Appendix A: *Supplementary Information for "PLayer: A Plug-and-Play Embedded Neural System to Boost Neural Organoid 3D Reconstruction"*

Yuanzheng Ma[a,b,f], Davit Khutsishvili[a,b,f], Zihan Zang[c], Wei Yue[d], Zhen Guo[e], Tao Feng[a,b], Zitian Wang[a,b], Liwei Lin[d], Shaohua Ma[a,b,*], and Xun Guan[a,b,*]

[a]Tsinghua-Berkeley Shenzhen Institute, Tsinghua University, Shenzhen, 518055, China
[b]Tsinghua Shenzhen International Graduate School, Tsinghua University, Shenzhen, 518055, China
[c]Department of Bioengineering, University of California, Los Angeles, Los Angeles, CA, 90095, USA
[d]Department of Mechanical Engineering, University of California, Berkeley, CA, 94720, USA
[e]Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, Massachusetts, 02139, USA
[f]These authors contributed equally

[*]Correspondence: ma.shaohua@sz.tsinghua.edu.cn; xun.guan@sz.tsinghua.edu.cn

A.1 Specimen Preparation and Experimental Parameters

Unguided neural organoids for training (Fig. 1-3, Fig. S1) were produced using the STEMdiff™ Cerebral Organoid Kit (STEMCELL Technologies Catalog # 08570), following the guidelines provided by the manufacturer and previous studies[56]. Regionalized neural organoids depicted in Fig. 4-7 and Fig. S2 were produced using STEMdiff™ Dorsal Forebrain Organoid Differentiation Kit (STEMCELL Technologies Catalog # 08620), adhering to the manufacturer's instructions and previous research methodologies[57]. For extensive characterization details ensuring the presented organoids correspond to reality, refer to the work by Wang et al.[17] and Tang et al.[20]. Cerebral Organoids of Day 19, Day 34, Day 71, and Day 112 age and Dorsal Forebrain Organoids of Day 82 age were collected on the same day, rinsed twice with PBS to eliminate any leftover medium, and then fixed with 4.0% (w/v) PFA at 4°C overnight.

After washing out the PFA with PBS, the samples were sequentially soaked in 10%, 20%, 27 and 30% (w/v) sucrose solutions, then submerged in Optimal Cutting Temperature (OCT) compound (Tissue-Tek), and left overnight at 4°C. The tissue samples were subsequently frozen, and 10μm, 20μm, 30 μm, and 60μm sections were obtained via cryosectioning. For staining, all sectioned slices were treated with 10% normal donkey serum containing 1.0% (v/v) Triton X-100 (Sigma-Aldrich). The primary antibodies (β-tubulin (TuJ1), Abcam, ab18207, and

BioLegend, 801201; Neurofilament Heavy, MilliporeSigma, AB5539; MAP2, Abcam, ab92434 and ab5392) were applied to the samples and left at 4°C for 3 days. The samples were then washed four times, each at different durations (1 min – 10 min – 10 min – 10 min). Following this, they were incubated with the corresponding secondary antibody (Goat Anti-Chicken IgY H&L, ab150169; Goat Anti-Mouse IgG H&L, ab150113; Goat Anti-Rabbit IgG H&L 488, ab150077 - Alexa Fluor® 488) for 2 hours at room temperature, which was then washed off in a similar manner with 1 min – 10 min – 1 hour – 5 hour durations. The slices were then stained with DAPI (Sigma-Aldrich) for 30 minutes and mounted with glycerol (Sigma-Aldrich). Finally, the samples were examined under a confocal microscope (Nikon) for imaging and analysis as shown in Fig. 1.

For the *in-silico* experiment, configurations including learning rate, number of epochs, loss function type, optimizer, data augmentations, and learning rate scheduler are provided in Tab. S1 below.

Learning Rate	0.0002
Number of Epochs	1000
Types of Loss Function	L1 loss
Optimizer	Adam
Data Augmentations	Random Rotation, Random Crop, Random Brightness/Contrast/Saturation
Learning Rate Scheduler	Cosine Annealing

Tab. S1 Key Parameters Used in the Experimental Setup

Random adjustments to brightness, contrast, and saturation were applied to mitigate brightness variability across layers, particularly in the presence of bright spots. This augmentation was essential, as normalization alone was insufficient to address such variations.

