GaLA-SVR: Feed-Forward 3D Gaussian Framework with Physics-Aware Latent Aggregation for Slice-to-Volume Reconstruction

Anonymous authors
Paper under double-blind review

Abstract

Reconstructing coherent 3D volumes from sparse, motion-corrupted MRI slices requires accurate motion estimation and faithful integration of incomplete observations. Existing learning-based SVR methods often reconstruct an intermediate volume assembled from motion-corrected slice intensities for subsequent refinement, rather than leveraging semantically rich slice features across stacks. This can limit their ability to exploit complementary observations from different views. In sparsely observed regions, refinement needs to hallucinate missing structures from incomplete volume intensity information. To address this, we propose **GaLA-SVR**, a feed-forward 3D **Ga**ussian framework with physics-aware **L**atent **A**ggregation for **S**lice-to-**V**olume **R**econstruction. GaLA-SVR bypasses existing intermediate volume reconstruction, with its core consisting of a physics-aware latent aggregation scheme and a dense patch-wise correspondence-based motion estimator. First, we propose a novel Point Spread Function (PSF)-aware feature aggregation scheme to learn a comprehensive 3D latent representation from 2D slice features, maintaining structural completeness even in sparsely observed or unobserved regions, which is then decoded into anisotropic 3D Gaussians for high-fidelity volumetric rendering. To achieve the precise spatial alignment essential for feature aggregation, we reformulate slice motion estimation as a dense patch-wise correspondence problem, facilitated by a novel Intra- and Inter-Slice Attention (IISA) block. Experiments across three MRI datasets demonstrate that GaLA-SVR achieves SOTA volumetric reconstruction quality with accurate slice motion estimation. Code will be released upon acceptance.

1 Introduction

Reconstructing a coherent 3D representation from sparse observations is a fundamental inverse problem in imaging. This challenge is particularly pronounced in Magnetic Resonance Imaging (MRI), where high-resolution 3D acquisition requires long scan times and is therefore highly susceptible to subject motion (Jiang et al., 2007). Rapid 2D acquisitions effectively freeze intra-slice motion, providing sparsely sampled stacks with high in-plane but low through-plane resolution. However, the residual rigid inter-slice motion still causes spatial misalignment across slices (Saleem, 2014). Recovering a geometrically consistent, high-resolution 3D volume from such observations, referred to as Slice-to-Volume Reconstruction (SVR), is therefore an important reconstruction problem across a broad range of brain MRI applications (Gholipour et al., 2010; Kainz et al., 2015).

Conventional SVR jointly solves slice-to-volume registration and super-resolution reconstruction through per-subject iterative optimization (Rousseau et al., 2006; Gholipour et al., 2010; Xu et al., 2023), which is computationally expensive and sensitive to initialization. Learning-based SVR methods accelerate this process and improve robustness by amortizing subject-specific inference. While some primarily target slice-to-volume registration while retaining iterative or model-based refinement for reconstruction (Hou et al., 2018; Xu et al., 2022), others employ separate networks for registration and reconstruction, requiring training multiple task-specific components (Young et al., 2024; Wu et al., 2025). However, all of these learning-based SVR methods perform volumetric reconstruction by first assembling motion-corrected slice intensities into an intermediate volume, utilizing model-based updates or a learned network for subsequent refinement. This early interme-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

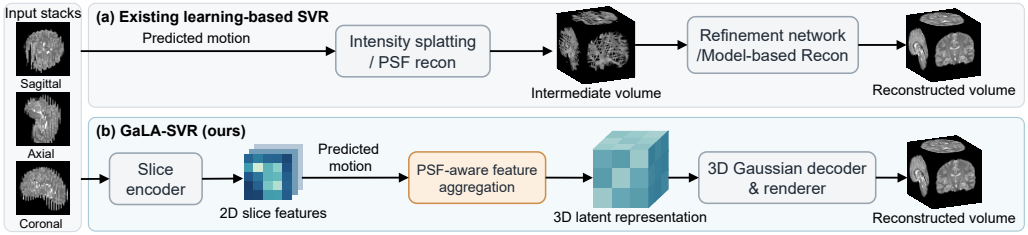

Figure 1: (a) Existing learning-based SVR methods assemble slice intensities into intermediate volumes that lack evidence in sparse regions. (b) GaLA-SVR actively aggregates semantically rich 2D features into a complete 3D latent representation before decoding. Input stacks are shown in through-plane views to illustrate inter-slice motion.

diate volume reconstruction discards the semantically rich slice features the network has already extracted. It not only obscures differences between complementary cross-stack observations, but also becomes even more detrimental in sparsely observed or unobserved regions due to insufficient slice coverage. The downstream refinement process is thus forced to hallucinate missing structures based solely on intermediate volumes, as shown in Figure 1. Consequently, this representational bottleneck severely degrades reconstruction quality. Moreover, these methods typically regress global rigid parameters or sparse anchor points and therefore provide overly sparse supervision that may hamper the precise spatial alignment essential for robust reconstruction.

To address these limitations, we propose GaLA-SVR, a feed-forward 3D Gaussian framework that learns a comprehensive and semantically rich 3D latent representation by aggregating contextual slice features. Specifically, we propose a novel Point Spread Function (PSF)-aware feature aggregation scheme that actively queries and fuses 2D features into a comprehensive 3D latent representation, adhering to slice acquisition physics. To achieve the precise spatial alignment required for feature aggregation, we depart from sparsely supervised motion regression, instead reformulating slice motion estimation as a dense patch-wise correspondence problem. This provides rich spatial supervision for correspondence learning facilitated by a novel Intra- and Inter-Slice Attention (IISA) block and an analytical solver for rigid transformation parameters. Finally, this comprehensive latent representation is decoded into anisotropic 3D Gaussians for high-fidelity volumetric rendering.

Our contributions are summarized as follows: (1) We propose GaLA-SVR. To the best of our knowledge, it is the first feed-forward 3D Gaussian framework for the SVR problem. (2) We propose a novel PSF-aware feature aggregation scheme to obtain a semantically rich and comprehensive 3D latent representation by fully leveraging rich context within sparsely acquired slice features. (3) We reformulate slice motion estimation by coupling dense patch-wise correspondences with an analytical Planar Kabsch solver, where a novel IISA block provides rich spatial supervision to achieve accurate and robust alignment. (4) Extensive experiments across fetal (FeTA), neonatal (dHCP), and adult (CamCAN) MRI datasets demonstrate that GaLA-SVR achieves state-of-the-art in volumetric reconstruction quality while delivering highly accurate motion estimation.

2 Related Works

Slice-to-Volume Reconstruction. Conventional SVR alternates between registration and super-resolution reconstruction, requiring computationally expensive subject-specific optimization (Rousseau et al., 2006; Gholipour et al., 2010). Learning-based approaches alleviate this burden by replacing parts of the optimization with learned predictors. SVRnet (Hou et al., 2018) regresses canonical anchor points from individual MRI slices, whereas PlaneInVol (Yeung et al., 2021) uses attention-based pairwise comparisons to localize ultrasound planes. Both primarily target slice motion prediction rather than learned volumetric reconstruction. SVoRT (Xu et al., 2022) incorporates volume estimation into Transformer-based registration, but retains alternating slice motion updates and model-based reconstruction using a conjugate-gradient solver. FC-SVR (Young et al., 2024) and SVRMamba (Wu et al., 2025) instead combine splat-slice UNet and Mamba-based motion predictors, respectively, with CNN-based interpolation of motion-corrected intensity splatting volumes. However, this early intensity fusion obscures complementary cross-stack information, forcing refinement networks to infer missing structures solely from intermediate volume intensities. Recently,

several works (Wang et al., 2026b; Dannecker et al., 2026) have integrated 3D Gaussian representations (Kerbl et al., 2023) into SVR to leverage their spatial locality and continuity. However, they remain constrained to per-subject iterative optimization, incurring substantial computational overhead at inference.

Feed-forward 3D Gaussian Splatting. Feed-forward 3D Gaussian Splatting (3DGS) replaces per-scene optimization with the learned prediction of Gaussian primitives for fast learned inference. In natural scene reconstruction, pixelSplat (Charatan et al., 2024), GS-LRM (Zhang et al., 2024), and GRM (Xu et al., 2024) predict pixel-aligned Gaussians. This tightly couples primitive allocation to input sampling, potentially introducing redundant primitives across overlapping views. TokenGS (Ren et al., 2026) decouples Gaussian prediction from pixels via learnable tokens that attend to multi-view image features. However, without an explicit 3D grid, local spatial relationships must be learned implicitly rather than being structurally encoded through voxel neighborhoods. AnySplat (Jiang et al., 2025) jointly predicts camera parameters and pixel-wise Gaussians, then refines the latter through voxel-wise attribute fusion. Its voxel-wise aggregation therefore operates on predicted Gaussian parameters rather than 3D features/voxels. In contrast, VolSplat (Wang et al., 2025b) employs a forward-lifting approach, unprojecting 2D image features into a 3D grid for refinement before Gaussian decoding. However, because this strategy depends strictly on input view coverage, 3D cells failing to receive direct projections are left empty when inputs are sparse. While MedGSSR (Wang et al., 2026a) and MedGMAE (Fu et al., 2026) employ feed-forward Gaussians for 3D medical tasks, they assume pre-existing, motion-free 3D inputs. Consequently, they cannot handle unknown slice motion or reconstruct directly from sparse, unaligned 2D slices.

3 METHOD

Given N acquired stacks of slices $\boldsymbol{y} = \{y_i\}_{i=1}^N$, our goal is to jointly reconstruct the unknown high-resolution canonical volume V and estimate the corresponding slice-wise rigid transformations $\{\boldsymbol{T}_i\}_{i=1}^N$ of subject movement during scan, i.e., rotations and translations. Our GaLA-SVR $\mathcal{F}_\theta$, parameterized by θ, maps the input slices directly to reconstructed volume $\hat{V}$ and estimated transformations $\{\hat{\boldsymbol{T}}_i\}_{i=1}^N$:

$$\left(\hat{V}, \{\hat{\boldsymbol{T}}_i\}_{i=1}^N\right) = \mathcal{F}_\theta\left(\{y_i\}_{i=1}^N\right). \tag{1}$$

The overall framework of GaLA-SVR is illustrated in Figure 2 (a). First, a shared CNN slice encoder extracts feature maps $\{F_i\}_{i=1}^N$ from each input slice. As detailed in Section 3.2, the motion estimator tokenizes these features and predicts their canonical 3D coordinates using our proposed IISA blocks to analytically resolve the rigid transformation parameters. With the estimated motions, our proposed PSF-aware feature aggregation scheme actively queries and fuses 2D slice features for each 3D latent candidate cell via through-plane PSF weights (Section 3.3) to form a comprehensive and semantically rich 3D latent representation. For efficient decoding, we employ M sparse Transformer blocks (Xiang et al., 2025) that operate only on active cells rather than the entire grid. Each active cell is then decoded into J anisotropic 3D Gaussians, which are finally processed by a differentiable renderer to obtain the reconstructed 3D volume.

