Processing Magazine* 38.2, 18–44.

Phung-Ngoc, C., A. Bousse, A. De Paepe, H.-P. Dang, O. Saut, and D. Visvikis (2026). "Joint Reconstruction of Activity and Attenuation in PET by Diffusion Posterior Sampling in Wavelet Coefficient Space". *IEEE Transactions on Radiation and Plasma Medical Sciences*. DOI: 10.1109/TRPMS.2026.3706239. URL: https://arxiv.org/abs/2505.18782.

Rezaei, A., M. Defrise, G. Bal, C. Michel, M. Conti, C. Watson, and J. Nuyts (2012). "Simultaneous reconstruction of activity and attenuation in time-of-flight PET". *IEEE transactions on medical imaging* 31.12, 2224–2233.

Rombach, R., A. Blattmann, D. Lorenz, P. Esser, and B. Ommer (2022). "High-resolution image synthesis with latent diffusion models". *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 10684–10695.

Song, J., C. Meng, and S. Ermon (2020). "Denoising diffusion implicit models". *arXiv preprint arXiv:2010.02502*.

References iii



Vazia, C., T. Dassow, A. Bousse, J. Froment, B. Vedel, F. Vermet, A. Perelli, J.-P. Tasu, and D. Visvikis (2026). "Material Decomposition in Photon-Counting Computed Tomography with Diffusion Models: Comparative Study and Hybridization with Variational Regularizers". *IEEE Transactions on Radiation and Plasma Medical Sciences*. DOI: 10.1109/TRPMS.2026.3651354. URL: https://arxiv.org/abs/2503.15383.



Vincent, P. (2011). "A connection between score matching and denoising autoencoders". *Neural computation* 23.7, 1661–1674.



Webber, G., Y. Mizuno, O. D. Howes, A. Hammers, A. P. King, and A. J. Reader (2025). "Likelihood-Scheduled Score-Based Generative Modeling for Fully 3D PET Image Reconstruction". *IEEE transactions on medical imaging*.



Thanks!

E-Mail : alexandre.bousse@univ-brest.fr

Web : https://abousse.github.io/

Appendix 1: diffusion models

Design of the diffusion model

- **Goal:** Starting from $\boldsymbol{x}_T \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$, we wish to sample $\boldsymbol{x}_{T-1}, \boldsymbol{x}_{T-2}, \ldots$ until $\boldsymbol{x}_0$.
- **Problem:** The reverse process $q(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t)$ is unknown
- **Solution:** $q(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t, \boldsymbol{x}_0)$ is known by the Bayes' Formula:

$$q(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t, \boldsymbol{x}_0) = \mathcal{N}\left(\boldsymbol{x}_{t-1} \mid \boldsymbol{\mu}(\boldsymbol{x}_t, \boldsymbol{x}_0, t), \sigma_t^2 \boldsymbol{I}\right),$$

  for some $\boldsymbol{\mu}$.
- **Problem:** $\boldsymbol{x}_0$ is unknown.
- **Solution:** $\boldsymbol{x}_0$ can be approximated by $\mathbb{E}[\boldsymbol{x}_0 \mid \boldsymbol{x}_t]$, known from Tweedie's Formula:

$$\mathbb{E}[\boldsymbol{x}_0 \mid \boldsymbol{x}_t] = \frac{\boldsymbol{x}_t + (1 - \bar{\alpha}_t) \nabla \log p_t(\boldsymbol{x}_t)}{\sqrt{\bar{\alpha}_t}}$$

  where $p_t(\boldsymbol{x}_t)$ is the PDF of $\boldsymbol{x}_t$ and $\nabla \log p_t(\boldsymbol{x}_t)$ is the score.

