PTNet3D: A 3D High-Resolution Longitudinal Infant Brain MRI Synthesizer Based on Transformers

Xuzhe Zhang, Graduate Student Member, IEEE, Xinzi He, Jia Guo, Nabil Ettehadi, Natalie Aw, David Semanek, Jonathan Posner, Andrew Laine, Life Fellow, IEEE, and Yun Wang

Abstract—An increased interest in longitudinal neurodevelopment during the first few years after birth has emerged in recent years. Noninvasive magnetic resonance imaging (MRI) can provide crucial information about the development of brain structures in the early months of life. Despite the success of MRI collections and analysis for adults, it remains a challenge for researchers to collect high-quality multimodal MRIs from developing infant brains because of their irregular sleep patterns, limited attention, inability to follow instructions to stay still during scanning. In addition, there are limited analytic approaches available. These challenges often lead to a significant reduction of usable MRI scans and pose a problem for modeling neurodevelopmental trajectories. Researchers have explored solving this problem by synthesizing realistic MRIs to replace corrupted ones. Among synthesis methods, the convolutional neural network-based (CNN-based) generative adversarial networks (GANs) have demonstrated promising performance. In this study, we introduced a novel 3D MRI synthesis framework—pyramid transformer network (PTNet3D)—which relies on attention mechanisms through transformer and performer layers. We conducted extensive experiments on high-resolution Developing Human Connectome Project (dHCP) and longitudinal Baby Connectome Project (BCP) datasets. Compared with CNN-based GANs, PTNet3D consistently shows superior synthesis accuracy and superior generalization on two independent, large-scale infant brain MRI datasets. Notably, we demonstrate that PTNet3D synthesized more realistic scans than CNN-based models when the input is from multi-age subjects. Potential applications of PTNet3D include synthesizing corrupted or missing images. By replacing corrupted scans with synthesized ones, we observed significant improvement in infant whole brain segmentation.

Index Terms—Infant brain MRI, MRI synthesis, neural network, performer, transformer.

I. INTRODUCTION

The first two years of life after birth mark rapid periods of postnatal growth and development for the human brain. The brain structures, functions, and neural pathways that develop during this time lay the foundation for the individuals that we will become. An important goal for many studies of early childhood is to identify early biomarkers of later cognitive functions, behaviors, or risks. Structural magnetic resonance imaging (MRI) has become an important non-invasive approach to investigate brain structural changes with high spatial resolution. Over the last decade, researchers have found a modest relationship between brain structure, cognition, and behavior [1–4], suggesting that with improved methodologies, early imaging biomarkers may be useful in predicting later risk.

Compared with adults, infant brains have 1) lower contrast-to-noise ratios due to the relative lack of myelination and shorter scan times [5]; 2) lower spatial resolution due to the smaller overall volume of the brain; and most importantly 3) tissue intensities that change dramatically over the first two years of life. In addition, given infants’ characteristics such as long preparation time (feeding and swaddling to induce sleep), irregular sleep patterns, and the inability to follow instructions to keep still, it is often difficult to collect high-quality multimodal MRI scans for infants [6]. Depending on the research goal of studies and the choice of MRI processing pipelines, researchers often prioritize only one modality / protocol. For example, studies with a focus on newborns will most likely prioritize the acquisitions of T2 weighted (T2w) over T1 weighted (T1w) scans if they have chosen to use the developing human connectome project (dHCP) structural pipeline [7] or T1w over T2w scans if using the Infant FreeSurfer pipeline [8]. Obtaining high-quality structural MRI scans for both modalities (T1w and T2w) may be impractical. Of note, structural MRI processing (tissue/region segmentation, surface reconstruction) is the first procedure for analyzing other MRI modalities, e.g., functional MRI and...
performer layers. We design a novel 3D framework, inheriting MRIs (T1w and T2w scans) using both transformer and
with a simplified self-attention and requires less computation
also introduced and applied to vision tasks [34], [35]; it is
performance in classification and segmentation in terms of
the computer vision domain and demonstrates outstanding
and convolution-free architecture, has been introduced to
providing more accurate whole brain segmentation [9]–[11].
Therefore, novel and robust methodologies, which can syn-
synthesize missing or corrupted infant MRI scans, can be
very helpful for developmental neuroscience and clinical
research [12].

