CONVERGENCE MAPS WITH DENOISING DIFFUSION PROBABILISTIC MODELS

Mila Lüscher

Supervisors:
Prof. Dr. Alexandre Refregier
Dr. Tilman Tröster
Arne Thomsen

ETH zürich

Denoising Diffusion Probabilistic Models

A New Era of Image Generation

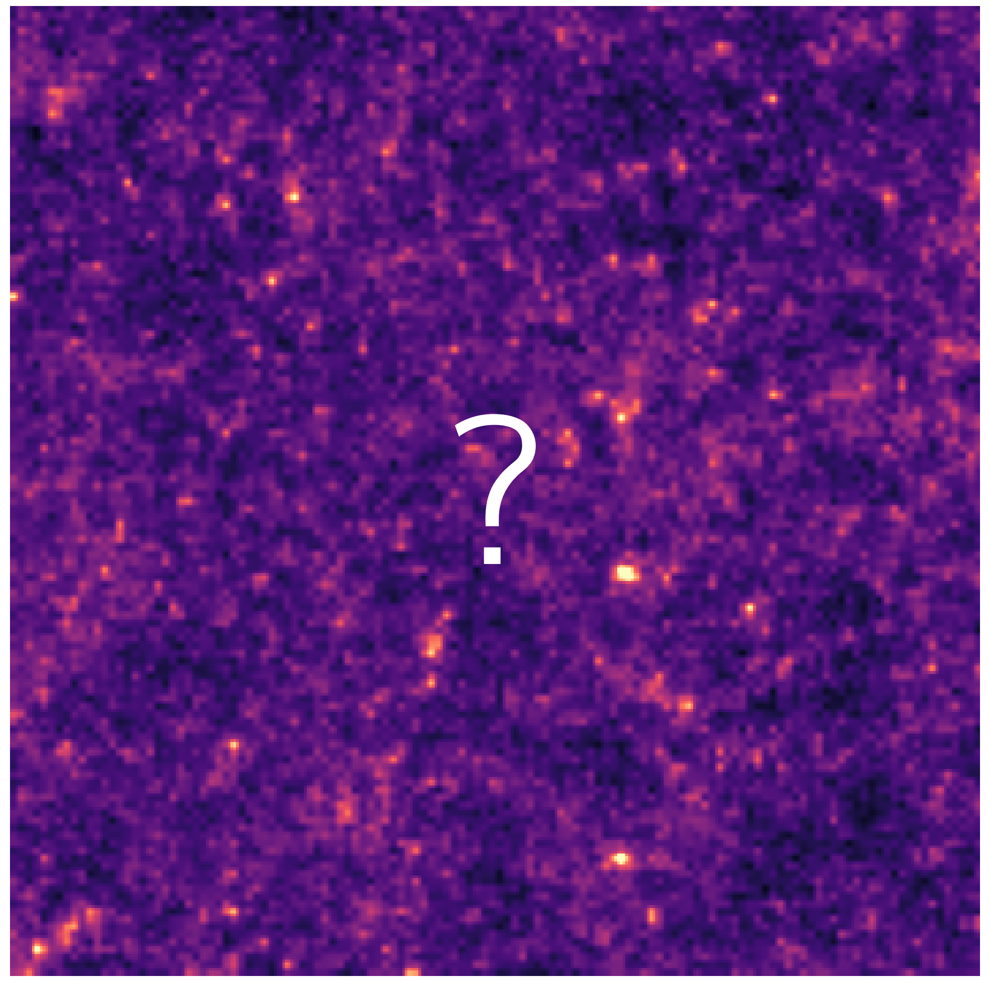
Corgis

Dall-E 2

Midjourney

Stable Diffusion

Convergence Maps



Contents

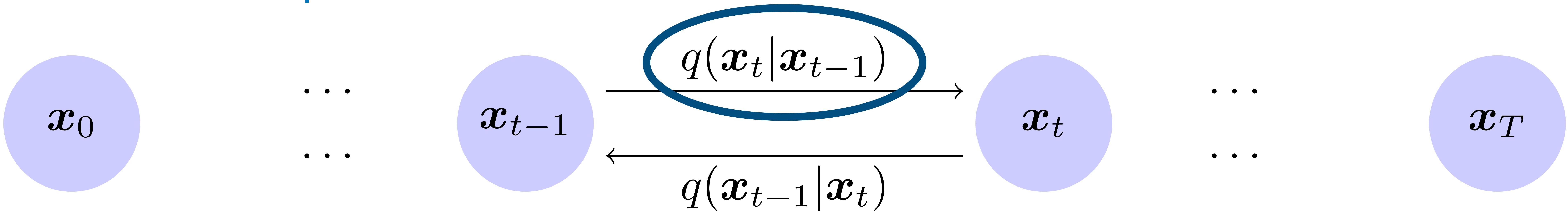
1. Denoising Diffusion Probabilistic Models (DDPM)
2. Conditional Wasserstein Generative Adversarial Networks (CWGAN) (Perraudin et al. 2020)
3. Training Data
4. Training Results
5. Discussion on Hyper-parameters and Sampling Methods
6. DDPM vs. CWGAN
7. Conclusion

Contents

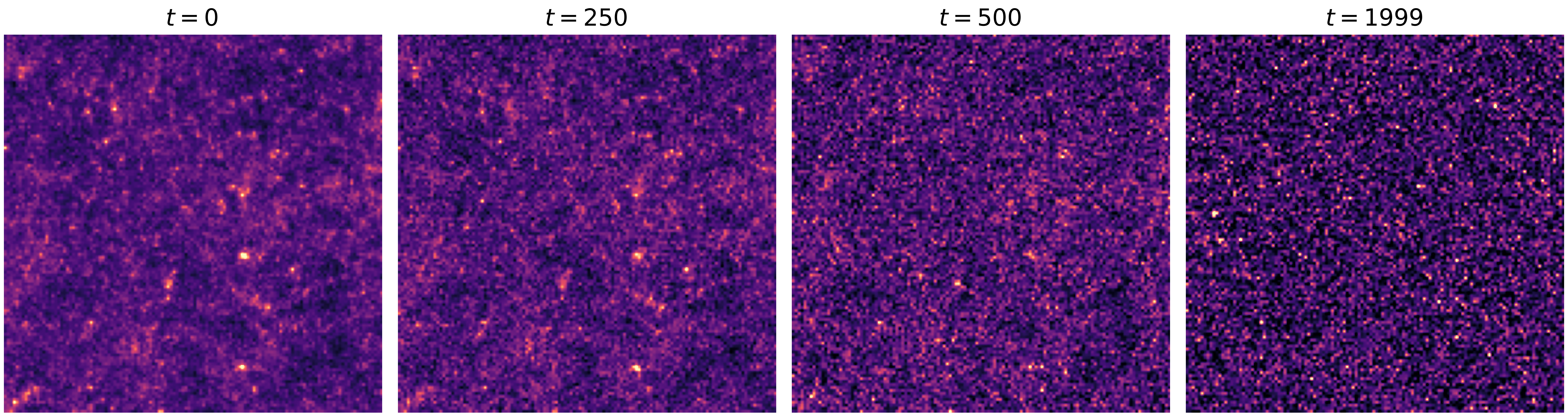
1. Denoising Diffusion Probabilistic Models (DDPM)
2. Conditional Wasserstein Generative Adversarial Networks (CWGAN) (Perraudin et al. 2020)
3. Training Data
4. Training Results
5. Discussion on Hyper-parameters and Sampling Methods
6. DDPM vs. CWGAN
7. Conclusion

Denoising Diffusion Probabilistic Models

Forward diffusion process



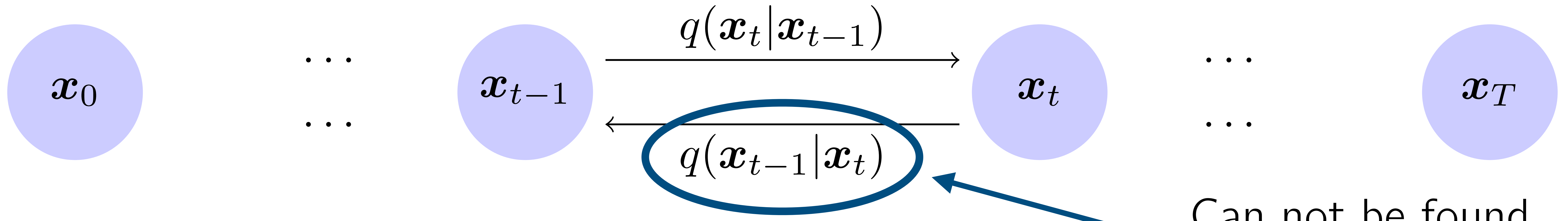
Conditional probability: $q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) = \mathcal{N}(\boldsymbol{x}_t; \sqrt{1-\beta_t}\boldsymbol{x}_{t-1}, \beta_t)\,,$ $\qquad \boldsymbol{x}_t = \sqrt{1-\beta_t}\boldsymbol{x}_{t-1} + \sqrt{\beta_t}\boldsymbol{\epsilon}_{t-1}$



(Ho, Jain, and Abbeel 2020)

Denoising Diffusion Probabilistic Models

Reverse diffusion process



Can not be found analytically

Need an approximate model:

$$q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t) \approx p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t) = \mathcal{N}(\boldsymbol{x}_{t-1}; \mu_\theta(\boldsymbol{x}_t, t), \beta_\theta(\boldsymbol{x}_t, t))$$

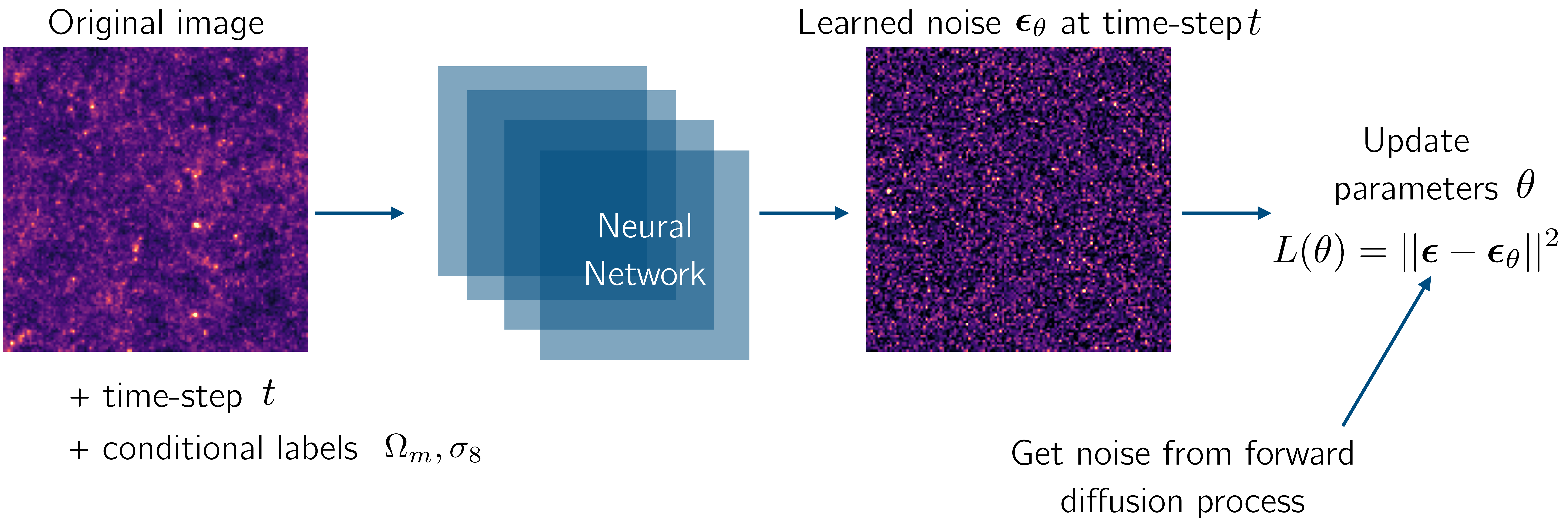
$$\mu_\theta(\boldsymbol{x}_t, t) = \frac{1}{\sqrt{\alpha_t}}\left(\boldsymbol{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(\boldsymbol{x}_t, t)\right)$$

Only need to learn noise term $\longrightarrow$ $\boxed{L(\theta) = \mathbb{E}_{t,\mathbf{x}_0,\epsilon}\left[||\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\boldsymbol{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t)||^2\right]}$

(Ho, Jain, and Abbeel 2020)