Design of the diffusion model

- **Goal:** Starting from $\boldsymbol{x}_T \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$, we wish to sample $\boldsymbol{x}_{T-1}, \boldsymbol{x}_{T-2}, \ldots$ until $\boldsymbol{x}_0$.
- **Problem:** The reverse process $q(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t)$ is unknown
- **Solution:** $q(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t, \boldsymbol{x}_0)$ is known by the Bayes' Formula:

$$q(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t, \boldsymbol{x}_0) = \mathcal{N}\left(\boldsymbol{x}_{t-1} \mid \boldsymbol{\mu}(\boldsymbol{x}_t, \boldsymbol{x}_0, t), \sigma_t^2 \boldsymbol{I}\right),$$

  for some $\boldsymbol{\mu}$.
- **Problem:** $\boldsymbol{x}_0$ is unknown.
- **Solution:** $\boldsymbol{x}_0$ can be approximated by $\mathbb{E}[\boldsymbol{x}_0 \mid \boldsymbol{x}_t]$, known from Tweedie's Formula:

$$\mathbb{E}[\boldsymbol{x}_0 \mid \boldsymbol{x}_t] = \frac{\boldsymbol{x}_t + (1 - \bar{\alpha}_t) \nabla \log p_t(\boldsymbol{x}_t)}{\sqrt{\bar{\alpha}_t}}$$

  where $p_t(\boldsymbol{x}_t)$ is the PDF of $\boldsymbol{x}_t$ and $\nabla \log p_t(\boldsymbol{x}_t)$ is the score.

Design of the diffusion model

- **Goal:** Starting from $\boldsymbol{x}_T \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$, we wish to sample $\boldsymbol{x}_{T-1}, \boldsymbol{x}_{T-2}, \ldots$ until $\boldsymbol{x}_0$.
- **Problem:** The reverse process $q(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t)$ is unknown
- **Solution:** $q(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t, \boldsymbol{x}_0)$ is known by the Bayes' Formula:

$$q(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t, \boldsymbol{x}_0) = \mathcal{N}\Big(\boldsymbol{x}_{t-1} \;\Big|\; \boldsymbol{\mu}(\boldsymbol{x}_t, \boldsymbol{x}_0, t), \sigma_t^2 \boldsymbol{I}\Big),$$

for some $\boldsymbol{\mu}$.

- **Problem:** $\boldsymbol{x}_0$ is unknown.
- **Solution:** $\boldsymbol{x}_0$ can be approximated by $\mathbb{E}[\boldsymbol{x}_0 \mid \boldsymbol{x}_t]$, known from Tweedie's Formula:

$$\mathbb{E}[\boldsymbol{x}_0 \mid \boldsymbol{x}_t] = \frac{\boldsymbol{x}_t + (1 - \bar{\alpha}_t) \nabla \log p_t(\boldsymbol{x}_t)}{\sqrt{\bar{\alpha}_t}}$$

where $p_t(\boldsymbol{x}_t)$ is the PDF of $\boldsymbol{x}_t$ and $\nabla \log p_t(\boldsymbol{x}_t)$ is the score.

Design of the diffusion model

- **Goal:** Starting from $\boldsymbol{x}_T \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$, we wish to sample $\boldsymbol{x}_{T-1}, \boldsymbol{x}_{T-2}, \ldots$ until $\boldsymbol{x}_0$.
- **Problem:** The reverse process $q(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t)$ is unknown
- **Solution:** $q(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t, \boldsymbol{x}_0)$ is known by the Bayes' Formula:

$$q(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t, \boldsymbol{x}_0) = \mathcal{N}\Big(\boldsymbol{x}_{t-1} \;\Big|\; \boldsymbol{\mu}(\boldsymbol{x}_t, \boldsymbol{x}_0, t), \sigma_t^2 \boldsymbol{I}\Big),$$

  for some $\boldsymbol{\mu}$.
- **Problem:** $\boldsymbol{x}_0$ is unknown.
- **Solution:** $\boldsymbol{x}_0$ can be approximated by $\mathbb{E}[\boldsymbol{x}_0 \mid \boldsymbol{x}_t]$, known from Tweedie's Formula:

$$\mathbb{E}[\boldsymbol{x}_0 \mid \boldsymbol{x}_t] = \frac{\boldsymbol{x}_t + (1 - \bar{\alpha}_t) \nabla \log p_t(\boldsymbol{x}_t)}{\sqrt{\bar{\alpha}_t}}$$

  where $p_t(\boldsymbol{x}_t)$ is the PDF of $\boldsymbol{x}_t$ and $\nabla \log p_t(\boldsymbol{x}_t)$ is the score.