Previous studies have demonstrated that synthesized single
or multimodal MRIs based on existing high-quality scans,
to some extent, improve biomedical imaging processing pro-
cedures, e.g., segmentation and registration. For example,
replacing corrupted fluid-attenuated inversion recovery MRI
scans with its synthesized version based on corresponding
T1w, T2w, and proton density (PD) scans can yield better
segmentation [13]. Similarly, a previous study has shown that
synthesized T1w scans can replace real T1w scans in inter-
modality and cross-subject brain MRI registration, and this
approach improves registration as compared with only using
real PD scans [14].

Prior to the rise of deep learning (DL), registration-based
and intensity-based transformation methods were prevalent in
this domain. Registration-based methods rely on a group atlas
as well as deformable registration to synthesize images with
different contrast [15]. Although registration-based image
synthesis provides promising performance in synthesizing com-
puted tomography and positron emission tomography from
MRI [16], [17], it may not be applicable to infant MRI
synthesis because of 1) lack of an accurate and longitudinal
infant brain MRI atlas; 2) more profound variations in the
infant brain at different ages which may introduce more error
when registering to an atlas. Intensity-based transformation
methods often utilize image analogies, sparse reconstruction,
non-linear regression, as well as neural network to achieve
image synthesis [13], [14], [18]–[22]. However, an earlier
study has concluded that these methods, whether dictionary
reconstruction, random forest regression, or neural network-
based, tend to lose fine details and yield suboptimal results in
synthesis [12].

Given the success of generative adversarial network (GAN)
in image synthesis, translation, and manipulation [23]–[26],
recent studies have attempted to introduce the convolution
neural network (CNN)-based GAN framework into medical
image synthesis and have shown improved performance com-
pared with aforementioned methods [12], [20], [22], [27]–[29].
Recently, the transformer layer, which is a self-attention
and convolution-free architecture, has been introduced to
the computer vision domain and demonstrates outstanding
performance in classification and segmentation in terms of
accuracy and efficiency [30]–[33]. The performer layer is
also introduced and applied to vision tasks [34], [35]; it is
a similar attention-based architecture to the transformer but
with a simplified self-attention and requires less computation
than the transformer.

In this study, we focus on synthesizing infant brain structural
MRIs (T1w and T2w scans) using both transformer and
performer layers. We design a novel 3D framework, inheriting
the U-Net-like as well as multi-resolution pyramid structures
[25], [36], and utilizing performer encoder (PFE), performer
decoder (PFD), and transformer bottleneck to synthesize
high-quality infant MRI. We conduct extensive experiments
based on a large-scale high-resolution infant MRI dataset – the
Developing Human Connectome Project (dHCP) dataset [7] as
well as another longitudinal infant MRI dataset – the Baby
Connectome Project (BCP) dataset [37], and compare our
model’s performance with other methods including pix2pix,
pix2pixHD, and StarGAN [25], [26], [38]. We demonstrate
that our proposed model can synthesize realistic T1w scans
based on T2w scans and vice versa. Compared with CNN-
based models, our framework is superior in various metrics
when validated on the unseen test dataset. More importantly,
our PTNet3D can provide good synthetic results across different
ages while CNN-based models fail on scans from subjects
<= 6 months old. We also have experimentally shown that using
PTNet3D to synthesize corrupted modality (T1w) based
on good-quality T2w improves dual-channel segmentation
using 3D U-Net.

II. RELATED WORKS

A. GAN-Based MRI Synthesis

CNN-based GAN is the most prevalent framework in the
image translation and synthesis domain. It utilizes adversarial
training, which uses the discriminator network’s feedback to
generate images similar to the training data. During the train-
ing, two subnetworks: generator and discriminator, are trained
simultaneously. The generator employs a decoder (original
GAN) or an encoder-decoder (conditional GAN) architecture.
The original GAN was proposed to unconditionally generate
images from latent space noise vector [23]. The discriminator
is a classifier trained by the real and synthesized image. The
discriminator has access to the true label during the training.
The generator is trained using the feedback from the discrimi-
nator and aims to “fool” the discriminator and generate images
that cannot be distinguished from real images. The conditional
GAN has been used in various downstream applications, such
as super-resolution, style transfer, sketch-to-image generation,
and image inpainting [26], [38], [39]. However, the training
of a GAN model can be unstable. Stability is improved in a
conditional GAN model as the input is not random noise but
informative images. Other advancements in conditional GAN
include using a unified generator for multi-domain synthesis
and reducing the data required for training a GAN [38], [40].