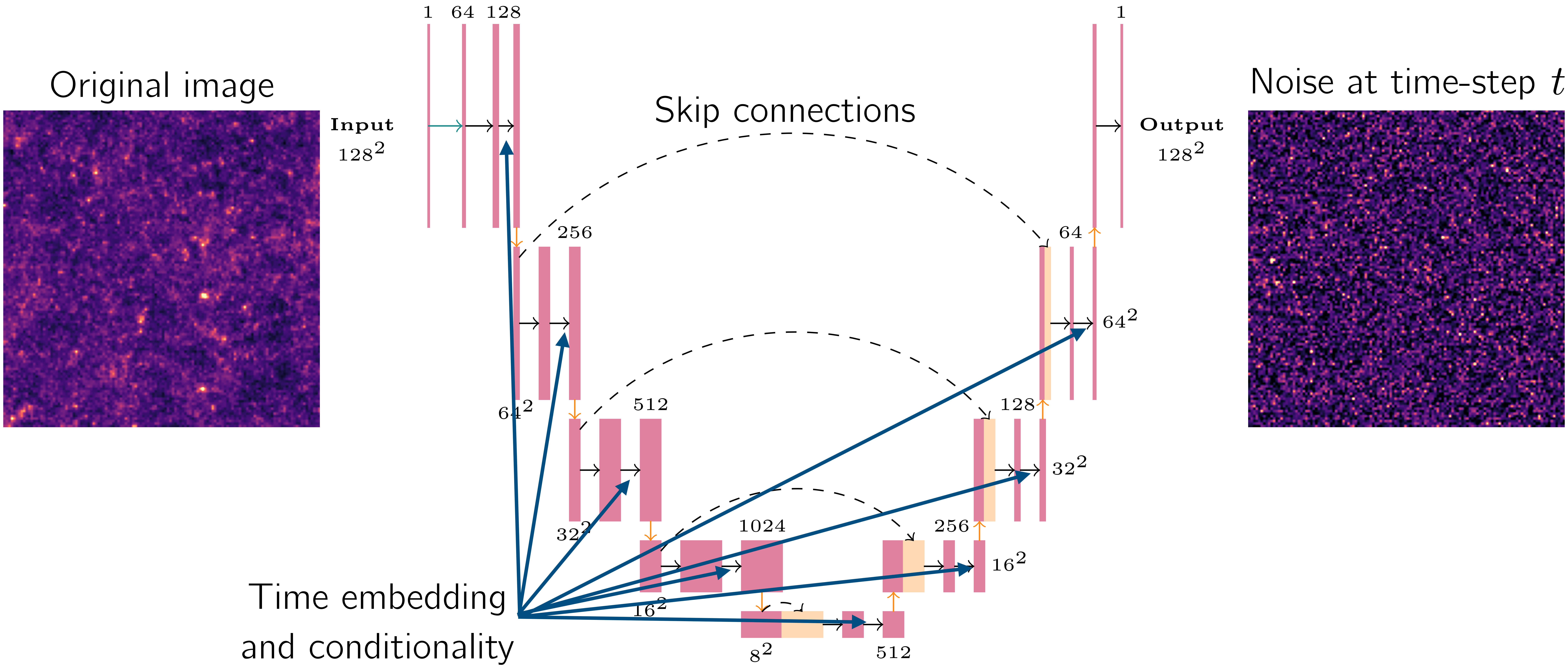
Neural Networks: DDPM

Architecture



Neural Networks: DDPM

UNet



Denoising Diffusion Probabilistic Models

Ancestral sampling

Sampling Algorithm:

Call model

1. Sample noise $\boldsymbol{x}_T \sim \mathcal{N}(0,1)$

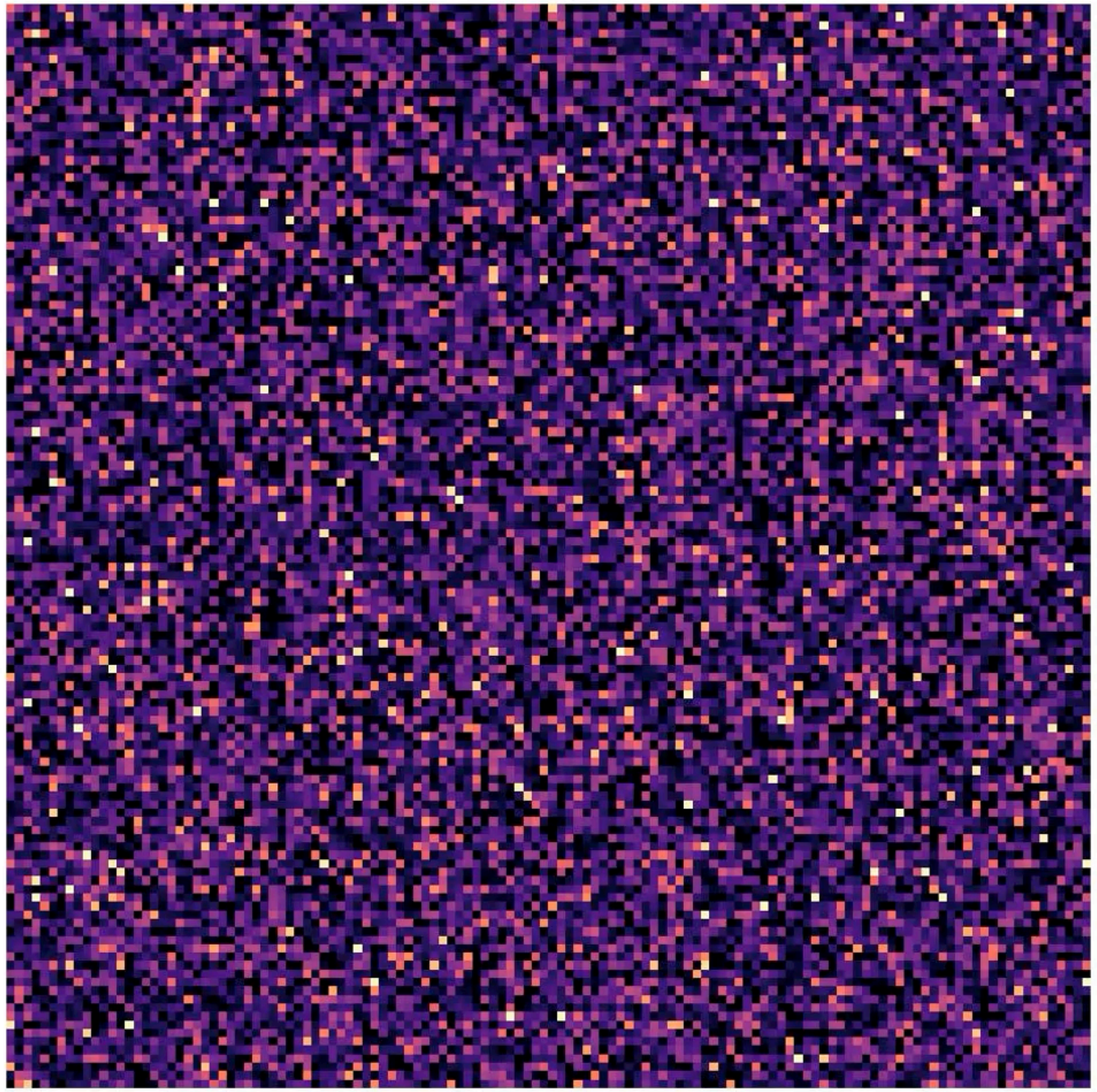
2. for each time-step $t = T, \ldots, 1$ do

$$\boldsymbol{x}_{t-1} = \frac{1}{\sqrt{\bar{\alpha}}}(\boldsymbol{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(\boldsymbol{x}_t, t))$$

   if $t > 1$ do $\boldsymbol{x}_{t-1} = \boldsymbol{x}_{t-1} + \sqrt{\tilde{\beta}_t}\boldsymbol{z}$ where $\boldsymbol{z} \sim \mathcal{N}(0,1)$

3. Return denoised image $\boldsymbol{x}_0$

$t = 1995$



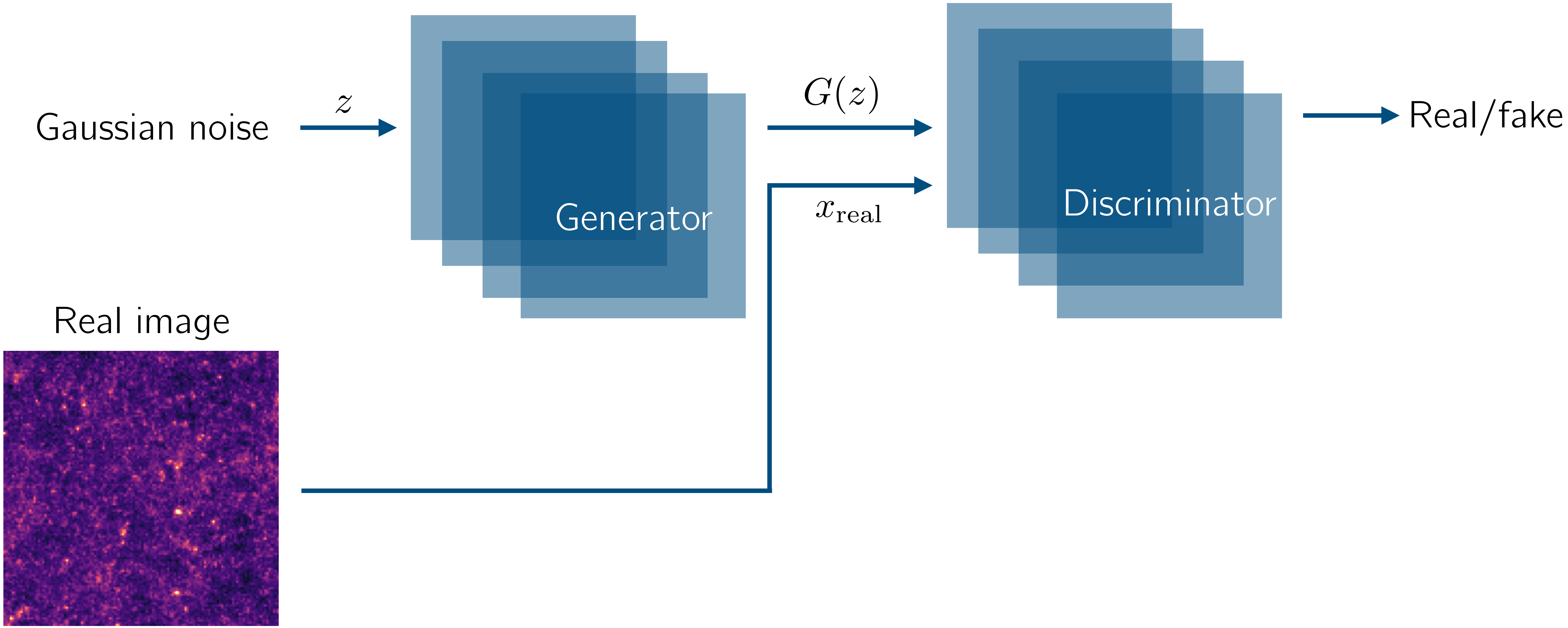
(Ho, Jain, and Abbeel 2020)

Contents

1. Denoising Diffusion Probabilistic Models (DDPM)
2. Conditional Wasserstein Generative Adversarial Networks (CWGAN) (Perraudin et al. 2020)
3. Training Data
4. Training Results
5. Discussion on Hyper-parameters and Sampling Methods
6. DDPM vs. CWGAN
7. Conclusion