Design of the diffusion model

- **Goal:** Starting from $\boldsymbol{x}_T \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$, we wish to sample $\boldsymbol{x}_{T-1}, \boldsymbol{x}_{T-2}, \ldots$ until $\boldsymbol{x}_0$.
- **Problem:** The reverse process $q(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t)$ is unknown
- **Solution:** $q(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t, \boldsymbol{x}_0)$ is known by the Bayes' Formula:

$$q(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t, \boldsymbol{x}_0) = \mathcal{N}\Big(\boldsymbol{x}_{t-1} \;\Big|\; \boldsymbol{\mu}(\boldsymbol{x}_t, \boldsymbol{x}_0, t), \sigma_t^2 \boldsymbol{I}\Big),$$

  for some $\boldsymbol{\mu}$.
- **Problem:** $\boldsymbol{x}_0$ is unknown.
- **Solution:** $\boldsymbol{x}_0$ can be approximated by $\mathbb{E}[\boldsymbol{x}_0 \mid \boldsymbol{x}_t]$, known from Tweedie's Formula:

$$\mathbb{E}[\boldsymbol{x}_0 \mid \boldsymbol{x}_t] = \frac{\boldsymbol{x}_t + (1 - \bar{\alpha}_t) \nabla \log p_t(\boldsymbol{x}_t)}{\sqrt{\bar{\alpha}_t}}$$

  where $p_t(\boldsymbol{x}_t)$ is the PDF of $\boldsymbol{x}_t$ and $\nabla \log p_t(\boldsymbol{x}_t)$ is the score.

Design of the diffusion model

- **Problem:** The score $\nabla \log p_t(\boldsymbol{x}_t)$ is unknown.
- **Solution:** It can be approximated by a NN $\boldsymbol{s}_{\boldsymbol{\theta}}$ trained by score matching [Vincent 2011]:

$$\min_{\boldsymbol{\theta}} \mathbb{E}_{\boldsymbol{x}_t|\boldsymbol{x}_0, \boldsymbol{x}_0 \sim p^{\text{data}}, t} \left[\left\| \boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t) - \nabla \log p_t(\boldsymbol{x}_t) \right\|_2^2 \right]$$

$$\Leftrightarrow \quad \min_{\boldsymbol{\theta}} \mathbb{E}_{\boldsymbol{x}_t|\boldsymbol{x}_0, \boldsymbol{x}_0 \sim p^{\text{data}}, t} \left[\gamma_t \left\| \hat{\boldsymbol{x}}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t) - \boldsymbol{x}_0 \right\|_2^2 \right]$$

  $\hat{\boldsymbol{x}}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t) \approx \mathbb{E}[\boldsymbol{x}_0 \mid \boldsymbol{x}_t]$ being a denoiser.
- **Inference:** $\boldsymbol{x}_{t-1}$ is sampled from $\boldsymbol{x}_t$ with $q_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t)$ defined as

$$q_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t) = \mathcal{N}\left(\boldsymbol{x}_{t-1} \mid \boldsymbol{\mu}\left(\boldsymbol{x}_t, \hat{\boldsymbol{x}}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t), t \right), \sigma_t^2 \boldsymbol{I} \right)$$

