fastWDM3D: Fast and Accurate 3D Healthy Tissue Inpainting

Alicia Durrer[1] [0009-0007-8970-909X], Florentin Bieder[1] [0000-0001-9558-0623], Paul Friedrich[1] [0000-0003-3653-5624], Bjoern Menze[2] [0000-0003-4136-5690], Philippe C. Cattin[1] [0000-0001-8785-2713]⋆, and Florian Kofler[2,3,4,5] [0000-0003-0642-7884]⋆

[1] Department of Biomedical Engineering, University of Basel, Switzerland
[2] Department of Quantitative Biomedicine, University of Zurich, Switzerland
[3] Helmholtz AI, Helmholtz Zentrum München, Germany
[4] Department of Diagnostic and Interventional Neuroradiology, School of Medicine, Klinikum rechts der Isar, Technical University of Munich, Germany
[5] TranslaTUM - Central Institute for Translational Cancer Research, Technical University of Munich, Germany

Abstract. Healthy tissue inpainting has significant applications, including the generation of pseudo-healthy baselines for tumor growth models and the facilitation of image registration. In previous editions of the *BraTS Local Synthesis of Healthy Brain Tissue via Inpainting Challenge*, denoising diffusion probabilistic models (DDPMs) demonstrated qualitatively convincing results but suffered from low sampling speed. To mitigate this limitation, we adapted a 2D image generation approach, combining DDPMs with generative adversarial networks (GANs) and employing a variance-preserving noise schedule, for the task of 3D inpainting. Our experiments showed that the variance-preserving noise schedule and the selected reconstruction losses can be effectively utilized for high-quality 3D inpainting in a few time steps without requiring adversarial training. We applied our findings to a different architecture, a 3D wavelet diffusion model (*WDM3D*) that does not include a GAN component. The resulting model, denoted as *fastWDM3D*, obtained a SSIM of 0.8571, a MSE of 0.0079, and a PSNR of 22.26 on the *BraTS* inpainting test set. Remarkably, it achieved these scores using only two time steps, completing the 3D inpainting process in 1.81 s per image. When compared to other DDPMs used for healthy brain tissue inpainting, our model is up to ∼ 800× faster while still achieving superior performance metrics. Our proposed method, *fastWDM3D*, represents a promising approach for fast and accurate healthy tissue inpainting. Our code is available at https://github.com/AliciaDurrer/fastWDM3D.

Keywords: Healthy Tissue Inpainting · 3D Diffusion Model · Efficient.

⋆ equal contribution

1 Introduction

Tumor growth datasets often lack images of initially healthy brains, which would be essential for accurately predicting tumor progression [7]. Furthermore, many automated brain magnetic resonance imaging (MRI) analysis tools, e.g., for segmentation, are trained exclusively on healthy data and perform better when pathological tissue is replaced with healthy tissue [4]. Therefore, healthy tissue inpainting is a crucial step in both scenarios. The *Brain Tumor Segmentation (BraTS)* challenge [2,3,22] features an inpainting sub-challenge, highlighting the importance of this task. Inpainting challenge results [5,14,19,30] and a subsequent study [6], comparing different denoising diffusion probabilistic models (DDPMs) [13] on the *BraTS* inpainting test set, show that methods using DDPMs yield promising outcomes. However, a major drawback is their long inference time. In this work, we present a modification of an inpainting-specific 3D wavelet diffusion model (*WDM3D*) [6,9]. Our modification, called *fastWDM3D*, only requires two time steps compared to the original 1000, but still manages to achieve better scores on the *BraTS* inpainting test set.

1.1 Related Work

DDPMs [13] produce diverse, high-quality outputs but require long sampling times, whereas generative adversarial networks (GANs) [11] quickly generate high-quality samples but often face mode collapse. In [28], Xiao et al. summarize this as the *Generative Trilemma* and suggest modeling each denoising step in a diffusion model with multimodal conditional GANs (*DDGANs*) to generate high-quality, diverse outputs fast. By using *DDGANs* on wavelet transformed images (*WDDGAN*), Phung et al. [23] further increase inference speed. Speeding up DDPMs is a recurring research topic: Reducing the number of time steps in training and inference has also been explored by e.g., [8,16,29]. Furthermore, distillation [25] or an optimized choice of time steps [20,31] have been used to decrease inference time. Finding an efficient variance schedule, controlling the noise perturbation process, has been the focus of several works [18,21,26,27].