Generative Adversarial Network

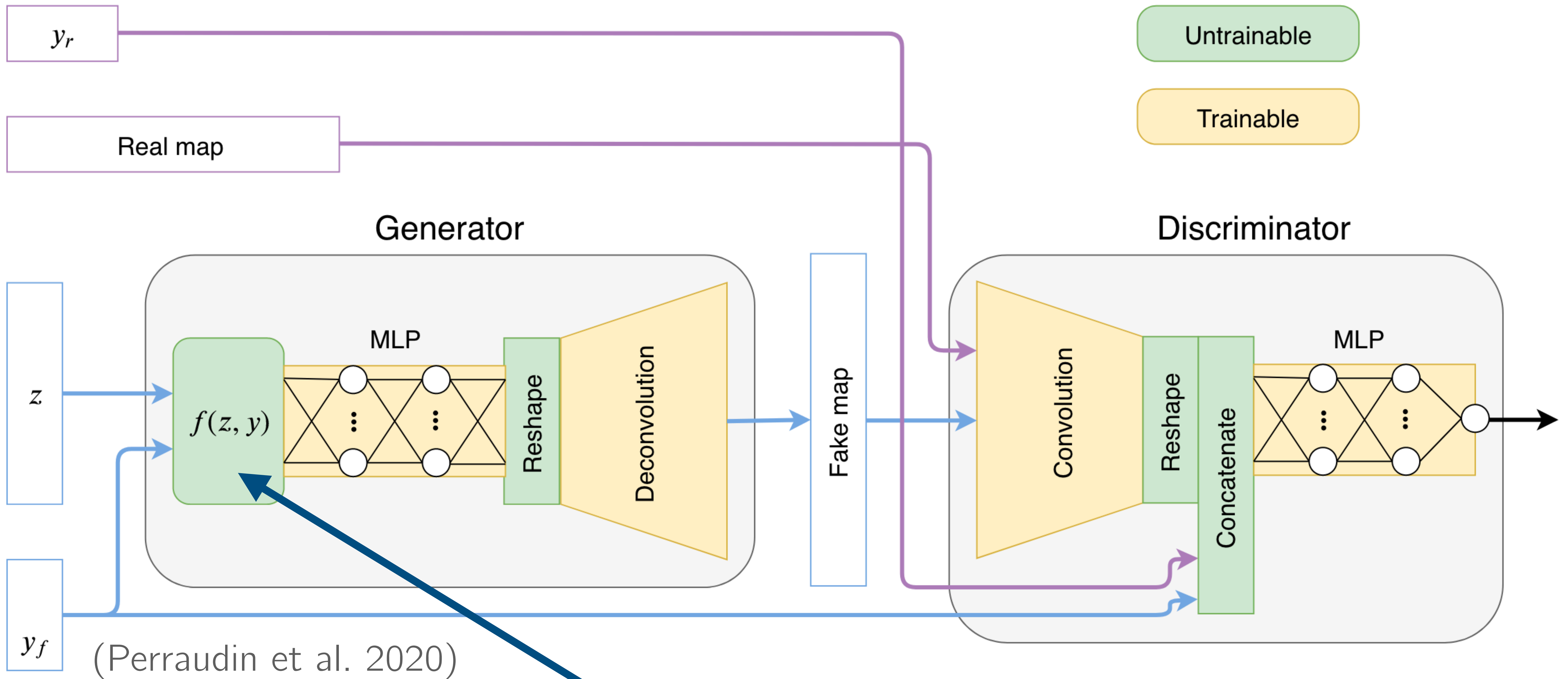
Architecture



Neural Networks: CWGAN

Architecture

Conditioning parameters



(Perraudin et al. 2020)

Loss Function:

Wasserstein loss function to combat mode collapse

Conditioning Function:

$$\hat{z} = f(z, y) = \left(l_0 + \frac{l_1 - l_0}{b - a}(y - a) \right) \frac{z}{\|z\|_2}$$

DDPM vs. CWGAN

Differences

GAN

+ high quality image generation

- hard to train

- mode collapse

+ fast sampling

- requires a lot of training data

DDPM

+ high quality image generation

+ easy training

+ no mode collapse

- slow sampling speed

+ more faithful to data

Contents

1. Denoising Diffusion Probabilistic Models (DDPM)
2. Conditional Wasserstein Generative Adversarial Networks (CWGAN) (Perraudin et al. 2020)
3. Training Data
4. Training Results
5. Discussion on Hyper-parameters and Sampling Methods
6. DDPM vs. CWGAN
7. Conclusion

Training Data

Convergence maps

N-Body simulations:

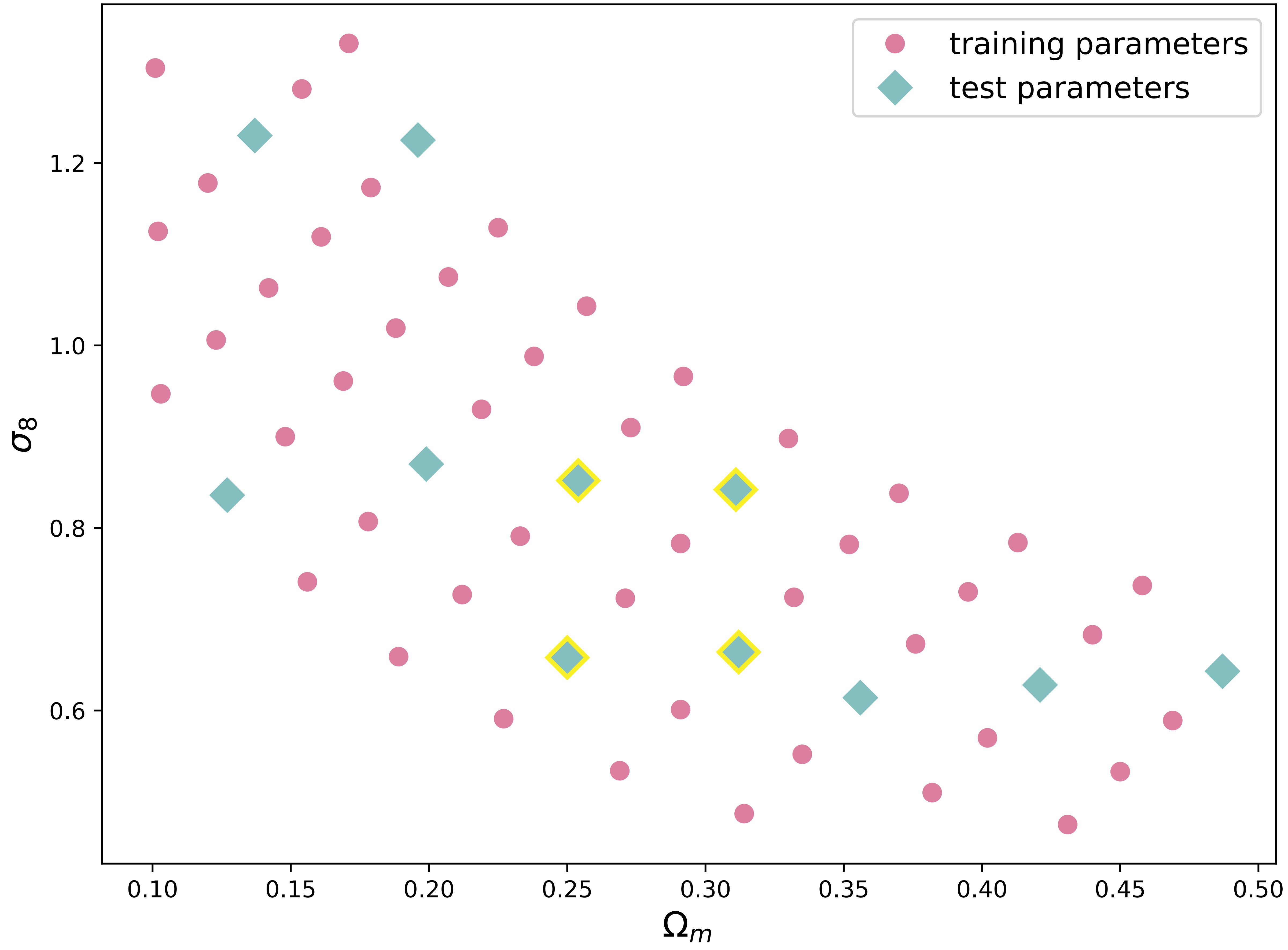
- PKDGrav3
- flat LambdaCDM universe
- 57 cosmologies: $\Omega_m,\ \sigma_8$

Convergence maps:

- 3D particle boxes to 2D convergence maps
- UFalcon

Training data:

- 46 training and 11 test cosmologies



Contents

1. Denoising Diffusion Probabilistic Models (DDPM)
2. Conditional Wasserstein Generative Adversarial Networks (CWGAN) (Perraudin et al. 2020)
3. Training Data
4. Training Results
5. Discussion on Hyper-parameters and Sampling Methods
6. DDPM vs. CWGAN
7. Conclusion

Results

Best Checkpoint

Preprocessing:

Standardised $\mathcal{N}(0,1)$

Beta scheduler:

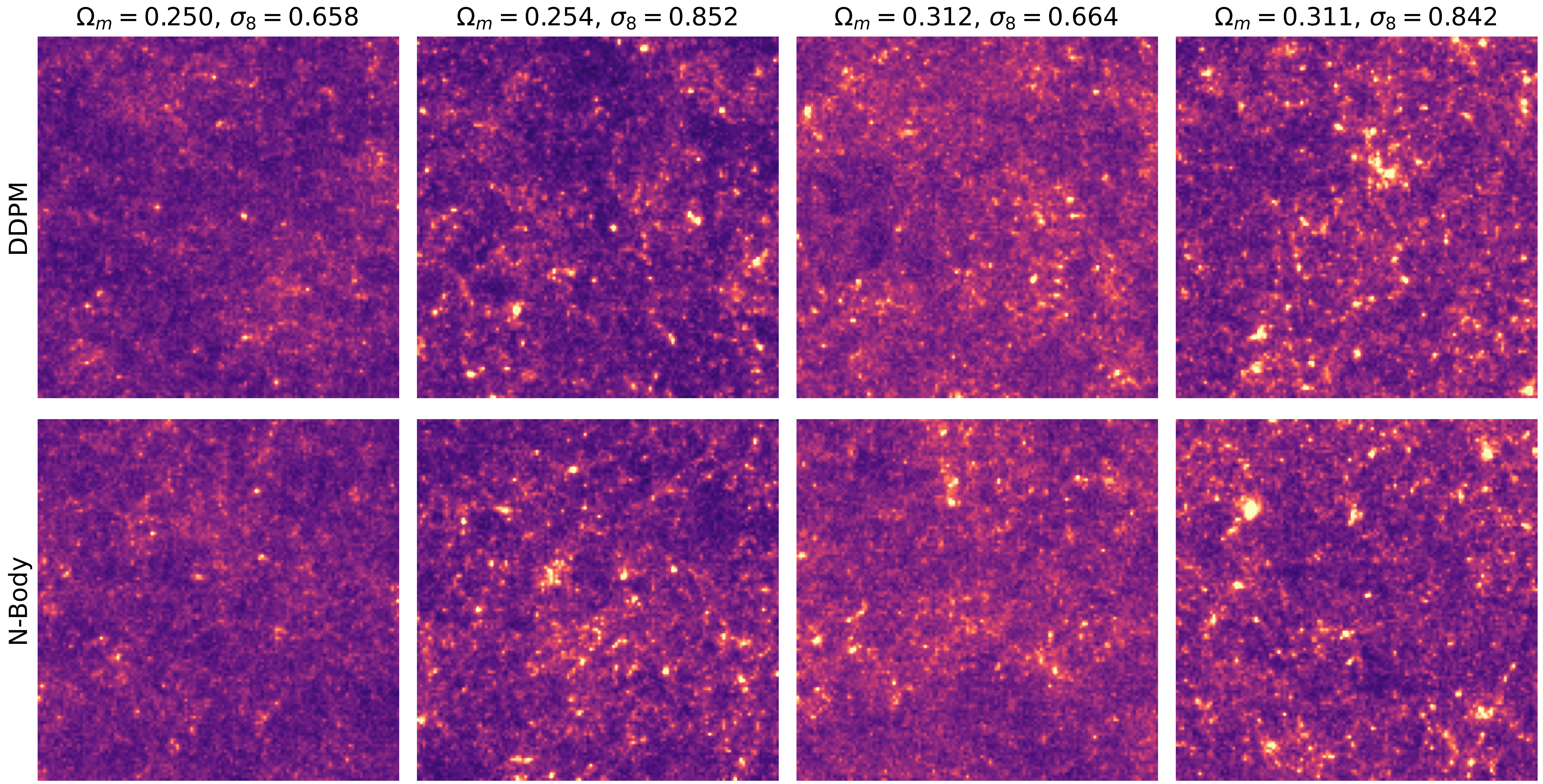
Linear beta scheduler:

$T = 2000,\ \beta_0 = 0.01,\ \beta_T = 0.00005$

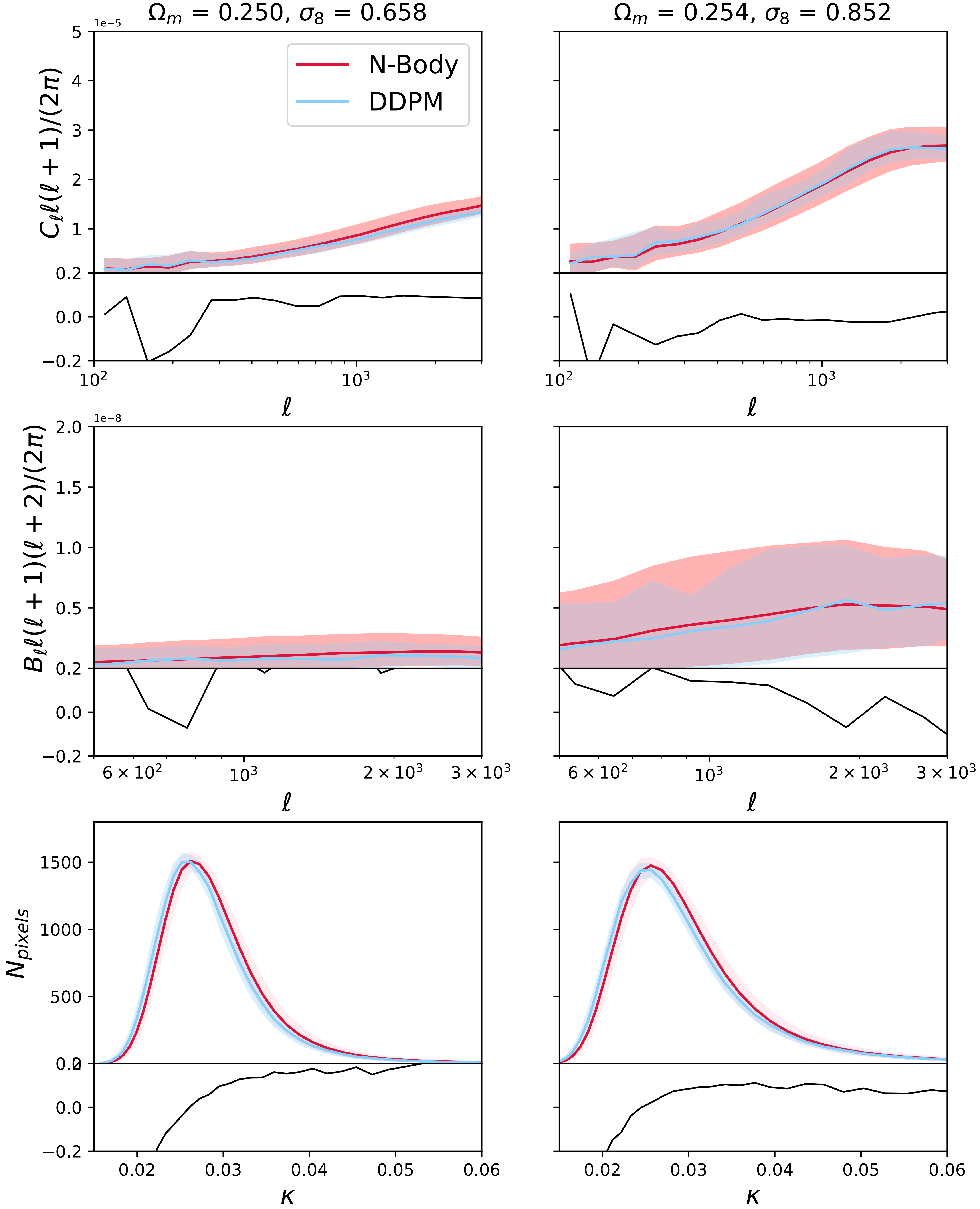
Architecture:

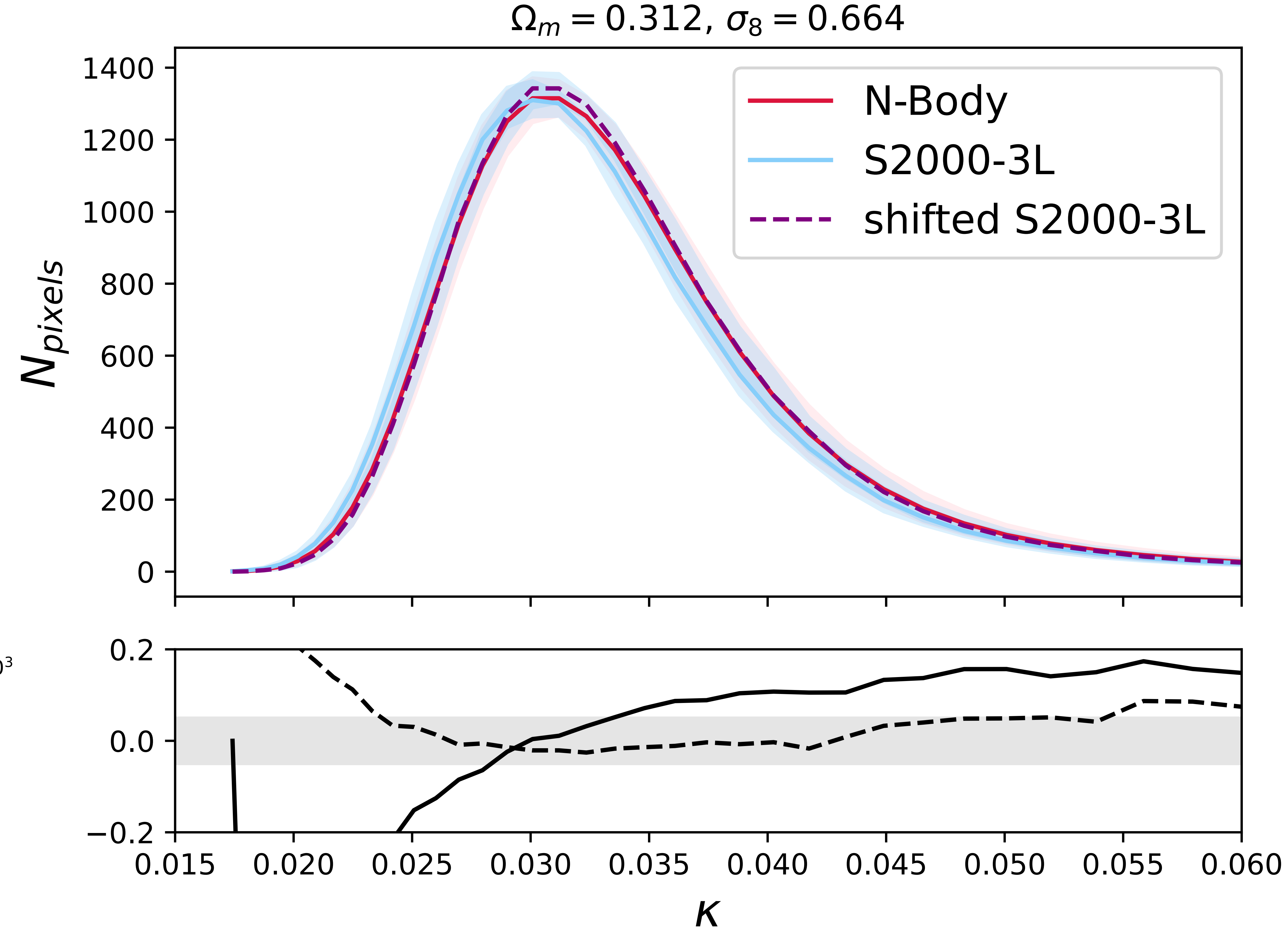
3 UNet layers

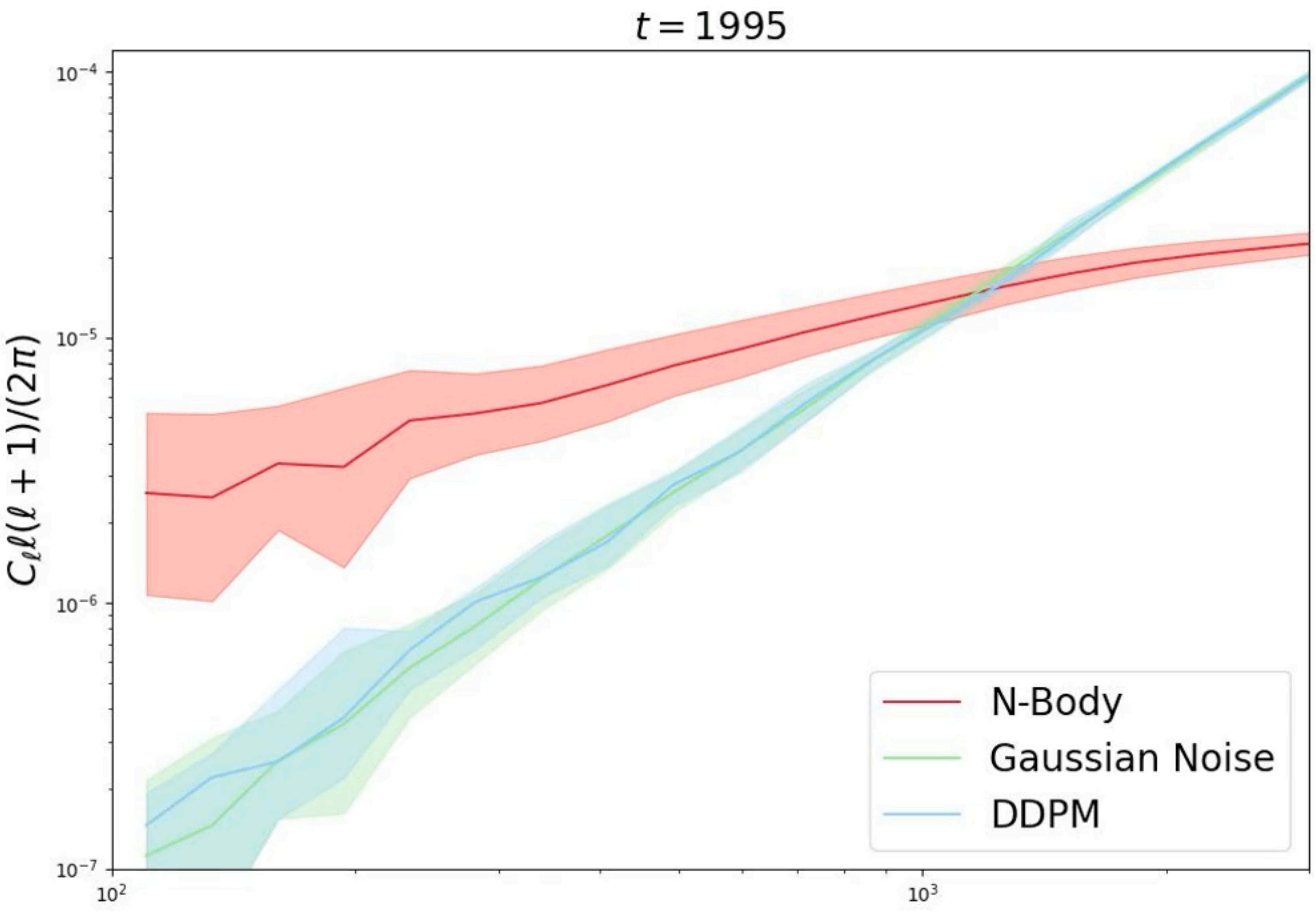
Test cosmologies

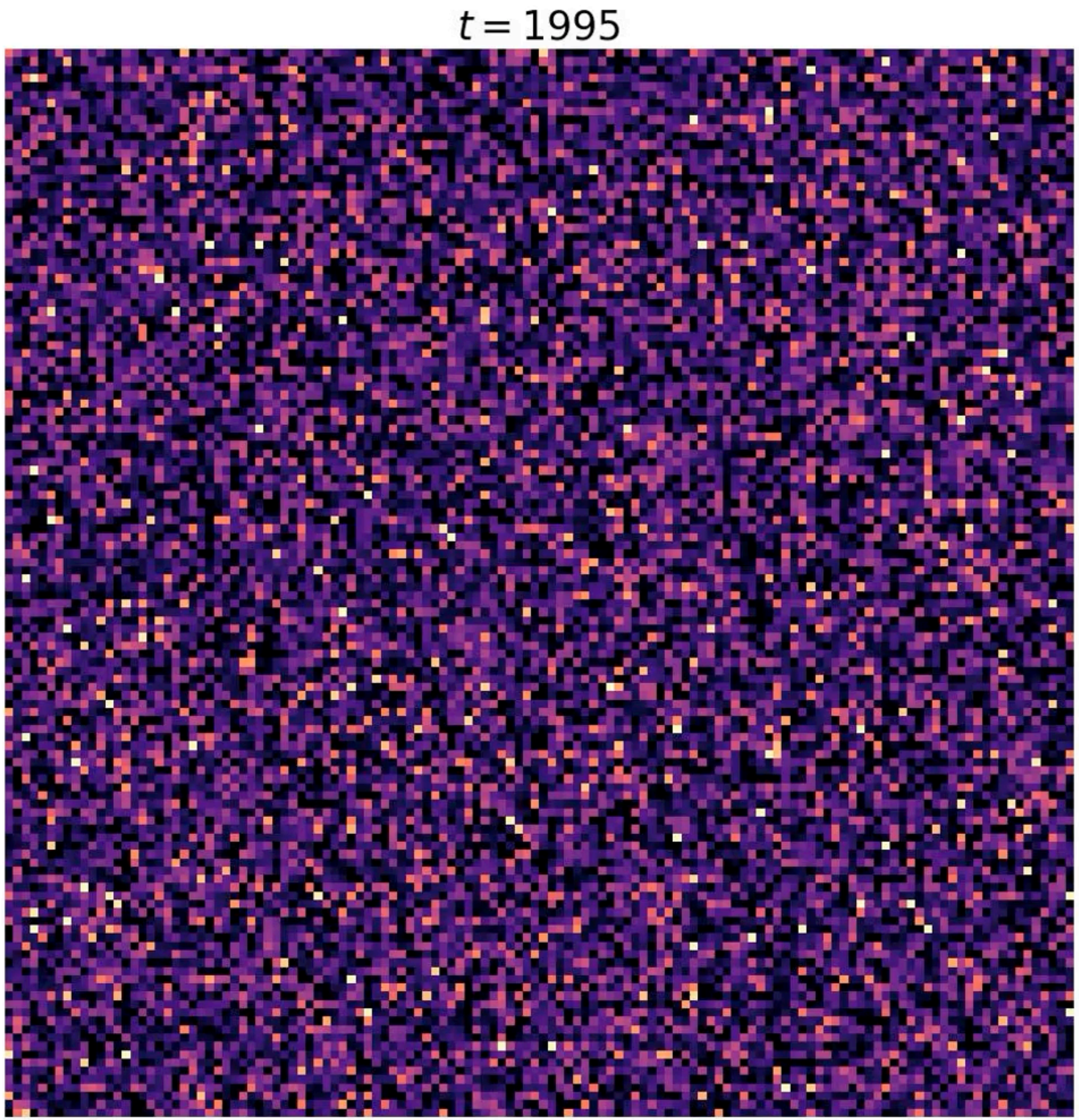


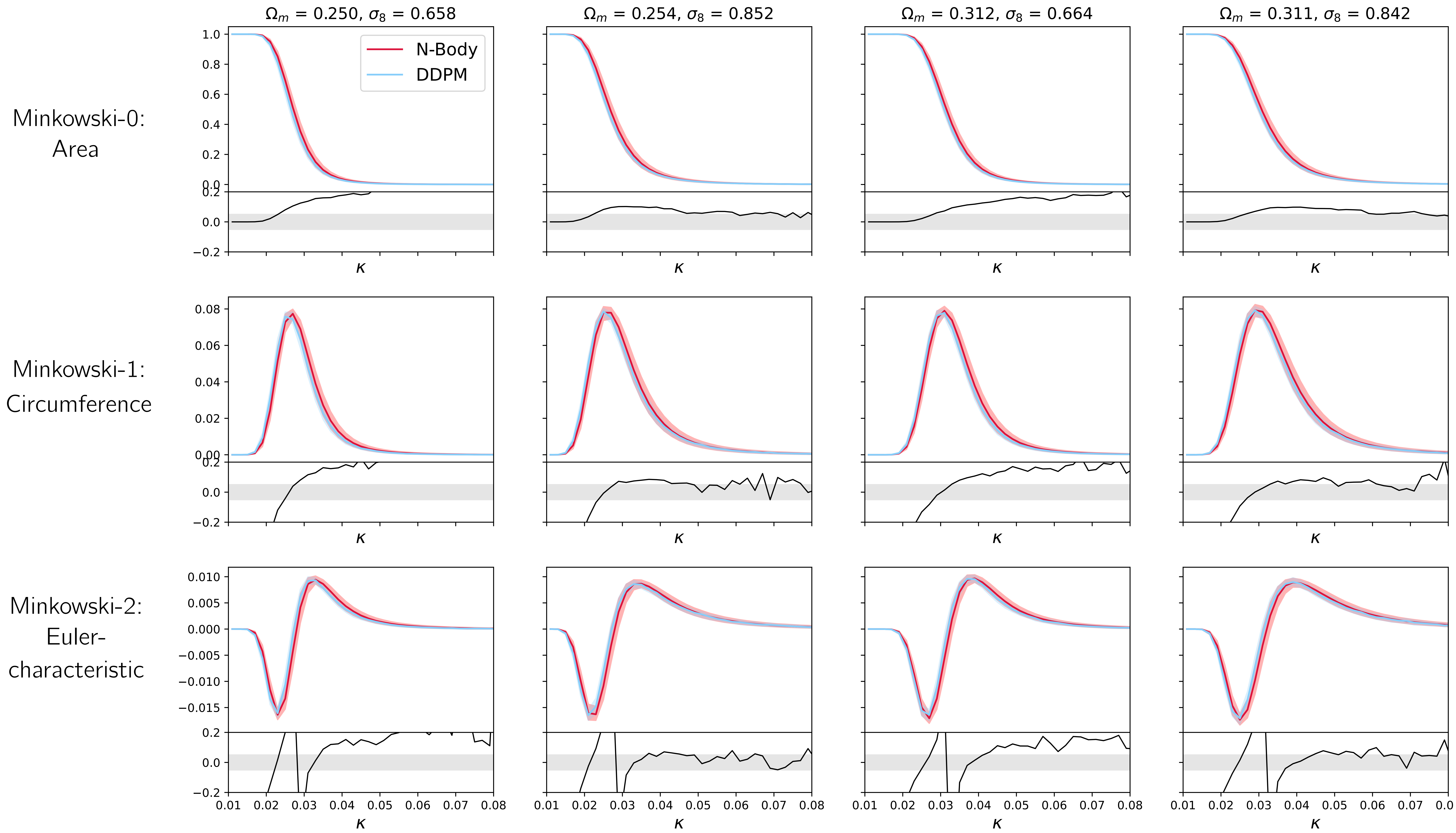
Summary Statistics



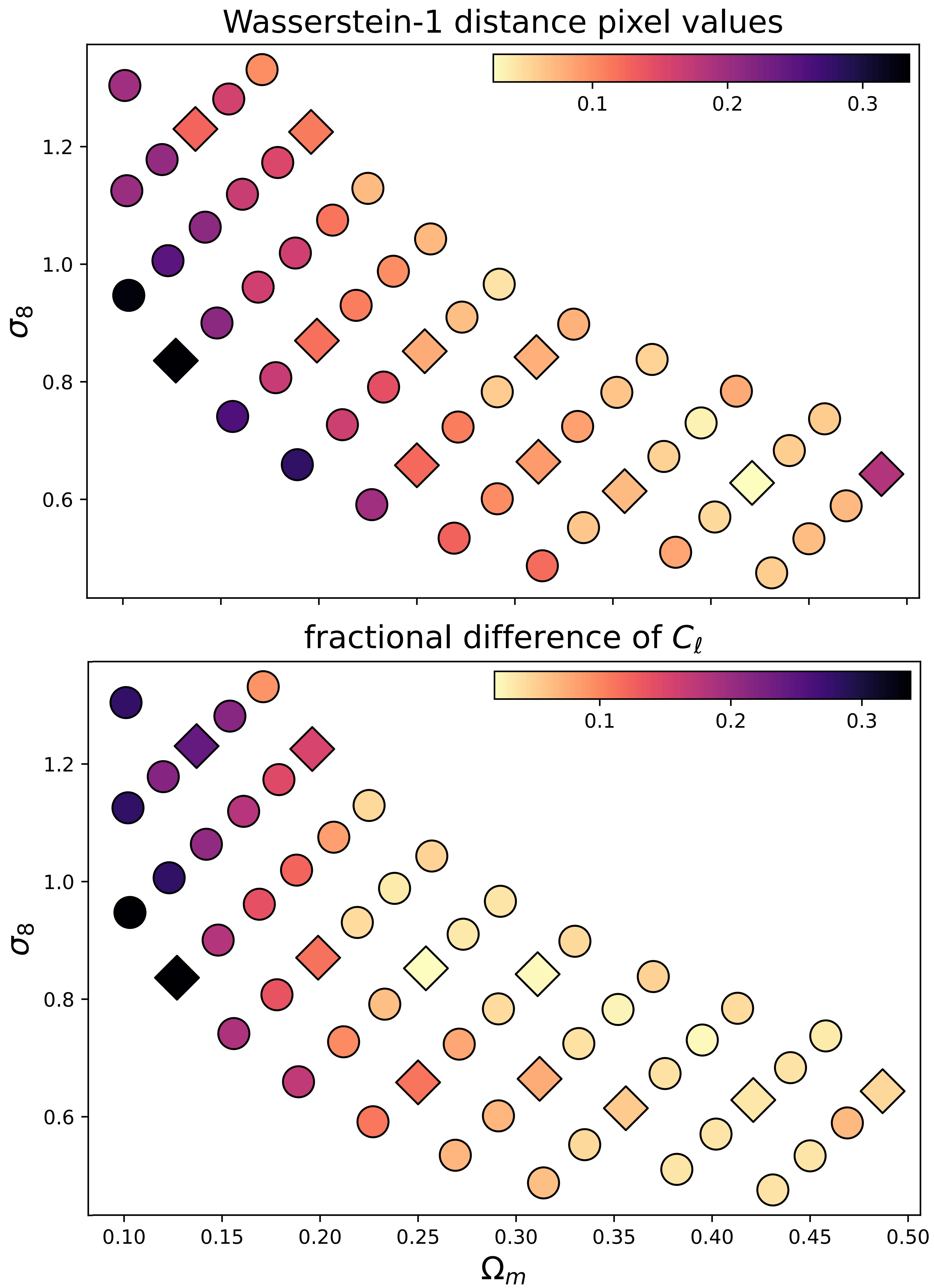


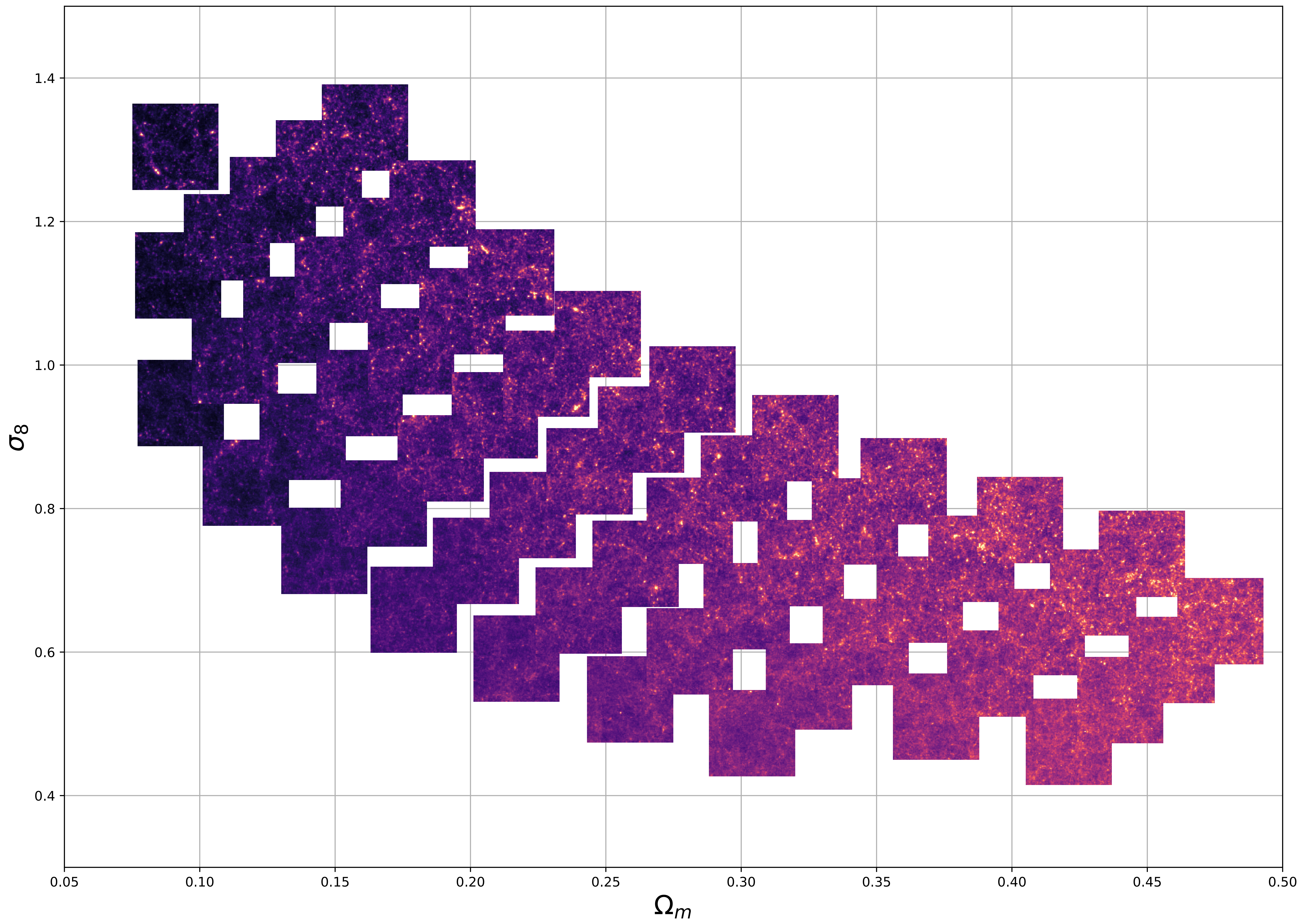

$t = 1995$
10^{-4}
10^{-5}
10^{-6}
10^{-7}
$C_\ell \ell(\ell+1)/(2\pi)$
10^2
10^3
N-Body
Gaussian Noise
DDPM