Design of the diffusion model

- **Problem:** The score $\nabla \log p_t(\boldsymbol{x}_t)$ is unknown.
- **Solution:** It can be approximated by a NN $\boldsymbol{s}_{\boldsymbol{\theta}}$ trained by score matching [Vincent 2011]:

$$\min_{\boldsymbol{\theta}} \mathbb{E}_{\boldsymbol{x}_t|\boldsymbol{x}_0, \boldsymbol{x}_0 \sim p^{\text{data}}, t} \left[\left\| \boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t) - \nabla \log p_t(\boldsymbol{x}_t) \right\|_2^2 \right]$$

$$\Leftrightarrow \quad \min_{\boldsymbol{\theta}} \mathbb{E}_{\boldsymbol{x}_t|\boldsymbol{x}_0, \boldsymbol{x}_0 \sim p^{\text{data}}, t} \left[\gamma_t \left\| \hat{\boldsymbol{x}}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t) - \boldsymbol{x}_0 \right\|_2^2 \right]$$

$\hat{\boldsymbol{x}}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t) \approx \mathbb{E}[\boldsymbol{x}_0 \mid \boldsymbol{x}_t]$ being a denoiser.

- **Inference:** $\boldsymbol{x}_{t-1}$ is sampled from $\boldsymbol{x}_t$ with $q_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t)$ defined as

$$q_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t) = \mathcal{N}\left(\boldsymbol{x}_{t-1} \mid \boldsymbol{\mu}\left(\boldsymbol{x}_t, \hat{\boldsymbol{x}}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t), t\right), \sigma_t^2 \boldsymbol{I}\right)$$

Design of the diffusion model

- **Problem:** The score $\nabla \log p_t(\boldsymbol{x}_t)$ is unknown.
- **Solution:** It can be approximated by a NN $\boldsymbol{s}_{\boldsymbol{\theta}}$ trained by score matching [Vincent 2011]:

$$\min_{\boldsymbol{\theta}} \mathbb{E}_{\boldsymbol{x}_t|\boldsymbol{x}_0, \boldsymbol{x}_0 \sim p^{\text{data}}, t} \left[\left\| \boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t) - \nabla \log p_t(\boldsymbol{x}_t) \right\|_2^2 \right]$$

$$\Leftrightarrow \quad \min_{\boldsymbol{\theta}} \mathbb{E}_{\boldsymbol{x}_t|\boldsymbol{x}_0, \boldsymbol{x}_0 \sim p^{\text{data}}, t} \left[\gamma_t \left\| \hat{\boldsymbol{x}}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t) - \boldsymbol{x}_0 \right\|_2^2 \right]$$

  $\hat{\boldsymbol{x}}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t) \approx \mathbb{E}[\boldsymbol{x}_0 \mid \boldsymbol{x}_t]$ being a denoiser.
- **Inference:** $\boldsymbol{x}_{t-1}$ is sampled from $\boldsymbol{x}_t$ with $q_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t)$ defined as

$$q_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t) = \mathcal{N}\Big(\boldsymbol{x}_{t-1} \;\Big|\; \boldsymbol{\mu}\big(\boldsymbol{x}_t, \hat{\boldsymbol{x}}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t), t\big), \sigma_t^2 \boldsymbol{I}\Big)$$

Denoising Diffusion probabilistic Model (DDPM) [Ho et al. 2020]

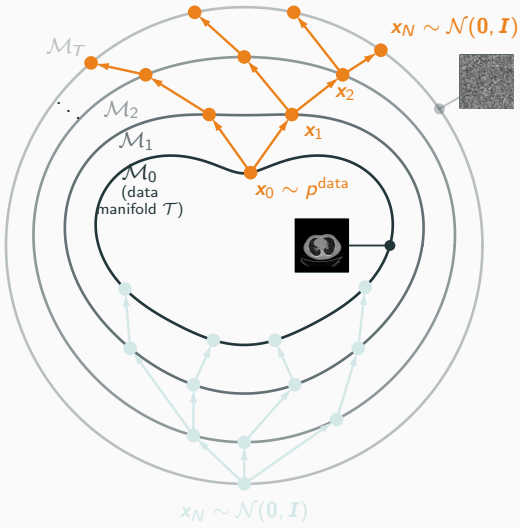
Forward diffusion

Require: $T, (\alpha_t)_{t=1}^T$
  $\boldsymbol{x}_0 \sim p^{\text{data}}$
  for $t = 1, \dots, T$ **do**
    $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$
    $\boldsymbol{x}_t = \sqrt{\alpha_t}\boldsymbol{x}_{t-1} + \sqrt{1-\alpha_t}\boldsymbol{\epsilon}$

