Pixel-Level Residual Diffusion Transformer

Scalable 3D CT Volume Generation

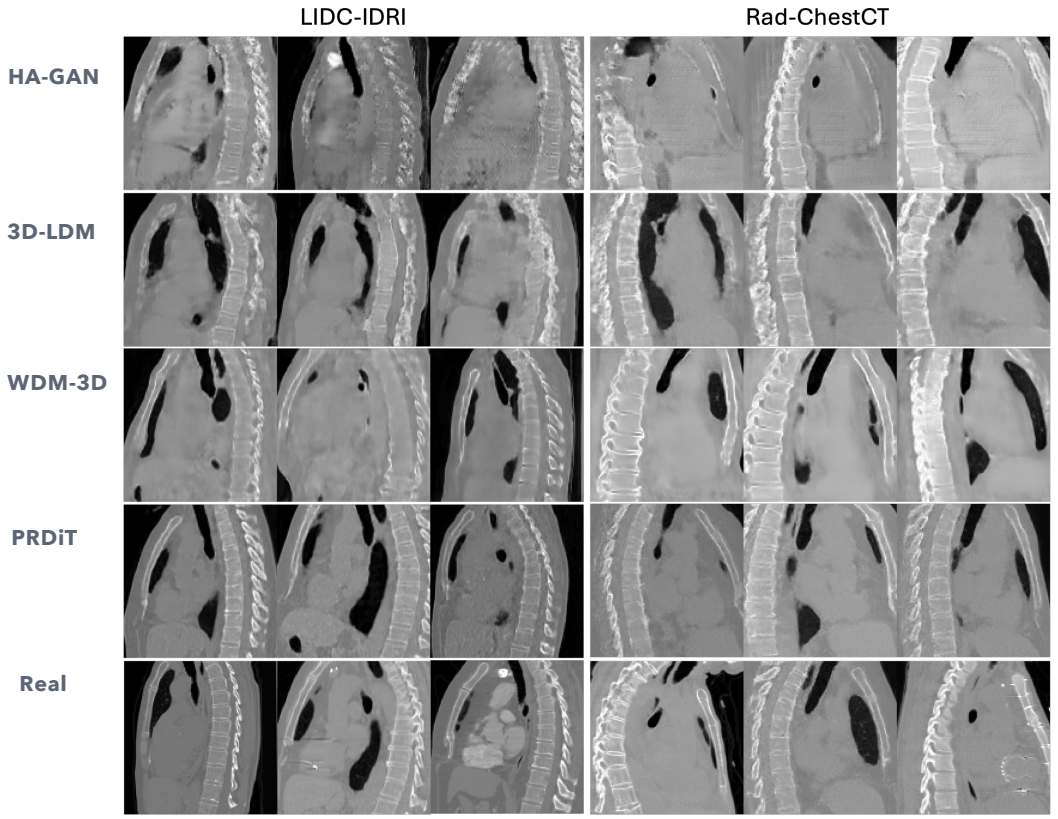
Zhenkai Zhang, Markus Hiller, Krista A. Ehinger, Tom Drummond
School of Computing and Information Systems, The University of Melbourne

Core idea
Local + global residuals

Resolution
128^3 **and** 256^3

Why it matters
No autoencoder bottleneck

This talk shows how PRDiT keeps local anatomy sharp, preserves global coherence, and scales high-resolution 3D CT synthesis without retraining everything from scratch.



Best 128^3 FID: 1.41 on LIDC-IDRI and 1.45 on RAD-ChestCT
Rows top to bottom: HA-GAN, 3D-LDM, WDM-3D, PRDiT, real data

Generated samples are **unconditional** and independent.

Motivation

Why 3D CT generation is still hard

High fidelity in 3D needs both local detail and long-range anatomical consistency.

Why prior 3D methods struggle

- 128^3 to 256^3 increases voxel count by 8x and attention cost by about 64x.
- Patching or downsampling reduces memory but weakens global anatomical context.
- Latent compression can discard subtle local detail in 3D CT volumes.

We want voxel-level synthesis without losing either local detail or whole-volume coherence.

PRDiT design target

Keep the easy part local

A lightweight denoiser estimates coarse patch content and local noise first.