$t = 1995$



Ω_m = 0.250, σ_8 = 0.658
Ω_m = 0.254, σ_8 = 0.852
Ω_m = 0.312, σ_8 = 0.664
Ω_m = 0.311, σ_8 = 0.842
N-Body
DDPM
Minkowski-0: Area
Minkowski-1: Circumference
Minkowski-2: Euler-characteristic
κ


Wasserstein-1 distance pixel values
0.1
0.2
0.3
1.2
1.0
0.8
0.6
σ_8
fractional difference of C_ℓ
0.1
0.2
0.3
1.2
1.0
0.8
0.6
σ_8
0.10 0.15 0.20 0.25 0.30 0.35 0.40 0.45 0.50
Ω_m


1.4
1.2
1.0
0.8
0.6
0.4
σ_8
0.05 0.10 0.15 0.20 0.25 0.30 0.35 0.40 0.45 0.50
Ω_m

Contents

1. Denoising Diffusion Probabilistic Models (DDPM)
2. Conditional Wasserstein Generative Adversarial Networks (CWGAN) (Perraudin et al. 2020)
3. Training Data
4. Training Results
5. Discussion on Hyper-parameters and Sampling Methods
6. DDPM vs. CWGAN
7. Conclusion

Hyper-parameters

Preprocessing, beta-scheduler, architecture

Preprocessing:

Standardised $\mathcal{N}(0,1)$

Scaled $\in [-1,1]$

Beta scheduler:

Linear beta scheduler:

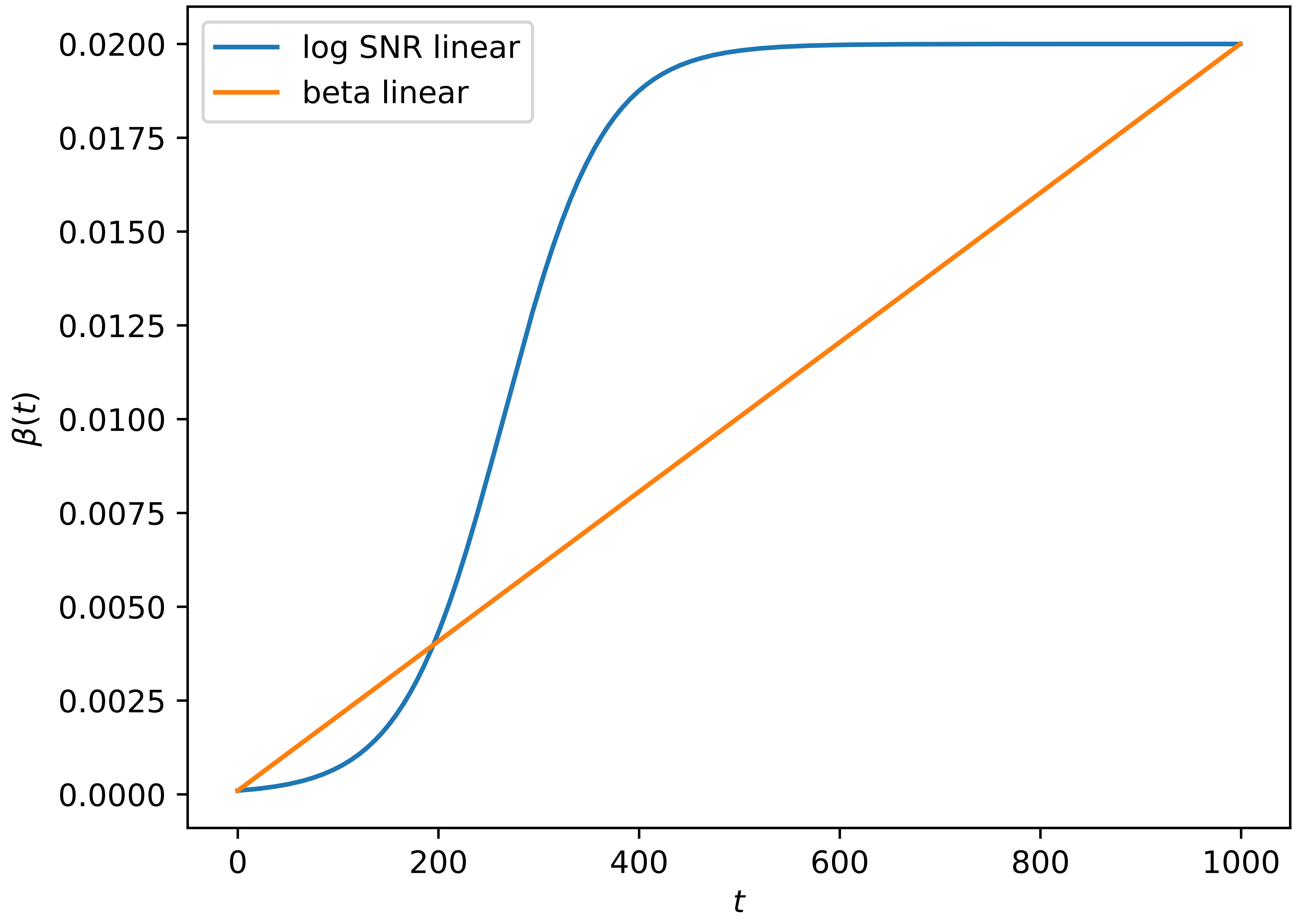
$$T = 1000,\ \beta_0 = 0.02,\ \beta_T = 0.0001$$

$$T = 2000,\ \beta_0 = 0.01,\ \beta_T = 0.00005$$

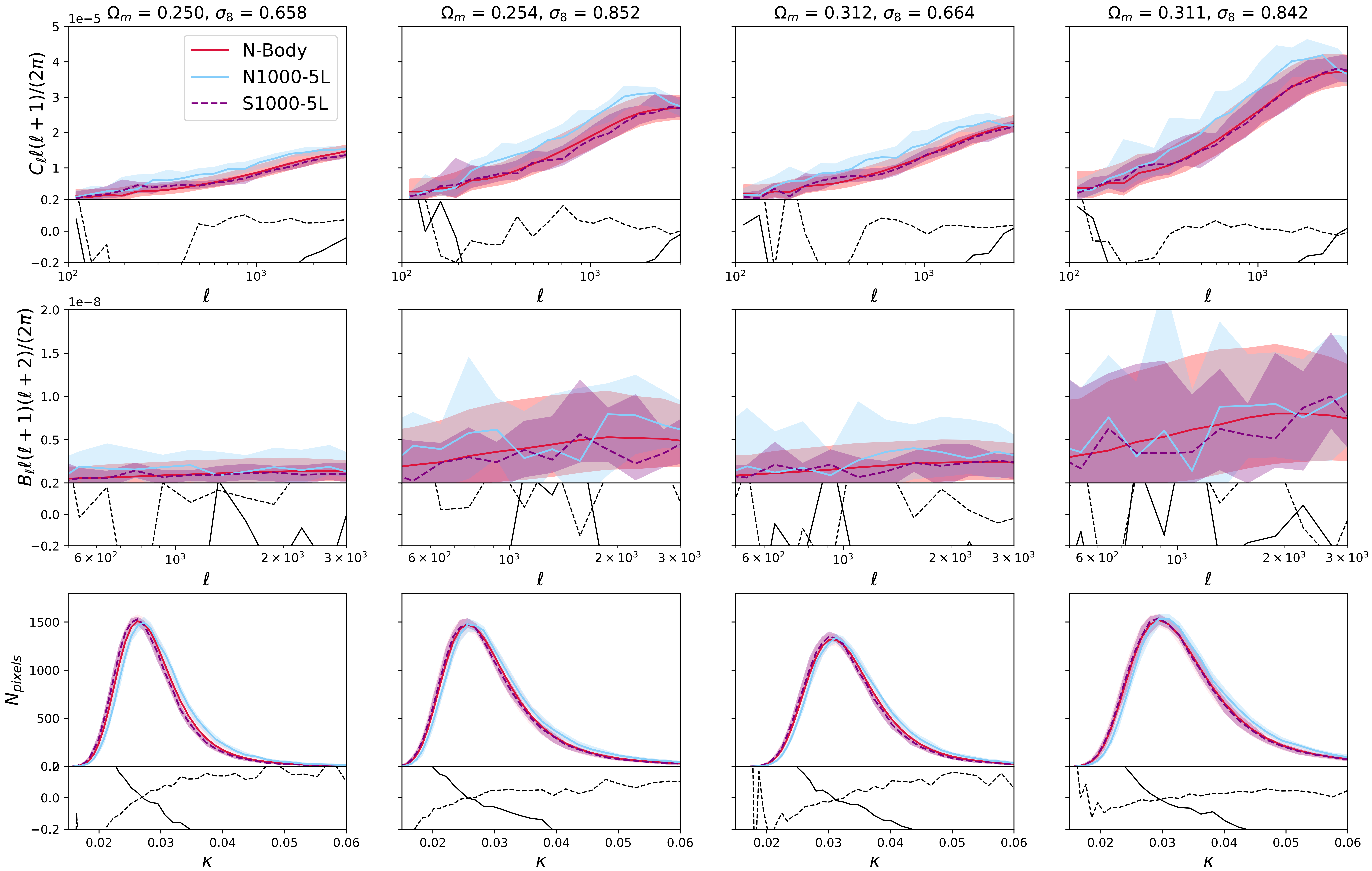
Signal-to-noise scheduler

Architecture:

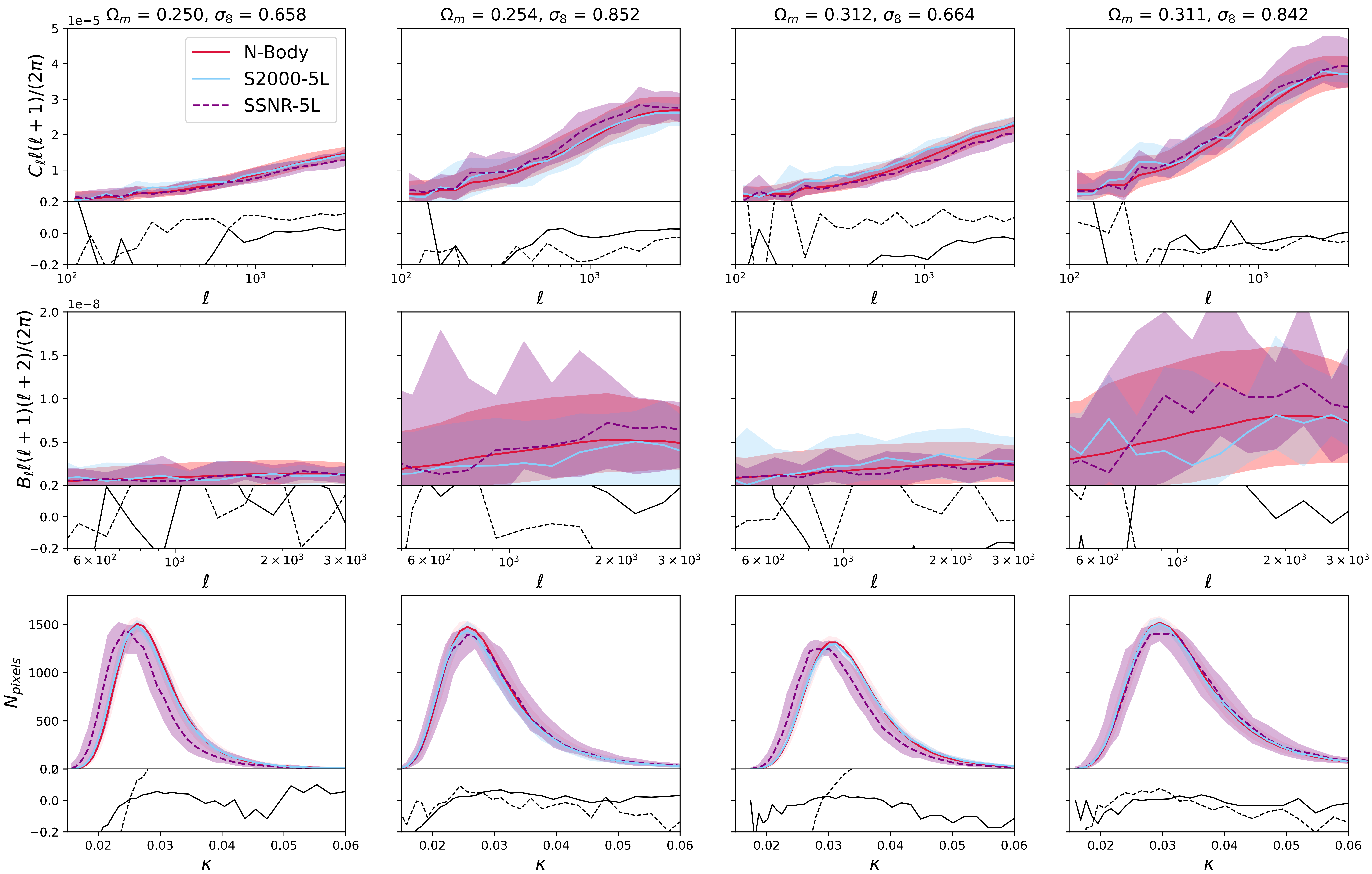
5, 4 and 3 UNet layers



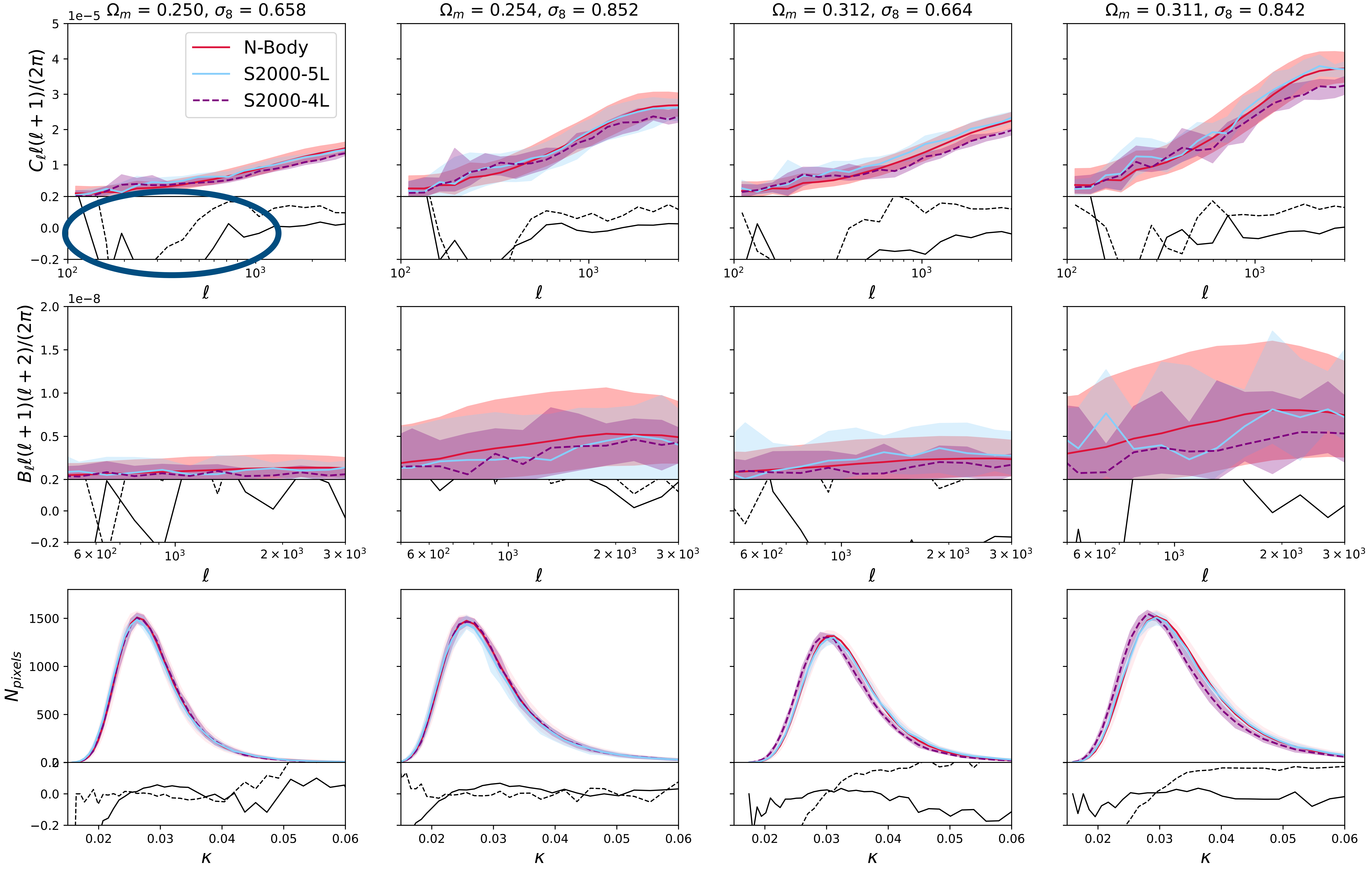
Preprocessing



Beta-Scheduler



Architecture



Results

Sampling methods

Reverse-Time SDE Sampling:

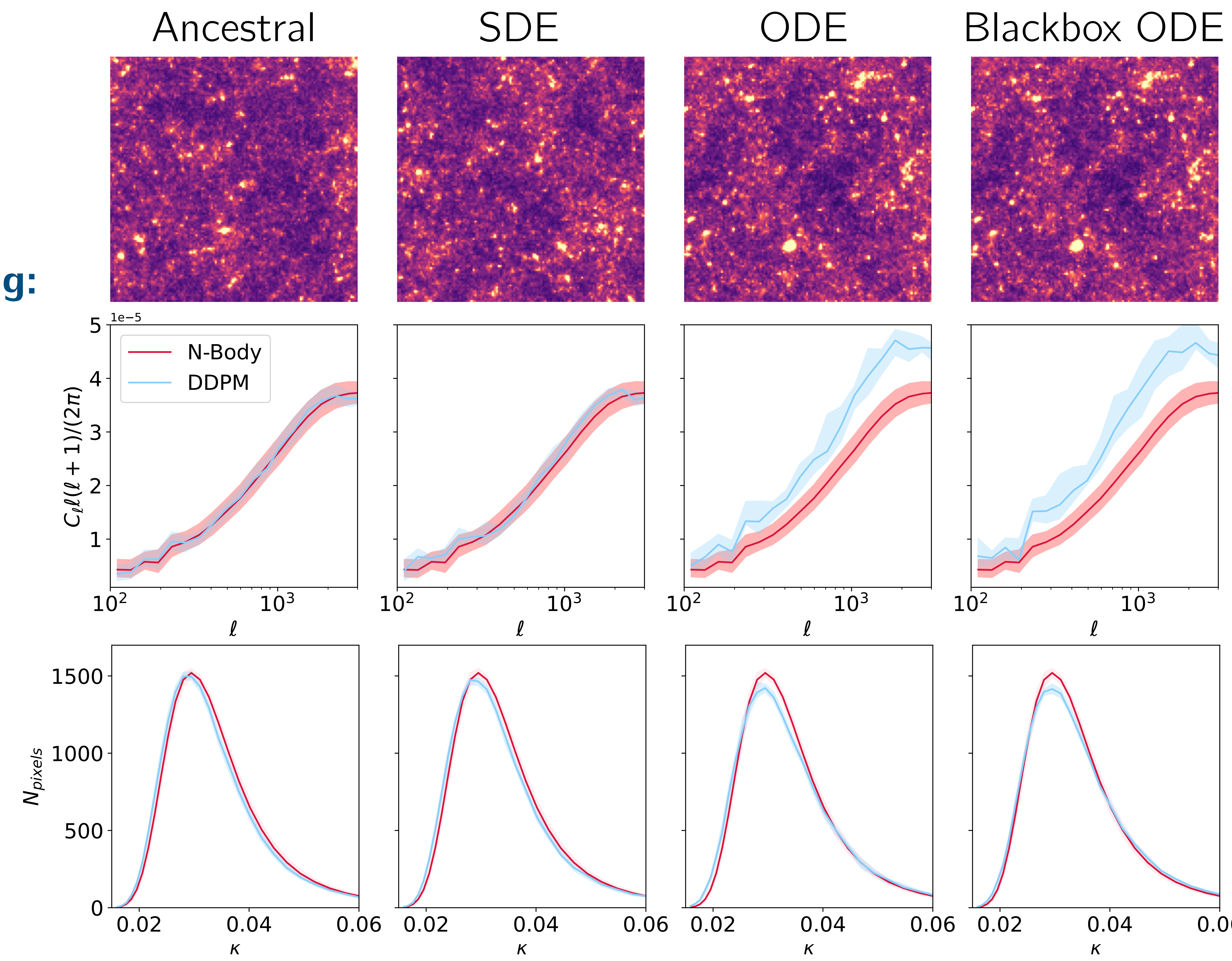
(Y. Song et al. 2021)

Iteratively solve discretised RT-SDE at each time-step

ODE Sampling:

(Y. Song et al. 2021)

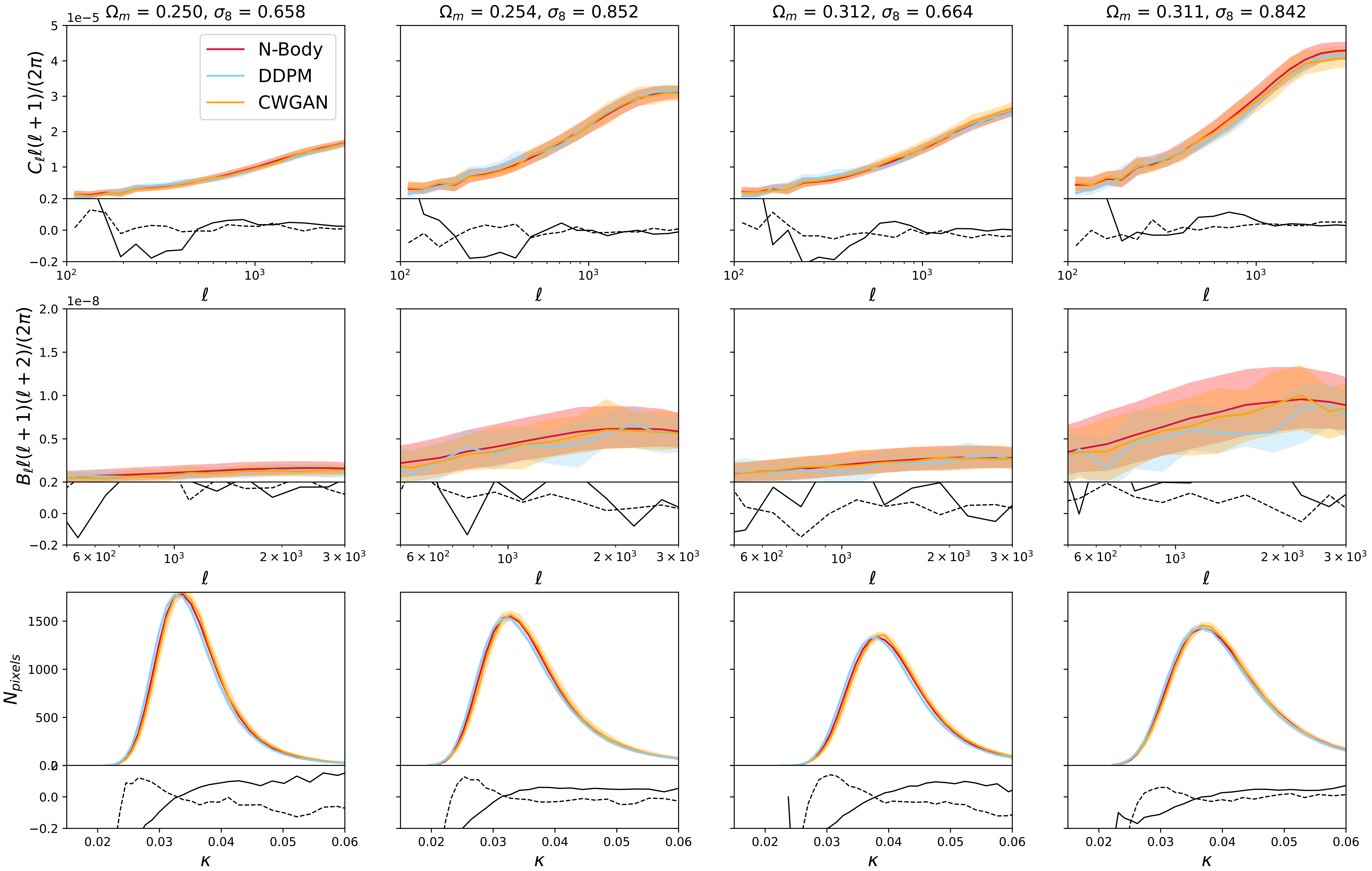
Iteratively solve discretised ODE or use blackbox solver (Runge-Kutta of order 5)

