Introducing 3D Representation for Dense Volume-to-Volume Translation via Score Fusion

Xiyue Zhu [1] **Dou Hoon Kwark** [1] **Ruike Zhu** [1] **Kaiwen Hong** [1] **Yiqi Tao** [1] **Shirui Luo** [2] **Yudu Li** [1]
Zhi-Pei Liang [1] **Volodymyr Kindratenko** [1 2]

https://score-fusion.github.io/

Abstract

In volume-to-volume translations in medical images, existing models often struggle to capture the inherent volumetric distribution using 3D voxel-space representations, due to high computational dataset demands. We present Score-Fusion, a novel volumetric translation model that effectively learns 3D representations by ensembling perpendicularly trained 2D diffusion models in score function space. By carefully initializing our model to start with an average of 2D models as in existing models, we reduce 3D training to a fine-tuning process, mitigating computational and data demands. Furthermore, we explicitly design the 3D model's hierarchical layers to learn ensembles of 2D features, further enhancing efficiency and performance. Moreover, Score-Fusion naturally extends to multi-modality settings by fusing diffusion models conditioned on different inputs for flexible, accurate integration. We demonstrate that 3D representation is essential for better performance in downstream recognition tasks, such as tumor segmentation, where most segmentation models are based on 3D representation. Extensive experiments demonstrate that Score-Fusion achieves superior accuracy and volumetric fidelity in 3D medical image super-resolution and modality translation. Additionally, we extend Score-Fusion to video super-resolution by integrating 2D diffusion models on time-space slices with a spatial-temporal video diffusion backbone, highlighting its potential for general-purpose volume translation and providing broader insight into learning-based approaches for score function fusion.

[1]University of Illinois at Urbana-Champaign [2]National Center for Supercomputing Applications. Correspondence to: Xiyue Zhu <xiyuez2@illinois.edu>.

Proceedings of the 42^{nd} International Conference on Machine Learning, Vancouver, Canada. PMLR 267, 2025.

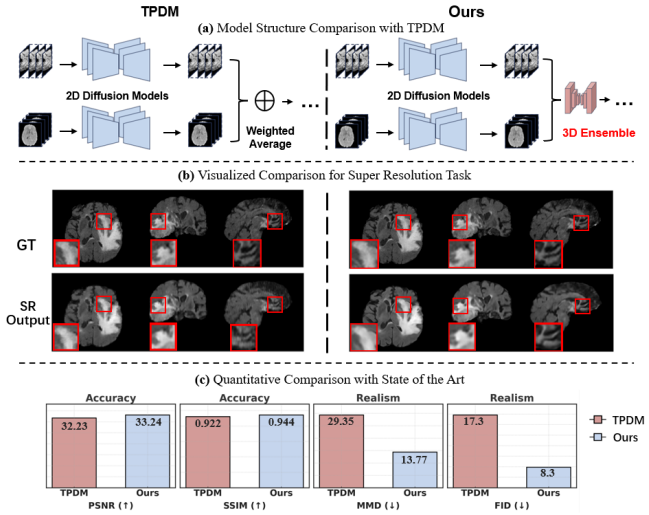

Figure 1: Comparison between TPDM (left) and Score-Fusion (right). Score-Fusion learns to ensemble pre-trained diffusion models with a 3D model, effectively utilizing 3D representations. Our model thus shows better 3D realism and demonstrates superior accuracy and realism metrics.

1. Introduction

Dense volume-to-volume translation is critical for volumetric medical imaging, such as magnetic resonance imaging (MRI) and X-ray computed tomography (CT). It addresses various inverse problems of image reconstruction (Hai-Miao Zhang, 2020; Liang et al., 2020; Wang et al., 2020), handling sparse (Kerstin et al.; Mardani et al., 2019), limited (Chung et al., 2023b; Lyu et al., 2020), and/or noisy (Chung et al., 2023b; Yang et al., 2019) imaging data. It also supports image synthesis, such as multi-contrast MRI (Zhang et al., 2022; Dar et al., 2018; Shi et al., 2021; Wolterink et al., 2017), CT-ultrasound (Vedula et al., 2017), and MR-histopathology (Leroy et al., 2021). In addition, by using the time axis as the third axis, video data can be considered as dense volumes. Therefore, dense volumetric translation can be useful in video data. Potential applications include video super-resolution (Li et al., 2023a; Zhou

et al., 2024; Li et al., 2025), video denoising (Zhang et al., 2023b; Fu et al., 2024), and video editing (Chai et al., 2023; Feng et al., 2024).

3D voxel-space representation has great potential to play a critical role in volume-to-volume translation tasks for various reasons. Firstly, medical images are inherently 3D-dense volumes. Using a 3D representation enables us to directly model the entire 3D distribution. Additionally, the 3D latent diffusion model (LDM) is a common technique to accelerate diffusion models. However, compared to LDMs, the 3D voxel-space diffusion model maintains better features in high-frequency details, which are essential in inverse problems that require highly accurate prediction, such as super-resolution. Moreover, volumetric translation models enable numerous downstream image processing tasks, ranging from image reconstruction (Meng et al., 2021) to analysis (Akrout et al., 2023; Fernandez et al., 2022)). Most models in analysis tasks, such as tumor segmentation(Hatamizadeh et al., 2022), are trained with 3D volumes using voxel-space 3D representations. Therefore, volume translation models using 3D representation may generate images that are more accurate when processed with such downstream task models.

However, previous works (Dorjsembe et al., 2024; Lee et al., 2023) highlight substantial challenges to utilize 3D representation in volumetric translation model for their increased demands for computational resources and large datasets that are costly to acquire in medical imaging. To the best of our knowledge, within the domain of 3D medical image inverse problems, no fully 3D models have demonstrated superior accuracy over 2D-based models due to these practical limitations. Weight inflation (Liu et al., 2023b) proposes designing a 3D network of the same size as 2D models, which is a promising approach given the rich 3D context and strong pre-trained 2D models. However, this is generally infeasible with existing 2D diffusion models (Saharia et al., 2022), which require around 500 GB of GPU memory for batch size 1 training and potentially an extremely long training time. As a result, current 3D diffusion models (Dorjsembe et al., 2024) are designed to be much smaller with insufficient capacity to demonstrate competitive performance in inverse problems. Recent advances in volume-to-volume translation have introduced methods that combine perpendicular 2D diffusion models (Lee et al., 2023; Chen et al., 2024), achieving improved accuracy and volumetric consistency. However, these methods cannot model the distribution of the entire volume since the generated images are produced by an averaging of the 2D networks without 3D representations, resulting in limited realism in 3D.

To effectively introduce 3D representations into volumetric translation, we present Score-Fusion, a pioneering model for volumetric translation that directly and effectively captures the distribution of 3D volumes. Score-Fusion adopts a two-stage training strategy: (1) It first trains multiple 2D diffusion models (Saharia et al., 2021) in perpendicular planes. (2) It then utilizes a 3D fusion network to produce the final translation in each diffusion step. Meanwhile, the Score-Fusion model is designed to start with a weighted average of 2D models following TPDM(Lee et al., 2023), which reduces 3D training to a fine-tuning process. The hierarchical layers of the 3D model are also reformulated to learn the ensemble of 2D features, further enhancing training efficiency and performance. Additionally, by ensembling diffusion models conditioned on various input modalities, Score-Fusion seamlessly supports multi-modality fusion.

Similarly, we find that time-space planes, i.e. x-t and y-t planes, can be crucial in video modeling, as demonstrated in previous works (Al-Sumaidaee et al., 2023; Otroshi-Shahreza et al., 2022). However, they remain underexplored in current video diffusion models. By extending Score-Fusion to video super-resolution, we trained additional 2D diffusion models on time-space planes and successfully demonstrated that introducing representations learned from time-space planes can enhance video super-resolution.

The mathematical intuition of Score-Fusion lies in the properties of diffusion models and their associated score functions (Song et al., 2021). As the score function models the gradient of the probability distribution, it is inherently suitable for an ensemble. Previous works (Chen et al., 2024; Chung et al., 2022) have also demonstrated this by showing strong performance with a straightforward weighted averaging of score functions. Consequently, Score-Fusion replaces the weighted averaging process with a 3D network, effectively incorporating 3D representations. To the best of our knowledge, Score-Fusion performs diffusion model fusion in the score function space, which provides new insights for diffusion model ensembling. Score-Fusion can also function as a plug-and-play mechanism compatible with various combinations of 2D models from previous studies (Chung et al., 2022; Chen et al., 2024; Li et al., 2024), consistently delivering performance improvements across various 2D backbones.

Score-Fusion has been evaluated in various MRI image processing tasks on the BraTS (Baid et al., 2021) and HCP (DC et al., 2013) datasets, including image super-resolution and modality translation. Our experimental results demonstrate that Score-Fusion performs superior volume translation over current state-of-the-art (SoTA) models in accuracy, realism, and downstream task performance. By learning to ensemble perpendicular 2D models conditioned on different input modalities, Score-Fusion shows strong performance without retraining new 2D models. Moreover, Score-Fusion has been extended to video super-resolution on the videoLQ dataset (Chan et al., 2022). Score-Fusion demonstrates the importance of time-space plane representation in video

super-resolution by showing improvement in temporal consistency and realism.

2. Related Work

3D medical image generation and translation. Attempts have been made to generate dense 3D volumes for medical imaging. Direct 3D-based diffusion models (Dorjsembe et al., 2024; Liu et al., 2023b) face difficulties due to high computational and dataset demands, resulting in moderate accuracy in tasks like super-resolution. Patch-wise, slice-wise, or cascaded generation strategies have been utilized to accommodate high-dimensional data(Uzunova et al., 2020). However, in such models, initial inaccuracies in the low-resolution base are propagated during the refinement stages and the patch-based refinement often struggles with maintaining global consistency across the image. Latent 3D models (Dorjsembe et al., 2024; Zhu et al., 2023a; Khader et al., 2023) have been exploited to compress the 3D data into a low-dimensional latent space and train diffusion models with this compressed latent space. However, the process of reducing dimensionality also has high computational and dataset demands and can lead to substantial reconstruction errors. Sequential slice generation from autoregressive models (Peng et al., 2023; Zhu et al., 2024) or simultaneous multiple slice generation may mitigate this issue of error accumulation over slices. Yet, these approaches suffer from challenges in maintaining coherence for long-range structures. More related to our approach, integrating multiple 2D models trained along perpendicular directions is a promising approach. TPDM (Lee et al., 2023) first proposes to combine two perpendicular 2D diffusion models to improve 3D imaging, where the weighted average of scores from pre-trained 2D models estimates the score function of a 3D model. Building on this concept,TOSM (Li et al., 2024) and MADM (Chen et al., 2024) further improve the model performance by including 2D models in all three directions and using multiple consecutive 2D slices in 2D models. These models generate highly accurate results by effectively leveraging the high-resolution information in each 2D plane.

Model ensembling. Ensemble techniques, which include bagging, boosting, and stacking, have been further developed through specialized algorithms like Random Forest, AdaBoost, XGBoost, and Mixture of Experts (MoE). These techniques also demonstrated remarkable efficacy in medical image analysis, particularly in brain tumor segmentation (Hatamizadeh et al., 2022; Zhou et al., 2019), hypertension detection (Fitriyani et al., 2019), and kidney stone identification (Kazemi & Mirroshandel, 2018). More recent research underscores the potential of ensembles as an effective strategy for scaling up large models(Jiang et al., 2024).

Diffusion model ensembling. Recently, diffusion models have shown great success. The ensembling methods for diffusion models have become a useful research topic. Most current works use weighted averages to ensemble different branches of diffusion models (Cheng et al., 2023; Lee et al., 2023). Collaborative Diffusion (Huang et al., 2023) is a learning-based ensembling method for diffusion; it trains an auxiliary model to estimate the confidence score for each branch of diffusion and ensemble based on the score. In this work, our approach uses information across all branches of diffusion models. This under-explored approach provides new insights for advancing diffusion model ensembling.

3. Score Fusion in 3D

Problem formulation. We formulate volume-to-volume translation as a conditional sampling problem. Specifically, let $\mathbf{x} \in \mathbb{R}^{b_1 \times b_2 \times b_3}$ be an input medical image volume, and let $\mathbf{y} \in \mathbb{R}^{b_1 \times b_2 \times b_3}$ be the corresponding target volume to be generated, where b_1, b_2, and b_3 denote the spatial dimensions of the volume. Our objective is to learn a conditional distribution $p(\mathbf{y} \mid \mathbf{x})$ that accurately captures the volumetric structure in three dimensions. The input volume $\mathbf{x}$ may consist of low-resolution data and/or a different imaging modality.

3.1. Overall Framework of Score-Fusion

We designed the Score-Fusion as a conditional diffusion model. Following DDPM and Palette (Ho et al., 2020; Saharia et al., 2022), our model gradually adds Gaussian noise to the target image in the training dataset during the *forward* or *diffusion process* as: $q\left(\boldsymbol{y}_t|\boldsymbol{y}_{t-1}\right) = \mathcal{N}\left(\boldsymbol{y}_t; \sqrt{1-\beta_t}\boldsymbol{y}_{t-1}, \beta_t \boldsymbol{I}\right)$; $q(\boldsymbol{y}_T|\boldsymbol{y}_0) = q(\boldsymbol{y}_0)\prod_{t=1}^{T} q(\boldsymbol{y}_t|\boldsymbol{y}_{t-1})$, where $\boldsymbol{y}_0 \sim q(\boldsymbol{y})$ is the target image and β_t is the variance of noise added at timestep t. The forward process produces a sequence of increasingly noisy variables $\boldsymbol{y}_1, ..., \boldsymbol{y}_T$, after sufficient noising steps, the process reaches a pure Gaussian noise, *i.e.*, $\boldsymbol{y}_T \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$.

During **training**, our denoising diffusion model, $\boldsymbol{\epsilon}_\theta(\boldsymbol{y}_t, \boldsymbol{x}, t)$, is trained to predict the noise added into $\boldsymbol{y}$, given $\boldsymbol{y}_t$. Demonstrated effective in existing work (Saharia et al., 2021), the sampling process can be guided by concatenating the noisy image $\boldsymbol{y}_t$ with condition $\boldsymbol{x}$. The loss used to optimize $\boldsymbol{\epsilon}_\theta(\boldsymbol{y}_t, \boldsymbol{x}, t)$ is: $\left\|\boldsymbol{\epsilon}_\theta(\sqrt{\alpha_t}\boldsymbol{y}_0 + \sqrt{1-\alpha_t}\boldsymbol{\epsilon}, \boldsymbol{x}, t) - \boldsymbol{\epsilon}\right\|_2^2$, where $\alpha_t := \prod_{i=1}^{t}(1-\beta_i)$, and we sample $\boldsymbol{y}_0, \boldsymbol{x} \sim p(\boldsymbol{y}_0, \boldsymbol{x})$, $\epsilon \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$.

During **sampling** in the *reverse* or *generative process*, we also follow Palette (Saharia et al., 2022) to generate images by iteratively removing the added noise in the sequence $\boldsymbol{y}_{T-1}, ..., \boldsymbol{y}_1, \boldsymbol{y}_0$, from a standard Gaussian prior $\boldsymbol{y}_T \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$. In addition, inspired by (Song et al., 2022; Chung et al., 2023a; Song et al., 2024), we explore self-consistency for solving inverse problems. More

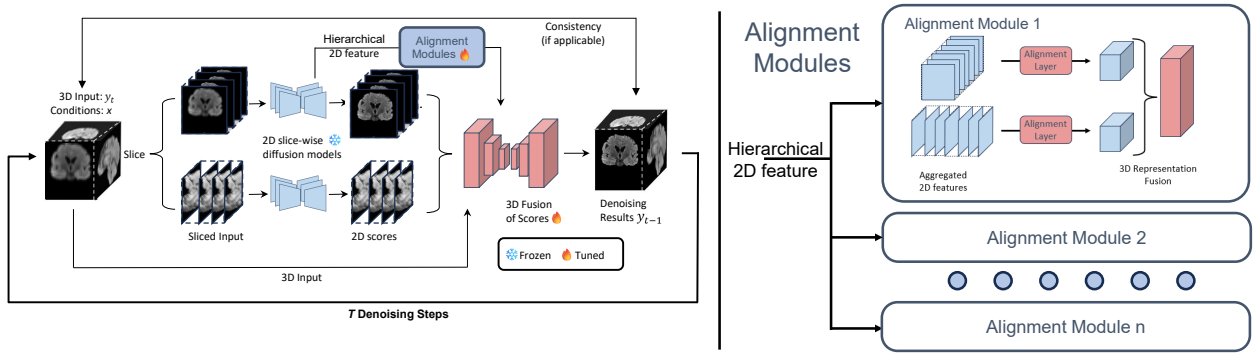

Figure 2: Overview of the Score-Fusion. At each denoising step, two pre-trained 2D models provide initial estimations of the scores in a slice-wise manner. Subsequently, a 3D network learns to integrate these estimations via 3D representation extracted from 3D input and aggregated 2D scores. In addition, the 3D model is initialized to output an average of 2D scores. Moreover, The 3D network layers are also reformulated to learn an ensemble of aggregated and aligned 2D features. These designs accelerate and stabilize the 3D training process.

specifically, in each diffusion sampling step, we estimate noise with our denoising model $\boldsymbol{\epsilon}_\theta(\boldsymbol{y}_t, \boldsymbol{x}, t)$. Therefore, we have the estimated $\hat{\boldsymbol{y}}_0(t)$ at the t-th denoising step as $\hat{\boldsymbol{y}}_0(t) := (\boldsymbol{y}_t - \sqrt{1-\alpha_t}\boldsymbol{\epsilon}_\theta(\boldsymbol{y}_t, \boldsymbol{x}, t))/\sqrt{\alpha_t}$. In inverse problems, conditional input $\boldsymbol{x}$ is obtained through a known linear degradation process $\boldsymbol{x} = A\boldsymbol{y}$. At each diffusion step, we project the estimated $\hat{\boldsymbol{y}}_0(t)$ to a plausible $\hat{\boldsymbol{y}}_0(t)$, such that $\boldsymbol{x} = A\hat{\boldsymbol{y}}_0$, by $\hat{\boldsymbol{y}}_0(t) \leftarrow \hat{\boldsymbol{y}}_0(t) - A^T(AA^T)^{-1}(A\hat{\boldsymbol{y}}_0(t) - x)$. We provide more details in Sec. H. After the consistency projection, we obtain $\boldsymbol{y}_{t-1}$ by adding noise back: $\boldsymbol{y}_{t-1} = \sqrt{\alpha_{t-1}}\hat{\boldsymbol{y}}_0(t) + \sqrt{1-\alpha_{t-1}}\boldsymbol{\epsilon}$.