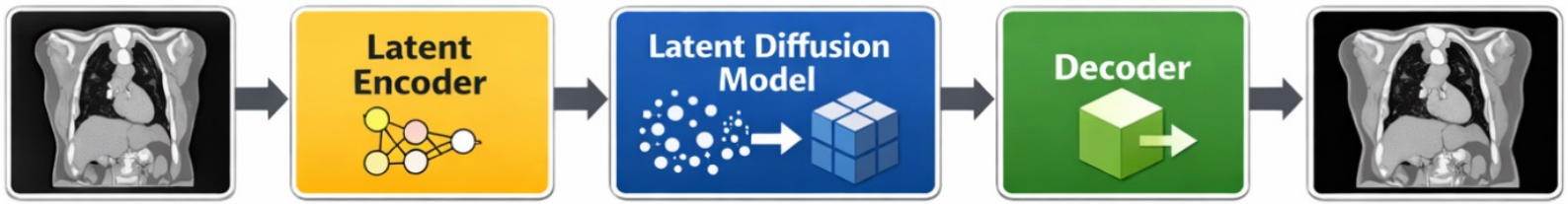
Refine only the missing global residual

A residual transformer spends capacity only on the full-volume corrections that remain.

Result: easier optimization

PRDiT in three steps

1. **Local Denoiser**
   Patch-wise clean signal and noise estimates
2. **Global Residual DiT**
   Global correction across all patches
3. **High-resolution reuse**
   Freeze low-res backbone and learn high-frequency detail