1.2 Contribution

In this study, we adapted the *WDDGAN* [23], originally designed for 2D natural image generation, for a 3D inpainting task. Our experiments revealed that the variance-preserving schedule [27] used in *WDDGAN* is crucial for performance, while the adversarial training does not yield significant benefits for our application. Consequently, we eliminated the discriminator from the architecture, not only simplifying the model but also accelerating the training process without compromising the quality of the generated outputs. To further substantiate the significance of the variance-preserving schedule and the reconstruction losses used, we integrated these components into *WDM3D* [6,9], a DDPM-only architecture without any GAN component. The resulting model, termed *fastWDM3D*, demonstrated superior performance compared to other DDPMs [6] evaluated on the *BraTS* inpainting test set. Notably, it outperformed the original *WDM3D* model, which employs 1000 time steps, a linear variance schedule, and a mean squared error (MSE) loss applied to wavelet coefficients [6,9]. In general, our *fastWDM3D* achieves better performance metrics than all other assessed DDPMs, while preserving 3D consistency and being up to $\sim 800\times$ faster.

2 Methods

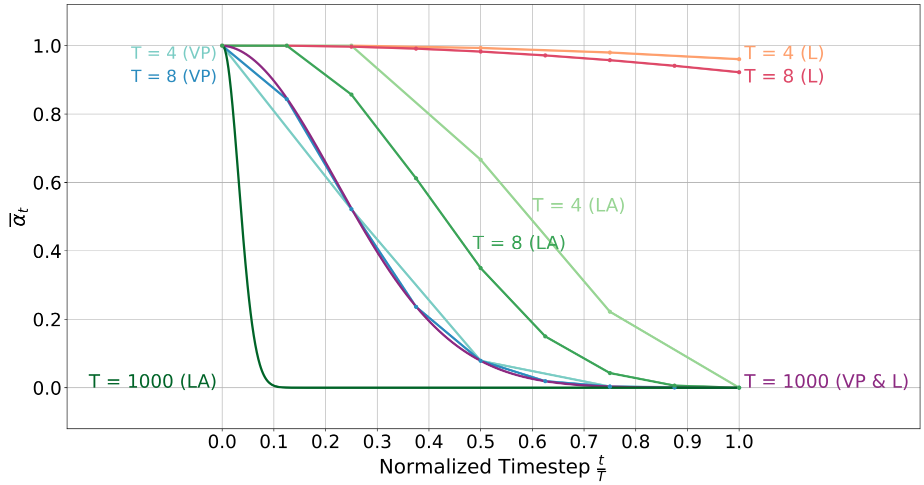

Fig. 1. Comparison of $\overline{\alpha}_t$ for the L-, LA- and VP schedule after normalizing all individual T to [0,1]. The L- and VP schedules have the same curve for $T = 1000$. The L schedule only provides full perturbation for large T while the LA schedule perturbs the image too early if T is large. The VP schedule is applicable for low and large T.

2.1 Denoising Diffusion Probabilistic Models

In DDPMs [13], noise gradually perturbs an input image x_0 for T time steps:

$$q(x_t|x_{t-1}) := \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t \boldsymbol{I}), \tag{1}$$

with $\boldsymbol{I}$ being the identity matrix and β_t being the variance at time step t. This is called the *forward process*. The *reverse* or *denoising process*

$$p_\theta(x_{t-1}|x_t) := \mathcal{N}\left(x_{t-1}; \mu_\theta(x_t, t), \sigma_t^2 \boldsymbol{I}\right) \tag{2}$$

follows a sequence of Gaussian distributions with mean μ_θ and variance σ_t^2, parameterized by a time-conditioned model $\epsilon_\theta(x_t, t)$. The goal of the reverse process is to match the true denoising distribution $q(x_{t-1}|x_t)$. The model $\epsilon_\theta(x_t, t)$ can be trained to predict x_0, which then can be used to sample x_{t-1} using the posterior distribution $q(x_{t-1}|x_t, x_0)$.