Reverse diffusion: DDPM [Song et al. 2020]

Require: $T, (\alpha_t)_{t=1}^T, (\sigma_t)_{t=1}^T$
  $\boldsymbol{x}_T \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$
  for $t = T, \dots, 1$ **do**
    $\hat{\boldsymbol{x}}_\theta \leftarrow \dfrac{\boldsymbol{x}_t + (1-\bar{\alpha}_t)\boldsymbol{s}_\theta(\boldsymbol{x}_t, t)}{\sqrt{\bar{\alpha}_t}}$ ▷ *Tweedie's estimate*
    $\boldsymbol{\mu}_t = \dfrac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})\boldsymbol{x}_t + \sqrt{\bar{\alpha}_{t-1}}(1-\alpha_t)\hat{\boldsymbol{x}}_\theta}{1-\bar{\alpha}_t}$
    $\boldsymbol{x}_{t-1} \sim \mathcal{N}\left(\boldsymbol{\mu}_t, \sigma_t^2 \boldsymbol{I}\right)$
  return $\boldsymbol{x}_0$

Faster alternative (although less stable): Diffusion denoising implicit model (DDIM) [Song et al. 2020].



Denoising Diffusion probabilistic Model (DDPM) [Ho et al. 2020]

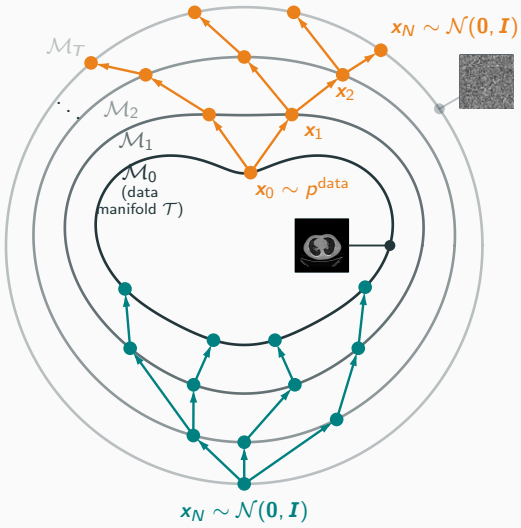
Forward diffusion

Require: $T, (\alpha_t)_{t=1}^T$
  $\boldsymbol{x}_0 \sim p^{\text{data}}$
  for $t = 1, \dots, T$ **do**
    $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$
    $\boldsymbol{x}_t = \sqrt{\alpha_t}\boldsymbol{x}_{t-1} + \sqrt{1-\alpha_t}\boldsymbol{\epsilon}$

Reverse diffusion: DDPM [Song et al. 2020]

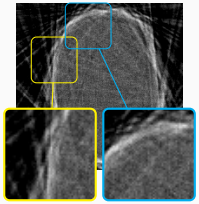
Require: $T, (\alpha_t)_{t=1}^T, (\sigma_t)_{t=1}^T$
  $\boldsymbol{x}_T \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$
  for $t = T, \dots, 1$ **do**
    $\hat{\boldsymbol{x}}_{\boldsymbol{\theta}} \leftarrow \dfrac{\boldsymbol{x}_t + (1-\bar{\alpha}_t)\boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)}{\sqrt{\bar{\alpha}_t}}$ ▷ *Tweedie's estimate*
    $\boldsymbol{\mu}_t = \dfrac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})\boldsymbol{x}_t + \sqrt{\bar{\alpha}_{t-1}}(1-\alpha_t)\hat{\boldsymbol{x}}_{\boldsymbol{\theta}}}{1-\bar{\alpha}_t}$
    $\boldsymbol{x}_{t-1} \sim \mathcal{N}\left(\boldsymbol{\mu}_t, \sigma_t^2 \boldsymbol{I}\right)$
  return $\boldsymbol{x}_0$

Faster alternative (although less stable): Diffusion denoising implicit model (DDIM) [Song et al. 2020].