Score-Fusion Models. The key component of this work lies in our denoising network $\boldsymbol{\epsilon}_\theta(\boldsymbol{y}_t, \boldsymbol{x}, t)$. In particular, our model $\boldsymbol{\epsilon}_\theta$ consists of two 2D diffusion denoising models, $\boldsymbol{\epsilon}^{2D(a)}_{\theta_a}$ and $\boldsymbol{\epsilon}^{2D(b)}_{\theta_b}$, and a 3D diffusion denoising model, $\boldsymbol{\epsilon}^{3D}_{\theta_{3D}}$, with θ_a, θ_b, and θ_{3D} being their trainable parameters, respectively. The 3D network is conditioned on two 2D diffusion models, trained to capture 2D image distributions along orthogonal planes to provide complementary views of the volumetric data. The 3D network is carefully initialized to start with a weighted average of 2D networks' estimation, such that Score-Fusion has the same performance with TPDM(Lee et al., 2023) before any 3D training. This design effectively constrains the 3D model, reduces the 3D training to a fine-tuning process, and thus promotes faster and stabilized training convergence. Additionally, the hierarchical representations from the 2D models are introduced to the layers in the 3D model with alignment projection layers. In this way, the 3D model's layers are reformulated to learn an ensemble of pre-trained 2D models' representations, instead of learning representations from scratch. Therefore, the training of the 3D model is further accelerated and stabilized by using the aligned 2D representation as a reference. We refer to this hybrid 2D/3D volumetric generative model as Score-Fusion. Fig. 2 provides a schematic overview of Score-Fusion.

3.2. 2D Score Models

The 2D diffusion models, $\boldsymbol{\epsilon}^{2D(a)}_{\theta a}$ and $\boldsymbol{\epsilon}^{2D(b)}_{\theta b}$, are trained on two perpendicular slices of the volumes using a standard conditional diffusion framework (Saharia et al., 2021). We take gradient descent steps on the following objectives for both 2D diffusion models during training:

$$\begin{aligned}\nabla_{\theta a}\left\|\boldsymbol{\epsilon}^{2D(a)}_{\theta a}(\boldsymbol{y}_t[:,i,:], \boldsymbol{x}[:,i,:], t) - \boldsymbol{\epsilon}\right\|_2^2 \\ \nabla_{\theta b}\left\|\boldsymbol{\epsilon}^{2D(b)}_{\theta b}(\boldsymbol{y}_t[:,:,j], \boldsymbol{x}[:,:,j], t) - \boldsymbol{\epsilon}\right\|_2^2\end{aligned} \tag{1}$$

Here, i and j are the indices for the slices along two perpendicular planes, which are sampled uniformly: $i \sim \text{Uniform}\{0, ..., b_2\}$, $j \sim \text{Uniform}\{0, ..., b_3\}$. After proper training, the high-capacity 2D model can provide a decently accurate estimation of $\boldsymbol{\epsilon}$ for every volume slice.

3.3. 3D Fusion Model

The 3D ensembling model, $\boldsymbol{\epsilon}^{3D}_{\theta_{3D}}$, is trained to fuse the pre-trained 2D diffusion models to capture the desired volumetric image distributions. In this stage, we first obtain the inference results, $\hat{\boldsymbol{Y}}^{2D(a)}$ and $\hat{\boldsymbol{Y}}^{2D(b)}$, from the 2D diffusion models, $\boldsymbol{\epsilon}^{2D(a)}_{\theta a}$ and $\boldsymbol{\epsilon}^{2D(b)}_{\theta b}$ by iterating through the sliced directions:

$$\begin{aligned}\hat{\boldsymbol{Y}}^{2D(a)}[:,i,:] &= \boldsymbol{\epsilon}^{2D(a)}_{\theta a}(\boldsymbol{y}_t[:,i,:], \boldsymbol{x}[:,i,:], t) \text{ for } i \in [0, b_2) \\ \hat{\boldsymbol{Y}}^{2D(b)}[:,:,j] &= \boldsymbol{\epsilon}^{2D(b)}_{\theta b}(\boldsymbol{y}_t[:,:,j], \boldsymbol{x}[:,:,j], t) \text{ for } j \in [0, b_3)\end{aligned} \tag{2}$$

During training of the 3D diffusion model, both 2D models return $\hat{\boldsymbol{Y}}$'s, which contains the predicted score and a hierarchical feature map of the model: $\hat{\boldsymbol{Y}}^{2D(a)} = (\hat{\epsilon}^{2D(a)}, \mathcal{F}^{2D(a)})$, $\hat{\boldsymbol{Y}}^{2D(b)} = (\hat{\epsilon}^{2D(b)}, \mathcal{F}^{2D(b)})$.

The 3D model is designed to learn an ensemble of multiple 2D models' score estimation with 3D representation. Specifically, at each diffusion step, the 3D model takes as input the original image $\boldsymbol{x}$, the noisy intermediate state $\boldsymbol{y}_t$, and the aggregated score estimation $\hat{\epsilon}$ obtained from the 2D

diffusion models. Furthermore, aggregated feature maps $\mathcal{F}$ from the 2D models are incorporated as supplementary information as in Fig. 2. Formally speaking, the 3D network is trained to perform the ensembling process using the following formulation using an L2 loss:

$$\nabla_{\theta_{3D}} \left\| \boldsymbol{\epsilon}^{3D}_{\theta_{3D}}(\boldsymbol{y}_t, \boldsymbol{x}, \hat{\boldsymbol{Y}}^{2D(a)}, \hat{\boldsymbol{Y}}^{2D(b)}, t) - \boldsymbol{\epsilon} \right\|_2^2 \quad (3)$$

where $\boldsymbol{y}_0, \boldsymbol{x} \sim p(\boldsymbol{y}_0, \boldsymbol{x})$, and $\epsilon \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$. Although, the inference results from 2D models, $\hat{\boldsymbol{Y}}^{2D(a)}$ and $\hat{\boldsymbol{Y}}^{2D(b)}$, already help the training of the 3D model, 3D ensemble model still needs to be trained from scratch. To improve training speed, we initially pre-train the model on 3D patches, $(\boldsymbol{y}_0, \boldsymbol{x}) = \text{crop}(\boldsymbol{y}_0, \boldsymbol{x})$, and then fine-tune it on the full volumes. Due to the translation invariance of our convolution-based networks, we empirically find that a naively pre-train on the patches results in a decently good network initialization, thereby effectively improving the training convergence. While existing works, such as (Wang et al., 2023), could potentially enhance this patch-wise diffusion training process, we leave such optimizations for future work.

In this work, the **network architecture** of the 3D model, $\boldsymbol{\epsilon}^{3D}_{\theta_{3D}}$, is a 3D Unet-like denoising model with time-step embeddings, the 3D input $\boldsymbol{x}$, the noisy target $\boldsymbol{y}_t$, and the noise estimated by the 2D models $\hat{\epsilon}^{2D(a)}, \hat{\epsilon}^{2D(b)}$ as the input of the 3D model. In the encoder, each downsampling block is enriched with corresponding feature maps from the features of both 2D models, $\mathcal{F}^{2D(a)}$, and $\mathcal{F}^{2D(b)}$. At each hierarchical level, the feature maps are first processed with MLP-based alignment layers, which align the 2D features with the 3D model and map them to an appropriate shape. The feature maps are then injected into 3D layers, providing the 3D layer with a reference to fused 2D features. Hence, the 3D layers are reformulated to learn an ensemble of the aggregated 2D features, which is easier than learning representation from scratch. In addition, using the feature maps mitigates the risk of information bottlenecks between the 2D and 3D stages, which could otherwise limit the performance. Additionally, rather than directly outputting the predicted noise, our 3D U-Net-like model produces two components: a weight vector $\boldsymbol{w}$, used to ensemble the estimations from the 2D models, and a residual term R, which is directly estimated by the 3D model. These two outputs are combined to form the final prediction:

$$\boldsymbol{\epsilon}^{3D}_{\theta_{3D}}(...) = (0.5 + \boldsymbol{w})\hat{\epsilon}^{2D(a)} + (0.5 - \boldsymbol{w})\hat{\epsilon}^{2D(b)} + \lambda R \quad (4)$$

where λ is a hyperparameter, whereas $\boldsymbol{w}$ and R are of the same size as the target noise $\epsilon \in \mathbb{R}^{b_1, b_2, b_3}$. This design enables the model to dynamically select the more reliable 2D estimation based on 3D context and allows the 3D model to contribute 3D-specific content R. Meanwhile, a tunable weight parameter, λ, controls the model's reliance on the 3D output, R. In addition, inspired by ControlNet (Zhang et al., 2023a), a zero-initialized convolution layer at the end of the model smoothly initializes $\boldsymbol{w}$ and R as all-zero vectors. This makes the 3D training a fine-tuning process starting with an average weighting strategy, and thereby stabilizing the 3D model training. The pseudo-code for training and inference with Score-Fusion is provided in Algorithm 1 and Algorithm 2.

3.4. Score-Fusion for Video Super-resolution

We extend the Score-Fusion framework to the task of video super-resolution. Since most existing diffusion-based video super-resolution methods incorporate both spatial and temporal features, we adopt MGLD-VSR (Yang et al., 2024) as our 3D model to fuse scores. MGLD-VSR builds upon a 2D latent diffusion model pretrained on the spatial (x–y) plane using image datasets and introduces inter-frame guidance to both the diffusion model and decoder, enabling temporal consistency in the output. However, like other diffusion-based video super-resolution methods, MGLD-VSR does not explicitly learn representations in the time-space planes, despite prior works emphasizing its importance (Al-Sumaidaee et al., 2023; Otroshi-Shahreza et al., 2022).

To address this limitation, we train two additional 2D diffusion models directly in the video latent space by slicing video latent embeddings along time-space planes. The denoising outputs from these time-space models are then used as auxiliary conditions to fine-tune the 3D model. This integration allows the model to benefit from both time-space representation learning and cross-plane features.

3.5. Multi-modality Fusion

In volumetric translation for medical imaging, the conditions for translating a new image can be multifaceted. For instance, DDMM-Synth (Li et al., 2023b) suggested using both MRI and low-resolution CT scans to produce high-resolution CT images. Training a separate model for each possible combination of input conditions would result in exponential time complexity, making it generally impractical. On the other hand, training a foundation model to support various translation tasks also requires larger datasets, which are hard to acquire in medical imaging. Therefore, a model that can efficiently integrate pre-trained models across diverse conditions provides significant advantages. Score-Fusion achieves this by naturally integrating multiple diffusion models, each conditioned on individual modalities, through a 3D network architecture that functions similarly to fusing two 2D models described in 3.3. This approach leverages the accelerated 3D training of Score-Fusion. To further enhance the speed of multi-modality fusion, we employ a smaller variant of our model, adjusting the number of channels in each layer.

Algorithm 1 Training of Score Fusion

repeat
  $(\mathbf{x}, \mathbf{y_0}) \sim p(\mathbf{x}, \mathbf{y_0})$ ▷ sample from dataset
  if pretrain **then** ▷ pretrain on patch
    $(\mathbf{x}, \mathbf{y_0}) = \text{crop}(\mathbf{x}, \mathbf{y_0})$
  end if
  $t \sim \text{Uniform}(0, T)$; $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
  for $i = 0$ to b_2 **do**
    $\hat{\mathbf{Y}}^{2D(a)}[:, i, :] \leftarrow \boldsymbol{\epsilon}_{\theta a}^{2D(a)}(\mathbf{y}_t[:, i, :], \mathbf{x}[:, i, :], t)$
  end for
  for $j = 0$ to b_3 **do**
    $\hat{\mathbf{Y}}^{2D(b)}[:, :, j] \leftarrow \boldsymbol{\epsilon}_{\theta a}^{2D(b)}(\mathbf{y}_t[:, :, j], \mathbf{x}[:, :, j], t)$
  end for
  Take a gradient descent step on
  $\nabla_{\theta_{3D}} \left\| \boldsymbol{\epsilon}_{\theta_{3D}}^{3D}(\mathbf{y}_t, \mathbf{x}, \hat{\mathbf{Y}}^{2D(a)}, \hat{\mathbf{Y}}^{2D(b)}, t) - \epsilon \right\|_2^2$
until converged

Algorithm 2 Inference of Score Fusion

$(\boldsymbol{x}) \sim p(\boldsymbol{x})$ ▷ sample from dataset
$\boldsymbol{y_T} \sim N(0, 1)$
for t = T,...,1,0 **do**
  for i = 0,1,...,b_2 **do**
    $\hat{\boldsymbol{Y}}^{2D(a)}[:, i, :] \leftarrow \boldsymbol{\epsilon}_{\theta a}^{2D(a)}(\boldsymbol{y}_t[:, i, :], \boldsymbol{x}[:, i, :], t)$
  end for
  for j = 0,1,...,b_3 **do**
    $\hat{\boldsymbol{Y}}^{2D(b)}[:, :, j] \leftarrow \boldsymbol{\epsilon}_{\theta a}^{2D(b)}(\boldsymbol{y}_t[:, :, j], \boldsymbol{x}[:, :, j], t)$
  end for
  $\hat{\boldsymbol{\epsilon}}^{3D} = \boldsymbol{\epsilon}_{\theta}^{3D}(\boldsymbol{y}_t, \boldsymbol{x}, \hat{\boldsymbol{Y}}^{2D(a)}, \hat{\boldsymbol{Y}}^{2D(b)}, t)$
  $\hat{\boldsymbol{y}}_0 \leftarrow \frac{\boldsymbol{y}_t - \sqrt{1-\alpha_t}\boldsymbol{\epsilon}^{3D}}{\sqrt{\alpha_t}}$ ▷ get current estimation of y0
  if Inverse problem **then**
    $\hat{\boldsymbol{y}}_0 \leftarrow \hat{\boldsymbol{y}}_0 - A^T(AA^T)^{-1}(A\hat{\boldsymbol{y}}_0 - x)$
  end if
  $\boldsymbol{y}_{t-1} \leftarrow \sqrt{\alpha_{t-1}}\hat{\boldsymbol{y}}_0 + \sqrt{1-\alpha_{t-1}}\boldsymbol{\epsilon}$
end for
return $\boldsymbol{y}_0$

4. Experiments

4.1. Experimental Setup

Datasets. We conducted experiments using the BraTS 2021 training dataset (Baid et al., 2021), which includes 1,251 volumetric brain scans with tumors across 4 modalities: FLAIR, T1, T1ce, and T2. We randomly divided the dataset into a 0.8:0.2 split for training and evaluation purposes, allowing its use for downstream tasks as well. Each scan was center-cropped to a dimension of 192×192×152 to remove the blank background. For training 2D models, we sliced the 3D volumes in two directions—transverse and sagittal planes for both TPDM baselines and Score-Fusion. In the super-resolution experiment, FLAIR images were downsampled using [4×4×4] average pooling. For modality translation, T1ce images served as inputs with FLAIR images as targets. Additionally, we investigated a multi-condition task, using both low-resolution FLAIR and T1ce images as input to predict high-resolution FLAIR images. In addition, we investigate Score-Fusion's generalizability to different datasets by applying it, alongside related baselines, to a super-resolution task on the FLAIR modality of the HCP dataset (DC et al., 2013). This demonstrates Score-Fusion's potential for broader applicability. We present both quantitative and qualitative results for the HCP dataset in Section D.

Baselines. We reproduced diverse baseline methods across a diverse set of established 2D and 3D translation models to ensure a comprehensive comparison. For slice-wise 2D models, we utilized Pix2pix (Isola et al., 2017) as the representative GAN-based method, U-Net (Ronneberger et al., 2015) for supervised regression, Palette (Saharia et al., 2022) as a diffusion-based approach, and I2SB (Liu et al., 2023a) for optimal-translation-based modeling. Similarly, for 3D-based baselines, we used Pix2pix3D (Isola et al., 2017), U-Net3D (Ronneberger et al., 2015), Med-DDPM (Dorjsembe et al., 2024) (or Palette3D). As stated in Sec. 1, Med-DDPM uses a small denoising network and thus demonstrates limited performance. In addition, we used Palette-2.5D for another baseline, which uses multiple consecutive 2D slices as input. Several existing approaches closely related to our method combine multiple pre-trained 2D diffusion models in perpendicular orientations, demonstrating enhanced performance over other baselines. For instance, TPDM (Lee et al., 2023) combines two 2D diffusion models trained on perpendicular planes. To support modality translation, we adapted the TPDM's 2D backbone to Palette (Saharia et al., 2022) architecture in place for DPS (Chung et al., 2023a). Furthermore, TOSM (Li et al., 2024) employs three perpendicularly trained 2D diffusion models, whereas MADM (Chen et al., 2024) uses three 2.5D diffusion models. We perform a hyper-parameter search on the super-resolution task on the BraTS dataset for all baselines.

Model Architecture and Variants. To make a fair comparison, we use pre-trained 2D models from TPDM, TOSM, and MADM utilizing an existing 3D diffusion model architecture, Med-DDPM (Dorjsembe et al., 2024). The TPDM-based model is our primary model as it achieves 30% faster inference speed and smaller model size relative to the TOSM-based model, as shown in Tab. 10. Meanwhile, MADM-based and TOSM-based models are heavier variants that yield performance gains across all metrics. These consistent improvements in all three variants demonstrate that our approach can serve as a plug-in-and-play mechanism for multiple combinations of 2D/2.5D model backbones and existing 3D model architectures. We also include a more detailed model architecture in Sec. F.