Inspired by the previous success of conditional GANs in
natural image translation, early studies have explored their
application in medical image synthesis [12], [27]–[29]. Specif-
ically, studies [27], [12] have used a similar framework in
[26] and [25] such as pix2pix and pix2pixHD, respectively,
for MRI cross-modality synthesis. Dar et al. [12] has explicit-
ly shown that GAN-based methods outperform the
previous intensity-based transformation and neural network-
based methods (i.e., Replica and Multimodal) in MRI synthesis
[20], [22]. Reference [29] has utilized a unified generator
extended from the StarGAN [38]. Both [29], [28] have further
introduced supervision on latent features to improve synthetic
results.
B. Transformer in Computer Vision Tasks

The transformer is an architecture that solely relies on self-attention mechanisms (Fig. 1) and is completely convolution-free [30]. A transformer layer consists of a multi-head self-attention layer and a fully connected feed-forward network (multilayer perceptron). A residual connection and a layer normalization are applied on both components. The transformer model was originally designed for sequence processing and is becoming a popular and fundamental architecture for NLP tasks. Recently, it has been extended to computer vision tasks, such as image classification, image segmentation, image generation, and object detection [31], [35], [41]–[45]. In those applications, the transformer has demonstrated a great potential to achieve or outperform state-of-the-art CNN-based networks, largely because of its self-attention mechanism.

The self-attention mechanism is based on multiplicative attention through the dot-product of weights and values (of dimension \(d_k\)), where the weight matrix is calculated by a compatibility function of the query with the corresponding key (of dimension \(d_k = d_v\)). In practice, queries, keys, and values are packed together into a matrix \(Q, K, V\), respectively. The scaled dot-product attention is calculated using (1). Instead of performing the scaled dot-product attention one time, the original paper proposed a multi-head attention (MHA) module [46], which is more beneficial for capturing global dependencies. As shown in Fig. 1, \(Q, K, V\) are linearly projected \(H\) times, by linear projections \(W_Q, W_K, W_V\). For each head, the single head attention is calculated in parallel based on Eq. (2). The final output of MHA is given by the linear projection \(W_O\) of the concatenation of head attentions as shown in Eq. (3) below.

\[
\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V
\]

\[
head_i(Q, K, V) = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)
\]

\[
\text{MHA}(Q, K, V) = \text{Concat}(head_1, \ldots, head_H)W_O
\]

C. Performer Block for Simplified Attention Mechanism

The original transformer model employed a full-rank softmax attention. Despite the superior performance of the transformer block and its self-attention mechanism, the space and time complexity of computing the full-rank attention matrix quadratically grows with the number of tokens \(L\) (which is proportional to image size in vision tasks). To prove this, we rewrote Eq. (1) by decomposing the softmax into exponential and normalization components, yielding:

\[
\text{Attention}(Q, K, V) = D^{-1}AV, \quad A = \exp\left(\frac{QK^T}{\sqrt{d_k}}\right), \quad D = \text{diag}(A1_L).
\]

In Eq. (4), \(\exp(x)\) is element-wise exponential function, \(D^{-1}\) performs normalization where \(\exp(x)\) is a diagonal matrix with the input vector \(x\) as the diagonal and \(1_L\) is the all-ones vector of length \(L\). By definition, we have \(Q, K, V \in \mathbb{R}^{L \times d_k}\). Therefore, Eq. (1) and (4) require a time complexity of \(O(L^2d_k)\) and a space complexity of \(O(L^2 + d_k)\) because \(A\) has to be calculated and stored firstly (Fig.2). The quadratic complexity limits the application of the original transformer to large input sequence.

The performer block was proposed to approximate the regular full-rank softmax attention by Fast Attention Via positive Orthogonal Random features (FAVOR+) mechanism [34]. The FAVOR+ replace the regular attention matrix \(D^{-1}\) by approximating \(\exp(QK^T/\sqrt{d_k})\) through \(Q'(K')^T\). Therefore, we have:

\[
\hat{\text{Attention}}(Q, K, V) = \hat{D}^{-1}(Q'(K')^TV), \quad \hat{D} = \text{diag}(Q'(K')^T1_L).
\]

In Eq. (5), we replace the non-linear \(\exp(QK^T/\sqrt{d_k})\) with a linear operation \(Q'(K')^T\) so that we can switch the order of multiplication. As indicated by the brackets in Eq. (5) and depicted in Fig. 2, we first calculate the \((K')^TV\). Such an approximation reduces the time and space complexity compared to \(O(Ld_k^2)\) and \(O(Ld_k^2 + d_kL)\), respectively (Fig.2).