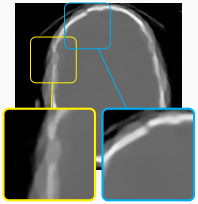


Appendix 2: additional results & quantitative analysis

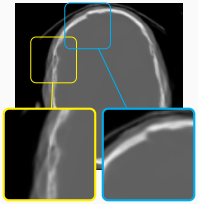
Fractured skull



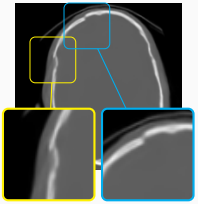
FDK



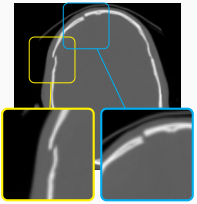
JRM-TV



JRM-ADM$_{js}$



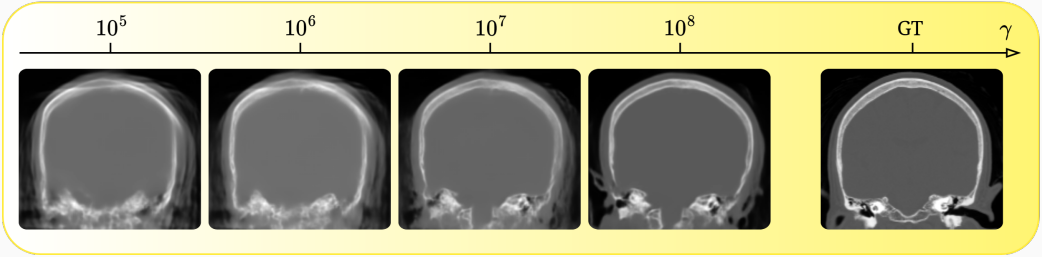
JRM-ADM



GT

Ablation study: can diffusion alone defeat motion blur?

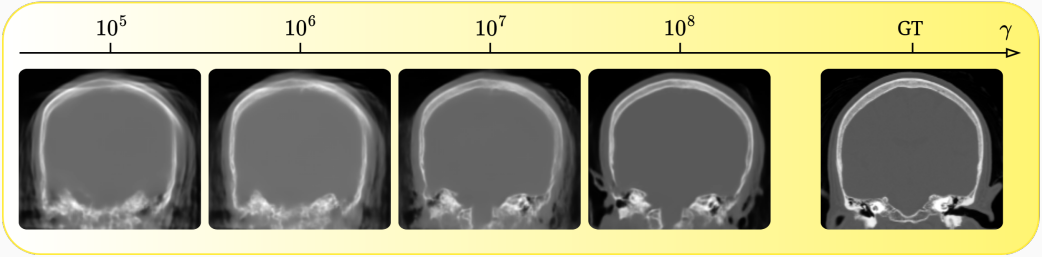
$\mathcal{W}_{\phi} = \mathrm{Id}$:



Answer: no.