Metrics. We used multiple metrics to assess both the accuracy and realism of generated MRI images. For accuracy, we used peak signal-to-noise ratio (PSNR) and the structural similarity index measure (SSIM), which are widely used in

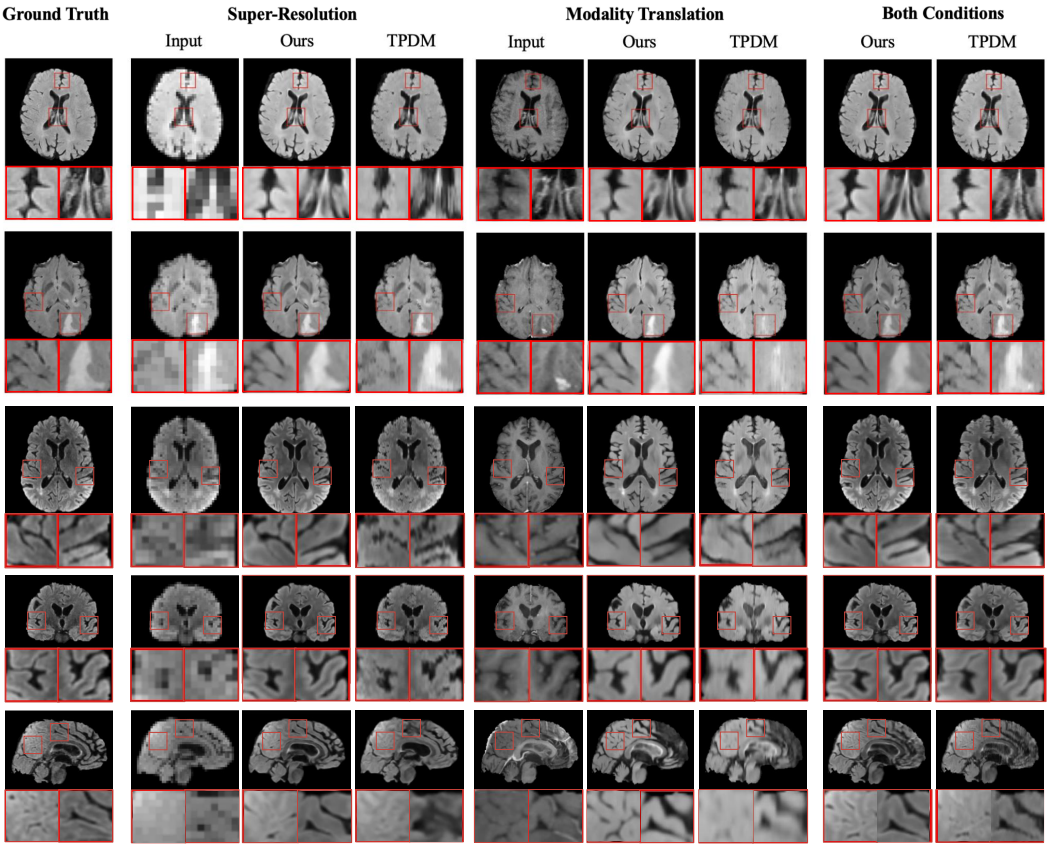

Figure 3: Visual comparison of generated samples for three different conditions. The first three rows show axial view slices from different MRI volumes. Neither Score-Fusion nor TPDM have a 2D model trained in this direction. The last three rows show slices for the same MRI volume in all three views. Score-Fusion reconstructs more realistic details with smoother edges and fewer artifacts.

medical imaging. To evaluate perceptual quality and realism, we used the maximum mean discrepancy (MMD) (Gretton et al., 2012) and the Fréchet inception distance (FID) metrics (Heusel et al., 2017). Lower MMD/FID scores imply the generated images are more realistic. To evaluate the FID score, following common practice (Dorjsembe et al., 2024; Sun et al., 2022), we adopted the same pre-trained model (Chen et al., 2019) to extract features and calculate the FID metrics in the feature space. Because diffusion-based models exhibit inherent stochasticity, we further assess uncertainty by performing inference multiple times with different noise realizations ϵ. From these runs, we calculate voxel-wise means and standard deviations, thereby providing uncertainty-aware metrics. Quantitative and qualitative results for these metrics can be found in Sec. B.

4.2. Experimental Results for Medical Image

We showcase the performance of Score-Fusion in solving various translation problems, including $4\times$ super-resolution, modality translation, and conditioned on both conditions in Fig. 3. We also include more randomly selected samples for more variants of our model in Sec. C. Fig. 3 shows the generation quality under various conditions and provides comparisons with other methods. The first two columns show the performance for super-resolution and modality translation, and the last columns show the model performance when using both conditions. Our approach excels in faithfully recovering intricate high-frequency details, particularly in tumor-affected areas where such details are complex and often underrepresented. In the super-resolution task, from the zoomed-in panel, Score-Fusion clearly distinguishes tissue boundaries across various tissue types, including tumor and white/grey matter. In the modality translation task, the distribution of contrast difference between modalities is also better captured, as stated in Sec. B. Furthermore, our method demonstrates superior volumetric consistency, while the baseline model exhibits noticeable artifacts. In the visualization of Score-Fusion, the fidelity of tissue texture and sharpness along all three orthogonal directions are well preserved, even though the 2D diffusion models in Score-Fusion are trained only on transverse and sagittal planes. Our model reconstructs tumor regions with clearer margins, fewer artifacts, and higher resolution for samples containing tumors, providing superior performance in all three orthogo-

nal directions. Overall, Score-Fusion generates images with higher accuracy, realism, and volumetric consistency. Tab. 1 summarizes the quantitative results of each translation task. Score-Fusion clearly surpasses other baseline models in most metrics, showing superior fidelity, structure, texture preservation, and noise suppression performance.

4.3. Downstream Task

We further evaluated the performance of Score-Fusion on a downstream task: tumor segmentation, where high-quality input modalities are crucial for accurately segmenting complex structures. Using the BraTS 2021 tumor segmentation dataset, we applied a pre-trained Swin-UNetR (Hatamizadeh et al., 2022) model with four modalities, replacing the ground-truth FLAIR modality with inferences from each model. Segmentation performance was assessed on Tumor Core (TC), Whole Tumor (WT), and Enhancing Tumor (ET) regions using two metrics: Dice score and Recovery rate. The Dice score measures segmentation quality, while the Recovery rate quantifies each model's segmentation score recovered from low-quality FLAIR to the generated FLAIR, with segmentation using ground truth (GT) FLAIR as the upper bound and downsampled FLAIR as the lower bound. The Recovery rate is defined as: Recovery Rate = (Prediction - Downsample) / (Ground-Truth - Downsample), where **Prediction** refers to the segmentation performance using predicted FLAIR, **Downsample** is the performance with downsampled FLAIR, and **Ground Truth** is the performance with ground-truth FLAIR. As shown in Tab. 2, our methods consistently outperformed other methods. Our method also qualitatively results in smoother tumor edges and more accurate structures demonstrated in Fig. 8 9. (see Sec. I for more details).

4.4. Multi-modality fusion

As discussed in Sec. 3.5, the Score-Fusion not only merges 2D models trained in different directions but also effectively integrates models pre-trained under various single conditions when faced with new combinations of input modalities given pre-trained 2D models on every single condition. We show the model's performance in Tab. 3. TPDM uses a weighted average for all 2D models, demonstrating limited performance. Using a 3D model of the original size, Score-Fusion learns to fuse scores estimated in two different conditions, demonstrating competitive performance without re-training 2D models. Score-Fusion-small further improves training speed with a marginal performance drop to flexibly support multi-modality fusion. The models on both conditions (last 2 rows) show the metrics when re-training every 2D model on both conditions using an early fusion strategy, representing an upper limit of multi-modality fusion performance. All training experiments are performed on Nvidia RTX A100-40G GPUs.

4.5. Training and Inference Speed.

We show Training and Inference Speed in Tab. 5. Previous 3D diffusion method struggles to use 3D representation, with extremely long training time (120 GPU days) and sub-optimal accuracy performance as in Tab. 1. In contrast, Score-Fusion effectively introduced 3D representation in just 16 days of extra 3D training time on top of TPDM. Score-Fusion-small further accelerates the 3D training for 4 times, achieving 30x more efficient than 3D diffusion baselines.

4.6. Video Super-resolution Results

Dataset. We extend Score-Fusion to video super-resolution. We treat videos as 3D dense volumes composed of 3 axes: x-axis, y-axis, and t-axis. We adopted a sliding window strategy on the time axis to get dense volumes of the same size. Following MGLD-VSR, we train our model on the REDS (Nah et al., 2019) dataset and evaluate on the VideoLQ (Chan et al., 2022) dataset.

Experiment Setup. Following previous works (Yang et al., 2024; Li et al., 2023a; 2025), we adopt a latent diffusion method, composed of a video auto-encoder and a diffusion model in the latent space. For the video autoencoder, we use an off-the-shelf spatial-temporal autoencoder from MGLD-VSR (Yang et al., 2024). We use Score-Fusion in the latent diffusion model by using a 3D latent model to fuse two perpendicular 2D latent models. For the 3D model, we also utilize the pretrained spatial-temporal latent diffusion model from MGLD-VSR (Yang et al., 2024). Such settings allow us to make a fair comparison with MGLD-VSR. For 2D models, we use pretrained models from Stable_SR (Wang et al., 2024), which share a similar architecture and model size with MGLD-VSR. During training, the hierarchical layer of the 3D model learns to incorporate the 2D features. Following llama-adapter, we use a zero-initialized gate on the 2D features, such that the model starts with its original states as in MGLD-VSR.

Method	DOVER(↑)
MGLD	0.748
Score-Fusion-MGLD	0.755

Table 4: Quantitative results for Video Super-Resolution.

We present our quantitative results in Tab. 4, using DOVER (Wu et al., 2023) metrics. DOVER focuses on video quality assessment by evaluating technical and aesthetic perspectives, which proves to be highly aligned with human preference. Our results show performance improvement, demonstrating the positive impact of time-space plane representations in video super-resolution.

Method	SR				MT				both condition			
	PSNR(↑)	SSIM(↑)	MMD(↓)	FID(1e-4)(↓)	PSNR	SSIM	MMD	FID	PSNR	SSIM	MMD	FID
Pix2pix(Isola et al., 2017)	28.75	0.889	512.2	25.9	22.25	0.812	8989.0	577.6	31.78	0.923	133.9	11.8
U-net(Ronneberger et al., 2015)	30.32	0.579	917.2	58.9	23.74	0.846	1829.0	320.3	33.58	0.931	83.5	36.6
Palette(Saharia et al., 2022)	29.26	0.894	40.9	13.5	22.68	0.784	284.4	85.9	33.6	0.939	34.9	9.3
I2SB(Liu et al., 2023a)	27.51	0.860	2644.5	47.7	20.75	0.738	35774.5	1343.6	31.3	0.905	1313.6	12.0
Palette-3D(Dorjsembe et al., 2024)	28.48	0.320	4222	88.7	Not Working				24.98	0.297	15926.0	463.1
Pix2pix-3D(Isola et al., 2017)	29.54	0.866	516.5	8.6	22.77	0.784	1974.0	342.2	31.86	0.900	87.81	56.2
U-net-3D(Ronneberger et al., 2015)	31.23	0.892	115.6	59.6	23.43	0.809	487.5	273.8	32.95	0.922	43.3	43.9
Palette-2.5D(Saharia et al., 2022)	29.76	0.834	35.37	12.3	23.04	0.728	1141.19	258.6	25.89	0.819	2858.37	138.92
TPDM(Lee et al., 2023)	32.23	0.922	29.35	17.3	25.35	0.868	176.3	185.5	35.12	0.945	14.5	22.2
Score-Fusion-TPDM	33.24	0.944	13.77	8.31	25.26	0.882	154.9	**48.2**	36.24	0.961	7.52	5.8
TOSM(Li et al., 2024)	32.76	0.932	24.17	24.87	**25.66**	0.881	1018.91	209.5	35.44	0.947	8.32	14.53
Score-Fusion-TOSM	33.30	**0.945**	13.62	**6.51**	25.24	**0.882**	**138.47**	136.06	**36.51**	0.963	5.926	**3.72**
MADM(Chen et al., 2024)	33.02	**0.945**	30.92	35.64	25.47	0.874	1419.4	251.4	35.21	0.946	8.21	13.6
Score-Fusion-MADM	**33.31**	**0.945**	**13.46**	6.57	25.13	0.876	192.51	130.33	36.37	**0.964**	**5.44**	3.81

Table 1: Quantitative evaluation of Score-Fusion on BraTS dataset. Best metrics are highlighted in **bold**. The proposed model achieves better accuracy (PSNR/SSIM) given more 3D context than their corresponding variant in most tasks. Moreover, thanks to 3D representation, Score-Fusion achieves significantly better 3D realism (MMD/FID). We demonstrate the standard deviation and uncertainty metrics in Tab. 6.

Method	Dice (%)			Recovery (%)		
	TC	WT	ET	TC	WT	ET
GT FLAIR	82.71	89.17	81.20	-	-	-
Downsampled GT	82.30	86.82	80.30	-	-	-
SR						
TPDM	82.49	87.77	80.49	46.27	40.46	20.62
TOSM	82.52	87.21	80.80	54.55	16.51	55.28
Score-Fusion-TPDM	**82.69**	87.85	**80.94**	**93.71**	43.80	**71.69**
Score-Fusion-TOSM	82.59	**87.86**	80.87	70.14	**44.38**	63.94
MT						
TPDM	77.28	77.74	78.37	-	-	-
TOSM	77.94	**79.21**	78.64	-	-	-
Score-Fusion-TPDM	77.88	78.51	78.22	-	-	-
Score-Fusion-TOSM	**78.84**	78.73	**79.52**	-	-	-
both condition						
TPDM	82.45	87.69	80.74	36.31	37.25	49.28
TOSM	82.54	87.27	80.82	57.81	19.25	57.79
Score-Fusion-TPDM	82.46	87.91	80.74	38.66	46.67	48.97
Score-Fusion-TOSM	**82.61**	**87.98**	**80.89**	**75.35**	**49.69**	**65.72**

Table 2: Segmentation performance with the FLAIR modality replaced by model predictions.

Method	PSNR	SSIM	MMD	Training Time (GPU days)
TPDM	32.43	0.929	25.05	0
Score-Fusion-small	35.34	0.956	8.64	4
Score-Fusion	35.6	0.958	8.82	16
TPDM-both_cond	35.12	0.945	14.5	16
Score-Fusion-both_cond	36.24	0.961	7.52	32

Table 3: Multi-modality fusion results for Score-Fusion.

Method	**2D Training** (GPU days)	**3D Training** (GPU days)	**Inference** (minutes/volume)
3D Palette	0	120	0.6
TPDM	16	0	1.72
Score-Fusion	16	16	2.34
Score-Fusion-small	16	4	1.92

Table 5: Training and Inference time results. GPUs are A100-40G. More complete comparison in Tab. 10

5. Conclusion

3D voxel-space representation can be essential in medical image translation and generation for both volumetric realism and downstream task performance. However, existing models struggle to use 3D representation due to computational challenges and data scarcity. In this work, we have introduced Score-Fusion to effectively introduce 3D voxel space representation into 3D medical image translation by fusing estimations from slice-wise 2D models in the score function space. Several key designs, including average-initialization, feature map fusion, patch-wise pre-training. In addition, our model integrated the strengths of both 2D and 3D diffusion models. Score-Fusion provides strong insights for diffusion model ensembling as the first work to adopt a learning-based fusion in the score function space. Empirical evaluations on various 3D MRI image translation tasks, including super-resolution and modality translation, have shown that Score-Fusion achieves unmatched accuracy, volumetric realism, and downstream task performance. In addition to computational and memory efficiency, the approach offers considerable flexibility in merging models conditioned on different domains.

Impact Statement

This paper presents work that aims to advance the field of Machine Learning and its application in medical imaging and video processing. There are several **Limitations** of our work. Unlike some other multi-stage models (Zhu et al., 2023b; Huang et al., 2023), Score-Fusion struggles with joint end-to-end training due to the substantial computational demands of simultaneously managing high-capacity 2D models and the volumetric complexities of 3D tasks. In addition, the model's dependency on patchwise pre-training for efficient 3D model learning presents limitations for tasks requiring the integration of long-range spatial information, such as large-area inpainting and compressed sensing MRI. Therefore, Score-Fusion may require longer training for such tasks. There are also many potential societal consequences of our work. However, direct application of our method to medical imaging should be approached with caution.

Acknowledgement

This work used the Delta system at the National Center for Supercomputing Applications through allocations CIS230243 and CIS240171 from the Advanced Cyberinfrastructure Coordination Ecosystem: Services & Support (ACCESS) program, which is supported by National Science Foundation grants #2138259, #2138286, #2138307, #2137603, and #2138296. We also thank Xiaoyue Li (xiaoyue98@zju.edu.cn) and Shen Yuan (yshen47@illinois.edu) for the valuable discussion.

References

Akrout, M., Gyepesi, B., Holló, P., Poór, A., Kincső, B., Solis, S., Cirone, K., Kawahara, J., Slade, D., Abid, L., et al. Diffusion-based data augmentation for skin disease classification: Impact across original medical datasets to fully synthetic images. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 99–109. Springer, 2023.

Al-Sumaidaee, S., Abdullah, M., Al-Nima, R., Dlay, S., and Chambers, J. Spatio-temporal modelling with multi-gradient features and elongated quinary pattern descriptor for dynamic facial expression recognition. *Pattern Recognition*, 142:109647, 2023. ISSN 0031-3203. doi: https://doi.org/10.1016/j.patcog.2023.109647. URL https://www.sciencedirect.com/science/article/pii/S0031320323003485.

Baid, U., Ghodasara, S., Mohan, S., Bilello, M., Calabrese, E., Colak, E., Farahani, K., Kalpathy-Cramer, J., Kitamura, F. C., Pati, S., et al. The rsna-asnr-miccai brats 2021 benchmark on brain tumor segmentation and radiogenomic classification. *arXiv preprint arXiv:2107.02314*, 2021.

Chai, W., Guo, X., Wang, G., and Lu, Y. Stablevideo: Text-driven consistency-aware diffusion video editing, 2023.

Chan, K. C., Zhou, S., Xu, X., and Loy, C. C. Investigating tradeoffs in real-world video super-resolution. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2022.

Chen, S., Ma, K., and Zheng, Y. Med3d: Transfer learning for 3d medical image analysis. *arXiv preprint arXiv:1904.00625*, 2019.

Chen, T., Hou, J., Zhou, Y., Xie, H., Chen, X., Liu, Q., Guo, X., Xia, M., Duncan, J. S., Liu, C., and Zhou, B. 2.5d multi-view averaging diffusion model for 3d medical image translation: Application to low-count pet reconstruction with ct-less attenuation correction, 2024. URL https://arxiv.org/abs/2406.08374.

Cheng, S.-I., Chen, Y.-J., Chiu, W.-C., Tseng, H.-Y., and Lee, H.-Y. Adaptively-realistic image generation from stroke and sketch with diffusion model. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 4054–4062, January 2023.

Chung, H., Sim, B., Ryu, D., and Ye, J. C. Improving diffusion models for inverse problems using manifold constraints, 2022.

Chung, H., Kim, J., Mccann, M. T., Klasky, M. L., and Ye, J. C. Diffusion posterior sampling for general noisy inverse problems, 2023a.

Chung, H., Lee, E. S., and Ye, J. C. Mr image denoising and super-resolution using regularized reverse diffusion. *IEEE Transactions on Medical Imaging*, 42(4):922–934, 2023b. doi: 10.1109/TMI.2022.3220681.

Dar, S. U. H., Yurt, M., Karacan, L., Erdem, A., Erdem, E., and Çukur, T. Image synthesis in multi-contrast mri with conditional generative adversarial networks. *IEEE Transactions on Medical Imaging*, 38:2375–2388, 2018. URL https://api.semanticscholar.org/CorpusID:3655281.

DC, V. E., SM, S., DM, B., TE, B., E, Y., K, U., and WU-Minn HCP Consortium. The wu-minn human connectome project: an overview, 2013.

Dorjsembe, Z., Pao, H.-K., Odonchimed, S., and Xiao, F. Conditional diffusion models for semantic 3d brain mri synthesis. *IEEE Journal of Biomedical and Health Informatics*, 2024.

Feng, R., Weng, W., Wang, Y., Yuan, Y., Bao, J., Luo, C., Chen, Z., and Guo, B. Ccedit: Creative and controllable video editing via diffusion models. *CVPR*, 2024.

Fernandez, V., Pinaya, W. H. L., Borges, P., Tudosiu, P.-D., Graham, M. S., Vercauteren, T., and Cardoso, M. J. Can segmentation models be trained with fully synthetically generated data? In *International Workshop on Simulation and Synthesis in Medical Imaging*, pp. 79–90. Springer, 2022.