The mapping from \(Q, K \in \mathbb{R}^{L \times d_k}\) to \(Q', K' \in \mathbb{R}^{L' \times d_k'}\) is achieved by kernel \(\phi: \mathbb{R}^{d_k} \rightarrow \mathbb{R}^{d_k'}\) so that each row in \(Q'\) and...
\( K' \) is given by \( \phi(q_i^T)^T \) and \( \phi(k_i^T)^T \), respectively (\( q_i \) and \( k_i \) denote rows in \( Q \) and \( K \)). The \( \phi \) is defined by

\[
\phi(x) = \frac{\exp(-\frac{k_i^T x}{\sqrt{m}})}{\sqrt{m}}(\exp(o_{i1}^T x), \cdots, \exp(o_{in}^T x)), \quad m = d_k^i.
\]

In Eq. (6), \( o_{i1}, \cdots, o_{in} \) are fixed, non-learnable, and random orthogonal vectors drawing from an isotropic distribution.

In the original paper, several different configurations for \( \phi \) were proposed and compared. We selected the one with positive and orthogonal feature maps, which provided the most negligible variance and provable accuracy while approximating the softmax kernel. Interested readers may find more detailed mathematical proofs and experiments in [34].

### III. METHODS

#### A. 3D Pyramid Transformer Net (PTNet3D) for MRI Synthesis

In this work, we introduce a novel 3D MRI synthesis framework: 3D Pyramid Transformer Net (PTNet3D). PTNet3D takes high-resolution \( 64 \times 64 \times 64 \) block as input and its architecture consists of transformer/performer layers, skip-connections, and a multi-scale pyramid representation. An overview of our proposed PTNet3D model is depicted in Fig. 3. Specifically, we exploit the transformer block in the bottleneck layer to take advantage of its self-attention mechanism on latent features (Fig. 4c). Considering its quadratic time and space complexity, it cannot be directly applied to the encoding/decoding path because of the higher spatial resolution of feature maps. Therefore, we design the performer-based encoder/decoder, which allows us to approximate full-rank softmax attention based on FAVOR+ in a linear time and space complexity (Fig. 4a&b). We adopt the successful U-shaped structure of U-Net and reshape the output tokens from each layer for skip connection, aiming to preserve fine structures of the brain.

Despite the flexibility of running on a high-resolution 3D image block \( (64^3) \) brought by the performer, it may not be able to capture global dependencies as well as the transformer because the performer is not equipped with full-rank attention [34]. Inspired by some previous studies in traditional computer vision and deep learning tasks [25], [47]–[52], we reintroduce the pyramid representation to our framework: 1) leverage the global information extracted by transformers to compensate for the potential information loss caused by the performer; 2) and alleviate the intensive computation need for high-resolution features. The entire framework operates in two levels: the original resolution and downsampled resolution (1/4 in x, y, and z axes). The original resolution image goes through the performer encoder, transformer bottleneck, and performer decoder. The downsampling image is unfolded directly and is fed into the transformer bottleneck to take advantage of its full-rank attention.

In the following sections, we introduce each component of our proposed PTNet3D model in detail. We introduced the performer-based encoder and decoder in Section A.1, transformer-based bottleneck in Section A.2, pyramid layer in Section A.3, and model details in Section B.

1) Performer Encoder and Decoder: The most significant challenges for applying the original transformer model in vision tasks are computational time and GPU memories when the input has high spatial resolution, such as in the case of brain MRIs. To solve this issue, instead of the transformer, we adapt the performer in our encoding and decoding blocks and name them as the Performer Encoder (PFE) and Performer Decoder (PFD), which are illustrated in Fig. 4. In PFE, for an input 3D tensor with a size of \( N \times C_{in} \times X \times Y \times Z \), we unfold the 3D matrix into a series of tokens using a window of \( n \) by \( n \) and a stride \( S \). The resultant tokens are with size \( N \times \frac{1}{S^3} XYZ \times C_{in} n^3 \) and are fed into the performer block (Fig. 2). The output from the performer is then transposed and reshaped to a size of \( N \times C_{out} \times X \times Y \times Z \) (Fig. 4a). During the encoding, the \( S \) is often set as 2, \( n \) is set as 3. In its counterpart PFD, the input tensors are first upsampled by a factor of \( S \) through trilinear interpolation and are then concatenated with the feature maps from the encoding path along the channel dimension (Fig. 2 and Fig. 4b). The concatenated feature maps are then fed into the performer block following the same unfolding process. In the end, a similar transpose and reshape are performed to form the output. During the decoding, the \( S \) is often set as 2, \( n \) is set as 1 (Fig. 4 panel b).