3.1 PRELIMINARIES

3D Gaussian Primitives and Volumetric Rendering. We parameterize each 3D Gaussian primitive G_j into its center $\boldsymbol{\mu}_j \in \mathbb{R}^3$, covariance matrix $\boldsymbol{\Sigma}_j \in \mathbb{R}^{3\times3}$, and a scalar intensity value I_j at its center. The covariance is decomposed as $\boldsymbol{\Sigma}_j = \boldsymbol{R}_j \boldsymbol{S}_j^2 \boldsymbol{R}_j^T$, where $\boldsymbol{S}_j = \mathrm{diag}(\boldsymbol{s}_j)$ is parameterized by a scale vector $\boldsymbol{s}_j \in \mathbb{R}^3$ and $\boldsymbol{R}_j$ by a quaternion $\phi_j \in \mathbb{R}^4$. Thus, each Gaussian is represented by $g_j = \{\boldsymbol{\mu}_j, \boldsymbol{s}_j, \phi_j, I_j\} \in \mathbb{R}^{11}$, whose contribution at a spatial point $\boldsymbol{x}$ is given by:

$$G_j(\boldsymbol{x}|g_j) = I_j \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_j)^T \boldsymbol{\Sigma}_j^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_j)\right). \tag{2}$$

For volumetric rendering, we define an intensity field $\mathcal{I}(\boldsymbol{x})$ over continuous spatial coordinates by locally aggregating contributions from neighboring Gaussians:

$$\mathcal{I}(\boldsymbol{x}) = \sum_{j:\, \|\boldsymbol{x}-\boldsymbol{\mu}_j\|_2 \le 3\sigma_j} G_j(\boldsymbol{x}|g_j), \tag{3}$$

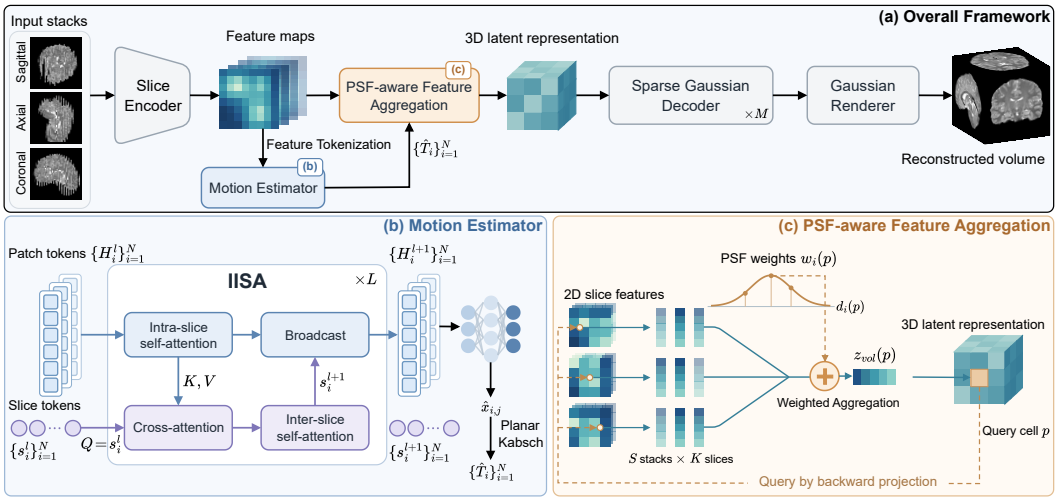

Figure 2: Overview of GaLA-SVR. Input stacks are shown in through-plane views to illustrate inter-slice motion. (a) GaLA-SVR aggregates contextual slice features into a 3D latent representation and decodes it into anisotropic 3D Gaussians for feed-forward volume reconstruction. (b) Motion estimation is formulated as dense patch-wise correspondence learning, using IISA blocks and an analytical Planar Kabsch solver for spatial alignment. (c) PSF-aware aggregation actively queries and fuses slice features at each active 3D cell according to slice acquisition physics, preserving rich structural context across sparse observations before Gaussian decoding.

where $3\sigma_j$ ($\sigma_j = \max(\boldsymbol{s}_j)$) is a spatial cutoff radius, truncating the negligible contributions of Gaussians. The discretized V is obtained by evaluating the intensity field $\mathcal{I}(\boldsymbol{x})$ at the voxel centers.

Forward Slice Acquisition Model. To mitigate motion artifacts in prolonged 3D scans, rapid 2D acquisitions are employed to freeze intra-slice motion, but at the cost of coarser through-plane resolution and inter-slice motion. To reconstruct the unknown discretized canonical volume V, multiple stacks of 2D slices $\{y_i\}_{i=1}^N$ are acquired from different orientations to provide complementary information. This forward acquisition process follows a linear model (Gholipour et al., 2010):

$$y_i = \boldsymbol{D}\boldsymbol{B}\boldsymbol{T}_i V + \epsilon_i, \tag{4}$$

where $\boldsymbol{T}_i$ denotes the rigid transformation mapping V into the local coordinate frame of slice i, $\boldsymbol{D}$ and $\boldsymbol{B}$ represent the slice sampling and Point Spread Function (PSF) blurring operators, respectively, and ϵ_i is the measurement noise. The discrete blurring operator $\boldsymbol{B}$ approximates convolution with a continuous PSF kernel $\mathcal{H}$. Following Kuklisova-Murgasova et al. (2012), $\mathcal{H}$ is modeled as a slice-aligned anisotropic Gaussian that factorizes into in-plane and through-plane components:

$$\mathcal{H}(\boldsymbol{\xi}, \zeta) = \mathcal{H}_{xy}(\boldsymbol{\xi})\, \mathcal{H}_z(\zeta), \tag{5}$$

where $\boldsymbol{\xi} \in \mathbb{R}^2$ and $\zeta \in \mathbb{R}$ denote the in-plane and through-plane offsets, respectively. The normalized 2D Gaussian kernel $\mathcal{H}_{xy}$ describes in-plane blurring, and the slice profile $\mathcal{H}_z$ describes signal integration along the slice normal. Setting the full width at half maximum (FWHM) of $\mathcal{H}_z$ to the nominal slice thickness τ gives (Kuklisova-Murgasova et al., 2012):

$$\mathcal{H}_z(\zeta) = \frac{1}{\sqrt{2\pi}\sigma_z} \exp\left(-\frac{\zeta^2}{2\sigma_z^2}\right), \qquad \sigma_z = \frac{\tau}{2\sqrt{2\ln 2}}. \tag{6}$$

3.2 Motion Estimation via Patch-wise Correspondences

Existing learning-based SVR methods typically estimate slice motion by regressing rigid transformation parameters or three canonical anchor points (Xu et al., 2022; Wu et al., 2025). While mathematically sufficient to define a plane, these compact targets provide only sparse supervision for spatial feature learning that may hamper precise spatial alignment. To overcome this, we reformulate slice motion estimation as a dense patch-wise canonical correspondence task, directly predicting the 3D canonical coordinates for a grid of local patch features.

Feature Tokenization. Let $\{F_i\}_{i=1}^{N}$ denote the set of feature maps extracted from the N input slices by the shared CNN encoder. For each slice i, we averagely pool its feature map into an $n \times n$ grid of patch features, denoted as $\{f_{i,j}\}_{j=1}^{n^2}$. Each patch feature is then processed by a projection function ϕ and augmented with a set of learned embeddings to explicitly encode its spatial and acquisition context. We inject embeddings for the 2D in-plane coordinates (e_j^{2D}), the stack identity (e_i^{stk}), and the relative through-plane slice position (e_i^{slc}), forming the initial patch tokens $H_i^0 = \{h_{i,j}^0\}_{j=1}^{n^2}$,

$$h_{i,j}^0 = \phi(f_{i,j}) + e_j^{2D} + e_i^{stk} + e_i^{slc}, \tag{7}$$

Details of the feature projection ϕ and positional encodings are provided in Appendix A.1.

Intra- and Inter-Slice Attention Block. To promote effective information exchange among patch tokens across different slices, we propose an intra- and inter-slice attention block (IISA) in an alternating fashion inspired by (Wang et al., 2025a). As shown in Figure 2 (b), at the l-th IISA block, we first apply intra-slice self-attention (Vaswani et al., 2017) independently to each slice, allowing patch tokens $h_{i,j}^l$ to capture local spatial dependencies. Unlike (Wang et al., 2025a), which performs global self-attention over patch tokens from all input views, we assign each slice a learnable slice token s_i^l, which cross-attends to its corresponding patch tokens to aggregate slice-level context. Inter-slice self-attention is then performed over these slice tokens to exchange information across slices. This rationale is twofold: 1) Operating on a single slice token per slice substantially reduces computational overhead compared to global patch-level attention. 2) Slice-level representations provide a more suitable inductive bias for estimating rigid transformations, which are inherently global geometric quantities. Finally, the updated slice token s_i^{l+1} is broadcast back to every patch token within the same slice via linear projection and addition to form $h_{i,j}^{l+1}$.

Canonical Coordinate Prediction. After the L IISA blocks, we employ a shared MLP coordinate head $g(\cdot)$ to directly regress the 3D canonical coordinates $\hat{\boldsymbol{x}}_{i,j}$ from final patch tokens $h_{i,j}^L$. Based on these predictions, we analytically solve the slice transformation $\hat{\boldsymbol{T}}_i = (\hat{\boldsymbol{R}}_i, \hat{\boldsymbol{c}}_i)$, with rotation $\hat{\boldsymbol{R}}_i$ and translation $\hat{\boldsymbol{c}}_i$, using a parameter-free weighted adaptation of the Kabsch algorithm for coplanar correspondences (Kabsch, 1976). Details are provided in Appendix A.3.

3.3 PSF-Aware Feature Aggregation

Given the estimated motion, we construct a latent volumetric representation by gathering 2D slice features at locations defined on a canonical $h \times w \times d$ grid. To ensure complete 3D feature coverage and mitigate the empty-cell issue in forward-lifting (Wang et al., 2025b) due to sparse input, we propose a PSF-aware feature aggregation scheme that actively queries 2D features for each candidate cell, fusing them in accordance with slice acquisition physics, as shown in Figure 2 (c).

Query by Backward Projection. Specifically, we construct a fixed atlas to define an approximate brain region within the latent grid, and the cells within the brain region form our dense 3D candidate query cell set $\mathcal{V}$. For each candidate cell $\boldsymbol{p} \in \mathcal{V}$, we query slice observations by back-projecting its canonical 3D coordinates into the local frame of slice i using the estimated transformations:

$$[u_i(\boldsymbol{p}) \quad v_i(\boldsymbol{p}) \quad d_i(\boldsymbol{p})]^T = \hat{\boldsymbol{R}}_i^T(\boldsymbol{p} - \hat{\boldsymbol{c}}_i). \tag{8}$$

where $\hat{\boldsymbol{R}}_i$ and $\hat{\boldsymbol{c}}_i$ denote the estimated rotation and translation parameters for slice i in canonical coordinates, $\boldsymbol{q}_i(\boldsymbol{p}) = (u_i(\boldsymbol{p}), v_i(\boldsymbol{p}))^T$ is its in-plane projection coordinate, and $d_i(\boldsymbol{p})$ is its signed distance to the slice plane. The details of atlas construction are provided in Appendix B.3.

PSF-aware Feature Aggregation. We derive the through-plane weighting of queried slice features from the forward acquisition model (Eq. 4). The forward operator specifies how volume points contribute to slice measurements. Its adjoint accumulates slice signals at volume points using the same coupling coefficients. For $\mathcal{A}_i = \boldsymbol{D}\boldsymbol{B}\boldsymbol{T}_i$, the adjoint is

$$\mathcal{A}_i^* = \boldsymbol{T}_i^*\boldsymbol{B}^*\boldsymbol{D}^* = \boldsymbol{T}_i^*\boldsymbol{B}\boldsymbol{D}^*, \tag{9}$$

where $\boldsymbol{B}^* = \boldsymbol{B}$ follows from the symmetric Gaussian convolution model. In the continuous slice-plane interpretation, the separable PSF in Eq. 5 allows the pointwise acquisition adjoint to factor into a through-plane weight $\mathcal{H}_z(d_i(\boldsymbol{p}))$ and an in-plane filtering term (Appendix A.4). We retain this through-plane factor to weight the queried slice features. Substituting $d_i(\boldsymbol{p})$ into Eq. 6 yields:

$$w_i(\boldsymbol{p}) = \frac{1}{\sqrt{2\pi}\sigma_z}\chi_i(\boldsymbol{p}) \exp\left(-\frac{d_i(\boldsymbol{p})^2}{2\sigma_z^2}\right), \tag{10}$$

where $\chi_i(\boldsymbol{p})$ indicates whether the projection lies within the field of view of slice i. For each query point $\boldsymbol{p}$, we select up to K valid slices from each stack with the highest PSF weights and denote the resulting set of selected slices by $\mathcal{N}_K(\boldsymbol{p})$, where $|\mathcal{N}_K(\boldsymbol{p})| \leq SK$ and S is the number of input stacks. We then bilinearly sample the feature map F_i of each selected slice at $\boldsymbol{q}_i(\boldsymbol{p})$, yielding the feature $\widetilde{f}_i(\boldsymbol{p})$. The PSF-aware aggregated feature at $\boldsymbol{p}$ is computed as

$$z_{\text{vol}}(\boldsymbol{p}) = \frac{\sum_{i\in\mathcal{N}_K(\boldsymbol{p})} w_i(\boldsymbol{p})\, \widetilde{f}_i(\boldsymbol{p})}{\sum_{i\in\mathcal{N}_K(\boldsymbol{p})} w_i(\boldsymbol{p})}. \tag{11}$$

Evaluating Eq. 11 across all candidate cells $\boldsymbol{p} \in \mathcal{V}$ constructs a comprehensive 3D latent representation of the underlying volume for the subsequent 3D Gaussian decoding.