A.2 GAN-Based Denoising for Improved Signal-to-Noise Ratio

The generated confocal images showed dense neuron-specific Class III TuJ1 signals attributed to high elevations of young neurons within the neural organoids. However, the

abundance of these signals, combined with the thickness of the slides, led to out-of-focus layers projecting their secondary antibody-coupled fluorophore signal. This resulted in visually unappealing images and introduced a pronounced background signal, posing challenges for the reconstruction and analysis of individual layers.

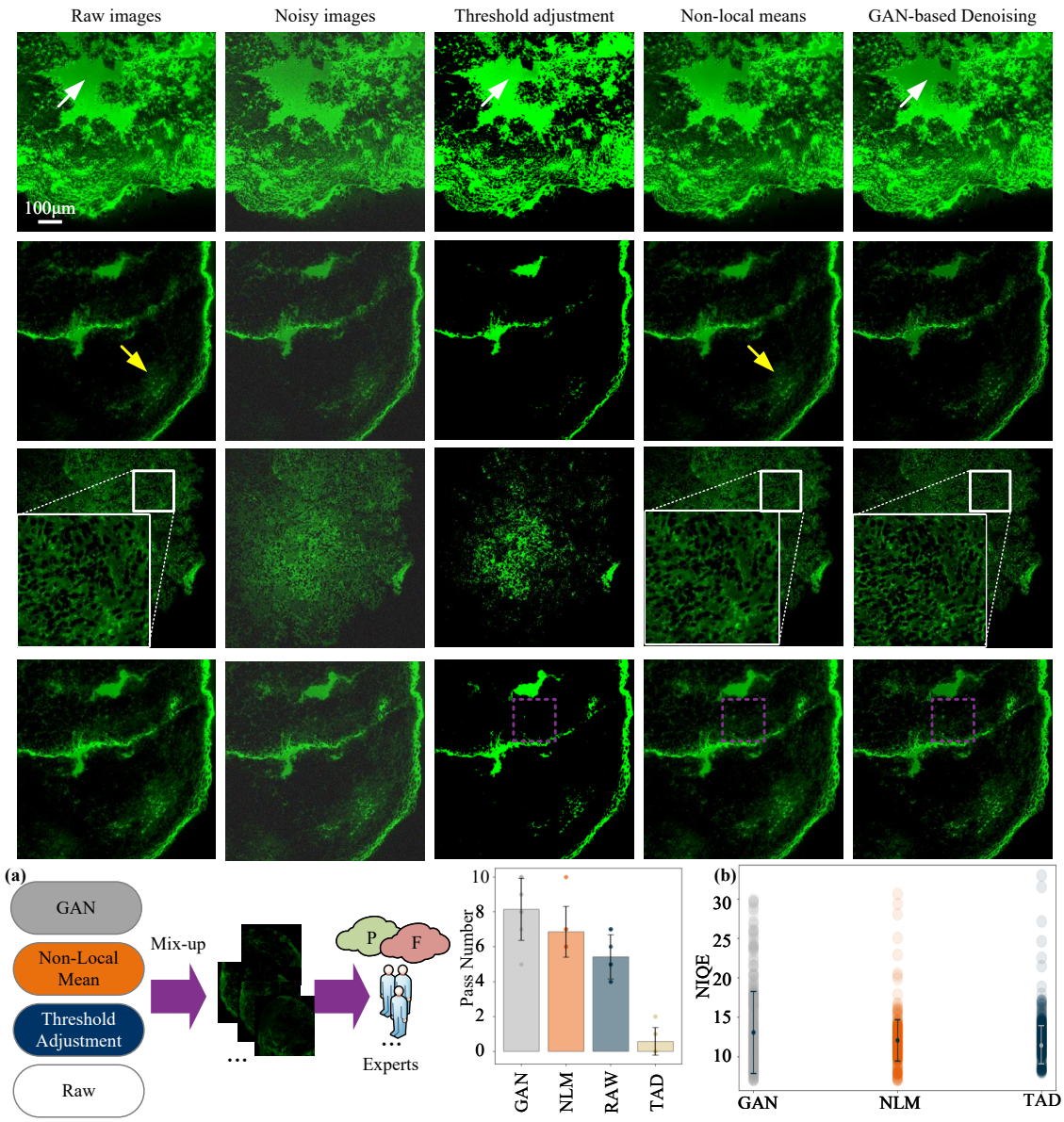

Fig. S1 A comprehensive comparative analysis of various image denoising methods. The figure consists of three sections: a visual representation at the top, (a), and (b). Columns in the top section, from left to right, show: 1st, raw unaltered images; 2nd, images with added noise; 3rd, images denoised using threshold adjustment; 4th, images denoised using non-local means; and 5th, images denoised using GAN-based methods. Rows, from top to bottom, represent four different samples. (a) The design of the expert evaluation study and pass rates for each denoising method. (b) NIQE evaluation results. Stained with anti-beta III tubulin primary, and a secondary antibody conjugated to 488 nm.

Since the primary sources of noise in confocal microscopy, including photon shot noise, electronic noise, and signal intensity fluctuations, typically adhere to Poisson and Gaussian distributions, we postulated that the raw image comprised the pure organoid image with additive noise: $I_{raw} = I_{pure} + G_1$. To mitigate Gaussian noise (G_1) within the raw data, we introduced additional Gaussian noise (G_2) to the raw image. Subsequently, we employed a Generative Adversarial Network (GAN)-based denoising neural network to restore I_{raw} from $I_{pure} + G_1 + G_2$, which was trained to learn the distribution of diverse Gaussian noise. This model was then utilized to process I_{raw} and remove the Gaussian noise G_1.

Our initial step involved applying the GAN-based denoising model to the raw image, ensuring subsequent Embedded Neural Network (ENN) was shielded from and not influenced by noise. This strategic denoising step enhanced the robustness of subsequent machine learning applications, thereby fostering improved accuracy and reliability in image analysis and interpretation. To show the efficacy of our chosen background denoising approach in enhancing the Signal-to-Noise Ratio of confocal images, we present a comparative analysis of various methods in Supplementary Figure 1 (Fig. S1).