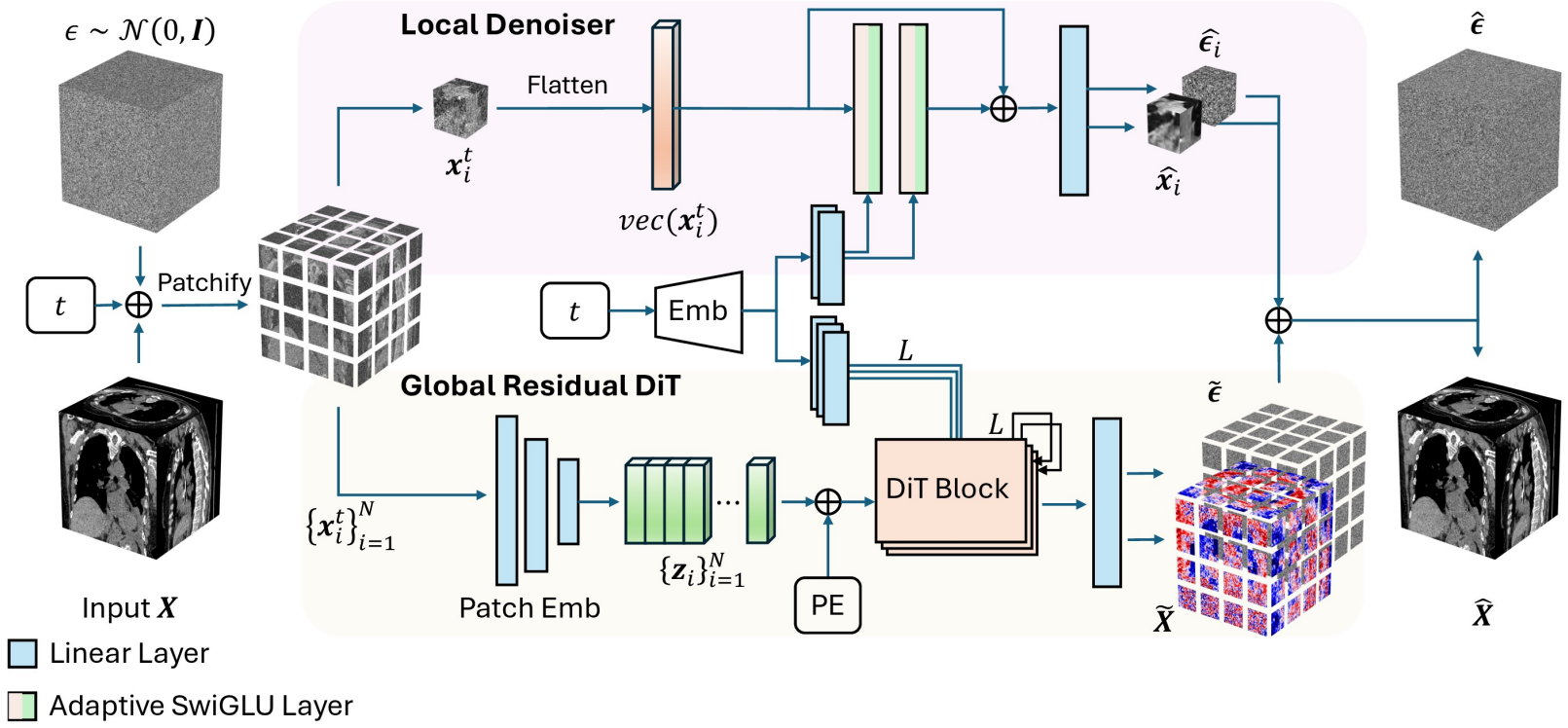


Existed method

Method

PRDiT: local first, global second

A strong local prior makes the global transformer lighter and easier to train.



Local Denoiser

Overlapping 3D patches produce clean-signal and noise estimates.

Global Residual DiT

The transformer attends globally and corrects only the missing residuals.

Why this helps

Voxel-level generation becomes easier because the model no longer learns the whole 3D signal at once.

Method

High-resolution reuse at 256^3

Freeze the 128^3 backbone and learn only the missing residual detail at high resolution.

Three-step recipe

1 Reuse 128^3 prior

Keep the pretrained low-resolution model frozen.

2 Add residual branch

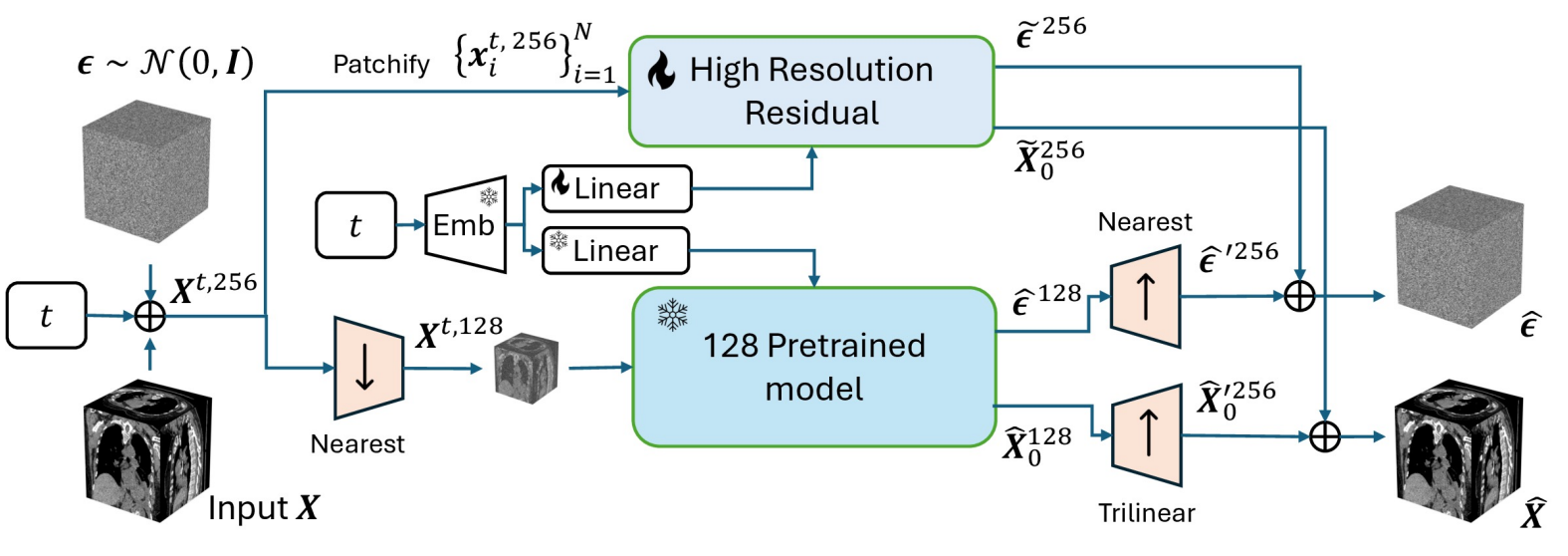
Predict only the missing 256^3 high-frequency detail.