Fitriyani, N. L., Syafrudin, M., Alfian, G., and Rhee, J. Development of disease prediction model based on ensemble learning approach for diabetes and hypertension. *Ieee Access*, 7:144777–144789, 2019.

Fu, Z., Guo, L., Wang, C., Wang, Y., Li, Z., and Wen, B. Temporal as a plugin: Unsupervised video denoising with pre-trained image denoisers, 2024. URL https://arxiv.org/abs/2409.11256.

Gretton, A., Borgwardt, K. M., Rasch, M. J., Schölkopf, B., and Smola, A. A kernel two-sample test. *The Journal of Machine Learning Research*, 13(1):723–773, 2012.

Hai-Miao Zhang, B. D. A review on deep learning in medical image reconstruction. *Journal of the Operations Research Society of China*, 8(2):311, 2020. doi: 10.1007/s40305-019-00287-4. URL https://www.jorsc.shu.edu.cn/EN/abstract/article_17509.shtml.

Hatamizadeh, A., Nath, V., Tang, Y., Yang, D., Roth, H., and Xu, D. Swin unetr: Swin transformers for semantic segmentation of brain tumors in mri images, 2022.

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models, 2020.

Huang, Z., Chan, K. C. K., Jiang, Y., and Liu, Z. Collaborative diffusion for multi-modal face generation and editing, 2023.

Isola, P., Zhu, J.-Y., Zhou, T., and Efros, A. A. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1125–1134, 2017.

Jiang, A. Q., Sablayrolles, A., Roux, A., Mensch, A., Savary, B., Bamford, C., Chaplot, D. S., de las Casas, D., Hanna, E. B., Bressand, F., Lengyel, G., Bour, G., Lample, G., Lavaud, L. R., Saulnier, L., Lachaux, M.-A., Stock, P., Subramanian, S., Yang, S., Antoniak, S., Scao, T. L., Gervet, T., Lavril, T., Wang, T., Lacroix, T., and Sayed, W. E. Mixtral of experts, 2024.

Kazemi, Y. and Mirroshandel, S. A. A novel method for predicting kidney stone type using ensemble learning. *Artificial intelligence in medicine*, 84:117–126, 2018.

Kerstin, H., Teresa, K., Erich, K., P., R. M., K., S. D., Thomas, P., and Florian, K. Learning a variational network for reconstruction of accelerated mri data. *Magnetic Resonance in Medicine*, 79(6):3055–3071. doi: 10.1002/mrm.26977. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/mrm.26977.

Khader, F., Müller-Franzes, G., Tayebi Arasteh, S., Han, T., Haarburger, C., Schulze-Hagen, M., Schad, P., Engelhardt, S., Baeßler, B., Foersch, S., et al. Denoising diffusion probabilistic models for 3d medical image generation. *Scientific Reports*, 13(1):7303, 2023.

Kuleshov, V., Fenner, N., and Ermon, S. Accurate uncertainties for deep learning using calibrated regression, 2018. URL https://arxiv.org/abs/1807.00263.

Lee, S., Chung, H., Park, M., Park, J., Ryu, W.-S., and Ye, J. C. Improving 3d imaging with pre-trained perpendicular 2d diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10710–10720, 2023.

Leroy, A., Shreshtha, K., Lerousseau, M., Henry, T., Estienne, T., Classe, M., Paragios, N., Grégoire, V., and Deutsch, E. Magnetic resonance imaging virtual histopathology from weakly paired data. In Atzori, M., Burlutskiy, N., Ciompi, F., Li, Z., Minhas, F., Müller, H., Peng, T., Rajpoot, N., Torben-Nielsen, B., van der Laak, J., Veta, M., Yuan, Y., and Zlobec, I. (eds.), *Proceedings of the MICCAI Workshop on Computational Pathology*, volume 156 of *Proceedings of Machine Learning Research*, pp. 140–150. PMLR, 27 Sep 2021. URL https://proceedings.mlr.press/v156/leroy21a.html.

Li, G., Ji, J., Qin, M., Niu, W., Ren, B., Afghah, F., Guo, L., and Ma, X. Towards high-quality and efficient video super-resolution via spatial-temporal data overfitting, 2023a.

Li, X., Shang, K., Wang, G., and Butala, M. D. Ddmm-synth: A denoising diffusion model for cross-modal medical image synthesis with sparse-view measurement embedding. *arXiv preprint arXiv:2303.15770*, 2023b.

Li, X., Liu, Y., Cao, S., Chen, Z., Zhuang, S., Chen, X., He, Y., Wang, Y., and Qiao, Y. Diffvsr: Revealing an effective recipe for taming robust video super-resolution against complex degradations, 2025. URL https://arxiv.org/abs/2501.10110.

Li, Z., Wang, Y., Zhang, J., Wu, W., and Yu, H. Two-and-a-half order score-based model for solving 3d ill-posed

inverse problems. *Computers in Biology and Medicine*, 168:107819, 2024.

Liang, D., Cheng, J., Ziwen, K., and Ying, L. Deep magnetic resonance image reconstruction: Inverse problems meet neural networks. *IEEE Signal Processing Magazine*, 37: 141–151, 01 2020. doi: 10.1109/MSP.2019.2950557.

Liu, G.-H., Vahdat, A., Huang, D.-A., Theodorou, E. A., Nie, W., and Anandkumar, A. I^2sb: Image-to-image schrödinger bridge. *ICML*, 2023a.

Liu, Y., Dwivedi, G., Boussaid, F., Sanfilippo, F., Yamada, M., and Bennamoun, M. Inflating 2d convolution weights for efficient generation of 3d medical images. *Computer Methods and Programs in Biomedicine*, 240:107685, October 2023b. ISSN 0169-2607. doi: 10.1016/j.cmpb.2023.107685. URL http://dx.doi.org/10.1016/j.cmpb.2023.107685.

Lyu, Q., Shan, H., and Wang, G. Mri super-resolution with ensemble learning and complementary priors. *IEEE Transactions on Computational Imaging*, 6:615–624, 2020. doi: 10.1109/TCI.2020.2964201.

Mardani, M., Gong, E., Cheng, J. Y., Vasanawala, S., Zaharchuk, G., Alley, M., Thakur, N., Han, S., Dally, W., Pauly, J. M., et al. Deep generative adversarial networks for compressed sensing automates mri. *IEEE Trans Med Imaging*, 2019.

Meng, Z., Guo, R., Li, Y., Guan, Y., Wang, T., Zhao, Y., Sutton, B., Li, Y., and Liang, Z.-P. Accelerating t2 mapping of the brain by integrating deep learning priors with low-rank and sparse modeling. *Magnetic Resonance in Medicine*, 85(3):1455–1467, March 2021. ISSN 0740-3194. doi: 10.1002/mrm.28526. Publisher Copyright: © 2020 International Society for Magnetic Resonance in Medicine Copyright: Copyright 2020 Elsevier B.V., All rights reserved.

Nah, S., Timofte, R., Gu, S., Baik, S., Hong, S., Moon, G., Son, S., and Lee, K. M. Ntire 2019 challenge on video super-resolution: Methods and results. In *CVPR Workshops*, June 2019.

Otroshi-Shahreza, H., Amini, A., and Behroozi, H. Feature-based no-reference video quality assessment using extra trees. *IET*, 2022.

Peng, W., Adeli, E., Bosschieter, T., Park, S. H., Zhao, Q., and Pohl, K. M. Generating realistic brain mris via a conditional diffusion probabilistic model. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 14–24. Springer, 2023.

Ronneberger, O., Fischer, P., and Brox, T. U-net: Convolutional networks for biomedical image segmentation, 2015.

Saharia, C., Ho, J., Chan, W., Salimans, T., Fleet, D. J., and Norouzi, M. Image super-resolution via iterative refinement, 2021.

Saharia, C., Chan, W., Chang, H., Lee, C., Ho, J., Salimans, T., Fleet, D., and Norouzi, M. Palette: Image-to-image diffusion models. In *ACM SIGGRAPH 2022 conference proceedings*, pp. 1–10, 2022.

Shi, Z., Mettes, P., Zheng, G., and Snoek, C. Frequency-supervised mr-to-ct image synthesis. In *Deep Generative Models, and Data Augmentation, Labelling, and Imperfections: First Workshop, DGM4MICCAI 2021, and First Workshop, DALI 2021, Held in Conjunction with MICCAI 2021, Strasbourg, France, October 1, 2021, Proceedings 1*, pp. 3–13. Springer, 2021.

Song, B., Kwon, S. M., Zhang, Z., Hu, X., Qu, Q., and Shen, L. Solving inverse problems with latent diffusion models via hard data consistency. *ICLR*, 2024.

Song, Y., Durkan, C., Murray, I., and Ermon, S. Maximum likelihood training of score-based diffusion models, 2021.

Song, Y., Shen, L., Xing, L., and Ermon, S. Solving inverse problems in medical imaging with score-based generative models, 2022.

Sun, L., Chen, J., Xu, Y., Gong, M., Yu, K., and Batmanghelich, K. Hierarchical amortized gan for 3d high resolution medical image synthesis. *IEEE journal of biomedical and health informatics*, 26(8):3966–3975, 2022.

Uzunova, H., Ehrhardt, J., and Handels, H. Memory-efficient gan-based domain translation of high resolution 3d medical images. *Computerized Medical Imaging and Graphics*, 86:101801, December 2020. ISSN 0895-6111. doi: 10.1016/j.compmedimag.2020.101801. URL http://dx.doi.org/10.1016/j.compmedimag.2020.101801.

Vedula, S., Senouf, O., Bronstein, A. M., Michailovich, O. V., and Zibulevsky, M. Towards ct-quality ultrasound imaging using deep learning. *arXiv preprint arXiv:1710.06304*, 2017.

Wang, G., Ye, J. C., and De Man, B. Deep learning for tomographic image reconstruction. *Nature Machine Intelligence*, 2:737–748, 12 2020. doi: 10.1038/s42256-020-00273-z.

Wang, J., Yue, Z., Zhou, S., Chan, K. C., and Loy, C. C. Exploiting diffusion prior for real-world image super-resolution. 2024.

Wang, Z., Jiang, Y., Zheng, H., Wang, P., He, P., Wang, Z., Chen, W., and Zhou, M. Patch diffusion: Faster and more data-efficient training of diffusion models, 2023. URL https://arxiv.org/abs/2304.12526.

Wolterink, J. M., Dinkla, A. M., Savenije, M. H., Seevinck, P. R., van den Berg, C. A., and Išgum, I. Deep mr to ct synthesis using unpaired data. In *Simulation and Synthesis in Medical Imaging: Second International Workshop, SASHIMI 2017, Held in Conjunction with MICCAI 2017, Québec City, QC, Canada, September 10, 2017, Proceedings 2*, pp. 14–23. Springer, 2017.

Wu, H., Zhang, E., Liao, L., Chen, C., Hou, J. H., Wang, A., Sun, W. S., Yan, Q., and Lin, W. Exploring video quality assessment on user generated contents from aesthetic and technical perspectives. In *ICCV*, 2023.

Yang, X., He, C., Ma, J., and Zhang, L. Motion-guided latent diffusion for temporally consistent real-world video super-resolution. 2024.

Yang, Z., Zhuang, X., Sreenivasan, K., Mishra, V., Curran, T., and Cordes, D. A robust deep neural network for denoising task-based fmri data: An application to working memory and episodic memory. *Medical Image Analysis*, 60:101622, 11 2019. doi: 10.1016/j.media.2019.101622.

Zhang, H., Li, H., Dillman, J., Parikh, N., and He, L. Multi-contrast mri image synthesis using switchable cycle-consistent generative adversarial networks. *Diagnostics*, 12:816, 03 2022. doi: 10.3390/diagnostics12040816.

Zhang, L., Rao, A., and Agrawala, M. Adding conditional control to text-to-image diffusion models, 2023a.

Zhang, Z., Jiang, Y., Shao, W., Wang, X., Luo, P., Lin, K., and Gu, J. Real-time controllable denoising for image and video, 2023b. URL https://arxiv.org/abs/2303.16425.

Zhou, C., Chen, S., Ding, C., and Tao, D. Learning contextual and attentive information for brain tumor segmentation. In *Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries: 4th International Workshop, BrainLes 2018, Held in Conjunction with MICCAI 2018, Granada, Spain, September 16, 2018, Revised Selected Papers, Part II 4*, pp. 497–507. Springer, 2019.

Zhou, S., Yang, P., Wang, J., Luo, Y., and Loy, C. C. Upscale-a-video: Temporal-consistent diffusion model for real-world video super-resolution, 2024.

Zhu, L., Xue, Z., Jin, Z., Liu, X., He, J., Liu, Z., and Yu, L. Make-a-volume: Leveraging latent diffusion models for cross-modality 3d brain mri synthesis, 2023a.

Zhu, L., Codella, N., Chen, D., Jin, Z., Yuan, L., and Yu, L. Generative enhancement for 3d medical images. *arXiv preprint arXiv:2403.12852*, 2024.

Zhu, X., Zyrianov, V., Liu, Z., and Wang, S. Mapprior: Bird's-eye view map layout estimation with generative models, 2023b.

Method	SR			MT			both condition		
	PSNR(↑)	SSIM(↑)	MACE(1e-4)(↓)	PSNR	SSIM	MACE	PSNR	SSIM	MACE
TPDM(Lee et al., 2023)	32.23 ± 0.0066	0.922 ± 0.000300	53.16	25.35 ± 0.2363	0.868 ± 0.00112	279.3	35.12 ± 0.0154	0.945 ± 0.00056	48.72
Ours-TPDM	33.24 ± 0.0298	0.944 ± 0.000261	44.63	25.26 ± 0.664	0.882 ± 0.00357	268.3	36.24 ± 0.0389	0.961 ± 0.000249	38.49
TOSM(Li et al., 2024)	32.76 ± 0.02157	0.932 ± 0.000434	51.68	25.66 ± 0.2703	0.881 ± 0.00132	221.7	35.44 ± 0.0173	0.947 ± 0.00073	46.32
Ours-TOSM	33.30 ± 0.0296	0.945 ± 0.000240	42.89	25.24 ± 0.668	0.882 ± 0.00317	200.8	36.51 ± 0.0364	0.963 ± 0.000210	36.67
MADM(Chen et al., 2024)	33.02 ± 0.0276	0.946 ± 0.000436	83.20	25.47 ± 0.2573	0.874 ± 0.00143	251.4	35.21 ± 0.0165	0.946 ± 0.00063	47.89
Ours-MADM	33.31 ± 0.0308	0.945 ± 0.000278	42.03	25.13 ± 0.667	0.876 ± 0.00342	234.6	36.37 ± 0.0379	0.964 ± 0.00226	37.15

Table 6: Quantitative evaluation of Score-Fusion on BraTS dataset with uncertainty metrics. Our models' performance boost is significant, given low standard deviations. Our model can also estimate uncertainty better through the standard deviation obtained by inference multiple times.

A. Overview

In this supplementary material, we first discuss the uncertainty awareness results performed by our model and baselines by running the inference multiple times in Sec. B. We provide more randomly selected results (We do exclude samples with low-quality GT) for more baselines and our variants in Sec. C. Then, we provide our super-resolution result in an additional dataset, HCP dataset (DC et al., 2013) in Sec. D. We provide ablation studies on key techniques in Sec. E. We also provide more details on training and inference, including a detailed model architecture in Sec. F and training/inference speed in Sec. G. We finally introduce a more detailed method for self-consistency projection in Sec. H and downstream task results in Sec. I.

B. Uncertainty Awareness

As with most diffusion-based models, our models and some of our baselines can have uncertainty estimations. To study this uncertainty, we perform inference five times for each sample in our validation set. This gives us 5 PSNR and SSIM values for each data sample. We then calculate the standard deviation (std) of the PSNR and SSIM for each sample and include the mean std across the entire validation set in Tab. 6. This further validates that our performance boost in PSNR and SSIM is significant. For the main variant, TPDM and Ours-TPDM, in the super-resolution task, we have a 1.01 boost in PSNR, which is much larger than the std of PSNR for both models (0.0066 and 0.0298). Even for MADM and Ours-MADM, where we have the most marginal PSNR boost, the boost is still 0.3, around 10 times larger than the std for both models (0.0308 and 0.0276). In contrast, in modality translation, the std is significantly larger since the uncertainty in this task is much larger than in others, indicating the PSNR drop is not as significant. In fact, previous work (Saharia et al., 2021) argues that PSNR prefers blurry results, and highly diverse and realistic results typically have low PSNR in tasks with high uncertainty.

In addition, this inference also provides a mean μ_i and std estimation σ_i for each voxel. We use Mean Absolute Calibration Error (MACE) (Kuleshov et al., 2018) to measure the uncertainty awareness of our model and baseline. MACE measures the absolute difference between the predicted uncertainty and the actual error, as shown in Eq. 5.

$$\text{MACE} = \frac{1}{N}\sum_{i=1}^{N} |\sigma_i - |y_i - \mu_i|| \tag{5}$$

As demonstrated in Table 6, all variants of our model exhibit lower MACE values compared to their respective baselines. This indicates that the standard deviation (std) predicted by our model, derived from multiple inferences, provides a more accurate estimation of the true error relative to the ground truth. Consequently, our model exhibits improved uncertainty awareness. For qualitative results in uncertainty awareness, we demonstrate our model's results with the uncertainty map and error map across various tasks and variants in Fig. 14 15 16 17 18 19 20 21. As shown in the figures, the uncertainty map aligns well with the actual error map, demonstrating decent uncertainty awareness for all models. Notably, our model

usually has a higher uncertainty in modality translation tasks in Fig. 16 and 17. In the modality translation task, the $p(\boldsymbol{y}|\boldsymbol{x})$ should have a high variance in overall contrast. Our model outputs samples that are highly diverse in overall contrast, indicating that Score-Fusion is able to model the target 3D conditional distribution $p(\boldsymbol{y}|\boldsymbol{x})$ better. In contrast, our baselines tend to output the mean estimation for overall contrast, demonstrating higher PSNR but limited capability of generating diverse and realistic results.

C. Additional qualitative result

We show results for the variants that show the best metrics. Namely, we show results for MADM and Ours-MADM in the super-resolution task in Fig. 14 and 15, and show results for TOSM and Ours-TOSM for the other two tasks in Fig. 18 19 16 and 17. We include more samples in all 3 views in Fig. 20 and Fig. 21 in super-resolution tasks in addition to Fig. 3. Each figure contains 2 sample volumes, each of which contains visualizations in all three views in three rows. We show all results with uncertainty and error maps.

Similarly to Fig. 3, we find that both MADM and TOSM demonstrate similar artifacts as TPDM in high-frequency details due to a direct averaging in the score function. In contrast, Score-Fusion consistently demonstrates better 3D consistency and realism across all views by introducing pixel-space 3D representation and networks to replace the weighted averaging in the score function space. For example, in the 6-th row of Fig. 14 16, and 18, the results from baselines are blurry at the top left part of the brain, whereas Score-Fusion shows more smooth and consistent results.