2.2 Discrete Wavelet Transform

Using generative models on the wavelet coefficients of the input images has been described in several works [9,10,12,23]. The discrete wavelet transform (DWT) applies several low- and high-pass filters with stride 2, $l = \frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\end{bmatrix}$ and $h = \frac{1}{\sqrt{2}}\begin{bmatrix}-1 & 1\end{bmatrix}$, along all spatial dimensions. A 3D volume y is decomposed into 8 wavelet coefficients $(x_{lll}, x_{llh}, x_{lhl}, x_{lhh}, x_{hll}, x_{hlh}, x_{hhl}, x_{hhh})$ of half the spatial resolution of y. The coefficients can be concatenated into a single matrix x. The inverse discrete wavelet transform (IDWT) of x restores the original image y.

2.3 Variance-Preserving Noise Schedule

Song et al. [27] showed that the noise perturbations in DDPMs, described by Eq. 1, correspond to a discretization of the stochastic differential equation (SDE) $dx = -\frac{1}{2}\beta_t x\, dt + \sqrt{\beta_t}\, dw$, where w is the standard Wiener process. This SDE is called Variance-Preserving (VP), as it leads to a process with bounded variance as $t \to \infty$. Xiao et al. [28] rely on this SDE to compute the VP variance schedule

$$\beta_t = 1 - \exp\Big(- \beta_{\min}\frac{t}{T} - 0.5(\beta_{\max} - \beta_{\min})\frac{2t-1}{T^2}\Big) \quad \text{for } t = 1, 2, \ldots, T, \tag{3}$$

ensuring that the overall noise perturbation is independent of the number of diffusion steps, allowing full perturbation even if T is low. As β_t determines the amount of noise added at each t, $\alpha_t := 1 - \beta_t$ represents the original signal preservation at each t and $\overline{\alpha}_t := \prod_{s=1}^{t} \alpha_s$ accumulates the original signal preservation up until this t. Thus, the closer $\overline{\alpha}_t$ is to zero, the more the image is perturbed. The variance schedule defines the extent and speed of perturbation in the *forward process*, whereby full perturbation is required but should not happen too early. Figure 1 shows the comparison of the linear variance schedule L (with $\beta_1 = 10^{-4}$ and $\beta_T = 0.02$ as in [13]), and the VP schedule (Eqn. 3 with $\beta_{\min} = 0.1$ and $\beta_{\max} = 20$ to match the L schedule for $T = 1000$ [27]). The VP schedule allows efficient noise perturbation using small and big T. In contrast, the L schedule does not perturb the image if T is small. Scaling the L schedule by multiplying β_1 and β_T by $\frac{1000}{T}$ is not possible, as $\beta > 1$ if $T < 20$. Thus, we also provide an adapted linear schedule (LA), with $\beta_1 = 10^{-4}$ and $\beta_T = 0.9999$. We chose this β_T to ensure that the image is fully perturbed in the *forward process* when using small T. However, this is not applicable for large T, shown in Fig. 1 for $T = 1000$, as it destroys all image information in early stages already.

2.4 Denoising Diffusion Generative Adversarial Networks

In *DDGAN*, the denoising distribution of a DDPM is modeled with an expressive multimodal distribution using a GAN. Fake samples from $p_\theta(x_{t-1}|x_t)$ are evaluated against real samples from $q(x_{t-1}|x_t)$. An adversarial loss, $\mathcal{L}_{adv}$, minimizes the softened reverse KL divergence per denoising step. The generator is based on the NCSN++ architecture [27] and is conditioned on a latent variable provided to the NCSN++ using a mapping network, originating from StyleGAN [17]. *WDDGAN* further reduces sampling times by working in the wavelet domain.

2.5 Models Used for Healthy Tissue Inpainting

We employed three distinct network architectures, listed below, that we adapted for inpainting through *Palette*-conditioning [24]. For the training process, we used the ground truth image (g), a mask masking out some healthy tissue (m), and the resulting voided image (v). The wavelet coefficients of v, m, and the noisy g were concatenated to form a 24-channel input. This input was then processed by the respective model, generating an 8-channel output, which was subsequently subjected to the IDWT to reconstruct the inpainted prediction, denoted as $\hat{y}$.