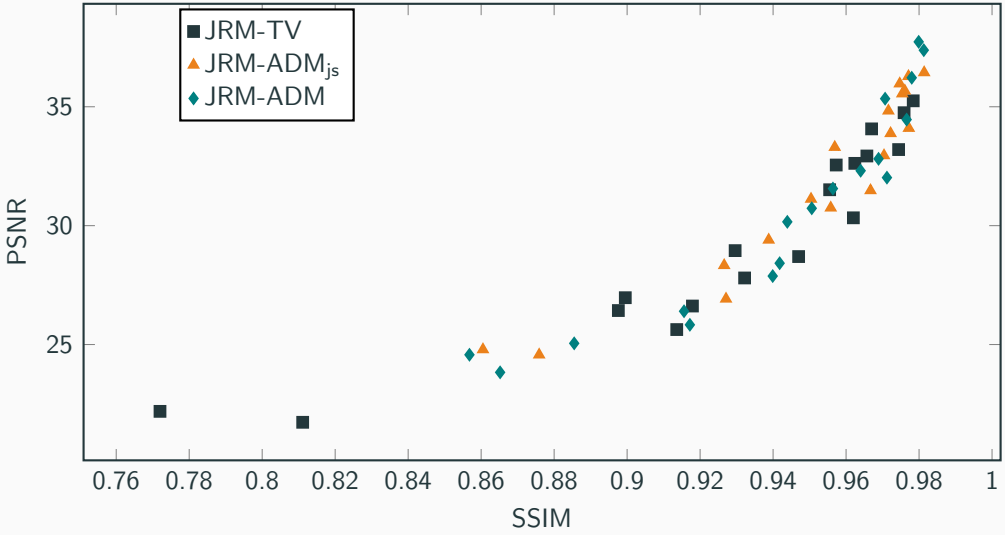
Ablation study: can diffusion alone defeat motion blur?

$\mathcal{W}_{\phi} = \mathrm{Id}$:

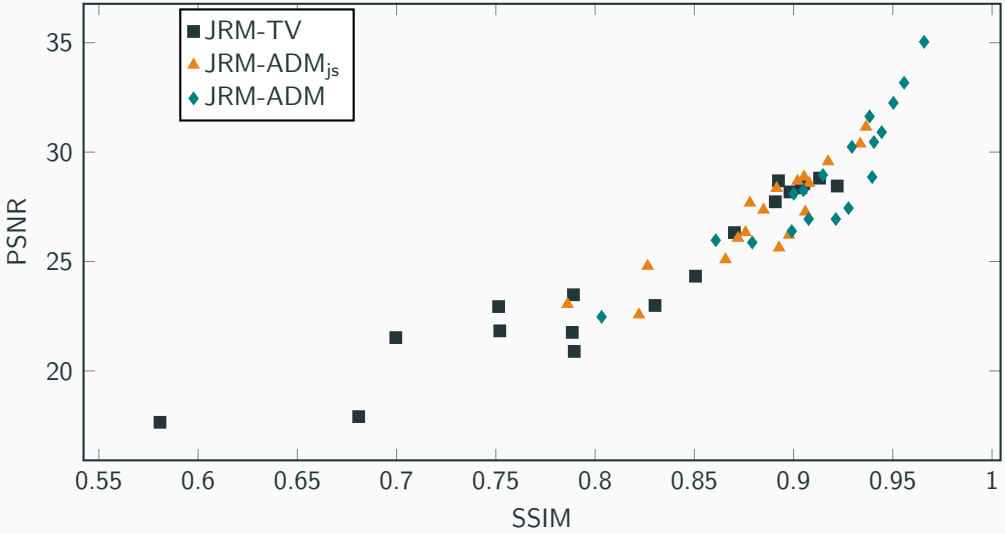


Answer: no.

60-view reconstructions: PSNR vs SSIM



20-view reconstructions: PSNR vs SSIM



n_a	Method	PSNR (↑)	SSIM (↑)	MAE τ (↓)	MAE ϑ (↓)
60	FDK	18.36	0.34	-	-
	JRM-TV	29.57	0.93	0.37	0.11
	JRM-ADM$_{js}$	**32.02**	**0.95**	**0.18**	0.12
	JRM-ADM	30.71	0.94	0.29	**0.10**
20	FDK	16.57	0.22	-	-
	JRM-TV	24.47	0.82	0.94	0.18
	JRM-ADM$_{js}$	27.09	0.88	0.36	0.19
	JRM-ADM	**28.88**	**0.92**	**0.32**	**0.13**