The first column of images displays the raw, unaltered images. In Fig. S1, discernible noise is evident in the initial images. The second column presents noisy images ($I_{pure} + G_1 + G_2$), showing the additive Gaussian noise. The third column employed a threshold adjustment denoising method[75], resulting in a significant modification of the image with increased contrast between light and dark regions. However, this approach adversely impacted organoid pixel intensity, as indicated by the white arrows in the first row of the third column. The fourth column introduced the non-local means denoising method from OpenCV[76], which demonstrated superior noise reduction compared to the threshold adjustment method. However, the presence of varying levels of Gaussian noise within the organoid's diverse fibers posed a challenge for the hard-coded nonlocal means method, especially when closely resembling those

of the organoid, as indicated by the yellow arrows in the second row. Notably, the GAN-based denoising method not only surpassed noise reduction but also exhibited the capability to discern specific characteristics within the organoid image, enhancing structural clarity, as evidenced in the white rectangle in the fourth row.

To provide an objective evaluation of these outcomes, considering the absence of a reference point in the denoised images, seven biomedical experts participated in the assessment of super-resolution images within a diverse dataset. This dataset encompassed images processed through various methods, including GAN-based denoising, non-local means denoising, noise suppression through threshold adjustment, and raw, unprocessed images. Fig. S1(a) presents statistical Pass/Fail results across different methods, as evaluated by the biomedical experts. Fig. S1(b) shows the statistical outcomes of various denoising methods using the NIQE. Notably, in Fig. S1(b), our GAN-based denoising method emerged as the most effective in reducing noise in organoid images, demonstrating the highest mean score. However, it is essential to acknowledge its difficulty in handling certain noisy images, as reflected by a considerable variance.

A.3 Neural Network Structure of the Restorer

As shown in the main Fig. 2, the overall structure of the *Restorer* begins by using multi-scale attention to extract features from the 2D layers. This enhances feature representation and optimizes the neural network structure. Additionally, we incorporate an additive attention block, as highlighted in the main Fig. 2 and Fig. 3, to further refine the *Restorer's* performance. All attention blocks are based on convolution layers, which makes them easier to reduce in size during model pruning and quantization.

Inspired by TransUNet[77, 78], we introduced a transformer block within the UNet structure to enhance the pyramid-like deep reconstruction at different scales. Each level of the

block contains a Transformer block[79], followed by a convolution block. This attention-based architecture not only achieves strong performance in reconstruction but also maintains a relatively compact model size, suitable for deployment.