3 Upsample and fuse

Combine reused structure with residual corrections.

This avoids retraining the full high-resolution backbone from scratch.

256^3 residual pipeline



Key idea

- Reuse coarse structure from 128^3, and spend new capacity only on the residual detail.
- This keeps 256^3 practical while preserving fine voxel detail.

Method

Predictor-corrector sampling

We first jump deterministically, then add controlled stochastic correction to recover diversity.

Sampling process

1 Cold predictor

Take a deterministic jump using the model prediction.

2 Hot corrector

Add controlled stochastic correction to preserve variance and improve exploration.

Why this helps

This balances stability and diversity better than cold sampling alone.

Algorithm 1: Predictor-Corrector Sampling

Input: Initial sample $x_T \in \mathbb{R}^{B\times C\times D\times H\times W}$;
Diffusion model f_θ with $f_\theta(x_t, t) := (\hat{\epsilon}_t, \hat{x}_0^t)$;
Predictor step multiplier $k \geq 1$.

Output: Sample sequence $\mathcal{X} = (x_t)_{t=0}^T$ and predictions $\hat{\mathcal{X}}_0 = (\hat{x}_0^t)_{t=1}^T$.

Initialize: $\mathcal{X} \leftarrow x_T$, angle map $\beta_t \leftarrow \frac{\pi}{2}\frac{t}{T}$.
for $t = T, T-1, \ldots, 1$ **do**
  $x_t \leftarrow \text{last}(\mathcal{X})$
  $(\hat{\epsilon}_t, \hat{x}_0^t) \leftarrow f_\theta(x_t, t)$ ▷ Model predictions at time t
  $g_t \leftarrow \sin(\beta_t)\,\hat{x}_0^t - \cos(\beta_t)\,\hat{\epsilon}_t$ ▷ Gradient direction in cosine plane
  $k \leftarrow \min(k, t)$ ▷ Avoid $t-k<0$, when t is small
  $\Delta\beta \leftarrow \beta_t - \beta_{t-k}$
  $x_{t-k} \leftarrow x_t - \Delta\beta \cdot g_t$ ▷ Predictor: k-step jump to time $(t-k)$
  $\Gamma_t^{(k)} \leftarrow \cos(\beta_{t-1})/\cos(\beta_{t-k})$ ▷ Scaling for variance preservation
  $\epsilon' \sim \mathcal{N}(0, I)$
  $x_{t-1} \leftarrow \Gamma_t^{(k)} x_{t-k} + \sqrt{1-(\Gamma_t^{(k)})^2}\,\epsilon'$ ▷ Corrector: variance preservation
  $\mathcal{X} \leftarrow \mathcal{X} \cup \{x_{t-1}\}$
end
return $(\hat{\mathcal{X}}_0, \mathcal{X})$

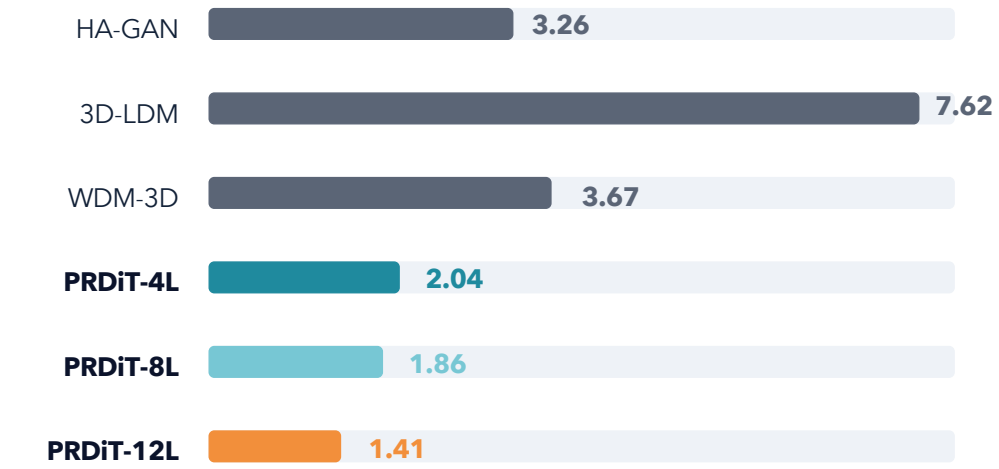
Experimental Results

PRDiT reaches state-of-the-art fidelity at 128^3

Even the 4-layer model beats prior work, and deeper variants keep improving.