D. Result on HCP dataset

We present our super-resolution results on the FLAIR modality in the HCP (DC et al., 2013) dataset to show our model is generalizable across datasets. The HCP dataset consists of 1251 MRI volumes with a resolution of 192x152x152. In contrast to the BraTs dataset, HCP comprises healthy brains with no brain tumors. Experiments results in Tab. 7 show that our model shows around 1.5 performance boost in PSNR. We also present the qualitative results in Fig. 4, including all 3 views. Again, Score-Fusion shows better 3D consistency and realism. For example, in the top-right part in the third view, our baseline demonstrates jittering and artifacts, while our model produces more realistic detail and smoother edges.

Method	PSNR	SSIM	MMD	FID(1e-4)
TPDM	28.17	0.890	81.81	35.70
Ours-TPDM	29.62	0.914	67.96	22.12

Table 7: Super-resolution result in HCP dataset.

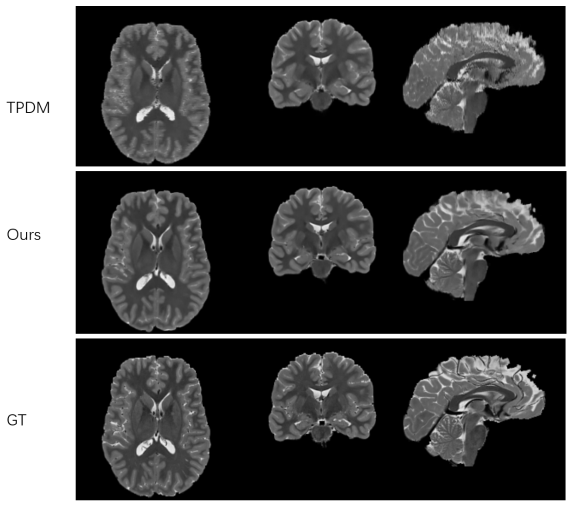

Figure 4: Qualitative results in HCP dataset. The input is a 4x4x4 downsampled version of the ground truth.

E. Ablation Studies

We provide an ablation study in Tab. 8 on the key design elements for Score-Fusion, which includes: (1) **Feature merging**: The 2D models not only contribute their outputs but also pass their feature maps to the 3D model. (2) **Finetune**: We initially pre-train the model on 3D patches and then fine-tune it on the full volume to speed up training. (3) **Consistency**: Inspired by DPS (Chung et al., 2023a) and score-SDE (Song et al., 2022), we implement self-consistency projections at each denoising step. All these designs show performance gain in the super-resolution task. In addition, we benchmark the smaller variant in Sec. 4.4 for comparison, which shows a moderate performance drop compared to our best model.

Smaller model	Consistency	Finetune	Feature	PSNR	SSIM	MMD
–	–	–	–	32.83	0.935	25.45
–	✓	–	–	32.88	0.94	18.84
–	✓	✓	–	32.97	0.941	15.24
–	✓	–	✓	33.04	0.942	17.76
–	✓	✓	✓	33.24	0.944	13.77
✓	✓	✓	–	32.8	0.937	16.78

Table 8: Ablation studies of additional design elements in Score-Fusion.

F. Detailed model architecture

In this section, we show detailed model architecture for 2D, 3D, and the smaller variant of the 3D model in Tab. 11, Tab. 12, and Tab. 13, respectively. In addition, we show other related hyper-parameters in Tab. 9. We modified the architecture of the 2D diffusion model from Palette (Saharia et al., 2022) and the 3D models from med-ddpm (Dorjsembe et al., 2024).

Given the differences in problem setting and dataset between our work and that of Palette, we conduct a comprehensive hyper-parameter search based on the super-resolution tasks. This search explores various configurations, including the number of channels, transformer layers, and learning rate, among others. The hyper-parameter search is conducted to optimize the performance of our baseline models, Palette2D, Palette3D, and Palette2.5D, in Tab. 1. While such a search could potentially enhance the performance of our proposed model, we do not perform a hyper-parameter search to optimize the performance of Score-Fusion, TPDM, TOSM, and MADM. This practice ensures a fair comparison between our model and their corresponding baselines, TPDM, TOSM, and MADM. Moreover, this shows that our model can be a plug-in-and-play mechanism for existing pre-trained 2D and 3D model architecture.

Table 9: Other hyper-parameters

Parameter	**2D Network**	**3D Network**
Batch size	4	1
Diffusion steps	1000	1000
Inference steps (DDIM)	50	50
Noise scheduler	Linear	Linear
Learning rate	0.00005	0.0001
Optimizer	Adam	Adam

G. Training and inference speed

We present training inference speed in Tab. 10. All experiments are done with RTX A100-40GB GPU. Since Score-Fusion needs to train an additional model on top of the baselines, our training time is inevitably higher. We need 16 GPU days to train our 3D models, which results in a 16-day increase in training time for most model variants compared to their corresponding baselines. Our models are also relatively slower in inference since we need to perform inference for an additional 3D model. However, as mentioned in Sec. 1, the 3D model is naturally limited in size due to computational challenges in training. Therefore, 3D inference is more efficient than slice-wise 2D inference. As a result, the increase in inference time is significantly smaller than in training. As shown in Tab. 10, our 3D model is around 30% faster than one 2D model and, therefore, leads to a 36% increase in inference time for Ours-TPDM and 26% for Ours-MADM and Ours-MADM.

Moreover, we find that the TPDM-based models are significantly faster than other variants of the models. Given the advantage of computational efficiency, we use TPDM and Ours-TPDM as our main variables for the model and the baseline.

Furthermore, to perform a more complete ablation study, the smaller 3D model decreases the inference and training time of the 3D model by 75% while showing a consistent performance boost over TPDM and a moderate performance drop compared to Ours-TPDM as shown in Tab. 1. Moreover, compared to 3D Palette baseline, our model effectively decreased 3D training from 120 GPU days to 16/4 days, addressing the computational challenge of 3D diffusion training.

Table 10: Training and Inference time for each model, GPUs are A100 with 40G memory.

Time	**Training time** (GPU days)	**Inference time** (minutes per volume)
2D Palette	8	0.85
2D I2SB	5	4.58
3D Pix2pix	6	0.0398
3D Unet	6	0.0398
3D Palette	120	0.6
TPDM	16	1.72
Ours-TPDM	32	2.34
Ours-TPDM-small	20	1.92
TOSM	24	2.55
Ours-TOSM	40	3.23
MADM	36	2.76
Ours-MADM	52	3.56

H. Details for consistency projection

In this section, we provide the exact definition and detail for self-consistency projection mentioned in Sec. 3.1. In this work, we address the inverse problem using a diffusion model with consistency projections. The goal is to recover a high-resolution image, $\boldsymbol{y}$, from its low-resolution observation $\boldsymbol{x}$, which is obtained through a linear degradation process. Specifically, the degradation process is modeled as: $\boldsymbol{x} = A\boldsymbol{y}$.

In the 3D case, the degradation operator A represents a downsampling operation that reduces the resolution of a volume $\boldsymbol{y}$ by a factor of 4 along each spatial dimension (x, y and z) and resizes it back to the original resolution. This means that each voxel in the low-resolution volume $\boldsymbol{x}$ corresponds to the average of a [4×4×4] region in the high-resolution volume $\boldsymbol{y}$. Specifically, let $\boldsymbol{y}, \boldsymbol{x} \in \mathbb{R}^{b_1 \times b_2 \times b_3}$. The operator matrix $A \in \mathbb{R}^{b_1 \times b_2 \times b_3, b_1 \times b_2 \times b_3}$ downscales the high-resolution volume $\boldsymbol{y}$ into the low-resolution volume $\boldsymbol{x}$ by averaging over [4x4x4] blocks of voxel of $\boldsymbol{y}$. Therefore, A is a sparse matrix where each non-zero entry corresponds to the average of a block of [4x4x4] voxels in $\boldsymbol{y}$ being averaged to form a block of voxel in $\boldsymbol{x}$. Therefore, A is $\frac{1}{64}$ for the places where $\boldsymbol{x}$ and $\boldsymbol{y}$ belong to the same block, and A would be 0 elsewhere:

$$\begin{aligned} A[(i,j,k),(p,q,r)] &= \frac{1}{64} \quad \text{if } \boldsymbol{x}(i,j,k), \boldsymbol{y}(p,q,r) \in \mathit{block} \\ A[(i,j,k),(p,q,r)] &= 0 \qquad\qquad\qquad\quad \text{Otherwise} \end{aligned} \tag{6}$$

Since we are doing average over [4x4x4], $\boldsymbol{x}(i,j,k)$ and $\boldsymbol{y}(p,q,r)$ are in the same block if and only if $i//4 == p//4$, $j//4 == q//4$, and $k//4 == r//4$.

In our diffusion process, we use $\hat{\boldsymbol{y}}_0(t) \leftarrow \hat{\boldsymbol{y}}_0(t) - A^T(AA^T)^{-1}(A\hat{\boldsymbol{y}}_0(t) - x)$ to make every of our mean prediction of $\hat{y}_0$ a plausible estimation with $\boldsymbol{x} = A\hat{y}_0(t)$

To compute matrix multiplication more efficiently in a super-resolution setting, we actually use $\hat{y}_0(t) \leftarrow \hat{y}_0(t) - (A\hat{y}_0(t) - x)$ in our code. This works in the average pooling downsample because $AA\boldsymbol{y} = A\boldsymbol{y}$ since A represents the degradation process composed of average pooling followed by resizing the image back to its original resolution.

I. Details in downstream task

In Section 4.3, we evaluate tumor segmentation performance using three types of FLAIR inputs: the ground truth FLAIR modality, 4x downsampled FLAIR modality (as described in Section 4.1), and the 4x super-resolution FLAIR prediction

on the downsampled FLAIR. Accurate tumor segmentation is crucial in medical imaging, and its performance heavily relies on the quality of the input data. It requires High-quality inputs for precise localization and delineation of tumor boundaries, while, depending on its degradation level, the degraded inputs could significantly lower segmentation accuracy and reliability. We use a robust pre-trained segmentation model, SwinUNet (Hatamizadeh et al., 2022), which takes four modalities (T1, T1ce, T2, and FLAIR) as input. For this downstream task, our objective is to assess how well the models can recover segmentation performance when working with degraded inputs. Segmentation is performed with other modalities with ground truth inputs and a FLAIR input from the ground truth FLAIR, downsampled FLAIR, or the model-predicted FLAIR. Note that because there is no degraded FLAIR modality available in the modality translation task, only dice scores are reported. For other tasks, including the super-resolution and both condition tasks, performance is measured using two metrics: (1) **Dice Score**, the primary metric of the segmentation model, and (2) **Recovery Rate**, a measure of how well model predictions improve upon degraded FLAIR inputs. The recovery rate is calculated as:

$$\text{Recovery Rate} = \frac{\text{Prediction} - \text{Downsample}}{\text{Ground Truth} - \text{Downsample}}$$

where **Prediction** refers to the segmentation performance using predicted FLAIR, **Downsample** is the performance with downsampled FLAIR, and **Ground Truth** is the performance with ground truth FLAIR.

Fig. 5, 6, 7 illustrate the Dice score and Recovery rate comparisons across tumor categories. Dashed lines represent the lower and upper bounds. They show that segmentation performance with the predicted FLAIR modality from Score-Fusion-based models outperforms other methods, as Score-Fusion-based models are constantly positioned higher than others.

We also show qualitative results in the tumor segmentation task, on TPDM, TOSM, and Score-Fusion built based on these two models. Fig 8 and Fig 9 show the results in super-resolution, Fig 10 and Fig 11 show the results in modality translation, and Fig 12 and Fig 13 show the results given both conditions. In Fig 8, 13, 12, and 13, in the sagittal plane, we can observe that our models help segmentation model capture a branch coming out of the whole tumor, indicated by a red bounding box. This branch is only partially captured or entirely missed in predictions from other models. This shows that our model yields more precise predictions, allowing the tumor segmentation model to delineate the entire tumor boundary more accurately.

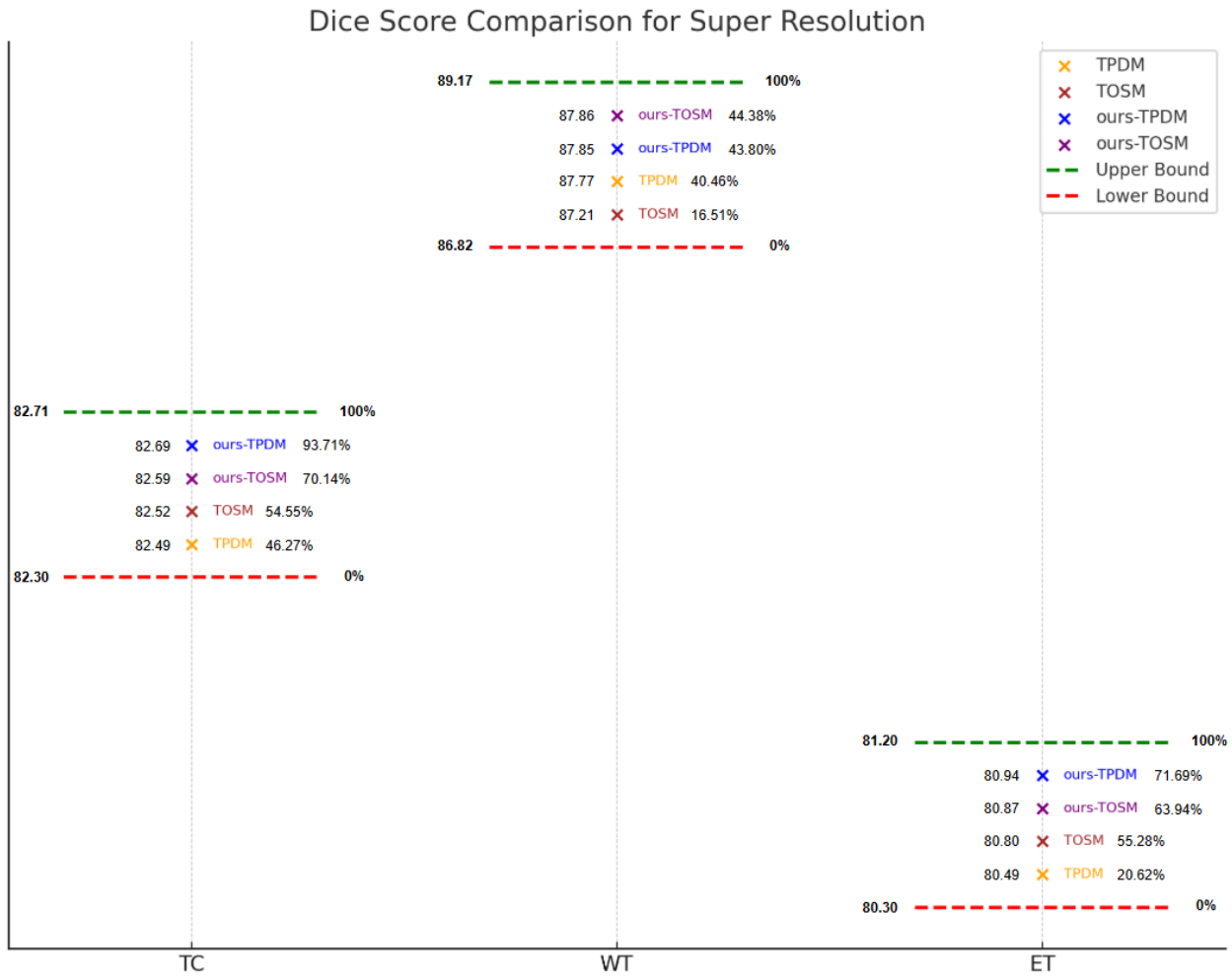

Figure 5: Comparison of Dice scores and recovery rates for super-resolution. The value on the left represents the Dice score, while the value on the right represents the recovery rate.

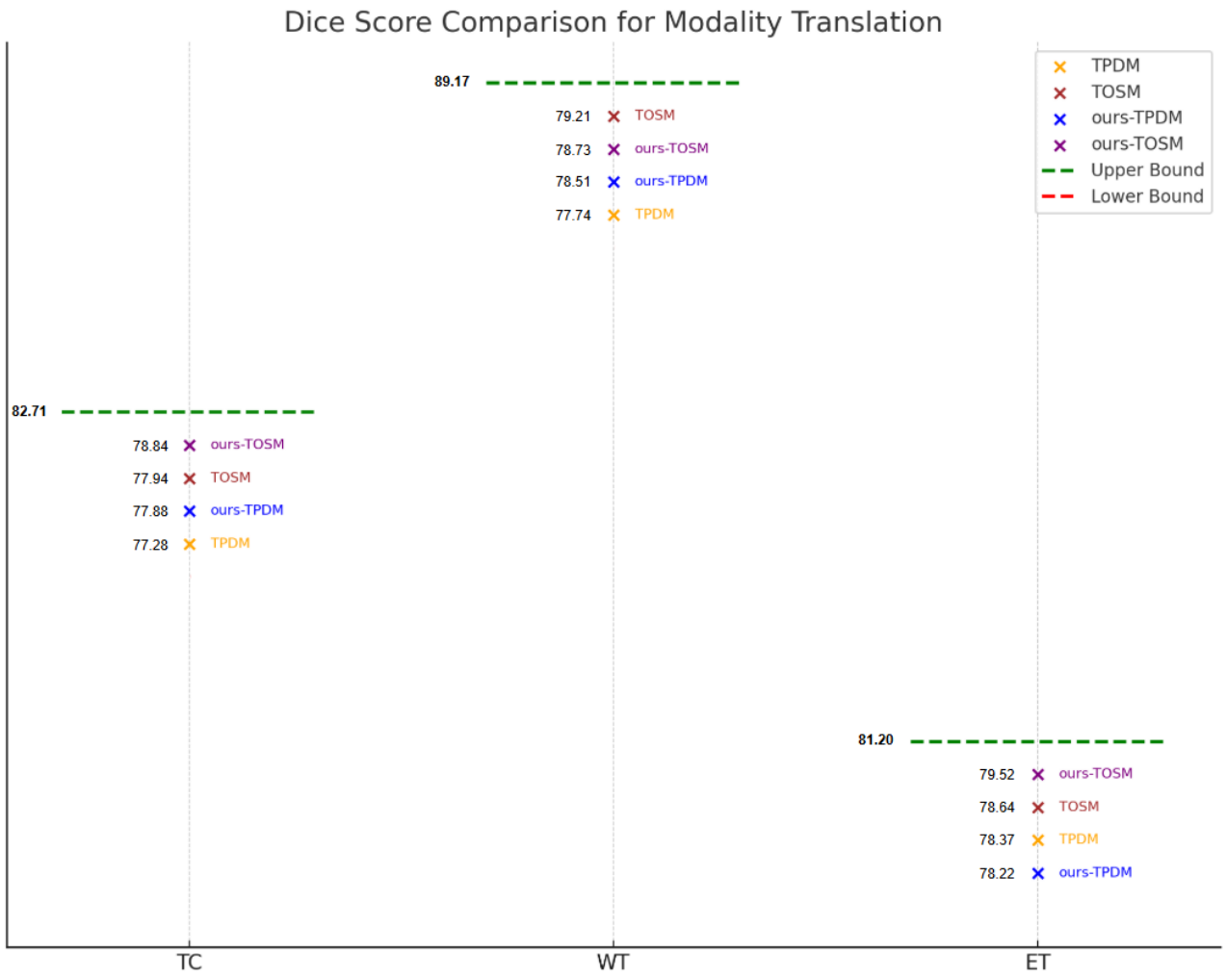

Figure 6: Comparison of Dice scores and recovery rates for modality translation. The value on the left represents the Dice score.