- *WDDGAN3D*: We extended the *WDDGAN* to a 3D inpainting network. In addition to the adversarial loss $\mathcal{L}_{adv}$ we incorporated a reconstruction loss $\mathcal{L}_{\hat{y}}$ between g and $\hat{y}$. Additionally, we computed a region-specific reconstruction loss $\mathcal{L}_{\hat{y}_m}$ focused on the masked area m of both g and $\hat{y}$. The total loss $\mathcal{L}$ was defined as $\mathcal{L} = \mathcal{L}_{adv} + \mathcal{L}_{\hat{y}} + \mathcal{L}_{\hat{y}_m}$. We assessed the performance for $T \in \{4, 8\}$ in different setups, including adjustments to the frequency of generator updates relative to the discriminator and variations in learning rates. However, all of them led to the same outcome: $\mathcal{L}_{adv}$ did not decrease during training. We report the vanilla configuration in Table 1.
- *GO3D*: Given that $\mathcal{L}_{adv}$ did not decrease during training, we removed the discriminator and the associated $\mathcal{L}_{adv}$, resulting in a simplified loss function defined as $\mathcal{L} = \mathcal{L}_{\hat{y}} + \mathcal{L}_{\hat{y}_m}$. This model is referred to as *Generator-only 3D* (*GO3D*) and we evaluted its performance for $T \in \{2, 4, 8, 16, 64, 256, 1000\}$.
- *fastWDM3D*: Building on the work of Durrer et al. [6], who demonstrated the efficacy of the wavelet diffusion model 3D (*WDM3D*) by Friedrich et al. [9] for inpainting tasks, we modified the noise schedule of the *WDM3D* by implementing the variance-preserving (VP) schedule. Additionally, we replaced the MSE loss on the wavelet coefficients with the loss function $\mathcal{L} = \mathcal{L}_{\hat{y}} + \mathcal{L}_{\hat{y}_m}$. Our training utilized time steps $T \in \{2, 4, 8\}$, contrasting with the $T = 1000$ employed in [6]. This model is termed *fastWDM3D.*

3 Experimental Details and Results

Dataset We used the publicly available dataset of the *BraTS 2023 Local Synthesis of Healthy Brain Tissue via Inpainting Challenge* [2,3,15,22] for training. It contains T1 scans showing brain tumors, as well as masks of the tumor and masks of some regions showing only healthy tissue (called healthy masks). From the 1251 patient scans, we used 1200 to train our models and 51 for validation to observe model performance during training. The non-public *BraTS* inpainting test set contains T1 scans, healthy masks and tumor masks of 568 patients. All scans had an initial resolution of $240 \times 240 \times 155$. We removed the top and bottom 0.5 percentile of voxel intensities and normalized them between -1 and 1. For training, we cropped out the region defined by the healthy mask m in the T1 image g, generating a voided image v. The images v, m and g were all cropped to $128 \times 128 \times 128$ with the region to be inpainted in the center. Each prediction $\hat{y}$ was normalized back to the individual input range of v for evaluation.

Experiments and Results All models were trained on a NVIDIA A100 (40 GB) GPU. All implementation details can be found at https://github.com/AliciaDurrer/fastWDM3D. We first compared *WDDGAN3D* and *GO3D*, see Table 1. For *GO3D* we explored the LA and the VP schedule, for *WDDGAN3D* we used the VP schedule only. We evaluated the models after 100 epochs, as we could observe convergence followed by overfitting for all model types. We report training duration and memory requirement, as well as SSIM, MSE and PSNR on the *BraTS* inpainting test set. *GO3D(LA)* showed the best performance regarding MSE, however, it was worse than *WDDGAN3D* and *GO3D* regarding SSIM and PSNR. As *GO3D(VP)* performed better than *WDDGAN3D* across all metrics while requiring less memory and significantly less training time, we evaluated *GO3D(VP)* for further time steps. An additional reason for keeping the VP instead of the LA schedule was that the VP schedule can be used independent of the number of time steps T without requiring any modifications (see Fig.1). Since our goal is to keep T as low possible, this is not decisive, but we want to provide a robust method that allows changes if required. We then compared *GO3D(VP)* to *fastWDM3D*, also using the VP schedule, in Table 2. We report SSIM, MSE, PSNR, and average sampling time per volume on the *BraTS* inpainting test set. For these experiments, the batch size was 3 instead of 2 used for the experiments in Table 1. We report the scores obtained on the test set after training the models for 100 epochs ($40 \cdot 10^3$ iterations) as there was no further improvement of *GO3D* on the validation set after these epochs. For *fastWDM3D*, we additionally provide the results after 300 epochs ($120 \cdot 10^3$ iterations), as we saw that its performance on the validation set kept increasing after 100 epochs. We also report the results for *GO3D* after 300 epochs for $T \in \{64, 256, 1000\}$ to demonstrate that more training iterations do not improve performance compared to lower T using less iterations. *GO3D* requires 38.8 GiB and *fastWDM3D* requires 18.3 GiB during training. Based on Tables 1 and 2, we observe that *GO3D* trained with a batch size of 2 and $T = 8$ achieves the best SSIM, while *fastWDM3D* trained for $120 \cdot 10^3$ iterations with a batch of size 3