3.4 Training Objectives

The entire framework jointly estimates slice motions and reconstructs the volume in a single forward pass, without constructing any intermediate intensity volume. The training objective of the slice motion estimation can be formulated as:

$$\mathcal{L}_{\text{motion}} = \frac{1}{NP}\sum_{i=1}^{N}\sum_{j=1}^{P}\|\hat{\boldsymbol{x}}_{i,j}-\boldsymbol{x}_{i,j}\|_1+\frac{1}{N}\sum_{i=1}^{N}\left[\lambda_1 \arccos\left(\frac{\text{tr}\left(\hat{\boldsymbol{R}}_i^T\boldsymbol{R}_i\right)-1}{2}\right)+\lambda_2\|\hat{\boldsymbol{c}}_i-\boldsymbol{c}_i\|_1\right], \tag{12}$$

where $P = n^2$. The first term computes the mean L1 coordinate error between the predicted and ground-truth 3D canonical patch coordinates. The second term measures the geodesic rotation loss on $\text{SO}(3)$ (Salehi et al., 2018), where $\text{tr}(\cdot)$ denotes the matrix trace. The final term penalizes the L1 translation error between the estimated and ground-truth slice centers. Moreover, the volumetric reconstruction training objective can be summarized as,

$$\mathcal{L}_{volume} = \lambda_3\|\hat{V}-V\|_1 + \lambda_4 D\text{-SSIM}(\hat{V}, V) \tag{13}$$

where the two terms denote the L1 and D-SSIM loss between the predicted volume $\hat{V}$ and the ground-truth volume V, respectively. Therefore, the overall training objective of GaLA-SVR can be formulated as follows:

$$\mathcal{L}_{overall} = \mathcal{L}_{motion} + \mathcal{L}_{volume}. \tag{14}$$

$\lambda_1, \lambda_2, \lambda_3$, and λ_4 are hyperparameters that balance the respective loss components.The training of GaLA-SVR is performed in an end-to-end manner without any iterative refinement.

4 Experiments

4.1 Experiments Settings

Datasets and Evaluation Metrics We conduct extensive experiments on three T2-weighted (T2w) brain MRI datasets covering distinct developmental stages: fetal brains from the Fetal Tissue Annotation Challenge (FeTA) dataset (Payette et al., 2021), neonatal brains from the developing Human Connectome Project (dHCP) dataset (Hughes et al., 2017), and adult brains from the Cambridge Centre for Ageing and Neuroscience (CamCAN) dataset (Taylor et al., 2017; Shafto et al., 2014). For all datasets, we simulate three image stacks per subject with orthogonal acquisition orientations, where each stack contains 15–30 slices with a size of 128×128 following (Xu et al., 2022) (detailed slice simulation is provided in Appendix B.2). We evaluate reconstruction quality using Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index Measure (SSIM), and Normalized Root Mean Squared Error (NRMSE). Motion estimation accuracy is evaluated using: Anchor-point Euclidean distance (ED) measures the mean distance between three corresponding anchor points under the estimation and ground-truth, and Geodesic Distance (GD) measures the angular discrepancy between the estimated and ground-truth rotation matrices. We also report inference runtime (Inf) to compare computational efficiency across all methods. Dataset details are presented in Appendix B.1.

Comparison Methods. We compare GaLA-SVR with an optimization-based method, SVRTK (Uus et al., 2020), and with five learning-based SVR approaches: SVRnet (Hou et al., 2018), PlaneInVol (Yeung et al., 2021), SVoRT (Xu et al., 2022), FC-SVR (Young et al., 2024), and SVR-Mamba (Wu et al., 2025). All learning-based baselines are trained separately on the three datasets

Table 1: Quantitative comparison of reconstruction performance on FeTA, CamCAN, and dHCP. Values are reported as mean ± standard deviation. Best and second-best results are shown in **bold** and underlined, respectively.

Method	FeTA			CamCAN			dHCP		
	PSNR↑	SSIM↑	NRMSE↓	PSNR↑	SSIM↑	NRMSE↓	PSNR↑	SSIM↑	NRMSE↓
Optimization-based									
SVRTK	27.22 ±3.78	0.870 ±0.030	0.270 ±0.105	22.94 ±2.32	0.806 ±0.063	0.286 ±0.084	23.51 ±2.99	0.803 ±0.051	0.406 ±0.139
Learning-based									
SVRnet	24.06 ±1.71	0.816 ±0.021	0.367 ±0.061	21.90 ±0.96	0.716 ±0.025	0.311 ±0.030	24.45 ±0.91	0.806 ±0.013	0.346 ±0.036
PlaneInVol	25.10 ±1.77	0.836 ±0.020	0.324 ±0.037	22.90 ±1.09	0.750 ±0.028	0.278 ±0.031	24.65 ±0.92	0.810 ±0.014	0.338 ±0.036
SVoRT	23.58 ±1.80	0.780 ±0.035	0.387 ±0.051	25.28 ±1.20	0.838 ±0.017	0.198 ±0.023	26.10 ±1.26	0.834 ±0.020	0.288 ±0.044
FC-SVR	26.62 ±2.41	0.862 ±0.018	0.278 ±0.041	23.02 ±1.19	0.797 ±0.027	0.276 ±0.037	24.27 ±0.92	0.820 ±0.012	0.356 ±0.039
SVRMamba	25.87 ±1.80	0.856 ±0.018	0.298 ±0.049	24.16 ±1.33	0.808 ±0.027	0.241 ±0.033	26.93 ±1.09	0.854 ±0.014	0.261 ±0.034
GaLA-SVR	**29.11** ±2.08	**0.881** ±0.011	**0.198** ±0.021	**27.18** ±1.21	**0.873** ±0.015	**0.171** ±0.023	**29.47** ±0.87	**0.883** ±0.007	**0.194** ±0.022

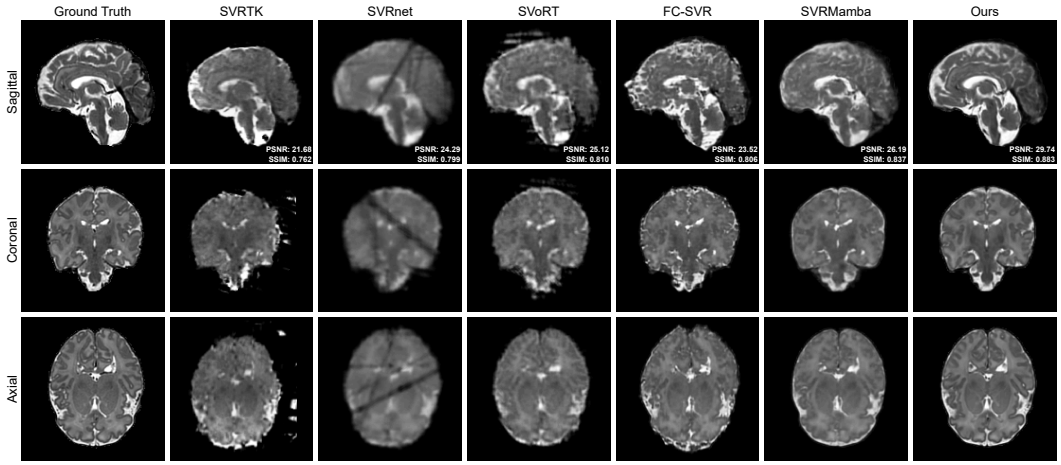

Figure 3: Qualitative results on the dHCP dataset. The figure displays the reconstructed middle slices in the sagittal, coronal, and axial planes of a single subject from the test set. The left corner of the sagittal view shows the volume PSNR and SSIM of the current predicted volume.

using the same splits and simulated slices. The description and implementation details of the baseline methods are provided in Appendix B.6.

Implementation Details. The shared CNN-based slice encoder encodes each 128×128 slice into a 1024×32×32 feature map. The number of IISA blocks L is set to 2, with a hidden dimension of 256 and eight attention heads. The dimension of the hidden layer of the canonical coordinate head is 256. The spatial dimensions of the 3D latent grid are set to one-quarter of the reference volume. The Gaussian decoder consists of 12 sparse Transformer blocks. The detailed implementation for the slice encoder and sparse transformer is provided in Appendix B. We set the number of anisotropic Gaussians per active cell to 8. An 88-dimensional output head predicts eight 11-parameter Gaussians per active cell. λ_1, λ_2, λ_3, and λ_4 are set to 0.5, 0.2, 2.0, and 0.4, chosen through empirical tuning on a validation subset. Experiments are conducted on an NVIDIA A6000 Ada GPU using the AdamW optimizer (Loshchilov & Hutter, 2019) with a batch size of 1. Baseline learning rates follow their official default configurations. For GaLA-SVR, the learning rate is set to 1×10^{-5} for the slice encoder, and 1×10^{-4} for both the motion estimation module and the Gaussian decoder.

4.2 Comparison Studies

Evaluation on Reconstruction accuracy. Table 1 presents the quantitative reconstruction performance of GaLA-SVR alongside baseline methods across the FeTA, CamCAN, and dHCP datasets. We observe that GaLA-SVR achieves the best reconstruction accuracy across all evaluated datasets and metrics. On the FeTA dataset, GaLA-SVR achieves a PSNR of 29.11 dB and an SSIM of 0.881,

Table 2: Quantitative comparison of transformation errors and inference time on FeTA, CamCAN, and dHCP. ED and GD are reported as mean,±,standard deviation, while inference time is mean runtime per subject. Best and second-best transformation results are shown in **bold** and underlined.