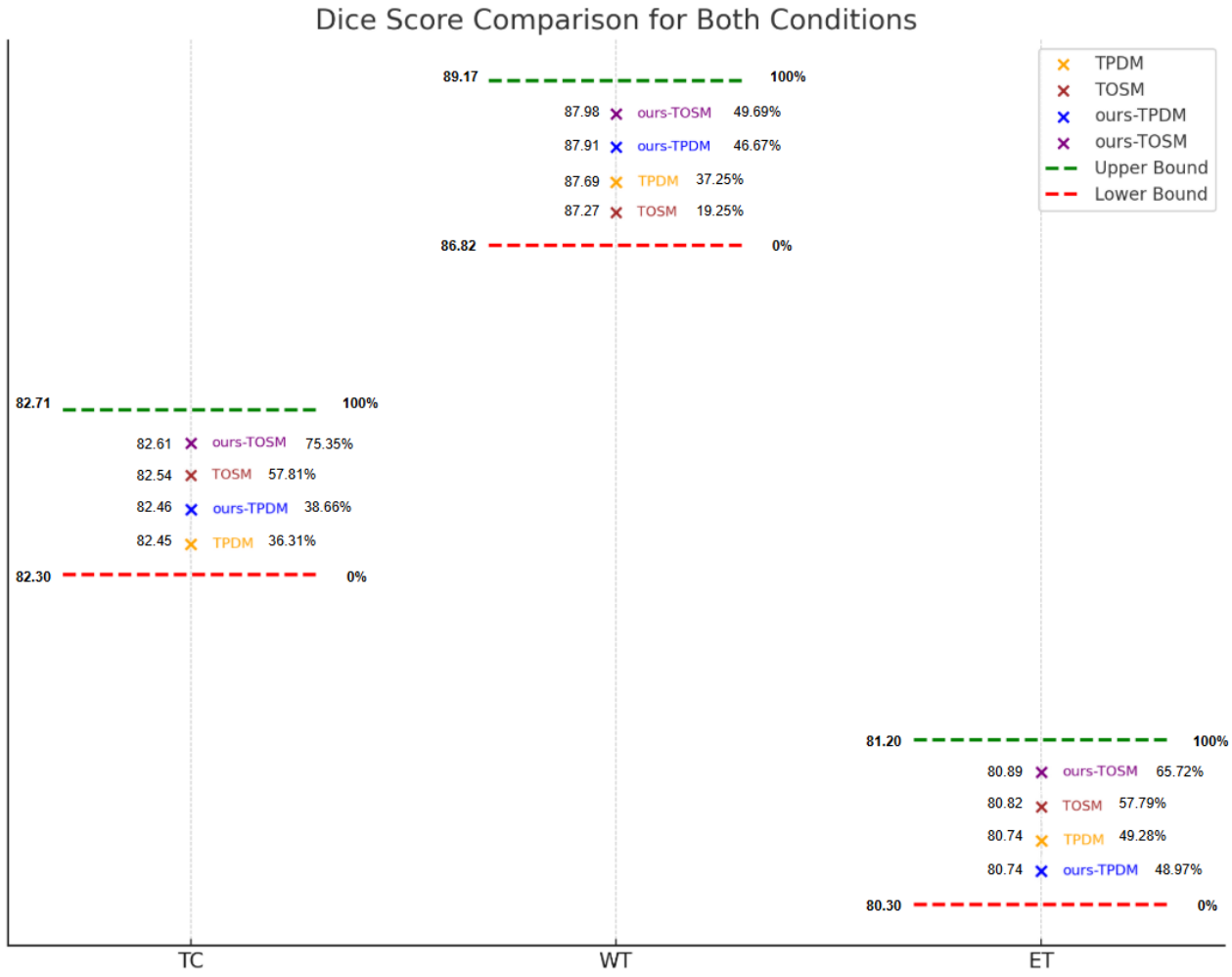

Figure 7: Comparison of Dice scores and recovery rates for both conditions. The value on the left represents the Dice score, while the value on the right represents the recovery rate.

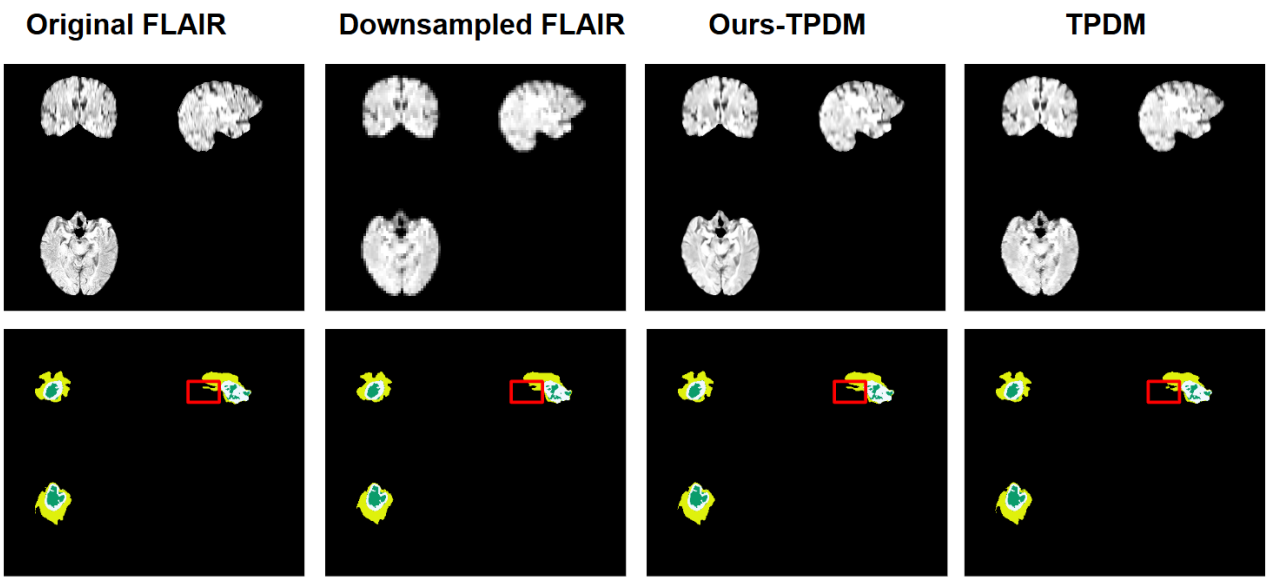

Figure 8: Qualitative results for the downstream task, tumor segmentation, in super-resolution task.

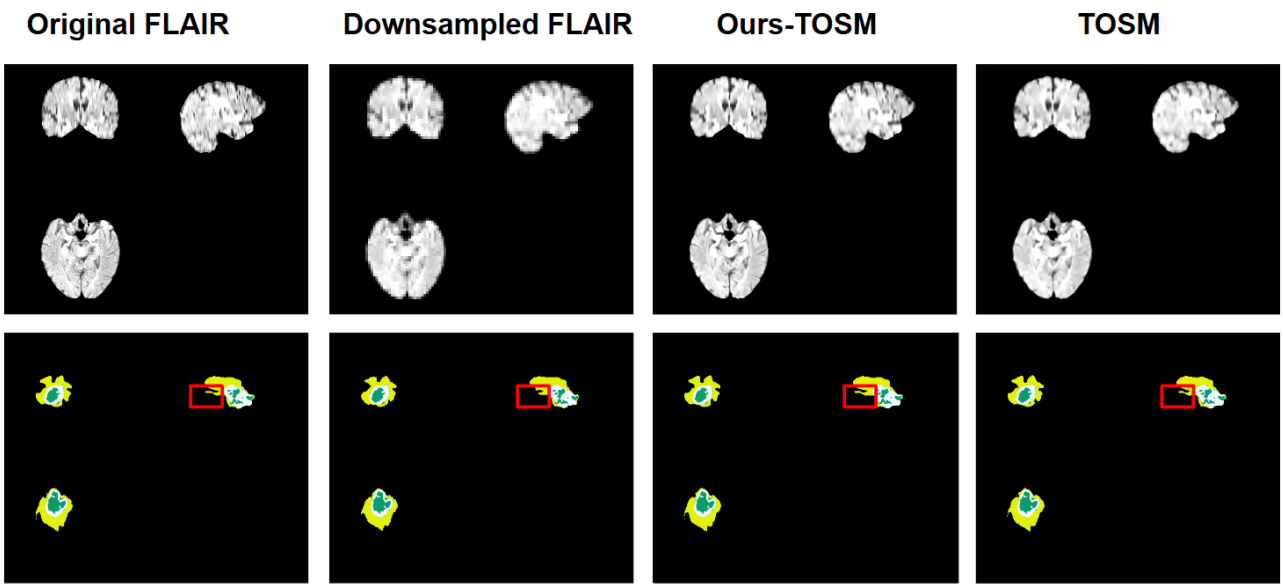

Figure 9: Qualitative results for the downstream task, tumor segmentation, in super-resolution task

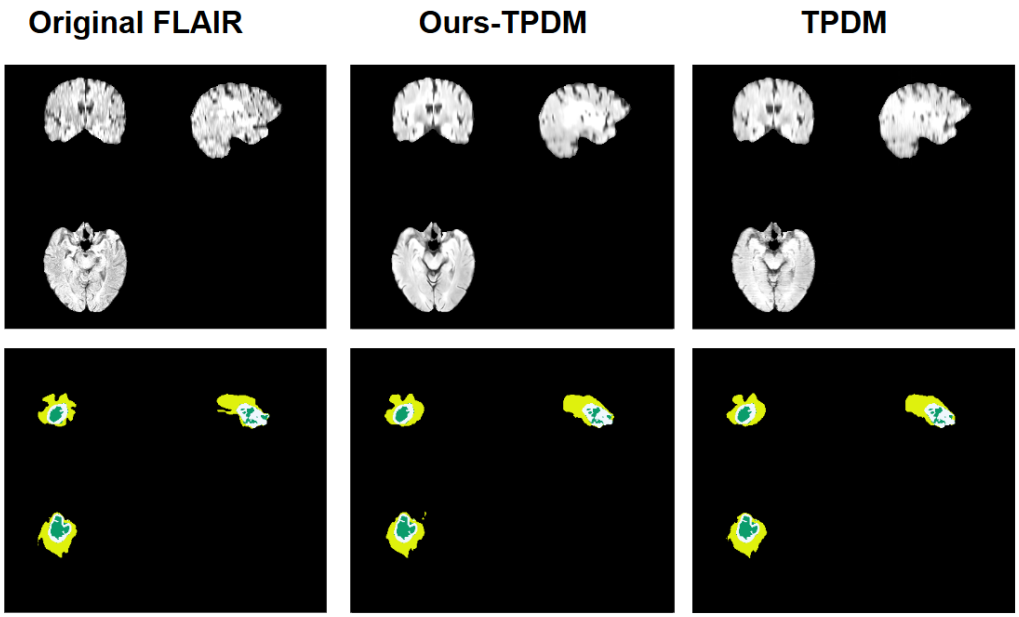

Figure 10: Qualitative results for the downstream task, tumor segmentation, in modality translation task

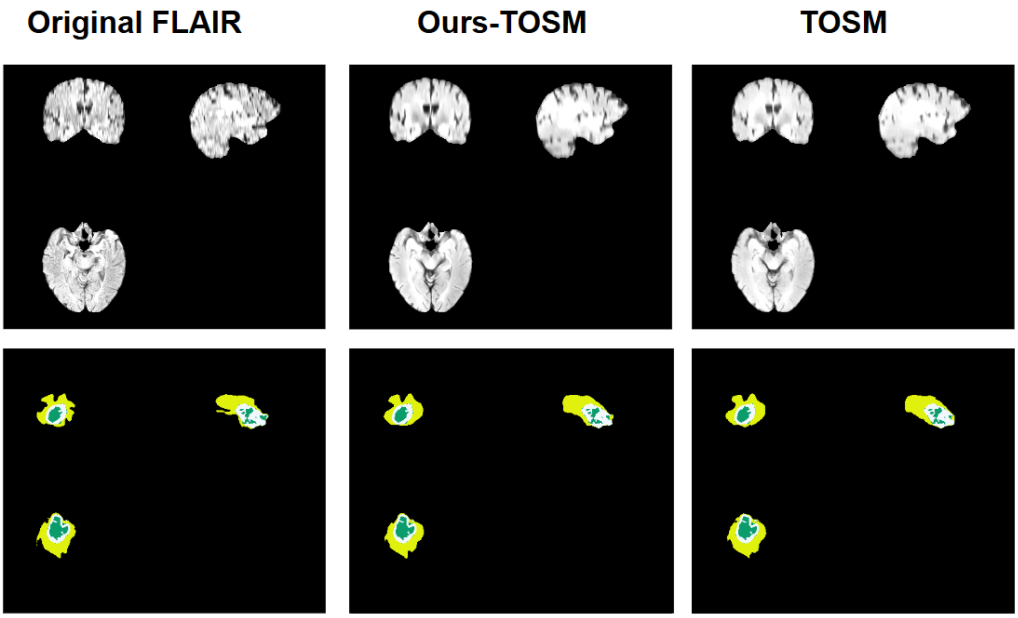

Figure 11: Qualitative results for the downstream task, tumor segmentation, in modality translation task

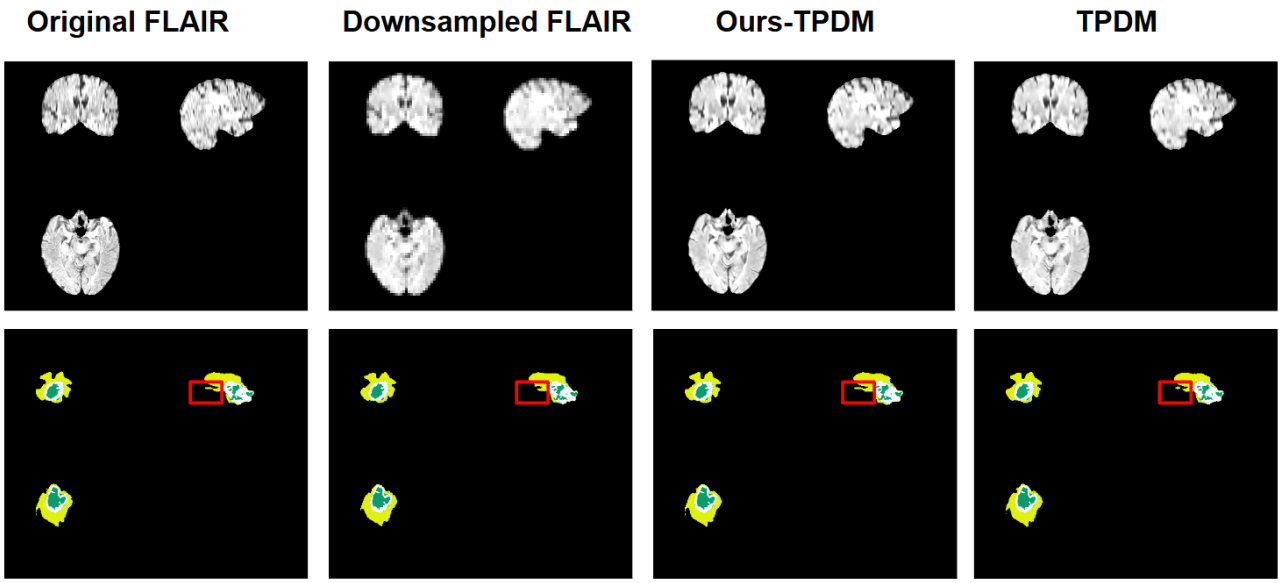

Figure 12: Qualitative results for the downstream task, tumor segmentation, in both-condition task

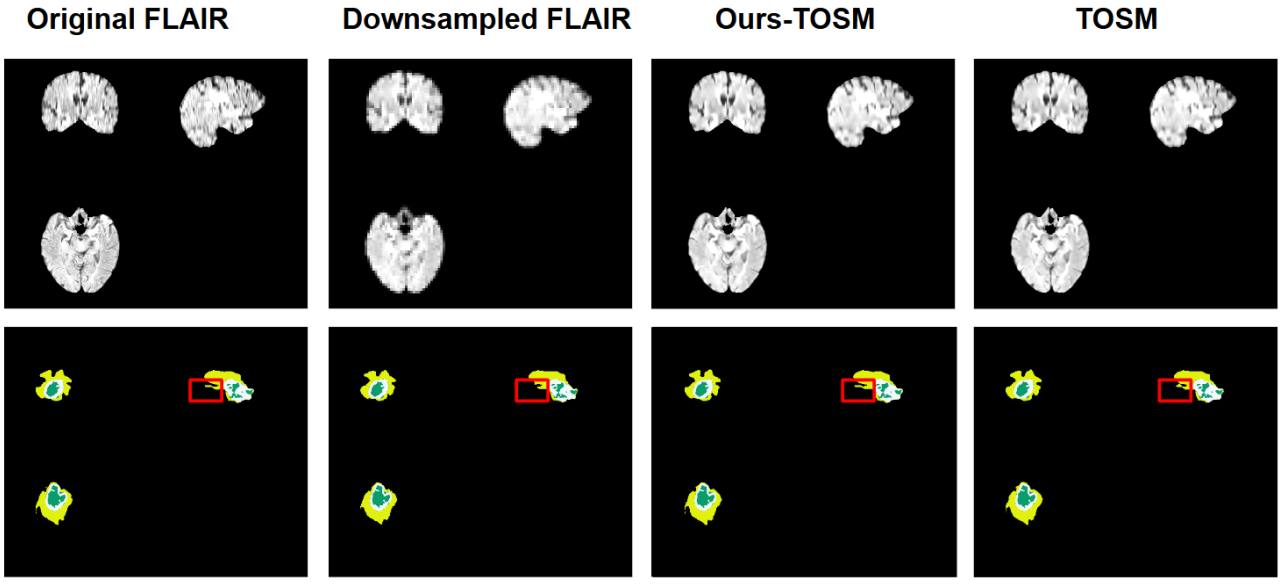

Figure 13: Qualitative results for the downstream task, tumor segmentation, in both-condition task

Table 11: Architecture for 2D diffusion model. Each ResnetBlock consists of 3 conv2D layers of the same channel and a skip connection. All ResnetBlocks are used with time embedding.