2) Transformer Bottleneck: In the bottleneck, we employ the original transformer blocks (Fig. 1) as the input feature maps are already of low spatial size. Such a transformer bottleneck (Fig. 4c) allows to better capture any global dependencies across the bottleneck features. The input of transformer bottleneck is a series of tokens formed by an unfolding process similar to the PFE and PFD. The output from the last PFE is unfolded with \( S = 2 \) and \( n = 3 \). After unfolding, a fully connected layer is applied to linearly project the token from \( N \times \frac{1}{S^3} XYZ \times C_{in} \) to \( N \times \frac{1}{S^3} XYZ \times C_{embd} \), where \( C_{in} \) equals the channel number from the last PFE multiplied by \( n^3 \), and \( C_{embd} \) represents the embedding dimension throughout the transformer blocks. The embeddings are then fed into \( M \) transformer blocks, in which \( M \) is set as 9. Before feeding the embedded tokens into the transformer blocks, following the previous studies [30], we add the positional encoding (PE). PE is proposed simultaneously with the transformer and aims to provide some information about the relative or absolute position of the tokens in the sequence. In this work, we utilize the same sine and cosine functions as the previous study suggested [30] to generate the positional encoding according to Eq. (7) and (8), where \( pos \) is the position and \( i \) is the dimension. After going through \( M \) transformer blocks, the output is projected, transposed, and reshaped as usual and is fed into the first PFD as shows in Fig. 2.

\[
P E_{pos, 2i} = \sin(pos/10000^{2i/C_{embd}})
\]

\[
P E_{pos, 2i+1} = \cos(pos/10000^{2i/C_{embd}})
\]

3) Pyramid Layer: The PFE and PFD reduce the computation complexity and allow the PTNet3D to operate on high-resolution 3D blocks. However, performer is theoretically less powerful than transformer in terms of capturing global dependencies as it is approximating the full rank attention of transformer [34]. Therefore, to avoid the potential infor-
The input and output channels (from the previous layers is unfolded with all PFDs set \( n \)). We parallelize the conversion at two distinct resolutions and concatenate them before feeding into the transformer bottleneck. The detailed structures of each component are illustrated in Fig. 4 below. The spatial projection is a fully-connected layer that reduces the channel to output channel number.

### B. Model Details

We provided detailed configurations of the proposed PTNet3D in this section. As illustrated in Fig. 3, PTNet3D has 3 PFEs and 4 PFDs. All PFEs have a window with \( n = 3 \) and except for the first PFE which utilizes an \( s = 1 \), others set \( s \) as 2 to reduce feature maps’ spatial dimension. All PFDs set \( n \) as 1 and except for the last PFD, others firstly upsample the input by a ratio of 2 through trilinear interpolation. Prior to the transformer bottleneck, the output from the previous layers is unfolded with \( s = 2 \) and \( n = 3 \). The input and output channels (\( C_{in} \) and \( C_{out} \)) for PFEs are 1, 16, 32 and 16, 32, 64 respectively. The pyramid layer first unfolds the downsamplers image with \( n = 3 \) and \( s = 1 \), and then linearly projects the unfolded tokens to an embedding dimension of 64. The projected tokens are then fed into 9 transformer blocks. The output is then linearly projected to a dimension of 32 and it is thereby reshaped and concatenated back to the original branch prior to the transformer bottleneck. The concatenated feature maps are unfolded with \( n = 3 \) and \( s = 2 \) and are linearly projected to an embedding dimension of 256. After going through 9 transformer blocks, resultant tokens are projected to a dimension of 96 and are feed into the decoding path. The input and output channels (\( C_{in} \) and \( C_{out} \)) for PFDs are 192, 64, 32, 17 and 32, 16, 16, 16 respectively. The final output of PFDs is linearly projected to 1-channel and reshaped as an output image.

### C. Datasets

1) Developmental Human Connectome Project—dHCP: We used 459 paired T1w and T2w high-resolution infant brain MRI scans from dHCP v1.0.2 data release (0.5 × 0.5 × 0.5 \( mm^3 \)). The structural T1w and T2w scans from dHCP were collected within one month after birth. The average postmenstrual age of infants at scan is 40.65 ± 2.19 weeks. More details about the image acquisition can be found in the original work [7]. Necessary data exclusion based on quality and data preprocessing were performed and details as well as example were provided in the Appendix A. Afterwards, we split the data with a ratio of 7:1:2.