Method	FeTA			CamCAN			dHCP		
	ED (mm)↓	GD (rad)↓	Inf. (s)↓	ED (mm)↓	GD (rad)↓	Inf. (s)↓	ED (mm)↓	GD (rad)↓	Inf. (s)↓
Optimization-based									
SVRTK	8.27 ±3.84	0.129 ±0.067	113.80	9.06 ±6.03	0.111 ±0.080	416.10	11.75 ±7.40	0.174 ±0.138	827.90
Learning-based									
SVRnet	15.63 ±5.25	0.221 ±0.111	0.03	9.77 ±3.82	0.097 ±0.050	0.09	10.75 ±4.70	0.156 ±0.101	0.06
PlaneInVol	14.33 ±4.82	0.203 ±0.105	0.03	7.38 ±3.14	0.071 ±0.035	0.10	9.45 ±4.74	0.138 ±0.102	0.07
SVoRT	11.68 ±4.30	0.182 ±0.088	0.17	**2.83** ±1.12	**0.037** ±0.014	0.71	5.28 ±4.36	0.092 ±0.091	0.47
FC-SVR	6.74 ±2.91	0.114 ±0.057	0.70	4.05 ±1.34	0.044 ±0.015	1.30	5.84 ±3.59	0.092 ±0.075	0.88
SVRMamba	10.80 ±3.94	0.197 ±0.075	0.05	5.46 ±1.65	0.076 ±0.023	0.14	5.80 ±4.02	0.105 ±0.081	0.12
GaLA-SVR	**6.10** ±2.78	**0.110** ±0.054	0.11	3.22 ±1.04	0.041 ±0.014	0.27	**3.79** ±3.62	**0.066** ±0.079	0.17

comfortably outperforming both the optimization-based SVRTK and the recent learning-based FC-SVR. This substantial performance advantage extends to the adult CamCAN and neonatal dHCP datasets. While SVoRT and SVRMamba serve as the strongest baselines on CamCAN and dHCP, respectively, GaLA-SVR significantly outperforms them by +1.90 dB and +2.54 dB in mean PSNR. Correspondingly, our method reduces the NRMSE by approximately 14% on CamCAN and 26% on dHCP compared to these two baselines. These substantial quantitative gains validate the superiority of our framework. Unlike baselines that rely on intermediate volume reconstruction, our PSF-aware 3D aggregation integrates cross-stack evidence directly in the latent space, forming a comprehensive and 3D latent representation of the volume. This can also be demonstrated from the qualitative results in Figure 3, where GaLA-SVR reconstructs more fine-grained details compared to the baseline methods. Moreover, compared to SVRTK and FC-SVR, which suffer from severe ghosting and intensity artifacts, GaLA-SVR reconstructs significantly more coherent and smooth canonical volumes, benefiting from the inherent continuity of 3D Gaussian representations. More qualitative results on all three datasets are provided in Appendix C.2.

Evaluation on Motion Estimation. Table 2 summarizes the quantitative slice-wise transformation errors across the three datasets. Overall, GaLA-SVR demonstrates exceptional robustness in slice motion estimation, achieving the lowest transformation errors on both the FeTA and dHCP datasets. Specifically, on the challenging FeTA dataset, our method yields an ED of 6.10 mm and a GD of 0.110 rad, distinctly outperforming the recent FC-SVR and the optimization-based SVRTK. Similarly, on the neonatal dHCP dataset, GaLA-SVR achieves the best motion estimation with an ED of 3.79 mm and a GD of 0.066 rad. On the adult CamCAN dataset, while SVoRT records marginally lower transformation errors, GaLA-SVR remains highly competitive and significantly surpasses all other baselines. Crucially, as previously shown in Table 1, our accurate motion predictions perfectly complement our PSF-aware feature aggregation module. This synergy enables GaLA-SVR to vastly outperform SVoRT in final reconstruction quality despite the minor deficit in standalone transformation metrics. This confirms that formulating motion estimation via dense patch-wise correspondences and an analytical Planar Kabsch solver provides highly reliable geometric priors.

Regarding inference efficiency shown in Table 2, all learning-based frameworks achieve massive speedups over the optimization-based SVRTK. While SVRnet and PlaneInVol are nominally the fastest (≤ 0.10 s), this is because they only estimate slice motion and rely on a basic PSF adjoint operator for reconstruction without any learned post-processing. Among complete learned pipelines, GaLA-SVR is only fractionally slower than SVRMamba, yet it drastically outperforms it in accuracy, nearly halving the motion errors on FeTA. Moreover, our method runs $2\times$ to $5\times$ faster than SVoRT and FC-SVR while delivering superior reconstruction and motion metrics, striking an ideal balance between efficiency and accuracy.

4.3 Ablation Studies

Ablation Study on PSF-aware Feature Aggregation. To verify the effectiveness of our proposed PSF-aware 3D feature aggregation, We ablated the proposed aggregation scheme with three al-

Table 3: Ablation studies on the dHCP dataset. (a) PSF-aware 3D feature aggregation. (b) IISA block and patch-wise feature correspondence. Values are reported as mean ± standard deviation.

(a) Feature aggregation				(b) Motion estimation		
Method	PSNR↑	SSIM↑	NRMSE↓	Method	ED(mm)↓	GD(rad)↓
Uniform-K	28.10 ±0.91	0.871 ±0.008	0.227 ±0.025	w/o intra-SA	3.99 ±3.52	0.070 ±0.078
Nearest-1	28.37 ±0.92	0.874 ±0.008	0.221 ±0.027	w/o inter-SA	5.33 ±3.85	0.096 ±0.081
Forward-lifting	27.79 ±1.03	0.868 ±0.012	0.234 ±0.028	Direct-regress	5.71 ±4.12	0.109 ±0.085
GaLA-SVR	**29.47** ±0.87	**0.883** ±0.007	**0.194** ±0.022	GaLA-SVR	**3.79** ±3.62	**0.066** ±0.079

ternatives: (1) *Uniform-K*, which averages the K valid slice features instead of using PSF-aware weighting. (2) *Nearest-1*, which uses only the nearest valid slice per stack of each query cell. (3) *Forward lifting*, which directly projects 2D features into 3D cells and averages them within each cell instead of querying 2D features from predefined 3D cells. Table 3 (a) presents the quantitative results of the three ablation experiments on reconstruction accuracy on the dHCP dataset. We can observe that employing either *Uniform-K* or *Nearest-1* leads to a clear drop in reconstruction accuracy. This demonstrates the superiority of our proposed PSF-aware 3D feature aggregation, confirming that adhering to the physics of slice acquisition is the optimal strategy for feature fusion. Furthermore, replacing our aggregation scheme with *Forward lifting* led to the most significant performance drop. This demonstrates the benefits of querying 2D features from predefined 3D cells. Conversely, forward-lifting projection inherently leaves a portion of grid cells within the brain region empty due to input sparsity.

Ablation Study on Motion Estimator. To verify the effectiveness of our proposed IISA block, we evaluated four ablation variants: (1) *w/o intra-slice attention*, which restricts each patch token to attending only to itself. (2) *w/o inter-slice attention*, which restricts each slice token to attending only to itself. (3) *Direct-regress*, which replaces the patch-wise feature correspondences by concatenating all patch tokens within a slice and employing an MLP to directly regress the rotation and translation parameters. Table 3 (b) presents the quantitative results of these three ablation experiments. *Direct-regress* shows the most severe performance degradation. This confirms that explicitly predicting dense patch-wise correspondences is fundamentally more robust than directly regressing rigid transformation parameters. Furthermore, *w/o inter-SA* causes a substantial drop in accuracy, underscoring the critical importance of cross-slice context for resolving spatial ambiguities and ensuring global geometric consistency. Lastly, *w/o intra-SA* leads to a more modest performance drop. Its inclusion strictly refines the spatial reasoning within individual slices. By synergistically combining these components, the full GaLA-SVR framework achieves the lowest overall transformation errors, validating the design of our proposed motion estimator. Further analysis on resolution on patch tokens is provided in Appendix C.1.

5 Conclusion

In conclusion, we propose GaLA-SVR, a novel feed-forward 3D Gaussian framework with physics-aware Latent aggregation for slice-to-volume reconstruction. By reformulating slice motion estimation as a dense patch-wise correspondence problem, our IISA-equipped estimator robustly and analytically recovers rigid slice transformations using a parameter-free Planar Kabsch solver. Furthermore, by adhering to the physics of slice acquisition, our PSF-aware feature aggregation actively queries and fuses 2D slice features into a comprehensive 3D latent representation. This comprehensive and semantically rich 3D latent representation is then decoded into anisotropic Gaussians using a sparse Transformer for high-fidelity volumetric rendering. Extensive experiments across three diverse datasets demonstrate that GaLA-SVR consistently outperforms existing state-of-the-art methods in both volumetric reconstruction accuracy and motion estimation robustness. By seamlessly bridging physical acquisition models with 3D Gaussian representations, GaLA-SVR offers a highly effective and unified solution for slice-to-volume reconstruction.

AI USE STATEMENT

In this work, we used generative AI tools, including ChatGPT and Codex, to assist with limited portions of the experimental code, including evaluation utilities and parts of the training and evaluation pipelines for baseline methods. We did not use generative AI tools to develop theoretical or conceptual frameworks, formulate or prove mathematical claims, propose research hypotheses, design the methodology or experiments, generate or process datasets, or interpret experimental results. Additionally, ChatGPT was used for literature summarization, brainstorming, manuscript structure and wording suggestions, and language editing. All AI-assisted code was manually reviewed and tested by at least two authors of the paper. Baseline implementations were verified against the corresponding published methods and, where available, official implementations. We take responsibility for the final content of this work, including any text, code, or other artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This research complies with the ICLR Code of Ethics. We use existing human MRI data from the publicly available FeTA, dHCP, and CamCAN datasets. The original data were collected under the relevant ethical approvals and informed-consent procedures of the respective studies. All datasets were accessed and used in accordance with their corresponding data-use and access conditions. All datasets are de-identified and contain no personally identifiable information. No direct human subject recruitment was involved.

REPRODUCIBILITY STATEMENT

We provide detailed descriptions of the datasets, preprocessing and slice simulation procedures, model architectures, hyperparameter settings, training procedures, and evaluation protocols in Sections 3 and 4 and Appendices A and B. Implementation and training details for the compared methods are also provided in Appendix B. Code and trained model weights will be released upon acceptance.

REFERENCES

David Charatan, Sizhe Lester Li, Andrea Tagliasacchi, and Vincent Sitzmann. pixelsplat: 3d gaussian splats from image pairs for scalable generalizable 3d reconstruction. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19457–19467. IEEE, 2024.

Maik Dannecker, Steven Jia, Nil Stolt-Ansó, Nadine Girard, Guillaume Auzias, François Rousseau, and Daniel Rueckert. Fast and explicit: Slice-to-volume reconstruction via 3d gaussian primitives with analytic point spread function modeling. In *Medical Imaging with Deep Learning*, 2026.

Xueming Fu, Fenghe Tang, Rongsheng Wang, Yingtai Li, Lixia Han, Jian Lu, Zihang Jiang, and S Kevin Zhou. Medgmae: Gaussian masked autoencoders for medical volumetric representation learning. In *The Fourteenth International Conference on Learning Representations*, 2026.

Ali Gholipour, Judy A Estroff, and Simon K Warfield. Robust super-resolution volume reconstruction from slice acquisitions: application to fetal brain mri. *IEEE transactions on medical imaging*, 29(10):1739–1758, 2010.

Benjamin Hou, Bishesh Khanal, Amir Alansary, Steven McDonagh, Alice Davidson, Mary Rutherford, Jo V Hajnal, Daniel Rueckert, Ben Glocker, and Bernhard Kainz. 3-d reconstruction in canonical co-ordinate space from arbitrarily oriented 2-d images. *IEEE transactions on medical imaging*, 37(8):1737–1750, 2018.

Emer J Hughes, Tobias Winchman, Francesco Padormo, Rui Teixeira, Julia Wurie, Maryanne Sharma, Matthew Fox, Jana Hutter, Lucilio Cordero-Grande, Anthony N Price, et al. A dedicated neonatal brain imaging system. *Magnetic resonance in medicine*, 78(2):794–804, 2017.

Lihan Jiang, Yucheng Mao, Linning Xu, Tao Lu, Kerui Ren, Yichen Jin, Xudong Xu, Mulin Yu, Jiangmiao Pang, Feng Zhao, et al. Anysplat: Feed-forward 3d gaussian splatting from unconstrained views. *ACM Transactions on Graphics (TOG)*, 44(6):1–16, 2025.

Shuzhou Jiang, Hui Xue, Alan Glover, Mary Rutherford, Daniel Rueckert, and Joseph V. Hajnal. MRI of moving subjects using multislice snapshot images with volume reconstruction (SVR): Application to fetal, neonatal, and adult brain studies. *IEEE Transactions on Medical Imaging*, 26(7):967–980, 2007. doi: 10.1109/TMI.2007.895456.