Table 1. Comparison of *WDDGAN3D* using the VP schedule, *GO3D* using the LA schedule, and *GO3D* using the VP schedule on the *BraTS* inpainting test set. For all experiments: batch size: 2, residual blocks in generator: 4, learning rate generator: $2 \cdot 10^{-5}$, trained for 100 epochs ($60 \cdot 10^3$ iterations). Abbreviations: T = time steps, TT = training time, Mem = memory required during training.

Type	T	TT (↓)	Mem (↓)	SSIM (↑)	MSE (↓)	PSNR (↑)
WDDGAN3D(VP)	4	∼112 h	31.69 GiB	0.8562 ± 0.1169	0.0082 ± 0.0063	21.92 ± 3.74
WDDGAN3D(VP)	8	∼112 h	31.69 GiB	0.8587 ± 0.1165	0.0081 ± 0.0064	22.08 ± 3.84
GO3D(VP)	4	∼11 h	24.24 GiB	0.8595 ± 0.1162	0.0079 ± 0.0062	22.17 ± 3.84
GO3D(VP)	8	∼11 h	24.24 GiB	0.8606 ± 0.1145	0.0079 ± 0.0061	22.19 ± 3.83
GO3D(LA)	4	∼11 h	24.24 GiB	0.8521 ± 0.1242	0.0066 ± 0.0065	21.69 ± 4.10
GO3D(LA)	8	∼11 h	24.24 GiB	0.8438 ± 0.1409	0.0061 ± 0.0049	21.83 ± 3.54

Table 2. Comparison of *GO3D* and *fastWDM3D* on the *BraTS* inpainting test set, both use the VP schedule. For all experiments: batch size: 3, residual blocks: 4, learning rate: $2 \cdot 10^{-5}$. Abbreviations: T = time steps, Iter = training iterations (as a multiple of 10^3), AST = average sampling time per volume (can vary as it depends on the overall server utilization).

Type	T	Iter	SSIM (↑)	MSE (↓)	PSNR (↑)	AST (↓)
GO3D	2	40	0.8553 ± 0.1165	0.0084 ± 0.0064	21.77 ± 3.68	1.45 s
GO3D	4	40	0.8596 ± 0.1147	0.0080 ± 0.0061	22.03 ± 3.69	2.37 s
GO3D	8	40	0.8596 ± 0.1156	0.0080 ± 0.0062	22.08 ± 3.74	4.25 s
GO3D	16	40	0.8599 ± 0.1155	0.0079 ± 0.0061	22.11 ± 3.73	7.95 s
GO3D	64	40	0.8567 ± 0.1175	0.0083 ± 0.0643	21.93 ± 3.77	62.87 s
GO3D	256	40	0.8518 ± 0.1200	0.0090 ± 0.0071	21.65 ± 3.91	118.46 s
GO3D	1000	40	0.8483 ± 0.1206	0.0092 ± 0.0070	21.52 ± 3.92	526.67 s
GO3D	64	120	0.8496 ± 0.1228	0.0085 ± 0.0067	21.91 ± 3.94	62.87 s
GO3D	256	120	0.8502 ± 0.1226	0.0084 ± 0.0066	21.97 ± 3.98	118.46 s
GO3D	1000	120	0.8507 ± 0.1213	0.0084 ± 0.0066	21.97 ± 3.97	526.67 s
fastWDM3D	2	40	0.8541 ± 0.1177	0.0090 ± 0.0066	21.42 ± 3.56	1.81 s
fastWDM3D	4	40	0.8366 ± 0.1281	0.0121 ± 0.0091	20.17 ± 3.70	3.78 s
fastWDM3D	8	40	0.8072 ± 0.1481	0.0167 ± 0.0135	18.96 ± 3.97	4.10 s
fastWDM3D	2	120	0.8571 ± 0.1193	0.0079 ± 0.0063	22.26 ± 3.97	1.81 s
fastWDM3D	4	120	0.8566 ± 0.1185	0.0081 ± 0.0065	22.20 ± 4.03	3.78 s
fastWDM3D	8	120	0.8561 ± 0.1192	0.0079 ± 0.0063	22.24 ± 4.01	4.10 s