2) Baby Connectome Project—BCP: To further evaluate our model’s performance at different developing ages and different datasets, we used the Baby Connectome Project (BCP) dataset [37]. Image synthesis tasks on longitudinal datasets are more challenging owing to the varying contrast of the developing brain. Therefore, we believe that re-evaluating our PTNet3D on the BCP dataset can further prove its value. BCP adopted a mixed study design containing both longitudinal and cross-sectional time points, ranging from birth to 72 months. The BCP scans have an isotropic resolution of 0.8 × 0.8 × 0.8 \( mm^3 \). We employed similar preprocessing and exclusion to those used in dHCP dataset. We also designed a fair and rigorous data split to ensure each available age is included in the training/validation/testing sets. The details can be found in the Appendix A.

### D. Experiments

1) Model Implementation: We first compared the proposed PTNet3D with other previously state-of-the-art CNN-based models, including pix2pix, pix2pixHD, and StarGAN. The pix2pix model is a U-Net-like model and its generator is an encoder-decoder that progressively downsampled the feature maps by a factor of 2 while increasing the dimension of the feature maps. It does not use any bottleneck layers but uses a skip connection between the encoder and decoder path. The pix2pixHD is an advanced version of pix2pix, which utilizes a residual block as the bottleneck layer’s backbone. It has two variants: the pix2pixHD-global only employs a single generator proposed by [54]; the pix2pixHD-local is additionally equipped with a local enhancer network that works on high-resolution feature maps [25]. The pix2pixHD-global employs 9 residual blocks in its bottleneck, and its local version utilized 3 and 6 residual blocks in high- and low-resolution branches, respectively. The StarGAN is an unified...
GAN model which enables a multi-domain image translation within a single network [38]. We borrowed the implementation for abovementioned models from their public GitHub repositories. Comparisons of computation cost of different models were provided in Appendix B.

2) Training Strategies: The pix2pix series need both adversarial loss ($L_{adv}$, Eq. (9)) and other regularization terms to stabilize the training. Therefore, a generator $G$ and discriminator $D$ are used during the training. We term $X$ as the input source image, and $Y$ as the target image. For the pix2pix, it utilizes the $L_{adv}$ and $L_1$ reconstruction loss (mean absolute error, Eq. (10), with a weight of 100) as the loss function described in Eq. (11); for the pix2pixHD, instead of incorporating an $L_1$ reconstruction loss, it incorporates the $L_1$ loss in the feature-level ($L_{feat}$, Eq. (12), with a weight of 10) with the $L_{adv}$ Eq. (13). It should be noted that, for Eq. (12), $D_i$ indicates the output from the $i$-th layer of the discriminator $D$. Specifically, $i$-th layer is the layer prior to the final patch-level class prediction and $i$ is set from 1 to 3.

$$L_{adv} = \mathbb{E}_{X,Y}[\log(D(X,Y)) + \log(1 - D(X,G(X)))] \quad (9)$$

$$L_{MAE} = \|G(x) - Y\|_1 \quad (10)$$

$$L_p = L_{adv} + 100 \times L_{MAE} \quad (11)$$

$$L_{feat} = \sum_{i=1}^{3} \|D_i(X,Y) - D_i(X,G(X))\|_1 \quad (12)$$

$$L_{PH} = L_{adv} + 10 \times L_{feat} \quad (13)$$

Unlike the pix2pix series, the StarGAN applies different adversarial loss and regularization term. StarGAN utilizes two different discriminators: $D_{src}$ to distinguish real and fake images, and $D_{cls}$ for domain class classification. Denoting the input source image $X$, source domain class $c_s$, target image $Y$, source domain class $c_y$, and generated image $G(X,c_y)$, the StarGAN is trained using Eq. (17):

$$L_{advStar} = \mathbb{E}_X[\log(D_{src}(X))] + \mathbb{E}_{X,c_s}[\log(1 - D_{src}(X,G(X),c_s))] \quad (14)$$

$$L_{cls} = \mathbb{E}_{X,c_s}[\log(D_{cls}(c_s|X))] + \mathbb{E}_{X,c_y}[\log(D_{cls}(c_y|G(X,c_y)))] \quad (15)$$

$$L_{cycle} = \|G(G(X,c_y),c_s) - X\|_1 \quad (16)$$

$$L_{Star} = L_{advStar} + L_{cls} + 10 \times L_{cycle} \quad (17)$$

PTNet3D also utilizes a hybrid loss function. It uses the same adversarial loss shown in Eq. (9) with a 3D patch-level discriminator similar to [25]. It uses a $L_2$ norm on pixel-wise reconstruction loss and feature-level perceptual loss. In addition to Eq. (9-11), we also use a 3D ResNet-18 model pre-trained on Kinetics-400 dataset as the external discriminator.
The pretrained ResNet-18 is frozen during training and only provides supervision in feature-level (Eq. (18)). We term all the feature-level regularizations as perceptual loss $L_{perc}$, which is defined as Eq. (19). And the loss function for PTNet3D is defined as Eq. (21)