A.4 Statistical Map Estimation

Within the intricate 3D volumes of organoids, discernible spatial continuity and clusterability among adjacent layers are evident in the organoid structure[80, 81]. This implies the existence of shared structural patterns or features among neighboring layers, particularly when the 3D volume reflects a coherent organoid structure. This insight prompts us to consider using vertical interpolation as an alternative to complex 3D convolutional neural networks. To achieve this, we employed the Inverse Proportional Function (IPF) to estimate the weights of different neighboring layers while interpolating into the target layer.

IPF assigns larger weights to nearby layers and smaller or even zero weights to layers distant from the target, as in the equation:

$$W_x = \frac{a}{D} + b \,, \tag{1}$$

where a and b are undetermined coefficients of IPF, estimated through curve fitting; D represents the distance between the target layer and its neighboring layer. This approach simplifies the mapping process and is well-suited for confocal microscopy. When trained on a high-resolution 3D volume with dimensions of 2048×2048×L (L being the number of layers), the GPU encountered challenges in processing the entire volume directly. Moreover, neural networks also encountered difficulties in managing the increased artifacts resulting from the previous interpolation when directly inputting the interpolated images into the network.

We applied downsampling to interpolate 2D layer images so that the neural network focused on the most confident information from the neighboring layers and mitigated artifact retention[82-84]. Importantly, the decrease in horizontal resolution reduced the demand for

computation power and memory usage during the reconstruction. Consequently, we investigated the utilization of a compact Convolutional Neural Network (CNN) architecture to encode the low-resolution version of the interpolated layer within the 3D volume. The defined loss function is expressed as:

$$Loss = L1\left(W_{n-2t} \times CNN\left(L_{n-2t}\right) + W_{n-t} \times CNN\left(L_{n-t}\right) + W_{n+t} \times CNN\left(L_{n+t}\right) + W_{n+2t} \times CNN\left(L_{n+2t}\right), I_{target}\downarrow\right), \quad (2)$$

where W_x, $x = n-2t, n-t, n+t, n+2t$, are the weights calculated with IPF; $n-2t$, $n-t$, $n+t$, and $n+2t$ are the layer indices, and t is the sampling interval (axial resolution magnification) of confocal microscopy. $I_{\text{target}}\downarrow$ symbolizes the target image. In our experiments, we employed t = 1,2,4 times downsampling to leave only the model's input layers, which interpolates into the target layer. The CNN model enabled us to encode the layer of low-resolution version for saving memory and accelerating computing. Additionally, it estimates the probability of specific pixels appearing on the target layer, earning the designation 'Statistical Map Estimation' (SME) for the target layer. Therefore, SME extracts features from neighboring layers prior to interpolation, expanding the field of view and enhancing interpolation efficiency, thereby improving the accuracy of the reconstruction process.

A.5 Neural Network Pruning and Quantization

Even though the model had already been optimized for computation efficiency by encoding the 3D volume by Statistical Map Estimation (SME) in low resolution, it was still oversized to be deployed in the embedded environment for fast reconstruction. We continued to apply parameter pruning and quantization to reduce the model size and accelerate inference speed, thus enhancing its computational efficiency[85].

Initially, we commenced the pruning process with a ratio of x = 21.83%, leveraging PyTorch's pruning functionalities to selectively remove elements[86]. It is essential to note that

the model tended to overfit in the context of biomedical imaging due to a limited dataset. Consequently, specific pruning ratios may lead to performance improvement. To find the best pruning ratio (x), we progressively increased it and iterated the pruning process until there was no further enhancement in model performance. We selected pruning ratios ranging from 0.6 to 0.995 to evaluate their impact on both model performance and deployment feasibility on resource-constrained devices, including Raspberry Pi 5. Through these experiments, we identified that a pruning ratio of 0.99 represents the optimal balance, as it is the first configuration that satisfies the Raspberry Pi's memory limitations while maintaining acceptable performance tradeoffs. Furthermore, while pruning inevitably leads to some loss in image quality, this tradeoff enables the practical application of the model in real-world scenarios where computational resources are limited. This rationale is critical to the framework's design, focusing on maximizing performance under strict deployment requirements (Fig.2(a); Tab. S2, 3, 4).