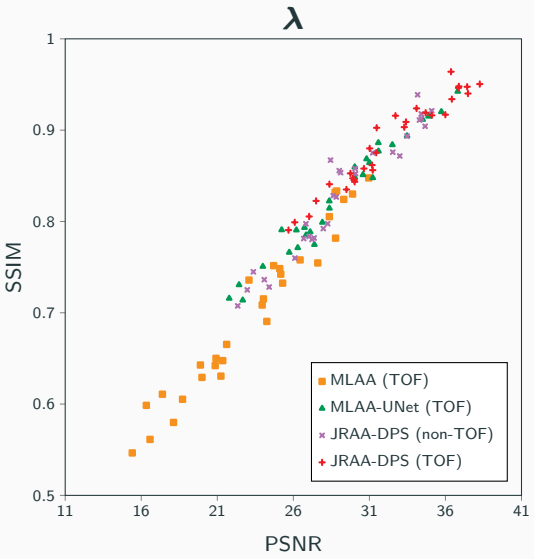
DPS for PCCT material decomposition — low-count: metrics

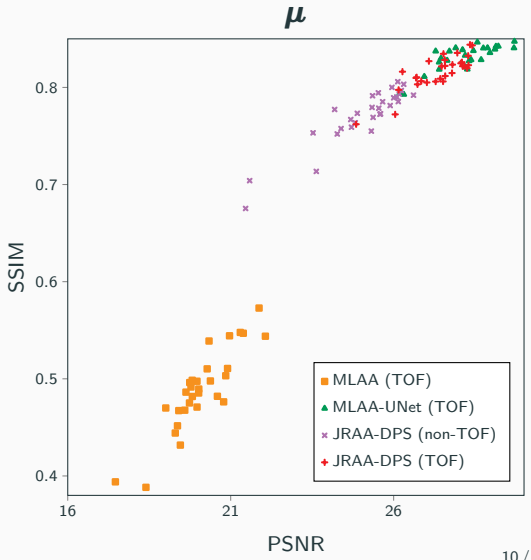
		FBP	LBFGS	U-Net-I	U-Net-P	ImTDPS	ProjTDPS	ODPS
Avg. Time (in sec)		0.06	28.59	95.91	88.26	90.51	89.3	134.08
PSNR ↑	Bones	26.41	37.97	39.99	**42.48**	30.12	41.39	42.13
	Tissues	20.74	31.31	33.06	33.05	25.78	34.52	**35.86**
SSIM ↑	Bones	0.26	0.89	0.90	**0.96**	0.64	0.91	0.95
	Tissues	0.12	0.80	0.86	0.79	0.56	**0.87**	**0.87**
LPIPS ↓	Bones	0.64	0.10	0.07	0.06	0.09	0.04	**0.02**
	Tissues	0.68	0.17	0.12	0.21	0.10	**0.04**	0.06

DPS for PCCT material decomposition — sparse-view: metrics

		FBP	LBFGS	U-Net-I	U-Net-P	ImTDPS	ProjTDPS	ODPS
Avg. Time (in sec)		0.05	23.67	104.38	105.35	90.38	82.97	88.89
PSNR ↑	Bones	25.22	35.09	37.52	**41.29**	30.09	37.69	40.58
	Tissues	19.13	29.57	33.35	32.46	25.55	31.48	**35.05**
SSIM ↑	Bones	0.22	0.87	0.93	**0.94**	0.64	0.90	**0.94**
	Tissues	0.11	0.78	**0.90**	0.83	0.55	0.82	0.86
LPIPS ↓	Bones	0.69	0.11	0.08	0.08	0.09	0.07	**0.04**
	Tissues	0.67	0.18	0.08	0.16	0.11	**0.06**	0.07

Results on Simulated Data





Results on Simulated Data (1% counts)