\[
L_{ResFeat} = \sum_{i=1}^{4} \|Res_i(Y) - Res_i(G(X))\|_1 \tag{18}
\]

\[
L_{Perc} = L_{feat} + L_{ResFeat} \tag{19}
\]

\[
L_{MSE} = \|G(x) - Y\|_2 \tag{20}
\]

\[
L_{PTNet} = L_{ado} + 10 \ast L_{perc} + 100 \ast L_{MSE} \tag{21}
\]

Five models were separately trained for T1w-to-T2w and T2w-to-T1w conversions. For T1w-to-T2w conversion, X was T1w scan and Y was the corresponding T2w scan, and vice versa. We used the default training strategies for pix2pix series and StarGAN as explained and demonstrated in the previous studies [25], [26], [38]. Detail training configurations can be found in the Appendix B.

3) Evaluation: To compare models’ performance on T1w-to-T2w and T2w-to-T1w synthesis tasks, four different metrics: the structural similarity index (SSIM), peak signal-to-noise ratio (PSNR), mean absolute error (MAE), and Fréchet Inception Distance (FID) were calculated on the test dataset [56]–[58]. We employed these four metrics to evaluate synthetic results from different perspectives. More detailed introductions to the metrics can be found in the Appendix D.

After directly comparing the models’ performance from the quantitative metrics, we evaluate each model based on the validity of its synthetic results indirectly. Specifically, using the same distribution introduced in Section C.1, we trained a 3D UNet which takes concatenated T1w and T2w blocks as inputs [59] to segment the entire brain into 87 brain regions based on the labels provided by dHCP studies. We compared the segmentation maps of real T1w + real T2w with those of real T1w + synthetic T2w. We hypothesized that the more valid synthetic scans are, the closer segmentation results are compared to those of real scans. Three different metrics were utilized in comparison: Dice score (DSC), average surface distance, and 95% Hausdorff distance (HD).

4) Ablation Experiments: To further study the PTNet3D where transformer and performer layers are first introduced to replace convolutional layers completely in MRI synthesis, we conducted ablation studies on loss functions, pyramid layers, and the dimensionality of the input image. We compared the PTNet3D performance when it was trained by MES loss (Eq. (20)) solely, by adversarial loss and MSE (with a weight of 100), and by a combination of adversarial loss, perceptual loss, and MSE loss together (Eq. (21)). Furthermore, we compared model performance with different pyramid layers: 0, 1, and 2. For the case of two layers, the second pyramid layer runs on a resolution of $8^3$, and the output is concatenated back to the main branch prior to the linear projection in the transformer bottleneck. We also evaluated PTNet3D on different input dimensions. As both transformer and performer layers take tokens as input, in this ablation experiment, we formed tokens through unfolding the 3D block ($64^3$) and 2D image ($224 \times 256$). The detailed results can be found in Section IV.

IV. RESULTS

A. Synthesis Results on dHCP Dataset

1) Visual Comparisons: We first visualized synthesized scans and calculated their absolute error maps between the original scan and synthesized one from each model (Fig. 5) for T1w-to-T2w synthesis. Absolute error is calculated between normalized ground truth and converted scans, ranging from [0,1] – lower values (white color) indicate minor difference. From Fig 5., we found that our model produces less extensive absolute error than the CNN-based models in general. Especially, the arrows indicated the region where our model produced a more realistic synthesis compared to other methods.

2) Quantitative Results: We quantitatively compared the results of our proposed PTNet3D model to results of other CNN-based models using the same high-resolution infant brain MRI dataset. The mean and standard deviation of SSIM, PSNR, MAE, and FID were calculated from the test dataset and were reported in Table 1. PTNet3D outperforms other models in almost all metrics, except for the FID in T2w-to-T1w synthesis. In the T1w-to-T2w synthesis scenario, compared to the pix2pix model, our PTNet3D achieves a 2.67% increase in SSIM, 4.45 dB rise in PSNR, and a 32% reduction in MAE. Similarly, for T2w-to-T1w synthesis, PTNet3D delivers a 1.5% increase in SSIM, 1.88 dB rise in PSNR, and an 18% decrease in MAE, when compared to the pix2pix model. The superior performance of PTNet3D was consistently observed for both T1w-to-T2w and T2w-to-T1w tasks. We noticed that pix2pixHD-local, StarGAN, and our
PTNet3D have very close FID in T2w-to-T1w synthesis. However, our PTNet3D has the best performance in SSIM, pSNR, and MAE, representing higher structural similarities.