We opted to reduce the input layer sizes of the model to accelerate the inference. By allowing the model to process smaller images, we significantly enhanced the processing speed compared to that of a full-sized organoid layer image. This modification involved receiving and predicting the 512×512 version of the organoid layer, utilizing BICUBIC interpolation to upscale the expected image size for display. This integration of parameter pruning and additional optimization was intended to balance model size and real-time processing efficacy in 3D organoid imaging. Herein, this refined model is referred to as a Reside-Embedded Neural Network (R-ENN).

To demonstrate variations in GPU performance (Fig. 6), we modeled a warm-up phase. Degraded reconstruction quality was observed in the first layer at 2×, 4×, and 8× compression factors (Tab. S2). These factors represent the level of data compression (e.g., 2× uses 10 input layers, while 8× uses 3). GPU optimization with a warm-up phase improved PSNR, SSIM,

MSE, NRMSE, and processing time, as shown in the table below. It should be noted that Fig. 6 in the manuscript highlights values without this phase since it better reflects general user behavior.

Compression Factor	2x		4x		8x	
Methods	w/	w/o	w/	w/o	w/	w/o
PSNR	30.8860	30.7900	29.8130	29.1220	28.1580	27.7670
SSIM	0.9624	0.9513	0.9374	0.9360	0.9273	0.9200
MSE	0.0008	0.0008	0.0011	0.0012	0.0015	0.0017
NRMSE	0.0286	0.0288	0.0323	0.0350	0.0392	0.0409
Time	14.3820	14.8110	14.3790	14.8000	14.3700	14.7670

Tab. S2 Comparison of Reconstruction Metrics With and Without GPU Warm-Up

A.6 Deployment of the Embedded Neural Network on Raspberry Pi

We trained the neural network using our server equipped with two Nvidia A6000 GPUs. After pruning the model (the pruning ratio is 21.83%), we deployed the neural network on the Raspberry Pi 5 with 8 GB of RAM. Using the Raspberry Pi's screen, we can directly display the real 3D volume of the organoid. The user can rotate, enlarge, and drag the image to different parts to observe details. Supplementary Figure 2 (Fig. S2) shows the display on the Raspberry Pi. More details are provided in the *Supplementary Video Three*.

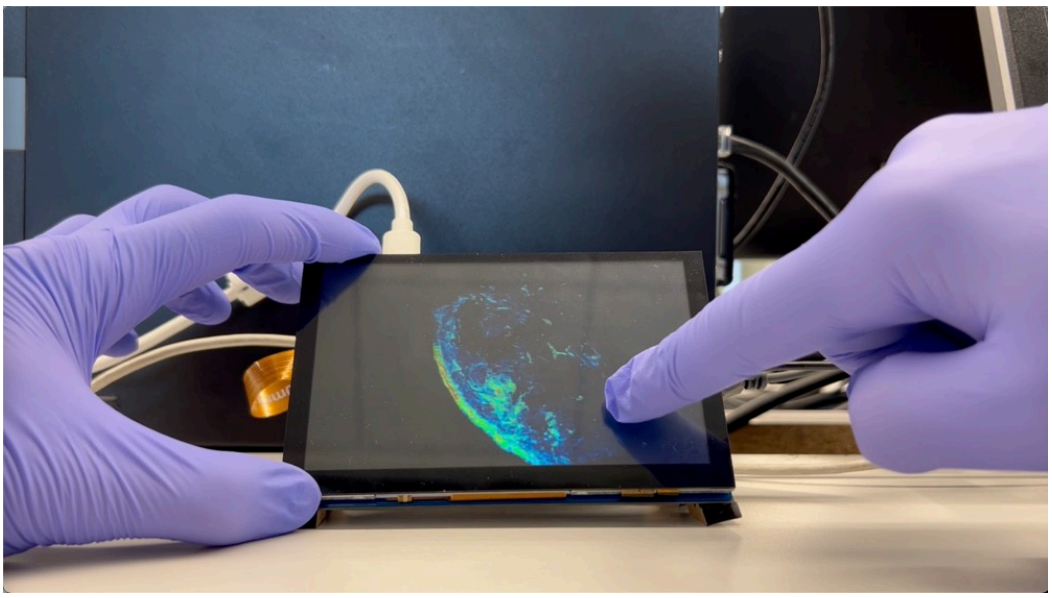

Fig. S2 Demonstration of Displaying the 3D Volume of the Organoid on an Embedded Device