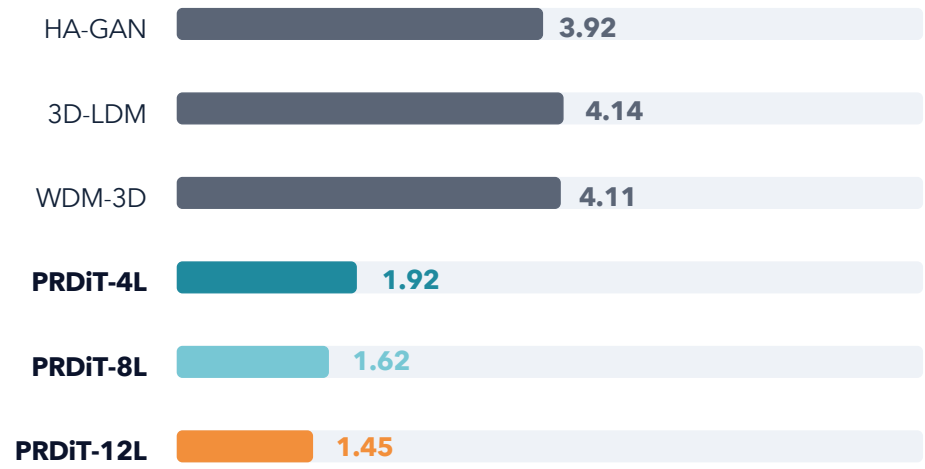
LIDC-IDRI FID

Lower is better



RAD-ChestCT FID

Lower is better



Best LIDC MMD: 0.1501

Best RAD FID: 1.45

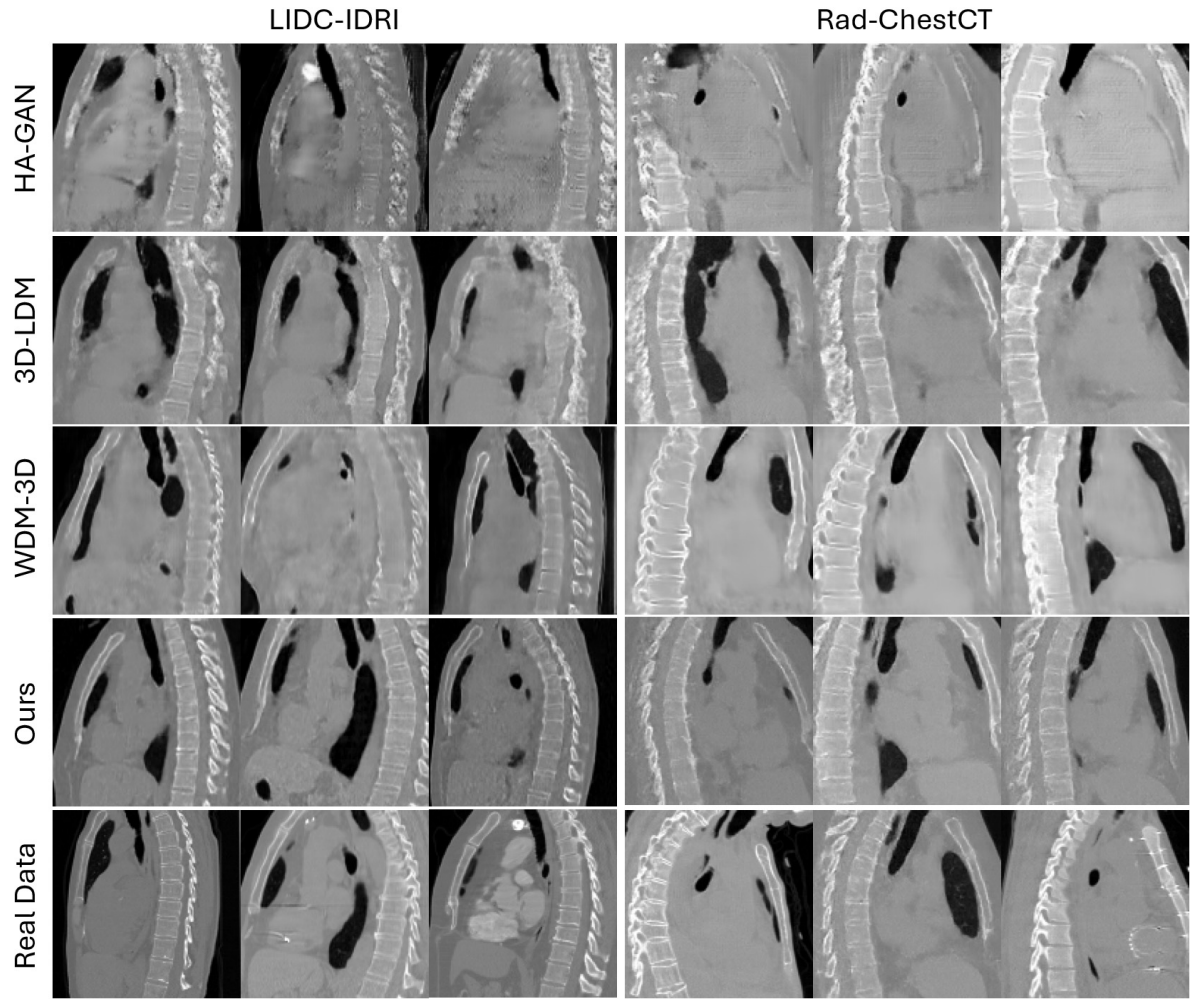
4L already beats all baselines

Depth from 4L to 12L keeps improving quality

Experimental Results

What the samples look like

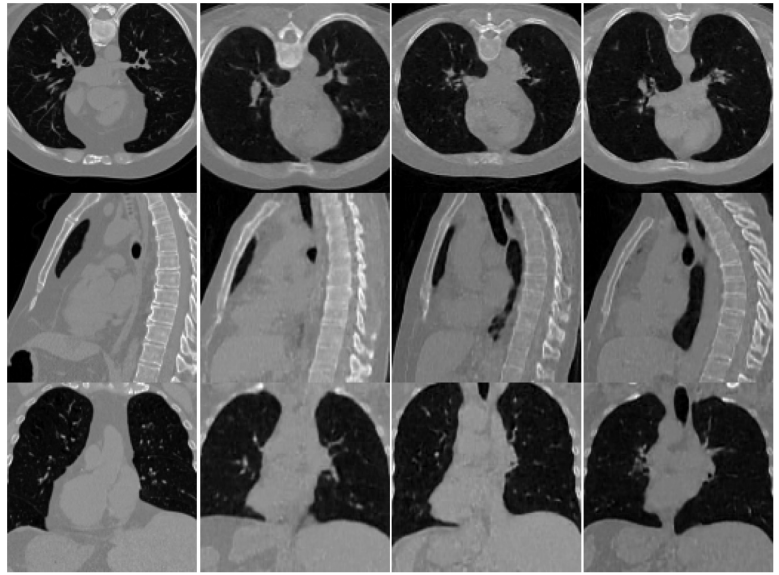
PRDiT produces sharper anatomical detail, and deeper variants make boundaries and fine structures cleaner.



Qualitative comparison on LIDC-IDRI and RAD-ChestCT across prior work, PRDiT, and real data.

Depth study

More layers sharpen the result



What stands out

- Sharper vertebral and bone edges
- Smoother organ boundaries with fewer blocky artifacts
- Clearer airways, vessels, and subtle structures

Samples are unconditional and should not be compared column-wise.

Thank you

Code, and contact information for PRDiT.

Code

Official PRDiT repository

GitHub

https://github.com/Fredy-Zhang/PRDiT

Public release and updates

Contact

Zhenkai Zhang

The University of Melbourne

zhenkaiz@student.unimelb.edu.au

Google Scholar

Questions and collaborations are welcome