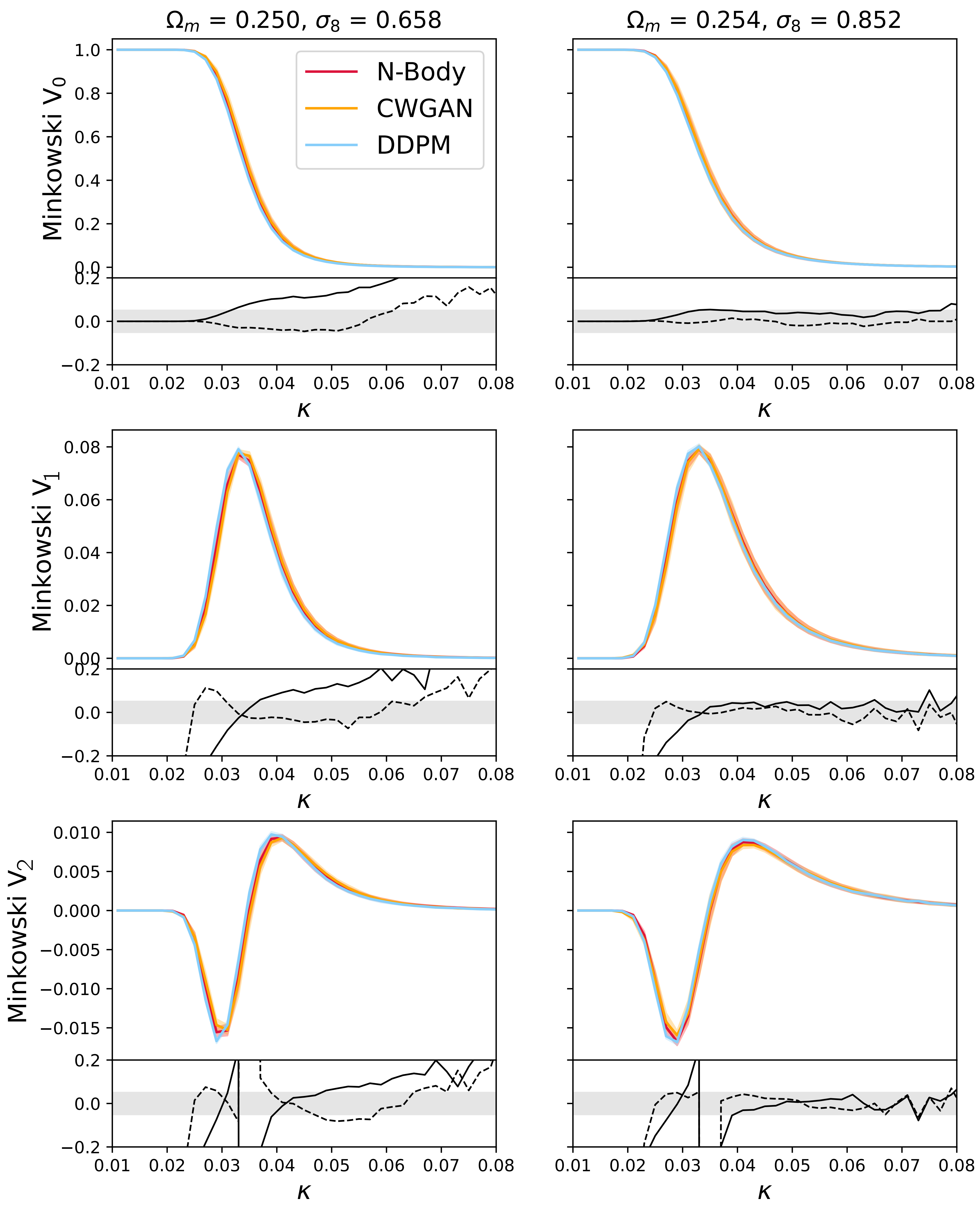


Contents

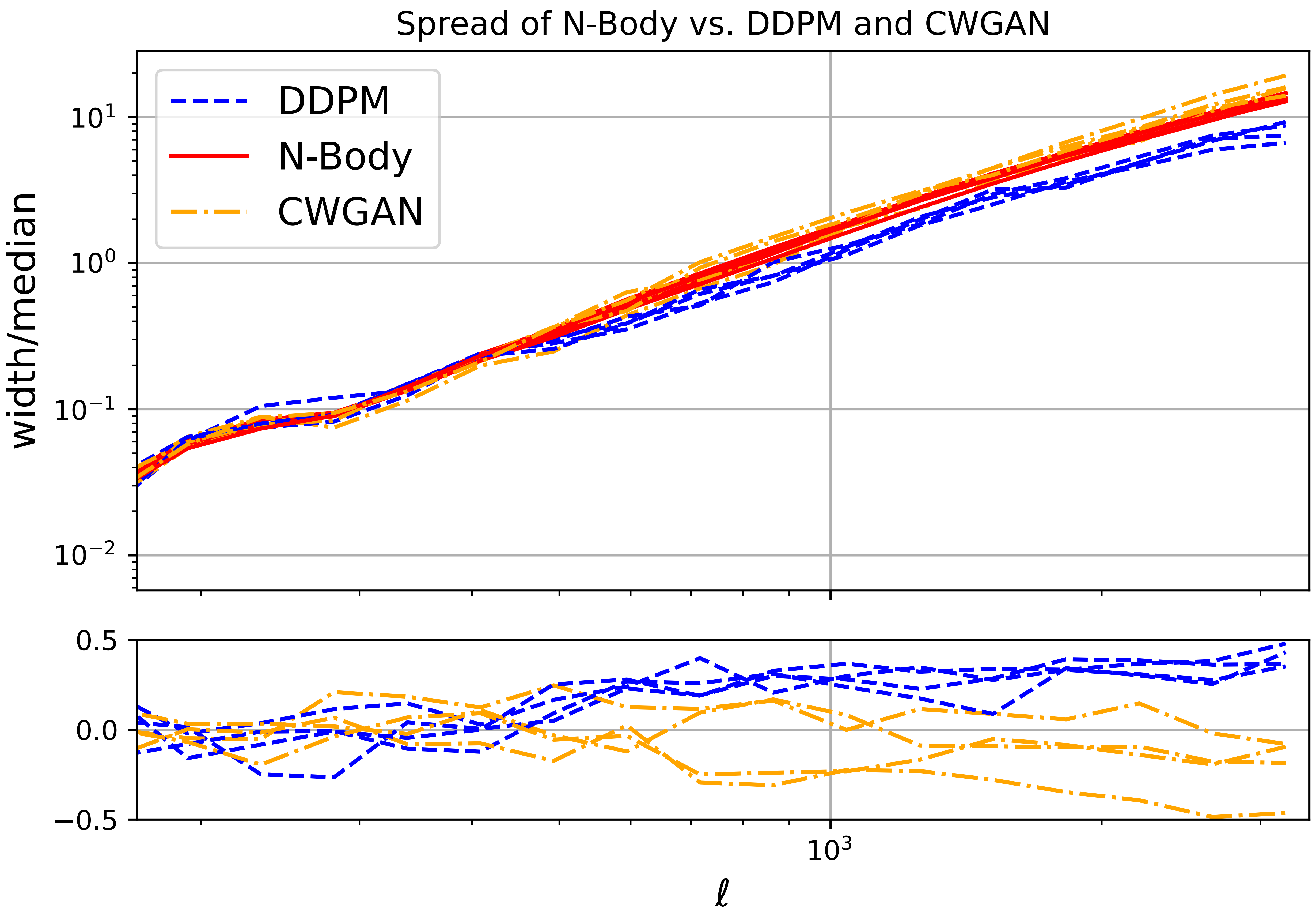
1. Denoising Diffusion Probabilistic Models (DDPM)
2. Conditional Wasserstein Generative Adversarial Networks (CWGAN) (Perraudin et al. 2020)
3. Training Data
4. Training Results
5. Discussion on Hyper-parameters and Sampling Methods
6. DDPM vs. CWGAN
7. Conclusion

Summary Statistics





Ω_m = 0.250, σ_8 = 0.658
Ω_m = 0.254, σ_8 = 0.852
N-Body
CWGAN
DDPM
Minkowski V_0
Minkowski V_1
Minkowski V_2
κ



Spread of N-Body vs. DDPM and CWGAN
DDPM
N-Body
CWGAN
width/median
ℓ

Contents

1. Denoising Diffusion Probabilistic Models (DDPM)
2. Conditional Wasserstein Generative Adversarial Networks (CWGAN) (Perraudin et al. 2020)
3. Training Data
4. Training Results
5. Discussion on Hyper-parameters and Sampling Methods
6. DDPM vs. CWGAN
7. Conclusion

Conclusion

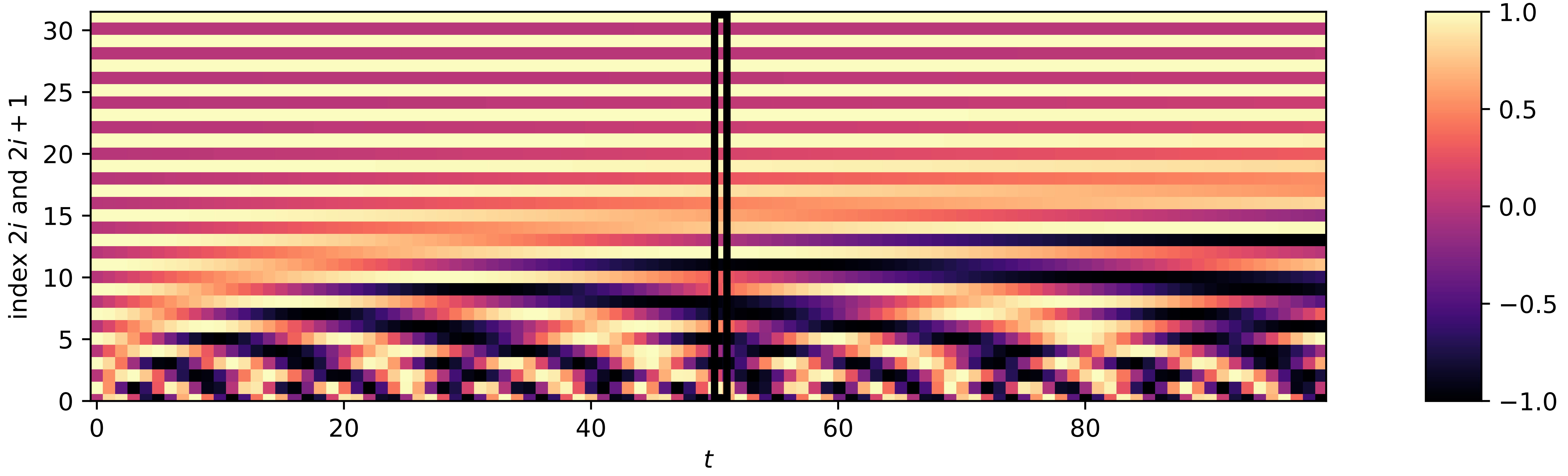
- Successfully trained a conditional DDPM to generate dark matter convergence maps
  - Linear beta scheduler, 2000 time-steps, 3 layer UNet, standardised data and ancestral sampling yield best results
  - Best results for higher matter density values
  - Large scale structure can be improved
- Slightly outperformed by CWGAN
  - With more time to fine-tune the DDPM could surpass the CWGAN

Neural Networks: DDPM

Sinusoidal position embedding

$$PE(\mathrm{t}, 2i) = \sin\left(\frac{\mathrm{t}}{n^{2i/d}}\right)$$

$$PE(\mathrm{t}, 2i+1) = \cos\left(\frac{\mathrm{t}}{n^{2i/d}}\right)$$



→ Encode each time-step onto a unique sinusoidal wave

→ Network learns relative position of one time-step to another