Table 3. Comparison of the *BraTS* 2023 inpainting challenge podium and our best model, *fastWDM3D* ($T = 2$), on the *BraTS* inpainting test set.

	SSIM (↑)	MSE (↓)	PSNR (↑)
Zhang et al. (1st 2023) [30]	0.91 ± 0.15	0.0049 ± 0.0016	23.59 ± 5.35
Durrer et al. (2nd 2023) [5]	0.86 ± 0.20	0.0100 ± 0.0016	20.42 ± 3.82
Huo et al. (3rd 2023) [14]	0.87 ± 0.18	0.0144 ± 0.0025	18.71 ± 4.01
Ours (*fastWDM3D* ($T = 2$))	0.86 ± 0.12	0.0079 ± 0.0063	22.26 ± 3.97

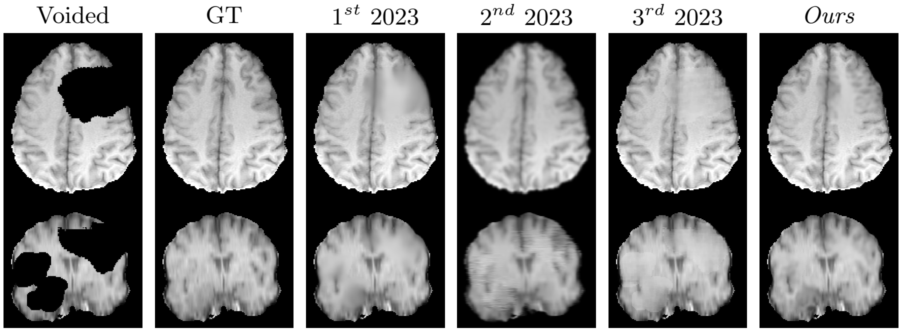

Fig. 2. Axial (top) and coronal (bottom) view of an image of the validation set. Comparison of the *BraTS* 2023 inpainting challenge podium and our best model, *fastWDM3D* ($T = 2$), given the voided input image (voided) and the ground truth (GT).

Table 4. Comparison of DDPM configurations by Durrer et al. [6] and our best model, *fastWDM3D* ($T = 2$), on the *BraTS* inpainting test set. Abbreviation: AST = average sampling time per volume (can vary as it depends on the overall server utilization).

	SSIM (↑)	MSE (↓)	PSNR (↑)	AST (↓)
DDPM 2D slice-wise [6]	0.78 ± 0.15	0.0160 ± 0.0118	18.71 ± 3.08	20 min
DDPM 2D seq-pos [6]	0.77 ± 0.20	0.0420 ± 0.1136	18.45 ± 5.28	25 min
DDPM Pseudo3D [6]	0.85 ± 0.12	0.0103 ± 0.0107	20.93 ± 3.38	25 min
DDPM 3D mem-eff [6]	0.70 ± 0.16	0.0523 ± 0.0222	14.14 ± 3.18	20 min
LDM3D [6]	0.60 ± 0.12	0.0700 ± 0.0280	8.77 ± 1.89	1 min
WDM3D [6]	0.61 ± 0.16	0.1060 ± 0.0757	10.57 ± 3.20	5 min
Ours (*fastWDM3D* ($T = 2$))	0.86 ± 0.12	0.0079 ± 0.0063	22.26 ± 3.97	1.81 s

and $T = 2$ obtains the best PSNR. The rounded MSE is the same for both. These methods are highlighted in yellow and green in Table 1 and 2, respectively. As *fastWDM3D* ($T = 2$) additionally has the lowest average sampling time (1.81s) and uses significantly less memory during training (18.3 GiB), we labeled this as our best configuration and compared it qualitatively and quantitatively to the *BraTS* 2023 inpainting challenge podium [5,14,19,30]. The challenge proceedings of 2024 are not available yet. For the qualitative comparison, we sampled images of the 2023 challenge podium methods using the publicly available algorithms [1] and provide exemplary axial and coronal slices of the inpainting generated by all methods in Fig. 2. The scores obtained on the *BraTS* inpainting test set are reported in Table 3. In a final step, we compared our best configuration to the different DDPMs by [6], summarized in Table 4. Among these configurations is their *WDM3D* using a linear variance schedule, an MSE loss on the wavelet coefficients and $T = 1000$.