Wolfgang Kabsch. A solution for the best rotation to relate two sets of vectors. *Foundations of Crystallography*, 32(5):922–923, 1976.

Bernhard Kainz, Markus Steinberger, Wolfgang Wein, Maria Kuklisova-Murgasova, Christina Malamateniou, Kevin Keraudren, Thomas Torsney-Weir, Mary Rutherford, Paul Aljabar, Joseph V. Hajnal, and Daniel Rueckert. Fast volume reconstruction from motion corrupted stacks of 2D slices. *IEEE Transactions on Medical Imaging*, 34(9):1901–1913, 2015. doi: 10.1109/TMI.2015.2415453.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, George Drettakis, et al. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023.

Maria Kuklisova-Murgasova, Gerardine Quaghebeur, Mary A Rutherford, Joseph V Hajnal, and Julia A Schnabel. Reconstruction of fetal brain mri with intensity matching and complete outlier removal. *Medical image analysis*, 16(8):1550–1564, 2012.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

Kelly Payette, Priscille de Dumast, Hamza Kebiri, Ivan Ezhov, Johannes C Paetzold, Suprosanna Shit, Asim Iqbal, Romesa Khan, Raimund Kottke, Patrice Grehten, et al. An automatic multi-tissue human fetal brain segmentation benchmark using the fetal tissue annotation dataset. *Scientific data*, 8(1):167, 2021.

Jiawei Ren, Michal Jan Tyszkiewicz, Jiahui Huang, and Zan Gojcic. Tokengs: Decoupling 3d gaussian prediction from pixels with learnable tokens. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15365–15375, June 2026.

Francois Rousseau, Orit A Glenn, Bistra Iordanova, Claudia Rodriguez-Carranza, Daniel B Vigneron, James A Barkovich, and Colin Studholme. Registration-based approach for reconstruction of high-resolution in utero fetal mr brain images. *Academic radiology*, 13(9), 2006.

Sahar N Saleem. Fetal mri: An approach to practice: A review. *Journal of advanced research*, 5(5): 507–523, 2014.

Seyed Sadegh Mohseni Salehi, Seyed Raein Hashemi, Clemente Velasco-Annis, Abdelhakim Ouaalam, Judy A Estroff, Deniz Erdogmus, Simon K Warfield, and Ali Gholipour. Real-time automatic fetal brain extraction in fetal mri by deep learning. In *2018 IEEE 15th international symposium on biomedical imaging (ISBI 2018)*, pp. 720–724. IEEE, 2018.

Meredith A Shafto, Lorraine K Tyler, Marie Dixon, Jason R Taylor, James B Rowe, Rhodri Cusack, Andrew J Calder, William D Marslen-Wilson, John Duncan, Tim Dalgleish, et al. The cambridge centre for ageing and neuroscience (cam-can) study protocol: a cross-sectional, lifespan, multidisciplinary examination of healthy cognitive ageing. *BMC neurology*, 14(1):204, 2014.

Jason R Taylor, Nitin Williams, Rhodri Cusack, Tibor Auer, Meredith A Shafto, Marie Dixon, Lorraine K Tyler, Richard N Henson, et al. The cambridge centre for ageing and neuroscience (cam-can) data repository: Structural and functional mri, meg, and cognitive data from a cross-sectional adult lifespan sample. *neuroimage*, 144:262–269, 2017.

Alena Uus, Tong Zhang, Laurence H Jackson, Thomas A Roberts, Mary A Rutherford, Joseph V Hajnal, and Maria Deprez. Deformable slice-to-volume registration for motion correction of fetal body and placenta mri. *IEEE transactions on medical imaging*, 39(9):2750–2759, 2020.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Chengkai Wang, Luoyu Hong, Yiting Zhao, Jiamin Wang, Xiang Feng, Feiwei Qin, Zhenzhong Kuang, Xuefei Yin, Ali Bashashati, and Yanming Zhu. Medgssr: Generalizable medical image super-resolution 3d reconstruction via hierarchical feed-forward gaussian splatting. In *European Conference on Computer Vision*, pp. 526–543. Springer, 2026a.

Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5294–5306. IEEE, 2025a.

Weijie Wang, Yeqing Chen, Zeyu Zhang, Hengyu Liu, Haoxiao Wang, Zhiyuan Feng, Wenkang Qin, Feng Chen, Jia-Wang Bian, Zheng Zhu, et al. Volsplat: Rethinking feed-forward 3d gaussian splatting with voxel-aligned prediction. *arXiv preprint arXiv:2509.19297*, 2025b.

Yinsong Wang, Thomas Fletcher, Xinzhe Luo, Áine Travers Dineen, Rhodri Cusack, and Chen Qin. Self-supervised slice-to-volume reconstruction with gaussian representations for fetal mri. In *2026 IEEE 23rd International Symposium on Biomedical Imaging (ISBI)*, pp. 01–05. IEEE, 2026b.

Jiangjie Wu, Hongjiang Wei, and Yuyao Zhang. Svrmamba: Slice-to-volume reconstruction from multiple mri stacks with slice sequence guided mamba. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 8404–8412, 2025.

Jianfeng Xiang, Zelong Lv, Sicheng Xu, Yu Deng, Ruicheng Wang, Bowen Zhang, Dong Chen, Xin Tong, and Jiaolong Yang. Structured 3d latents for scalable and versatile 3d generation. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21469–21480. IEEE, 2025.

Junshen Xu, Daniel Moyer, P Ellen Grant, Polina Golland, Juan Eugenio Iglesias, and Elfar Adalsteinsson. Svort: Iterative transformer for slice-to-volume registration in fetal brain mri. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 3–13. Springer, 2022.

Junshen Xu, Daniel Moyer, Borjan Gagoski, Juan Eugenio Iglesias, P. Ellen Grant, Polina Golland, and Elfar Adalsteinsson. NeSVoR: Implicit neural representation for slice-to-volume reconstruction in MRI. *IEEE Transactions on Medical Imaging*, 42(6):1707–1719, 2023. doi: 10.1109/TMI.2023.3236216.

Yinghao Xu, Zifan Shi, Wang Yifan, Hansheng Chen, Ceyuan Yang, Sida Peng, Yujun Shen, and Gordon Wetzstein. Grm: Large gaussian reconstruction model for efficient 3d reconstruction and generation. In *European Conference on Computer Vision*, pp. 1–20. Springer, 2024.

Pak-Hei Yeung, Moska Aliasi, Aris T Papageorghiou, Monique Haak, Weidi Xie, and Ana IL Namburete. Learning to map 2d ultrasound images into 3d space with minimal human annotation. *Medical Image Analysis*, 70:101998, 2021.

Sean I. Young, Yael Balbastre, Bruce Fischl, Polina Golland, and Juan Eugenio Iglesias. Fully convolutional slice-to-volume reconstruction for single-stack MRI. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11535–11545, 2024.

Kai Zhang, Sai Bi, Hao Tan, Yuanbo Xiangli, Nanxuan Zhao, Kalyan Sunkavalli, and Zexiang Xu. Gs-lrm: Large reconstruction model for 3d gaussian splatting. In *European Conference on Computer Vision*, pp. 1–19. Springer, 2024.

APPENDICES

A DETAILED FORMULATION

A.1 FEATURE PROJECTION AND POSITIONAL ENCODING

Feature projection. Given the context feature map $F_i \in \mathbb{R}^{1024\times32\times32}$ extracted by the slice encoder in 3.2, we apply adaptive average pooling to each F_i, producing an 8×8 grid. The shared feature projection maps each flattened patch feature $f_{i,j} \in \mathbb{R}^{1024}$ to the token dimension $D = 256$:

$$\phi(f_{i,j}) = \mathrm{GELU}\left(W_\phi\,\mathrm{LN}(f_{i,j}) + b_\phi\right), \tag{15}$$

where $W_\phi \in \mathbb{R}^{D\times1024}$, $b_\phi \in \mathbb{R}^D$, and LN denotes layer normalization over feature channels.

In-plane positional encoding. Let (a_j, b_j) denote the zero-based column and row indices of patch j on the $n \times n$ grid. We normalize its center coordinates and apply a learned affine projection:

$$\mathbf{r}_j = \begin{bmatrix} 2(a_j + \frac{1}{2})/n - 1 \\ 2(b_j + \frac{1}{2})/n - 1 \end{bmatrix}, \qquad e_j^{2D} = W_{2D}\mathbf{r}_j + b_{2D}, \tag{16}$$

where $a_j, b_j \in \{0, \ldots, n-1\}$, $W_{2D} \in \mathbb{R}^{D\times2}$, and $b_{2D} \in \mathbb{R}^D$. This encoding describes the patch location within the input image and is shared across slices.

Stack identity encoding. Let $s_i \in \{1, \ldots, S\}$ identify the stack containing slice i, with $S = 3$. We obtain its embedding through a learned lookup table:

$$e_i^{stk} = E_{\mathrm{stk}}[s_i], \qquad E_{\mathrm{stk}} \in \mathbb{R}^{S\times D}. \tag{17}$$

Within-stack slice encoding. Let k_i be the zero-based index of slice i within its stack, which contains N_{s_i} slices. We normalize this index and map it to the token space:

$$\rho_i = \begin{cases} 2k_i/(N_{s_i} - 1) - 1, & N_{s_i} > 1, \\ 0, & N_{s_i} = 1, \end{cases} \qquad e_i^{slc} = w_{\mathrm{slc}}\rho_i + b_{\mathrm{slc}}, \tag{18}$$

where $w_{\mathrm{slc}}, b_{\mathrm{slc}} \in \mathbb{R}^D$ are learned parameters. This encoding represents relative slice ordering, not the physical position of the slice in canonical space. The stack and slice embeddings are broadcast to all patches of their corresponding slice. Adding the three embeddings to the projected features, as in Eq. 7, produces $H_i^0 \in \mathbb{R}^{n^2\times D}$.

A.2 ATTENTION FORMULATION IN IISA

Given the patch tokens of slice i at block l H_i^l, and slice token s_i^l from Section 3.2. The initial slice token is $s_i^0 = s_{learn} + e_i^{stk} + e_i^{slc}$, where s_{learn} is a learned vector shared across slices. Each block $l = 0, \ldots, L-1$ performs the following four updates. We denote multi-head attention by $\mathrm{MHA}(Q, K, V)$ and use $\mathrm{SA}(X) = \mathrm{MHA}(X, X, X)$ for self-attention. LN denotes layer normalization. Each feed-forward network (FFN) consists of two linear layers separated by GELU, applied independently to each token.

Intra-slice self-attention. Patch tokens are updated independently within each slice using a pre-normalized Transformer layer:

$$\widetilde{H}_i^l = H_i^l + \mathrm{SA}_{\mathrm{intra}}^l\left(\mathrm{LN}_{\mathrm{p},1}^l(H_i^l)\right), \qquad \bar{H}_i^l = \widetilde{H}_i^l + \mathrm{FFN}_{\mathrm{intra}}^l\left(\mathrm{LN}_{\mathrm{p},2}^l(\widetilde{H}_i^l)\right). \tag{19}$$

We use the same layer parameters for all slices.

slice-token CA. The slice token queries the updated patch tokens of its own slice:

$$\widetilde{s}_i^l = s_i^l + \mathrm{MHA}_{\mathrm{read}}^l\left(\mathrm{LN}_{\mathrm{s}}^l(s_i^l), \mathrm{LN}_{\mathrm{h}}^l(\bar{H}_i^l), \mathrm{LN}_{\mathrm{h}}^l(\bar{H}_i^l)\right). \tag{20}$$

The slice token provides the query, while the patch tokens provide the keys and values. This stage contains a residual attention update without an additional FFN.