A.7 Comparison of PLayer with Interpolation Methods and Denoising Networks for 3D Reconstruction

Tab. S3 below presents a comparison of *PLayer* with other 3D reconstruction methods, evaluated using PSNR and SSIM metrics. As shown, *PLayer* consistently outperforms interpolation methods like IPF and Cubic. For example, in Sample O1 (Organoid 1), *PLayer* achieves a SSIM of 0.822 and a PSNR of 29.74, significantly higher than IPF (SSIM: 0.662, PSNR: 25.60) and Cubic (SSIM: 0.575, PSNR: 23.84).

Moreover, *PLayer* surpasses the pretrained Denoising UNet[5] across all samples, with Sample O3 showcasing *PLayer*'s SSIM of 0.856 and PSNR of 30.40, compared to UNet's lower SSIM of 0.680 and PSNR of 25.73. The superior performance of *PLayer* can be attributed to its ability to reconstruct end-to-end from low-resolution 3D volumes to high-resolution output. In contrast, the original Denoising UNet is limited in its ability to effectively capture long-range dependencies and global context[87]. The comparisons in the table show that the reconstruction of *PLayer* better aligns with reality than its counterparts.

Methods	***PLayer***			**IPF**			**GND**			**Cubic**			**Denoising UNet**		
Metrics / Samples	SSIM	PSNR	NRMSE	SSIM	PSNR	NRMSE	SSIM	PSNR	NRMSE	SSIM	PSNR	NRMSE	SSIM	PSNR	NRMSE
O1 (20 layers)	0.8220	29.740	0.0083	0.6620	25.597	0.0134	0.6170	24.550	0.0151	0.5750	23.841	0.0164	0.5570	23.391	0.0173
O2 (21 layers)	0.7780	29.240	0.0088	0.6720	25.387	0.0137	0.6410	24.334	0.0155	0.6430	24.829	0.0146	0.6040	23.619	0.0168
O3 (21 layers)	0.8560	30.400	0.0077	0.8000	28.397	0.0097	0.7660	27.532	0.0107	0.7520	27.277	0.0110	0.6800	25.729	0.0132
O4 (21 layers)	0.8540	31.100	0.0071	0.8290	30.932	0.0072	0.8200	30.346	0.0077	0.8160	30.448	0.0077	0.8160	30.209	0.0079
O5 (21 layers)	0.7440	30.800	0.0074	0.8160	30.377	0.0077	0.7850	29.233	0.0088	0.8030	29.538	0.0085	0.7650	28.431	0.0097

Tab. S3 Comparison Between *PLayer* and Other 3D Reconstruction Methods

A.8 Comparison of PLayer with Other Methods: Memory and Time Efficiency vs. Reconstruction Accuracy

Tab. S4 below compares the performance of *PLayer* with the Vision Transformer (ViT, a pretrained model) and our previous *LayerLink* method, focusing on memory usage, time efficiency, and reconstruction accuracy based on average SSIM, PSNR, MSE, and NRMSE.

Methods	Image ID	Avg Time (s/layer)	Avg SSIM	Avg PSNR (dB)	Avg MSE	Avg NRMSE	Memory Usage
PLayer **Prunning ratio = 0.99**	O1 (512×512)	26.5281	0.7500	26.4700	0.0022	0.1860	630.66 MB
	O2 (512×512)	30.2975	0.7500	26.6500	0.0022	0.1820	639.61 MB
	O3 (512×512)	31.4534	0.7400	25.9500	0.0025	0.1910	637.41 MB
	O4 (512×512)	30.9989	0.8200	28.7700	0.0013	0.1620	639.37 MB
ViT (w/ ***LayerLink*** **)**	O1 (512×512)	46.4001	0.8800	30.4700	0.0009	0.1480	720.41 MB
	O2 (512×512)	46.7148	0.8000	29.7700	0.0010	0.1530	724.50 MB
	O3 (512×512)	46.7220	0.8500	30.2300	0.0009	0.1500	724.88 MB
	O4 (512×512)	44.6817	0.8600	30.6600	0.0009	0.1470	726.35 MB
ViT (w/o ***LayerLink*** **)**	O1 (512×512)	25.5454	0.6000	21.7651	0.0066	0.3290	711.05 MB
	O2 (512×512)	29.6878	0.6200	22.3455	0.0058	0.3200	733.23 MB
	O3 (512×512)	31.7682	0.5900	20.4551	0.0090	0.3520	722.22 MB
	O4 (512×512)	33.1917	0.6700	22.9957	0.0050	0.3140	710.33 MB
LayerLink **Interpolation**	O1 (512×512)	1.7500	0.6100	21.3900	0.0072	0.3300	154.82 MB
	O2 (512×512)	1.9000	0.5900	21.0100	0.0079	0.3370	155.20 MB
	O3 (512×512)	1.8900	0.5300	20.8100	0.0083	0.3410	154.41 MB
	O4 (512×512)	1.8500	0.6100	22.3900	0.0058	0.3170	155.85 MB