layers		parameters
input	Conv3d	in_ch: 5, out_ch: 64, kernel: 3x3, stride: 1, pad: 1
Time_Embed	Linear	in_ch:64, out_ch: 256
	Activateion	Swish
	Linear	in_ch:256, out_ch: 256
Downsample_block_1	ResnetBlock	in_ch:64, out_ch: 64
	ResnetBlock	in_ch:64, out_ch: 64
	Downsample(Conv3d)	in_ch:64, out_ch: 64,kernel:3x3, stride:2)
Downsample_block_2	ResnetBlock	in_ch:64, out_ch: 128
	ResnetBlock	in_ch:128, out_ch: 128
	Downsample(Conv3d)	in_ch:128, out_ch: 128,kernel:3x3, stride:2
Downsample_block_3	ResnetBlock	in_ch:128, out_ch: 256
	ResnetBlock	in_ch:256, out_ch: 256
	Downsample(Conv3d)	in_ch:256, out_ch: 256,kernel:3x3, stride:2
Downsample_block_4	ResnetBlock	in_ch:256, out_ch: 512
	ResnetBlock	in_ch:512, out_ch: 512
Middle	ResnetBlock	in_ch:512, out_ch: 512
	ResnetBlock	in_ch:512, out_ch: 512
Upsample_block_1	ResnetBlock	in_ch:512, out_ch: 512
	ResnetBlock	in_ch:512, out_ch: 512
	Upsample	Conv3d and F.interpolate
Upsample_block_2	ResnetBlock	in_ch:512, out_ch: 256
	ResnetBlock	in_ch:256, out_ch: 256
	Upsample	Conv3d and F.interpolate
Upsample_block_3	ResnetBlock	in_ch:256, out_ch: 128
	ResnetBlock	in_ch:128, out_ch: 128
	Upsample	Conv3d and F.interpolate
Upsample_block_4	ResnetBlock	in_ch:128, out_ch: 64
	ResnetBlock	in_ch:64, out_ch: 64
Out	Normalize	64
	Activation	nn.SiLU
	Conv3d	in_ch:64, out_ch: 1, kernel: 3x3, stride: 1, pad: 1

Table 12: Architecture for 3D diffusion model. Each ResnetBlock consists of 2 conv3D layers of the same channel and a skip connection. All ResnetBlocks are used with time embed with an embedding layer, as well as gradient checkpoint

layers		parameters
input	Conv3d	in_ch: 5, out_ch: 64, kernel: 3x3, stride: 1, pad: 1
Time_Embed	Linear	in_ch:64, out_ch: 256
	Activateion	nn.SiLU
	Linear	in_ch:256, out_ch: 256
Downsample_block_1	ResnetBlock	in_ch:64, out_ch: 64
	ResnetBlock	in_ch:64, out_ch: 64
	Feature_injetced_from_2D	in_ch:64, out_ch: 64)
	Downsample(Conv3d)	in_ch:64, out_ch: 64,kernel:3x3, stride:2)
Downsample_block_2	ResnetBlock	in_ch:64, out_ch: 128
	ResnetBlock	in_ch:128, out_ch: 128
	Feature_injetced_from_2D	in_ch:128, out_ch: 128)
	Downsample(Conv3d)	in_ch:128, out_ch: 128,kernel:3x3, stride:2
Downsample_block_3	ResnetBlock	in_ch:128, out_ch: 192
	ResnetBlock	in_ch:192, out_ch: 192
	Feature_injetced_from_2D	in_ch:192, out_ch: 192)
	Downsample(Conv3d)	in_ch:192, out_ch: 192,kernel:3x3, stride:2
Downsample_block_4	ResnetBlock	in_ch:192, out_ch: 256
	ResnetBlock	in_ch:256, out_ch: 256
	Feature_injetced_from_2D	in_ch:256, out_ch: 256)
Middle	ResnetBlock	in_ch:256, out_ch: 256
	ResnetBlock	in_ch:256, out_ch: 256
Upsample_block_1	ResnetBlock	in_ch:256, out_ch: 256
	ResnetBlock	in_ch:256, out_ch: 256
	Upsample	Conv3d and F.interpolate
Upsample_block_2	ResnetBlock	in_ch:256, out_ch: 192
	ResnetBlock	in_ch:192, out_ch: 192
	Upsample	Conv3d and F.interpolate
Upsample_block_3	ResnetBlock	in_ch:192, out_ch: 128
	ResnetBlock	in_ch:128, out_ch: 128
	Upsample	Conv3d and F.interpolate
Upsample_block_4	ResnetBlock	in_ch:128, out_ch: 64
	ResnetBlock	in_ch:64, out_ch: 64
Out	Normalize	64
	Activation	nn.SiLU
	Conv3d	in_ch:64, out_ch: 2, kernel: 3x3, stride: 1, pad: 1

Table 13: Architecture for the smaller variant of 3D diffusion model. Again, each ResnetBlock consists of 2 conv3D layers of the same channel and a skip connection. All ResnetBlocks are used with time embedding with an embedding layer, as well as a gradient checkpoint. We used a smaller number of channels for each layer and omitted the feature injection from 2D

layers		parameters
input	Conv3d	in_ch: 5, out_ch: 32, kernel: 3x3, stride: 1, pad: 1
Time_Embed	Linear	in_ch:32, out_ch: 128
	Activateion	nn.SiLU
	Linear	in_ch:128, out_ch: 128
Downsample_block_1	ResnetBlock	in_ch:32, out_ch: 32
	ResnetBlock	in_ch:32, out_ch: 32
	Downsample(Conv3d)	in_ch:32, out_ch: 32,kernel:3x3, stride:2)
Downsample_block_2	ResnetBlock	in_ch:32, out_ch: 64
	ResnetBlock	in_ch:64, out_ch: 64
	Downsample(Conv3d)	in_ch:64, out_ch: 64,kernel:3x3, stride:2
Downsample_block_3	ResnetBlock	in_ch:64, out_ch: 64
	ResnetBlock	in_ch:64, out_ch: 64
	Downsample(Conv3d)	in_ch:64, out_ch: 64,kernel:3x3, stride:2
Downsample_block_4	ResnetBlock	in_ch:63, out_ch: 128
	ResnetBlock	in_ch:128, out_ch: 128
Middle	ResnetBlock	in_ch:128, out_ch: 128
	ResnetBlock	in_ch:128, out_ch: 128
Upsample_block_1	ResnetBlock	in_ch:128, out_ch: 128
	ResnetBlock	in_ch:128, out_ch: 128
	Upsample	Conv3d and F.interpolate
Upsample_block_2	ResnetBlock	in_ch:128, out_ch: 64
	ResnetBlock	in_ch:64, out_ch: 64
	Upsample	Conv3d and F.interpolate
Upsample_block_3	ResnetBlock	in_ch:64, out_ch: 64
	ResnetBlock	in_ch:64, out_ch: 64
	Upsample	Conv3d and F.interpolate
Upsample_block_4	ResnetBlock	in_ch:64, out_ch: 32
	ResnetBlock	in_ch:32, out_ch: 32
Out	Normalize	64
	Activation	nn.SiLU
	Conv3d	in_ch:32, out_ch: 2, kernel: 3x3, stride: 1, pad: 1