4 Discussion and Conclusion

Upon analyzing the results in Tables 1 and 2, we see that the SSIM, MSE and PSNR are on a similar level for all evaluated model types (*WDDGAN3D*, *GO3D*, *fastWDM3D*). However, the *fastWDM3D* has further benefits: The configuration achieving the highest scores, *fastWDM3D* ($T = 2$), is also the fastest in sampling, requiring on average only 1.81 s for the full 3D inpainting. In addition, its slimmer architecture uses much less memory than *WDDGAN3D* and *GO3D*. Comparing our *fastWDM3D* ($T = 2$) to the podium of the *BraTS 2023 Local Synthesis of Healthy Brain Tissue via Inpainting Challenge* [19], shown in Table 3, we see that our method outperforms the methods achieving the second [5] and third [14] place in terms of MSE and PSNR while being similar in terms of SSIM. The method placed first [30] in the challenge achieves quantitatively better scores than our proposed method. However, the qualitative results presented in Fig. 2 show that our method inpaints 3D realistic structures, while the methods achieving the first and third place create more blurry and less defined inpainting. The second place, a 2D DDPM, generates realistic inpainting in the axial plane, but, the 2D nature of the model leads to stripe artifacts visible in the coronal view. Compared to the diffusion model setups by [6], presented in Table 4, our method achieves better SSIM, MSE and PSNR scores while being up to $\sim 800\times$ faster during sampling. The key role of the VP schedule and the reconstruction losses is visible in Table 4 when comparing our scores to those of the *WDM3D* inpainting implementation by [6]. Changing the schedule and the loss allowed a vast improvement in the scores while using 500× fewer time steps.

Our proposed method, *fastWDM3D*, shows promising potential for clinical applications as it generates high-quality inpainting in a short amount of time. Fast training and sampling, as well as low memory requirements result in a smaller environmental footprint. Future research will include a closer investigation into why the VP schedule enables this performance and speed boost. Moreover, we aim to explore additional schedules, as the simply designed LA schedule already appears to be a promising approach. In addition, an ablation study disentangling the influence of the variance schedule, the loss, and the model architecture would be beneficial. Experimenting with different datasets and modalities, as well as exploring additional generative tasks (e.g., unconditional image generation), will further assess the robustness of our method.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. BraTS challenge: Top performing algorithms. https://github.com/BrainLesion/BraTS, last accessed 2025/06/24
2. Baid, U., et al.: The rsna-asnr-miccai brats 2021 benchmark on brain tumor segmentation and radiogenomic classification. arXiv preprint arXiv:2107.02314 (2021)
3. Bakas, S., et al.: Advancing the cancer genome atlas glioma mri collections with expert segmentation labels and radiomic features. Scientific data **4**(1), 1–13 (2017)
4. Dadar, M., Potvin, O., Camicioli, R., Duchesne, S., for the Alzheimer's Disease Neuroimaging Initiative: Beware of white matter hyperintensities causing systematic errors in freesurfer gray matter segmentations! Human Brain Mapping **42**(9), 2734–2745 (2021)
5. Durrer, A., Cattin, P.C., Wolleb, J.: Denoising diffusion models for inpainting of healthy brain tissue. In: International Challenge on Cross-Modality Domain Adaptation for Medical Image Segmentation, pp. 35–45. Springer (2023)
6. Durrer, A., et al.: Denoising diffusion models for 3d healthy brain tissue inpainting. In: MICCAI Workshop on Deep Generative Models. pp. 87–97. Springer (2024)
7. Ezhov, I., et al.: Learn-morph-infer: a new way of solving the inverse problem for brain tumor modeling. Medical Image Analysis **83**, 102672 (2023)
8. Frans, K., Hafner, D., Levine, S., Abbeel, P.: One step diffusion via shortcut models. arXiv preprint arXiv:2410.12557 (2024)
9. Friedrich, P., Wolleb, J., Bieder, F., Durrer, A., Cattin, P.C.: Wdm: 3d wavelet diffusion models for high-resolution medical image synthesis. In: MICCAI Workshop on Deep Generative Models. pp. 11–21. Springer (2024)