Tab. S4 Comparison of *PLayer* with Other Methods: Memory and Time Efficiency vs. Reconstruction Accuracy

PLayer demonstrates clear advantages in time and memory efficiency compared to ViT. *PLayer*'s model is significantly smaller (around 5 MB) than the ViT (around 35 MB), resulting in faster processing times per layer, around 30 seconds for *PLayer* compared to approximately 46 seconds for ViT across various samples. This efficiency is largely due to *PLayer*'s smaller parameter set and the use of pruning and quantization techniques, which reduce model complexity. In contrast, the ViT model consumes around 724 MB of memory on average, while *PLayer* requires about 639 MB.

Despite *PLayer*'s efficiency, it achieves slightly lower accuracy than the ViT w/ LayerLink due to the trade-offs from pruning. For instance, in Image O4, *PLayer* achieves an SSIM of 0.82 and a PSNR of 28.77 dB, compared to ViT's SSIM of 0.86 and PSNR of 30.66 dB. However, the non-pruned version of *PLayer* is also built on a ViT-based neural network, and performs comparably to the ViT, as shown in Fig. 7.

When compared to interpolation-based methods, *PLayer*'s performance is superior in terms of reconstruction accuracy. For example, in sample O1, *PLayer* achieves a PSNR of 26.47 dB and an SSIM of 0.75, outperforming the interpolation method, which only achieves a PSNR of 21.39 dB and an SSIM of 0.61. However, interpolation methods are more time-efficient, taking only around 1.75 seconds per layer and requiring significantly less memory (around 155

MB). This trade-off highlights that while interpolation is faster, *PLayer* offers far better reconstruction quality, making it the preferred choice for high-accuracy applications.

A.9 High-Resolution 3D Volume of the Organoid from Confocal Microscopy

We have made the entire 3D volume dataset available on Zenodo: https://zenodo.org/records/12786894. This dataset supports the study titled *'PLayer: A Plug-and-Play Embedded Neural System to Boost Neural Organoid 3D Reconstruction.'* It comprises a total of 539 high-resolution images, each with dimensions of 2048×2048 pixels. The image collection took place roughly over one month. Details are presented in the supplementary appendix *A.1 Specimen Preparation and Experimental Parameters.*

Caption List

Fig. S1 A comprehensive comparative analysis of various image denoising methods. The figure consists of three sections: a visual representation at the top, (a), and (b). Columns in the top section, from left to right, show: 1st, raw unaltered images; 2nd, images with added noise; 3rd, images denoised using threshold adjustment; 4th, images denoised using non-local means; and 5th, images denoised using GAN-based methods. Rows, from top to bottom, represent four different samples. (a) The design of the expert evaluation study and pass rates for each denoising method. (b) NIQE evaluation results. Stained with anti-beta III tubulin primary, and a secondary antibody conjugated to 488 nm.

Tab. S1 Key Parameters Used in the Experimental Setup

Tab. S2 Comparison of Reconstruction Metrics With and Without GPU Warm-Up

Tab. S3 Comparison Between *PLayer* and Other 3D Reconstruction Methods

Tab. S4 Comparison of *PLayer* with Other Methods: Memory and Time Efficiency vs. Reconstruction Accuracy

Fig. S2 Demonstration of Displaying the 3D Volume of the Organoid on an Embedded Device

Supplementary Video One 3D reconstruction

Supplementary Video Two The Comparison of Various Methods

Supplementary Video Three The Deployment