Inter-slice self-attention. We concatenate the updated slice tokens from all input slices into $\widetilde{S}^l \in \mathbb{R}^{N\times D}$. A second pre-normalized Transformer layer exchanges information across slices:

$$\bar{S}^l = \widetilde{S}^l + \mathrm{SA}^l_{\mathrm{inter}}\left(\mathrm{LN}^l_{\mathrm{g},1}(\widetilde{S}^l)\right), \qquad S^{l+1} = \bar{S}^l + \mathrm{FFN}^l_{\mathrm{inter}}\left(\mathrm{LN}^l_{\mathrm{g},2}(\bar{S}^l)\right). \tag{21}$$

The i-th row of S^{l+1} gives the updated slice token s_i^{l+1}. Attention operates jointly across slices from all stacks.

Broadcast update. Finally, we project each updated motion token and add it to every patch token in the corresponding slice:

$$h^{l+1}_{i,j} = \mathrm{LN}_{\mathrm{b}}\left(\bar{h}^l_{i,j} + \psi_{\mathrm{b}}(s_i^{l+1})\right), \qquad j = 1,\dots,P, \tag{22}$$

where $\bar{h}^l_{i,j}$ is the j-th row of $\bar{H}^l_i$ and ψ_{b} is a learned linear layer. The broadcast linear layer and the post-addition layer normalization are shared across IISA blocks. Attention and FFN modules use separate parameters for different stages and blocks.

A.3 Details of Planar Kabsch Algorithm

Given the predicted canonical coordinates $\{\hat{\boldsymbol{x}}_{i,j}\}_{j=1}^P$ from Section 3.2, let $\boldsymbol{q}_{i,j} = (u_{i,j}, v_{i,j}, 0)^T$ denote the corresponding local patch centers, where $P = n^2$. Both coordinate sets are expressed in millimetres before fitting. The slice transformation is obtained by solving

$$(\hat{\boldsymbol{R}}_i, \hat{\boldsymbol{c}}_i) = \underset{\boldsymbol{R}\in\mathrm{SO}(3),\, \boldsymbol{c}\in\mathbb{R}^3}{\arg\min} \sum_{j=1}^P \omega_{i,j} \left\|\boldsymbol{R}\boldsymbol{q}_{i,j} + \boldsymbol{c} - \hat{\boldsymbol{x}}_{i,j}\right\|_2^2, \tag{23}$$

where $\hat{\boldsymbol{c}}_i$ is the slice center in canonical space. Let $a_{i,j}$ be the mean absolute input intensity within patch j of slice i. We compute fixed, normalized correspondence weights as

$$e_{i,j} = \frac{a_{i,j}}{\max\left(P^{-1}\sum_{k=1}^P a_{i,k}, \varepsilon_e\right)}, \qquad \omega_{i,j} = \frac{e_{i,j} + \varepsilon_e}{\sum_{k=1}^P (e_{i,k} + \varepsilon_e)}, \tag{24}$$

with $\varepsilon_e = 10^{-3}$. This weighting reduces the contribution of low-signal patches while retaining positive weights for all correspondences. We compute the weighted centroids

$$\bar{\boldsymbol{q}}_i = \sum_{j=1}^P \omega_{i,j}\boldsymbol{q}_{i,j}, \qquad \bar{\boldsymbol{x}}_i = \sum_{j=1}^P \omega_{i,j}\hat{\boldsymbol{x}}_{i,j}. \tag{25}$$

For a fixed rotation, the optimal translation is $\boldsymbol{c} = \bar{\boldsymbol{x}}_i - \boldsymbol{R}\bar{\boldsymbol{q}}_i$. Substituting this expression into Eq. 23 leaves a rotation-only alignment problem between centered points. Since the local patch centers lie in the plane $z = 0$, only the first two columns of $\boldsymbol{R}$ enter the centered objective. We therefore construct the 3×2 cross-covariance matrix

$$\boldsymbol{M}_i = \sum_{j=1}^P \omega_{i,j}(\hat{\boldsymbol{x}}_{i,j} - \bar{\boldsymbol{x}}_i)(\boldsymbol{q}_{i,j} - \bar{\boldsymbol{q}}_i)^T_{1:2}. \tag{26}$$

Writing $\boldsymbol{B} = R_{:,1:2}$, minimizing the centered objective is equivalent to maximizing $\mathrm{tr}(B^T\mathbf{M}_i)$ subject to $\boldsymbol{B}^T\boldsymbol{B} = \boldsymbol{I}_2$. For a rank-two cross-covariance, the solution follows from the thin singular value decomposition:

$$\boldsymbol{M}_i = \boldsymbol{U}_i\boldsymbol{\Sigma}_i\boldsymbol{V}_i^T, \qquad \boldsymbol{B}_i = \boldsymbol{U}_i\boldsymbol{V}_i^T, \tag{27}$$

where $\boldsymbol{U}_i \in \mathbb{R}^{3\times 2}$ and $\boldsymbol{V}_i \in \mathbb{R}^{2\times 2}$. Let $b_{i,1}$ and $b_{i,2}$ denote the orthonormal columns of $\boldsymbol{B}_i$. We recover the slice normal by their cross product and assemble the rotation:

$$\hat{\boldsymbol{R}}_i = \begin{bmatrix} b_{i,1} & b_{i,2} & b_{i,1} \times b_{i,2} \end{bmatrix}. \tag{28}$$

This right-handed completion ensures $\hat{\boldsymbol{R}}_i^T\hat{\boldsymbol{R}}_i = \boldsymbol{I}_3$ and $\det(\hat{\boldsymbol{R}}_i) = 1$. The implementation additionally normalizes and re-orthogonalizes the axes to control floating-point errors. Finally, the canonical-space slice center is recovered as

$$\hat{\boldsymbol{c}}_i = \bar{\boldsymbol{x}}_i - \hat{\boldsymbol{R}}_i\bar{\boldsymbol{q}}_i. \tag{29}$$

A.4 Derivation of PSF-aware Feature Aggregation

This appendix derives the through-plane factor underlying Eq. 10 from the forward acquisition model. We distinguish the acquisition-adjoint identities from their use as a spatial weighting rule for learned features.

A.4.1 Acquisition Operator and Its Adjoint

For a fixed slice transformation, the forward acquisition operator in Eq. 4 is $\mathcal{A}_i = \boldsymbol{D}\boldsymbol{B}\boldsymbol{T}_i$. Under the standard Euclidean inner products, its adjoint is defined by $\langle \mathcal{A}_i V, \mathbf{g}_i \rangle = \langle V, \mathcal{A}_i^* \mathbf{g}_i \rangle$ for arbitrary vectorized volume V and slice signal $\mathbf{g}_i$. Applying the adjoint identity to each component gives

$$\begin{aligned} \langle \boldsymbol{D}\boldsymbol{B}\boldsymbol{T}_i V, \mathbf{g}_i \rangle &= \langle \boldsymbol{B}\boldsymbol{T}_i V, \boldsymbol{D}^* \mathbf{g}_i \rangle \\ &= \langle \boldsymbol{T}_i V, \boldsymbol{B}^* \boldsymbol{D}^* \mathbf{g}_i \rangle \\ &= \langle V, \boldsymbol{T}_i^* \boldsymbol{B}^* \boldsymbol{D}^* \mathbf{g}_i \rangle. \end{aligned} \tag{30}$$

Therefore,

$$\mathcal{A}_i^* = \boldsymbol{T}_i^* \boldsymbol{B}^* \boldsymbol{D}^*. \tag{31}$$

For real-valued matrices, the adjoint is the transpose. In particular, if $(\mathcal{A}_i)_{jk}$ denotes the contribution of volume element k to slice measurement j, then

$$(\mathcal{A}_i^* \mathbf{g}_i)_k = \sum_j (\mathcal{A}_i)_{jk} (\mathbf{g}_i)_j. \tag{32}$$

Thus, the adjoint accumulates slice signals at volume locations using the same coefficients as the forward acquisition operator. The adjoint of continuous convolution with a real kernel $h(\mathbf{r})$ is convolution with $h(-\mathbf{r})$. The centered Gaussian PSF satisfies $h(-\mathbf{r}) = h(\mathbf{r})$, so its convolution operator is self-adjoint. With a blur discretization and boundary treatment that preserve this symmetry, $\boldsymbol{B}^* = \boldsymbol{B}$, yielding

$$\mathcal{A}_i^* = \boldsymbol{T}_i^* \boldsymbol{B}\boldsymbol{D}^*. \tag{33}$$

A.4.2 Separation of the Through-Plane PSF Factor

Let $v(\boldsymbol{p})$ denote the continuous counterpart of V, and $\mathcal{A}_i^{\mathrm{c}}$ the slice-plane acquisition operator before discrete in-plane sampling. The slice-local coordinates $(\boldsymbol{q}_i(\boldsymbol{p}), d_i(\boldsymbol{p}))$ are defined in Eq. 8. Using the symmetry of the Gaussian PSF and the unit Jacobian of the rigid transformation, the continuous forward model can be written as

$$(\mathcal{A}_i^{\mathrm{c}} v)(\boldsymbol{q}) = \int_{\mathbb{R}^3} \mathcal{H}\big(\boldsymbol{q}_i(\boldsymbol{p}) - \boldsymbol{q}, d_i(\boldsymbol{p})\big) v(\boldsymbol{p}) \, \mathrm{d}\boldsymbol{p}. \tag{34}$$

Let g_i be a slice signal, extended by zero outside the acquired field of view. For finite-support, square-integrable signals, exchanging the integration order in $\langle \mathcal{A}_i^{\mathrm{c}} v, g_i \rangle$ identifies the adjoint. Substituting the separable PSF from Eq. 5 then gives

$$\begin{aligned} \big((\mathcal{A}_i^{\mathrm{c}})^* g_i\big)(\boldsymbol{p}) &= \int_{\mathbb{R}^2} \mathcal{H}\big(\boldsymbol{q}_i(\boldsymbol{p}) - \boldsymbol{q}, d_i(\boldsymbol{p})\big) g_i(\boldsymbol{q}) \, \mathrm{d}\boldsymbol{q} \\ &= \int_{\mathbb{R}^2} \mathcal{H}_{xy}\big(\boldsymbol{q}_i(\boldsymbol{p}) - \boldsymbol{q}\big) \mathcal{H}_z\big(d_i(\boldsymbol{p})\big) g_i(\boldsymbol{q}) \, \mathrm{d}\boldsymbol{q} \\ &= \mathcal{H}_z\big(d_i(\boldsymbol{p})\big) \int_{\mathbb{R}^2} \mathcal{H}_{xy}\big(\boldsymbol{q}_i(\boldsymbol{p}) - \boldsymbol{q}\big) g_i(\boldsymbol{q}) \, \mathrm{d}\boldsymbol{q}. \end{aligned} \tag{35}$$

Since $d_i(\boldsymbol{p})$ is independent of $\mathbf{q}$, the through-plane profile factors out of the in-plane integral. This provides the signal-level basis for the through-plane weighting in Eq. 10. Our feature aggregation retains this factor while using bilinear interpolation for in-plane feature sampling, without explicitly applying the $\mathcal{H}_{xy}$ convolution.

B Datasets and Implementation Details

B.1 Detailed Datasets Description.

We evaluate our method on three benchmark T2-weighted brain MRI datasets covering distinct human developmental cohorts:

- **FeTA (Payette et al., 2021):** We use reconstructed fetal T2-weighted brain MRI volumes. The source volumes are spatially cropped, reoriented, and resampled from 0.8 mm to 1.0 mm isotropic resolution using trilinear interpolation, yielding reference volumes of size $132 \times 124 \times 124$. The dataset is randomly partitioned into 65, 5, and 10 subjects for training, validation, and testing, respectively.
- **dHCP (Hughes et al., 2017):** We use neonatal T2-weighted brain MRI volumes registered to a common canonical space and resampled to 0.8 mm isotropic resolution, yielding volumes of size $182 \times 182 \times 136$. The dataset is similarly split into 675, 20, and 76 subjects for training, validation, and testing, respectively.
- **CamCAN (Taylor et al., 2017; Shafto et al., 2014):** We use adult T2w brain MRI with 1.0 mm isotropic reference volumes, with volume size being cropped to $160 \times 192 \times 160$. The dataset is divided into 591, 20, and 41 subjects for training, validation, and testing, respectively.