10. Gal, R., Hochberg, D.C., Bermano, A., Cohen-Or, D.: Swagan: A style-based wavelet-driven generative model. ACM Transactions on Graphics (TOG) **40**(4), 1–11 (2021)
11. Goodfellow, I.J., et al.: Generative adversarial nets. Advances in neural information processing systems **27** (2014)
12. Guth, F., Coste, S., De Bortoli, V., Mallat, S.: Wavelet score-based generative modeling. Advances in neural information processing systems **35**, 478–491 (2022)
13. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. Advances in neural information processing systems **33**, 6840–6851 (2020)
14. Huo, J., Liu, Y., Granados, A., Ourselin, S., Sparks, R.: Unleash the power of 2d pre-trained model for 3d t1-weighted brain mri inpainting. In: International Challenge on Cross-Modality Domain Adaptation for Medical Image Segmentation, pp. 3–10. Springer (2023)
15. Karargyris, A., Umeton, R., Sheller, M.J., et al.: Federated benchmarking of medical artificial intelligence with medperf. Nature Machine Intelligence **5**(7), 799–810 (2023)
16. Karras, T., Aittala, M., Aila, T., Laine, S.: Elucidating the design space of diffusion-based generative models. Advances in Neural Information Processing Systems **35**, 26565–26577 (2022)
17. Karras, T., Laine, S., Aila, T.: A style-based generator architecture for generative adversarial networks. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 4401–4410 (2019)
18. Kingma, D., Salimans, T., Poole, B., Ho, J.: Variational diffusion models. Advances in neural information processing systems **34**, 21696–21707 (2021)
19. Kofler, F., et al.: The brain tumor segmentation (brats) challenge 2023: Local synthesis of healthy brain tissue via inpainting. arXiv preprint arXiv:2305.08992 (2023)
20. Li, L., et al.: Autodiffusion: Training-free optimization of time steps and architectures for automated diffusion model acceleration. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 7105–7114 (2023)
21. Lin, S., Liu, B., Li, J., Yang, X.: Common diffusion noise schedules and sample steps are flawed. In: Proceedings of the IEEE/CVF winter conference on applications of computer vision. pp. 5404–5411 (2024)
22. Menze, B.H., et al.: The multimodal brain tumor image segmentation benchmark (brats). IEEE transactions on medical imaging **34**(10), 1993–2024 (2014)
23. Phung, H., Dao, Q., Tran, A.: Wavelet diffusion models are fast and scalable image generators. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 10199–10208 (2023)
24. Saharia, C., et al.: Palette: Image-to-image diffusion models. In: ACM SIGGRAPH 2022 conference proceedings. pp. 1–10 (2022)
25. Salimans, T., Ho, J.: Progressive distillation for fast sampling of diffusion models. arXiv preprint arXiv:2202.00512 (2022)
26. San-Roman, R., Nachmani, E., Wolf, L.: Noise estimation for generative diffusion models. arXiv preprint arXiv:2104.02600 (2021)
27. Song, Y., Sohl-Dickstein, J., Kingma, D.P., Kumar, A., Ermon, S., Poole, B.: Score-based generative modeling through stochastic differential equations. arXiv preprint arXiv:2011.13456 (2020)
28. Xiao, Z., Kreis, K., Vahdat, A.: Tackling the generative learning trilemma with denoising diffusion gans. arXiv preprint arXiv:2112.07804 (2021)

29. Ye, Z., Chen, Z., Li, T., Huang, Z., Luo, W., Qi, G.J.: Schedule on the fly: Diffusion time prediction for faster and better image generation. arXiv preprint arXiv:2412.01243 (2024)
30. Zhang, J., Chen, K., Weng, Y.: Synthesis of healthy tissue within tumor area via u-net. In: International Challenge on Cross-Modality Domain Adaptation for Medical Image Segmentation, pp. 233–240. Springer (2023)
31. Zheng, T., et al.: Beta-tuned timestep diffusion model. In: European Conference on Computer Vision. pp. 114–130. Springer (2024)