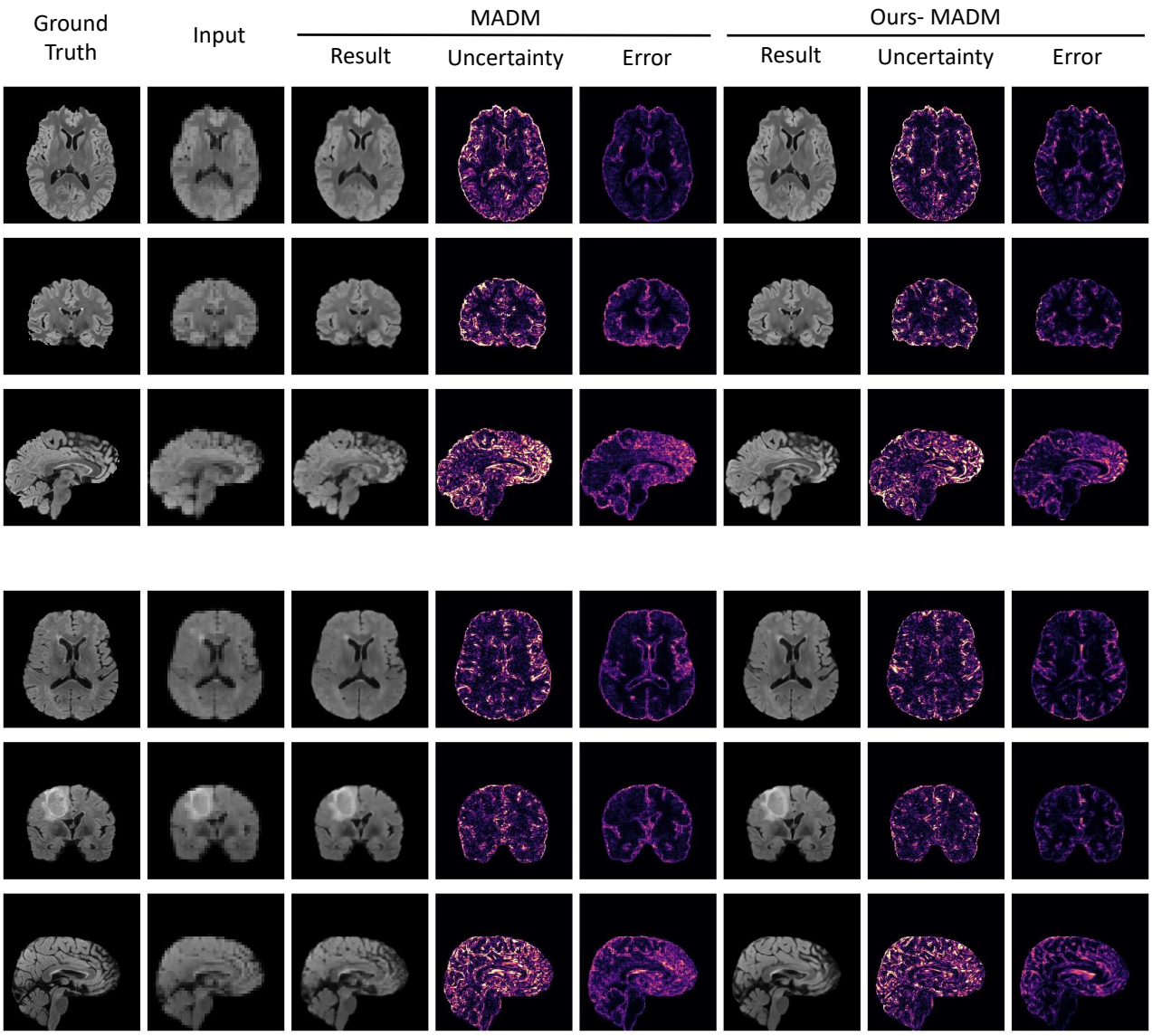

Figure 14: Uncertainty awareness results on super-resolution task

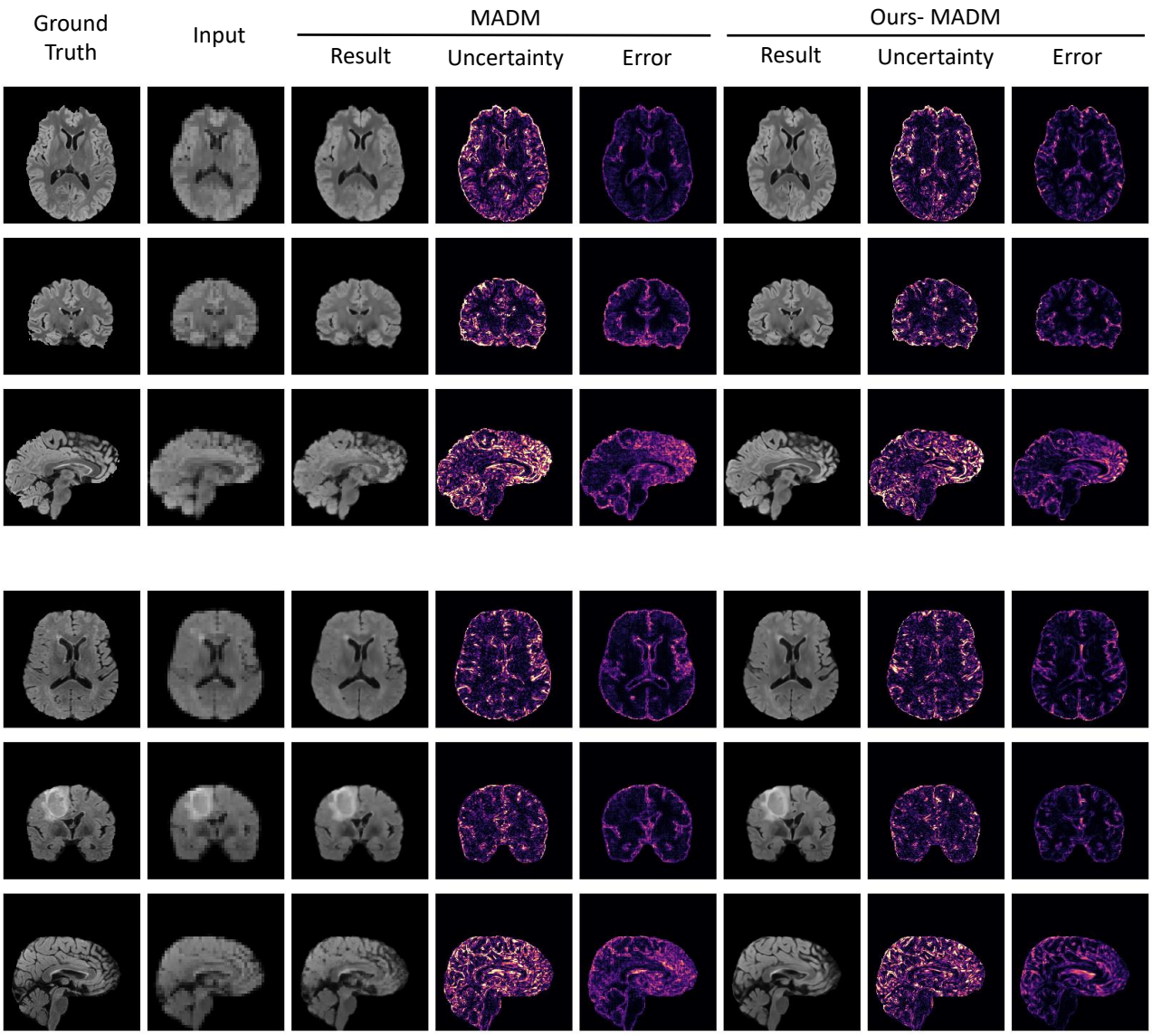

Figure 15: Uncertainty awareness results on super-resolution task

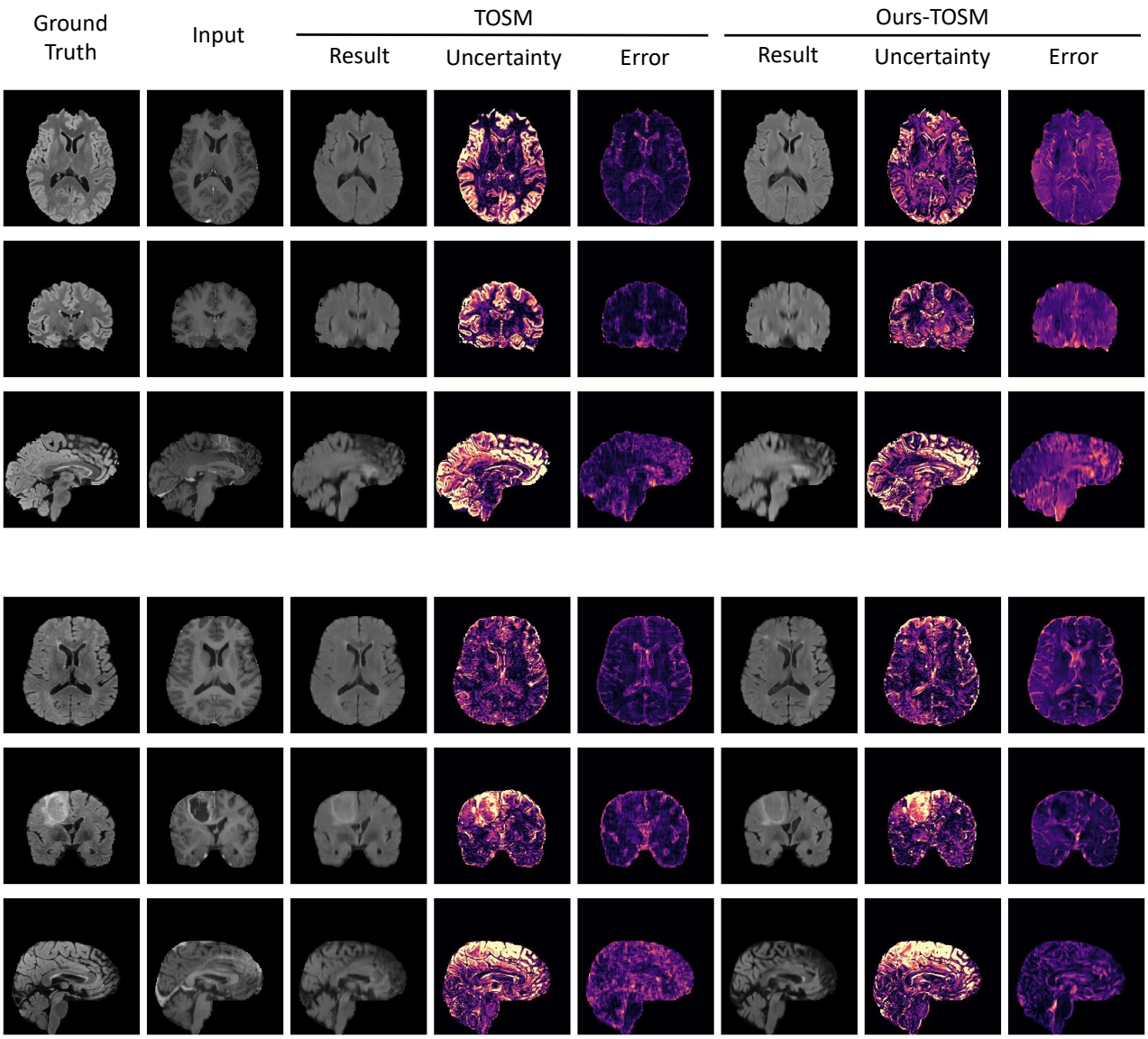

Figure 16: Uncertainty awareness results on modality translation task

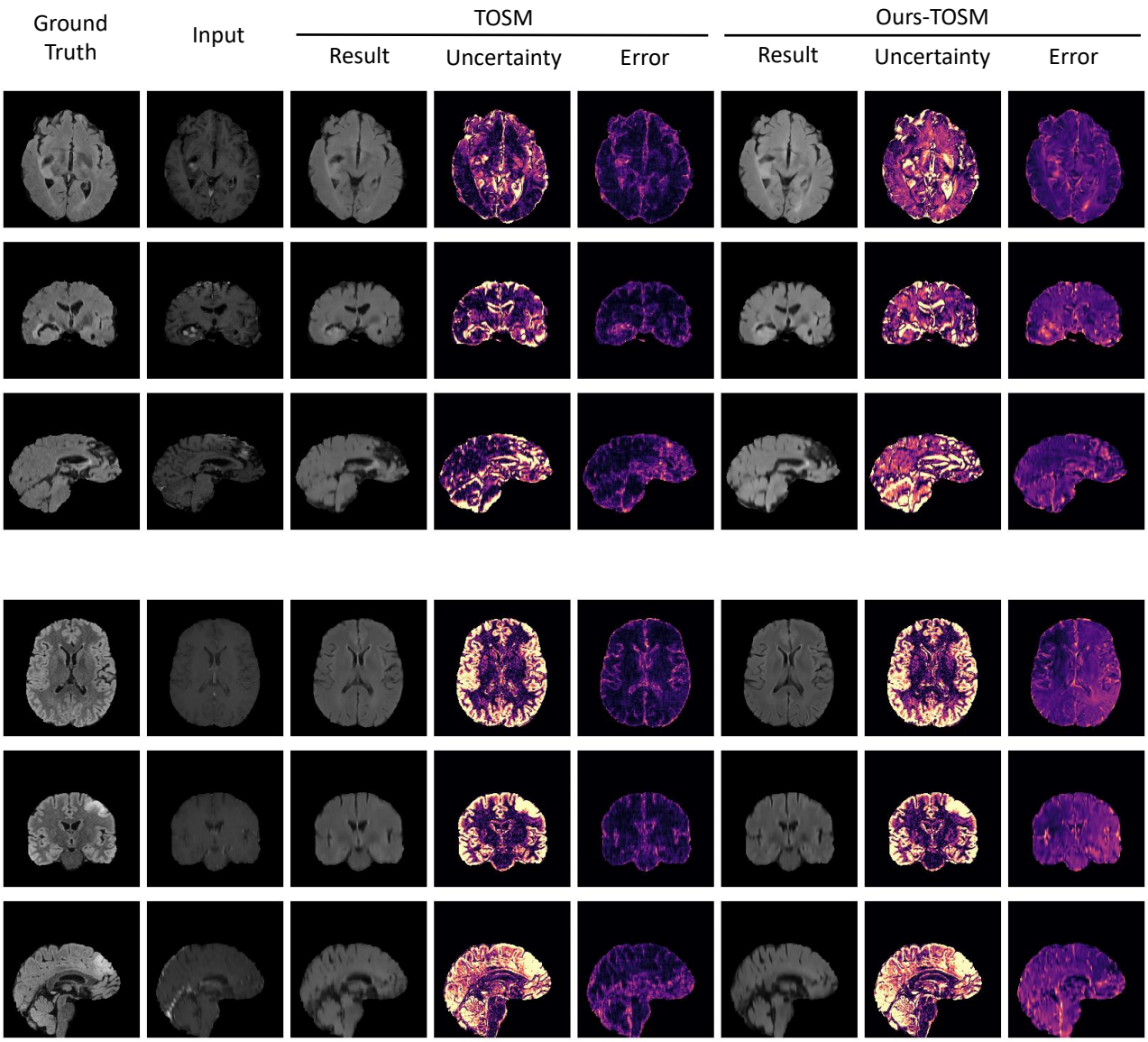

Figure 17: Uncertainty awareness results on modality translation task

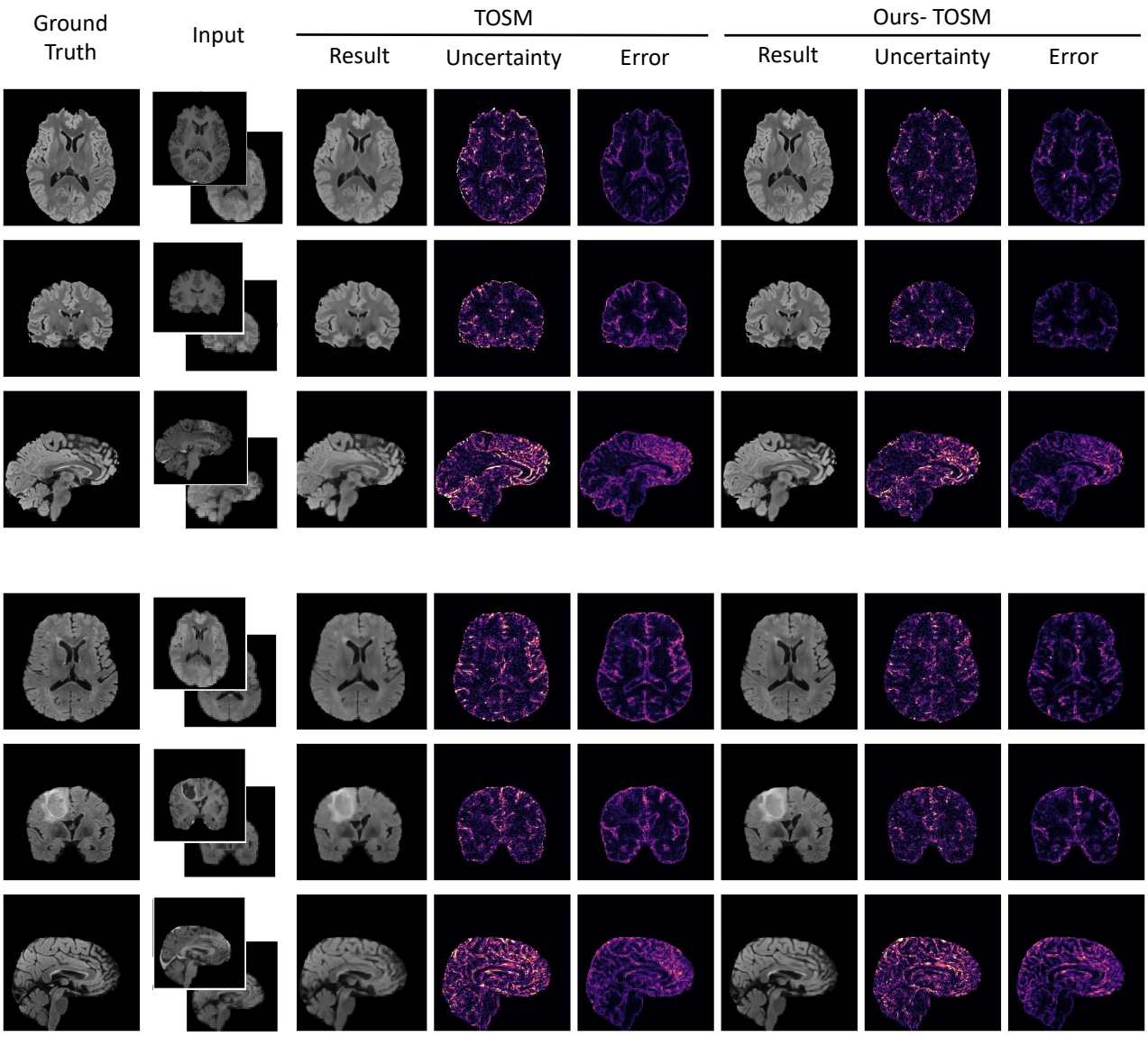

Figure 18: Uncertainty awareness results given both conditions

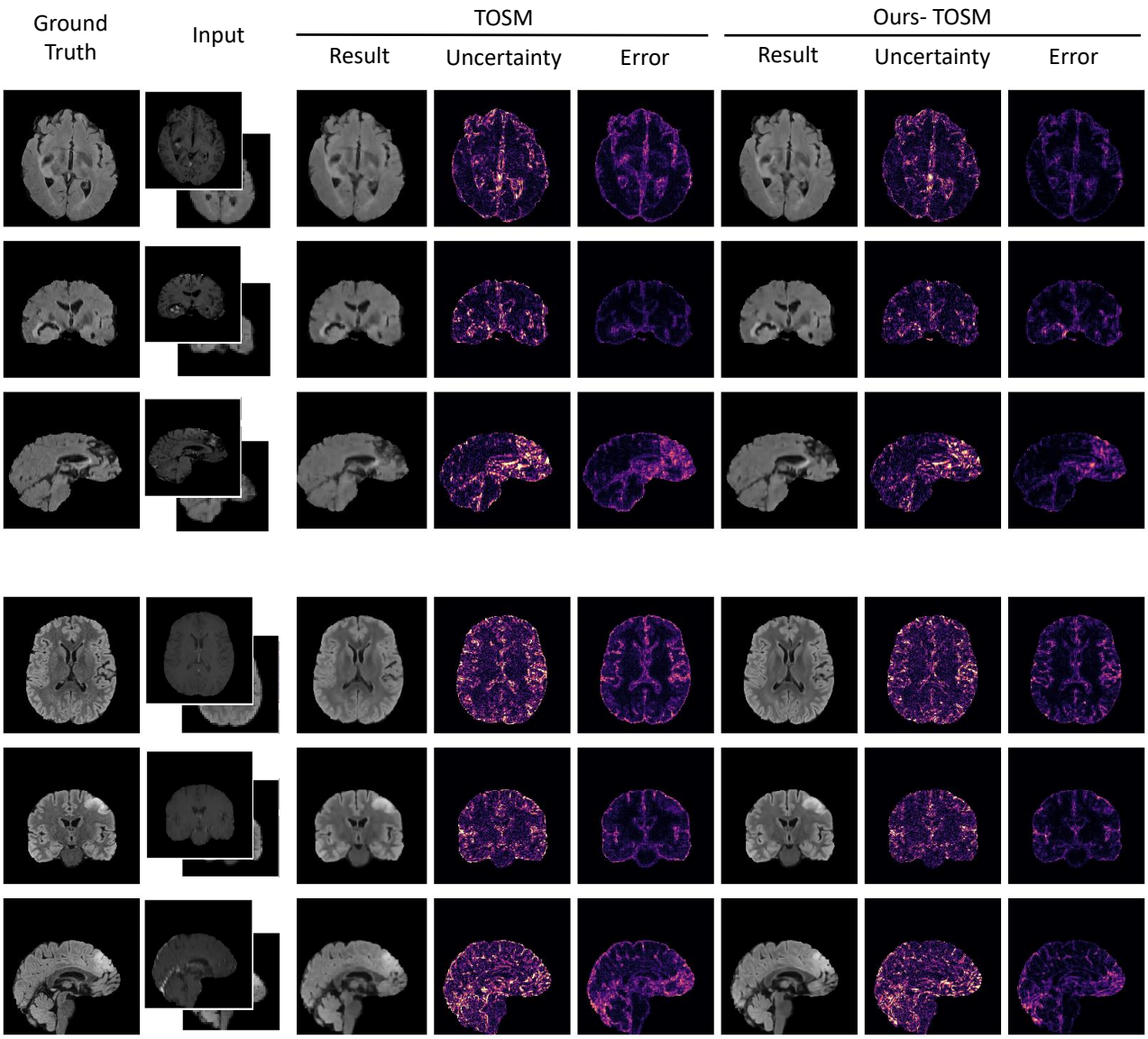

Figure 19: Uncertainty awareness results given both conditions

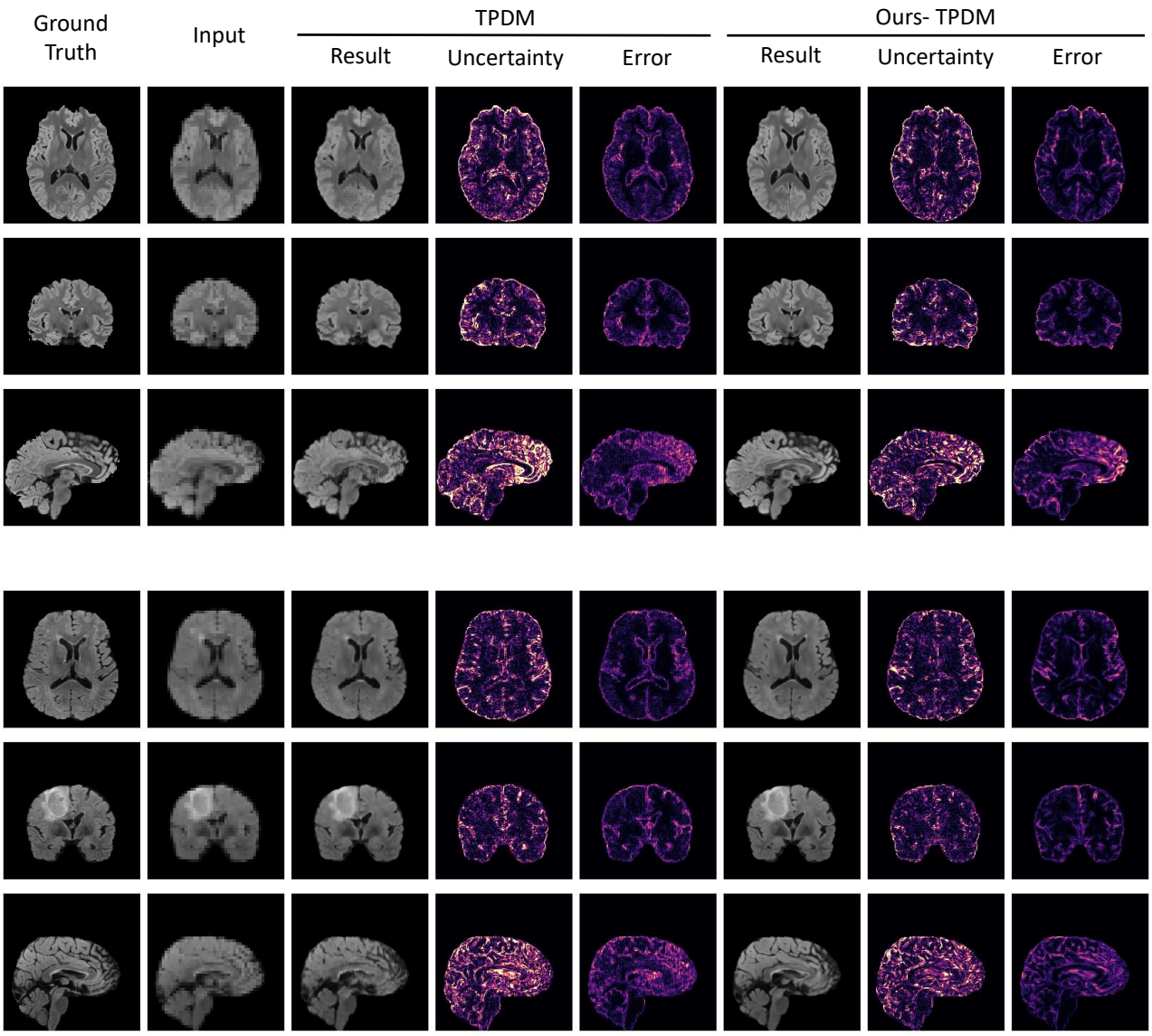

Figure 20: Uncertainty awareness results on super-resolution for TPDM and Ours-TPDM

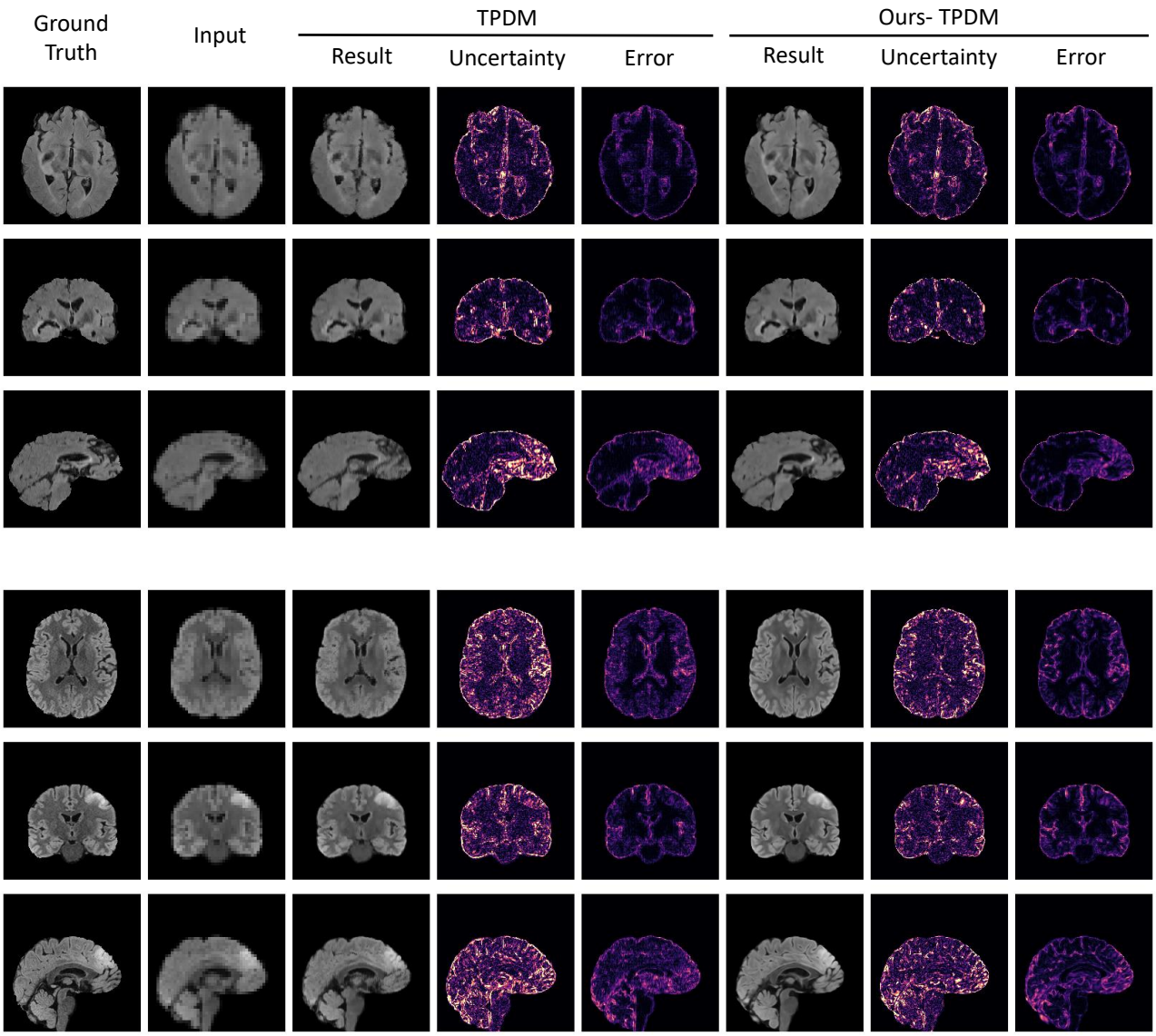

Figure 21: Uncertainty awareness results on super-resolution for TPDM and Ours-TPDM