B.2 Simulation of Slice Acquisition Process

Following the slice acquisition simulation protocol of SVoRT (Xu et al., 2022), we generate three motion-corrupted stacks (axial, coronal, sagittal) of 128×128 slices from the reference volume for each subject. The simulation pipeline incorporates slice-wise rigid motion, PSF-based acquisition, measurement noise, and signal attenuation.

Stack Geometry. The in-plane resolution is $1.25\,$mm for CamCAN and $1.0\,$mm for FeTA and dHCP. For each subject, the slice thickness $\tau \sim \mathcal{U}(2.5, 3.5)\,$mm and center-to-center spacing $\Delta \sim \mathcal{U}(2.5, \tau)\,$mm are randomly sampled and shared across all stacks. Additionally, each stack is perturbed by random in-plane translational offsets drawn from $\mathcal{U}(-3, 3)\,$mm.

Slice-Wise Motion. To simulate patient movement, rigid motion trajectories are independently sampled from a precomputed library, augmented via coordinate permutation, and normalized to their initial transformations. Motion states are sampled at intervals of $\mathcal{U}(1, 2)$ seconds and assigned to individual slices using an interleaving factor uniformly drawn from $\{2, \ldots, \lfloor\sqrt{N_s}\rfloor\}$ (N_s = total slice count) to mimic interleaved acquisition. This simulated motion is then composed with the nominal stack orientation to yield the final per-slice transformation. Intra-slice motion is not simulated.

PSF-Based Acquisition. Noise-free slices are generated using the forward operator (Eq. 4) with a slice-aligned anisotropic Gaussian PSF. Its physical standard deviations are $\sigma_x = \sigma_y = 1.2\delta/c$ and $\sigma_z = \tau/c$, where $c = 2\sqrt{2\ln 2}$ and δ denotes the in-plane voxel spacing. The kernel is discretized, truncated at 10^{-3} of its peak, and normalized to unit sum. The canonical volume is sampled at transformed PSF offsets via trilinear interpolation, accumulating weighted contributions.

Measurement Noise and Signal Attenuation. For each stack, a noise scale $\sigma_s \sim \mathcal{U}(0, 0.03)$ is drawn. Foreground pixels $y_i^0(\mathbf{u})$ are corrupted with Rician magnitude noise:

$$y_i^{\text{noise}}(\mathbf{u}) = \sqrt{(y_i^0(\mathbf{u}) + n_{1,i})^2 + n_{2,i}^2}, \qquad n_{1,i}, n_{2,i} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \sigma_s^2). \tag{36}$$

Subsequently, each slice has a 3% probability of undergoing local signal attenuation. Affected intensities are multiplicatively scaled by a 2D Gaussian void map:

$$v_i(\mathbf{u}) = 1 - A_i \exp\left[-\frac{1}{2}\left((u_i'/s_{x,i})^2 + (v_i'/s_{y,i})^2\right)\right], \tag{37}$$

where (u_i', v_i') are relative coordinates rotated by a random angle $\mathcal{U}(0, 2\pi)$. The void parameters are $A_i \sim \mathcal{U}(0.5, 1)$, $a_i \sim \mathcal{U}(5, 15)$ pixels, and $s_{x,i} \sim \mathcal{U}(1, 16)$ pixels, with $s_{y,i} = a_i^2/s_{x,i}$ maintaining area proportionality. This pipeline is applied identically for generating both training and test data.

B.3 Atlas Construction

We construct a separate, fixed binary occupancy atlas for each of the three datasets (FeTA, CamCAN, and dHCP) following the same procedure. Each atlas provides a coarse brain support region. For a given dataset, we randomly sample $M = 20$ training masks. The masks are obtained from the provided brain segmentations within the datasets. We adopt the canonical volume grids established during preprocessing, requiring no additional registration steps during atlas construction. Let m_j denote the j-th selected binary mask and $\mathcal{G}$ denote the target latent grid. We downsample each mask via adaptive 3D max pooling and compute their cell-wise union:

$$a_0(\mathbf{c}) = \max_{1 \le j \le M} \left[\mathrm{AdaptiveMaxPool}_{\mathcal{G}}(m_j) \right]_{\mathbf{c}}, \qquad \mathbf{c} \in \mathcal{G}. \tag{38}$$

Consequently, a coarse cell is marked as occupied if its corresponding pooling region contains foreground voxels in at least one of the selected masks. To ensure adequate boundary coverage, we expand this union by a one-cell margin via morphological dilation, ultimately defining the candidate query cell set $\mathcal{V}$:

$$a(\mathbf{c}) = \max_{\boldsymbol{\delta} \in \{-1,0,1\}^3} a_0(\mathbf{c} + \boldsymbol{\delta}), \qquad \mathcal{V} = \{ \boldsymbol{p}_{\mathbf{c}} \mid \mathbf{c} \in \mathcal{G},\ a(\mathbf{c}) = 1 \}, \tag{39}$$

where a_0 is zero-padded outside $\mathcal{G}$, and $\boldsymbol{p}_{\mathbf{c}}$ is the canonical-space center of cell $\mathbf{c}$. Practically, this dilation is implemented as a 3D max pooling operation with a kernel size of 3, stride of 1, and padding of 1. In (z, y, x) order, the dimensions of the latent grids $\mathcal{G}$ are $33 \times 31 \times 31$, $40 \times 48 \times 40$, and $46 \times 46 \times 34$ for FeTA, CamCAN, and dHCP, respectively. The atlas remains fixed throughout both training and inference. The atlas dictates the fixed spatial locations of the candidate cells; the actual slice associations and PSF weights are dynamically computed from the network-estimated motions, ensuring that the aggregated features are driven entirely by the input slice observations.

B.4 Slice Encoder.

The shared CNN-based slice encoder utilizes a four-stage residual architecture to transform each 128×128 input slice into a $1024 \times 32 \times 32$ feature map. The input is first projected to 64 channels using a 3×3 convolution, followed by GroupNorm and a SiLU activation. The main backbone has four residual stages with progressive channel widths of $(64, 128, 256, 512)$ and spatial strides of $(1, 2, 4, 8)$. These stages contain $(2, 2, 2, 1)$ residual blocks, respectively. Within each residual block, two 3×3 convolutions, each preceded by GroupNorm and SiLU, are paired with a skip connection. Transitions between successive stages are handled by 3×3 convolutions with a stride of 2. Throughout the network, all 3×3 convolutions use padding of 1, while all other convolutions use a stride of 1. To form the context feature map, the stride-8 features are bilinearly upsampled to a 32×32 resolution and concatenated with the stride-4 features. This resulting 768-channel tensor is subsequently processed by two 1×1 convolutional layers (separated by GroupNorm and SiLU), expanding the output to 1024 channels. Notably, GroupNorm operates with 8 groups in the initial convolutional layer and 32 groups throughout the rest of the architecture.

B.5 Sparse Transformer Decoder.

The decoder receives 1024-dimensional aggregated features at $|\mathcal{V}|$ active cells, together with their integer 3D grid coordinates. A linear projection maps the features to 768 dimensions, followed by the addition of fixed sinusoidal absolute positional encodings of the grid coordinates. The features pass through 12 sparse Transformer blocks with 12 attention heads and 64 channels per head. Attention is restricted to active cells within $4 \times 4 \times 4$ windows. Successive blocks alternate between window offsets $(0, 0, 0)$ and $(2, 2, 2)$, measured in latent-grid cells, allowing information exchange across window boundaries. The active coordinates and spatial resolution remain unchanged throughout the decoder. Window attention uses packed FlashAttention to accommodate variable numbers of active cells per window. Each block uses pre-normalization and residual connections around both the attention and feed-forward sublayers. The feed-forward network comprises an intermediate hidden layer of 3072 dimensions, with both input and output dimensions set to 768, and employs a GELU activation with the tanh approximation. The pre-normalization layers use non-affine LayerNorm. A final LayerNorm and a zero-initialized linear projection from 768 to 88 dimensions predict the parameters of eight 11-dimensional Gaussians per active cell.

B.6 Comparison Methods

SVRTK (Uus et al., 2020). SVRTK alternates slice-to-volume registration and super-resolution reconstruction. Using its rigid pipeline, we processed three input stacks per subject, performing three outer iterations, with 7 super-resolution iterations in intermediate stages and 21 in the final stage. The first stack's segmentation mask defined the reconstruction region, initialized with prescribed orthogonal acquisition orientations. Reconstruction resolution was set to 1.0 mm for FeTA and CamCAN, and 0.8 mm for dHCP.

SVRnet (Hou et al., 2018). SVRnet independently predicts slice locations using a CNN. We adopted the SVoRT repository's single-channel VGG16 implementation, which regresses three 3D anchor points per slice for conversion into rigid transformations and subsequent parameter-free PSF adjoint reconstruction. Models were trained from scratch for 200,000 steps per dataset using an anchor-point MSE loss (weight 0.1). Optimization utilized AdamW with an initial learning rate of 2×10^{-4}, a weight decay of 10^{-4}, and 500 warm-up steps followed by linear decay.

PlaneInVol (Yeung et al., 2021). PlaneInVol incorporates pairwise attention to model inter-slice relationships. We used its MRI adaptation from the SVoRT repository, which predicts three 3D anchor points per slice for subsequent PSF adjoint reconstruction. The model was trained from scratch for 200,000 steps per dataset, sharing the exact same objective, optimizer, and learning-rate schedule as SVRnet.

SVoRT (Xu et al., 2022). SVoRT iteratively refines slice transformations and volume reconstruction. Using its official implementation with four refinement iterations, each step performs PSF back-projection followed by learned super-resolution via two conjugate-gradient steps. Training jointly supervised anchor points, rotations, translations, and reconstructed volumes. Models were trained from scratch for 200,000 steps per dataset using AdamW (initial learning rate 2×10^{-4}, weight decay 0.01, linear warm-up and decay), with gradients accumulated over eight iterations.

FC-SVR (Young et al., 2024). FC-SVR estimates motion per stack and reconstructs volumes via spatial splatting and learned inpainting. Following its official implementation, the motion and inpainting networks were trained separately for 200,000 and 150,000 steps, respectively, using Adam with a learning rate of 10^{-4}. The inpainting network was supervised on sparse volumes generated from ground-truth motions. The results are evaluated by a ground-truth-flow-derived rigid correction to each stack prior to fusion to compensate for global rigid misalignment.

SVRMamba (Wu et al., 2025). SVRMamba models slice-sequence dependencies for motion estimation and volume reconstruction. Our reimplementation utilizes per-stack odd–even slice ordering, a single-channel ResNet50 encoder, positional encodings, and a four-layer Mamba block. A regression head predicts slice rotations and centers, followed by differentiable PSF stacking and a 3D U-Net. Models were trained from scratch for 200,000 steps using geodesic rotation, translation, and volume RMSE losses. Optimization used Adam with cosine decay from 3×10^{-4} to 10^{-4}.

C Additional Experiment

C.1 Effect of Patch Token Resolution

We investigate the impact of the patch-token grid resolution in the motion estimator by evaluating grids of 2×2, 4×4, 8×8, and 16×16 (corresponding to 4, 16, 64, and 256 tokens per slice, respectively). Across all variants, the decoder is fixed to generate eight Gaussians per active cell. As reported in Tables 4 and 5 on the complete dHCP test set of 76 subjects, coarser grids (2×2 and 4×4) result in noticeably lower reconstruction quality and larger transformation errors, emphasizing that robust motion estimation heavily relies on capturing sufficient local spatial detail. However, increasing the resolution further from 8×8 to 16×16 provides only marginal performance gains. Reconstruction PSNR improves by a negligible 0.01 dB, while mean ED decreases by merely 0.02 mm. We therefore adopt the 8×8 grid as our default configuration, as it strikes an optimal balance—achieving near-identical reconstruction fidelity and motion estimation accuracy to

Table 4: Reconstruction performance with different patch-token grids on dHCP. Values are mean(SD).

Grid	dHCP		
	PSNR ↑	SSIM ↑	NRMSE ↓
2×2	29.15(0.82)	0.880(0.007)	0.202(0.022)
4×4	29.27(0.82)	0.881(0.007)	0.199(0.022)
8×8	29.47(0.87)	**0.883(0.007)**	0.194(0.022)
16×16	**29.48(0.86)**	0.882(0.006)	**0.193(0.022)**

Table 5: Motion estimation performance with different patch-token grids on dHCP. Values are mean(SD).

Grid	dHCP		
	ED (mm) ↓	GD (rad) ↓	TED (mm) ↓
2×2	4.257(4.003)	0.074(0.083)	2.466(2.595)
4×4	3.936(3.724)	0.068(0.079)	2.281(2.577)
8×8	3.785(3.663)	0.066(0.079)	2.193(2.563)
16×16	**3.764(3.622)**	**0.065(0.076)**	**2.170(2.501)**

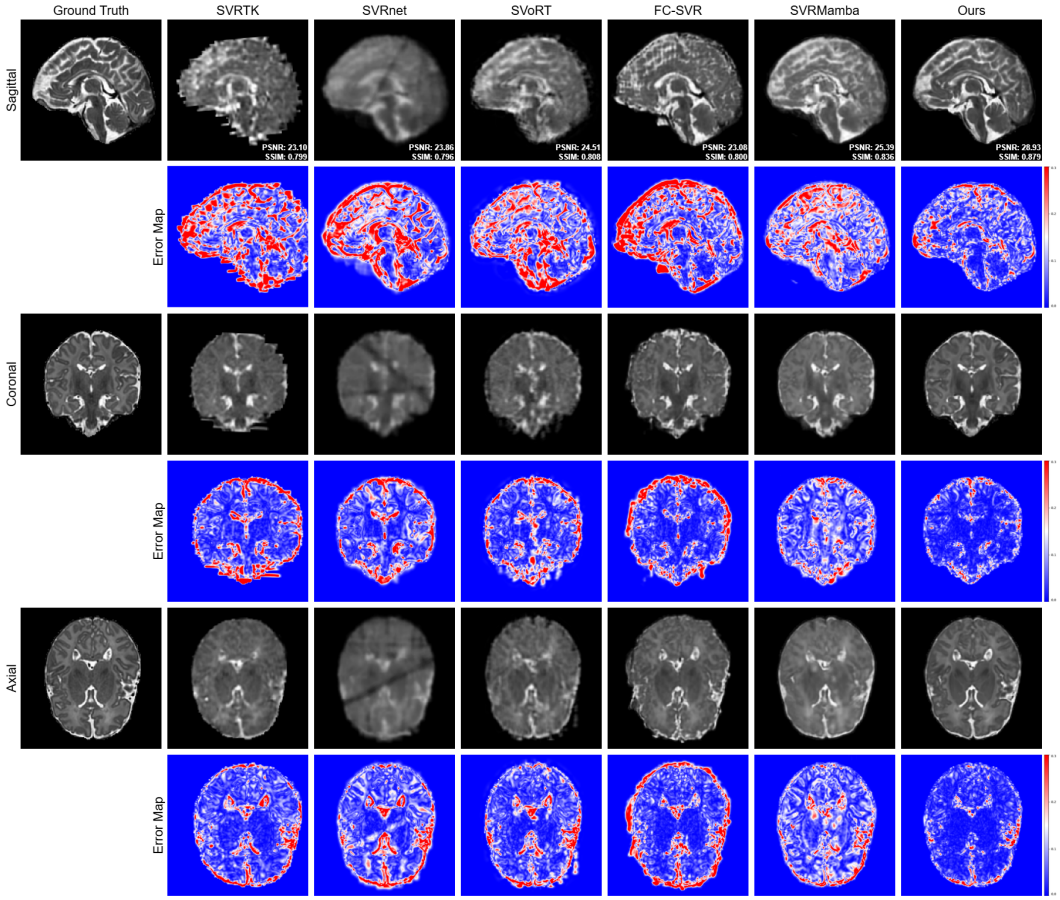

Figure 4: Qualitative results on the dHCP dataset. The figure shows reconstructed sagittal, coronal, and axial slices of a single test subject. Corresponding absolute error maps are displayed below each view. The lower-right corner of each sagittal view reports the full-volume PSNR (dB) and SSIM of the corresponding reconstruction.

the 16×16 grid while significantly reducing the computational burden by requiring only one-quarter of the patch tokens.

C.2 Additional Qualitative Results

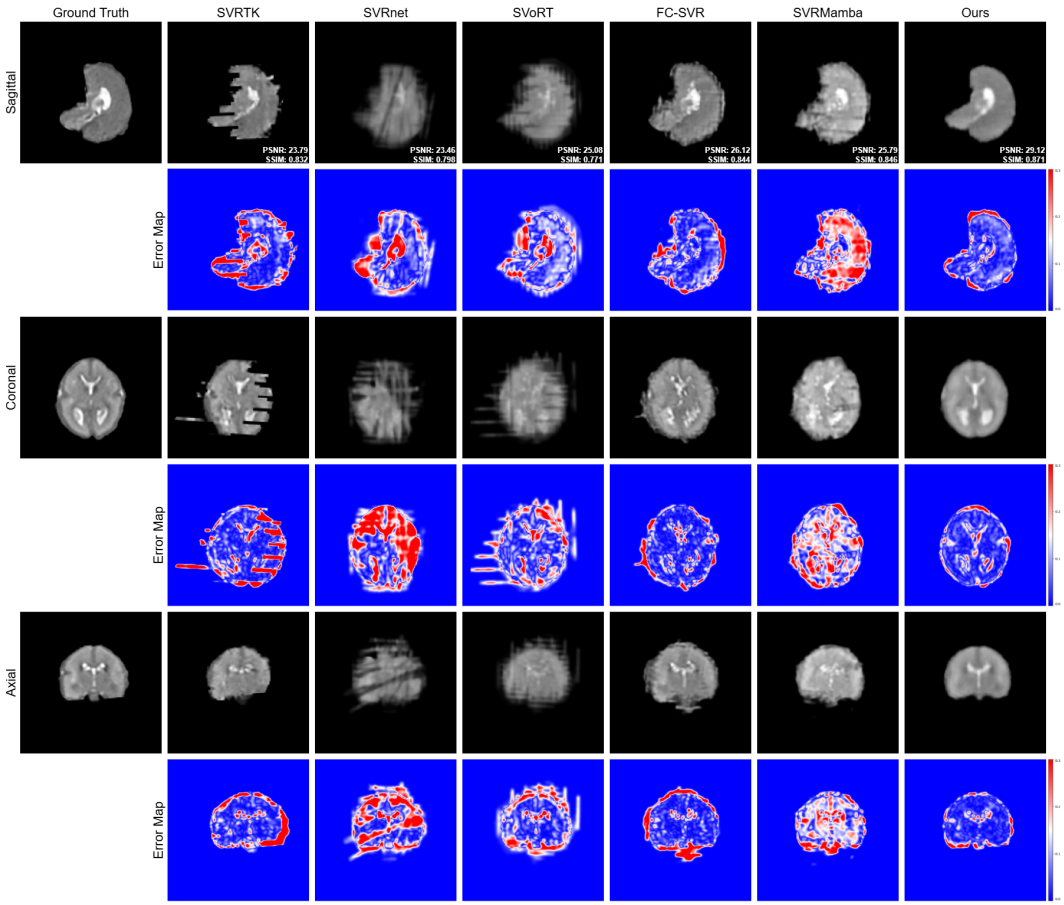

Figure 5: Qualitative results on the FeTA dataset. The figure shows reconstructed sagittal, coronal, and axial slices of a single test subject. Corresponding absolute error maps are displayed below each view. The lower-right corner of each sagittal view reports the full-volume PSNR (dB) and SSIM of the corresponding reconstruction.

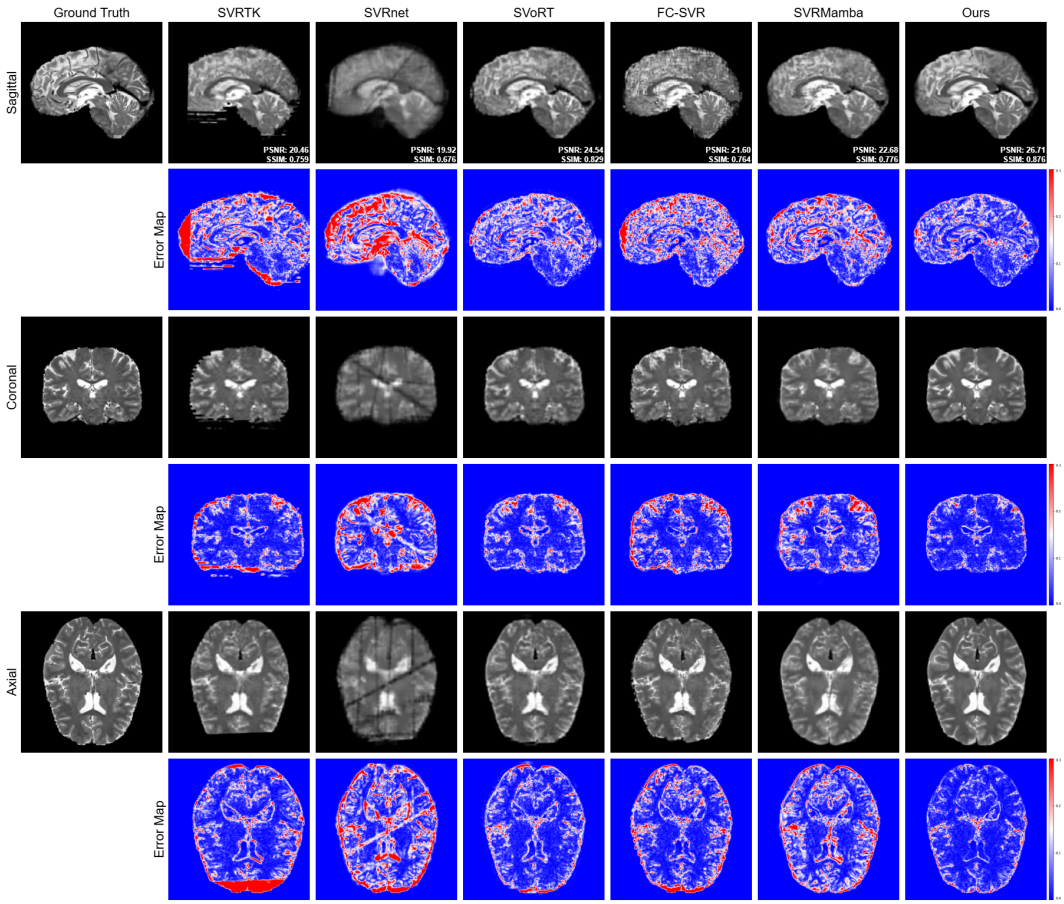

Figure 6: Qualitative results on the CamCAN dataset. The figure shows reconstructed sagittal, coronal, and axial slices of a single test subject. Corresponding absolute error maps are displayed below each view. The lower-right corner of each sagittal view reports the full-volume PSNR (dB) and SSIM of the corresponding reconstruction.