B. Synthesis Results on Longitudinal BCP Dataset

The detailed quantitative results from all 5 models were listed in Table 2. We reported the average SSIM, pSNR, MAE, and FID from 46 testing scans, which were acquired at different ages. We noticed that PTNet3D continues to outperform all other CNN-based counterparts on the longitudinal BCP dataset. In addition, we noticed that StarGAN’s performance significantly drops compared to the performance on dHCP dataset. Despite its relatively high SSIM, pSNR, and MAE on T2w-to-T1w synthesis task, StarGAN performed poorly in FID and failed on T1w-to-T2w synthesis tasks.

As introduced before, the longitudinal infant brain MRI poses a challenge in image synthesis because of the varying contrasts in rapidly developing brains. To investigate the possible impact of longitudinal data on synthetic quality and accuracy, we divided the test subjects into four different age groups: 0-6 months, 7-12 months, 13-24 months, and >24 months. A comparison of the model’s performance at each age group was conducted (Fig. 6).

Fig. 6 shows the distributions of SSIM, MAE, and FID for each model and age group separately. The pSNR is based on intensity difference as MAE does, so we provided it in Appendix C. In the left panel are the longitudinal results of T1w-to-T2w synthesis, while the right panel illustrates the T2w-to-T1w synthesis results. Our PTNet3D model consistently yielded higher SSIM, lower MAE and lower FID than CNN-based models.

Notably, the CNN-based models could not tackle the challenging longitudinal synthesis, especially when the input scans were from the 0-6 months group. An example was provided in Fig. 7. As a comparison, we provided the results from pix2pixHD-Local – the best CNN-based model (Table 2 and Fig. 6). The synthesized scan from PTNet3D, which is shown in the top middle, is less noisy, has fewer artifacts, and remarkably retains fine structural details. The superior synthetic quality of PTNet3D is clearer in the zoomed region, where pix2pixHD-Local loses detail of the cerebral cortex, that is preserved in the PTNet3D model.
C. Segmentation of 87 Brain Regions Using Synthesized Scans

Furthermore, we evaluated the validity of synthesized scans on real-world application – infant whole brain segmentation. We used the same dHCP dataset and followed the same data partition as Section A. To perform the segmentation task, we built a dual-channel 3D UNet model (detailed configurations in the Appendix E). After training on 291 pairs of multimodal MRI scans (T1w and T2w) and labels, we segmented six different sets of test scans, which contained real T1w and real T2w or real T1w + synthesized T2w. Each above synthesizer generated a unique pair of real T1w and synthesized T2w.

Segmentation accuracy using synthesized scans was evaluated by three metrics, including the Dice, Average Surface Distance (ASD), and 95% Hausdorff Distance (HD95), listed in Table 3. We noticed that PTNet3D outperforms other methods in all metrics, providing the closest performance to using real scans. We can conclude that PTNet3D synthesized more realistic scans based on the segmentation results.

Compared to pix2pixHD-Local, 87 out of 87 segmented regions from the derivative scans of PTNet3D are closer to those from real scans (significantly higher Dice at an FDR adjusted p value of 0.05, see Appendix F).

We further provided visualization of segmentation results in Fig. 8. For simplification, only segmentation maps from real scans, PTNet3D, and pix2pixHD-Local were provided. Compared to those using the real scans, PTNet3D and pix2pixHD-Local both performed well on segmenting white matter and gray matter. However, using the synthesized scans from PTNet3D generated more reliable segmentation, especially for small-to-medium structure regions. As highlighted in Fig. 8, using pix2pixHD-Local as the synthesizer led to an inaccurate segmentation of the brain stem and the ventral lateral nucleus within thalamus (the top row), a false negative segmentation of thalamus, and ambiguous boundary between parietal and occipital frontal lobe (middle row and bottom row).

D. Ablation Studies

To evaluate the contribution of each component in the proposed PTNet3D, we conducted ablation studies by comparing the synthetic results of PTNet3D under different configurations. Specifically, we investigated the influence of each component of the loss function; we justified the effectiveness of the proposed pyramid structure; we compared the proposed PTNet3D to its 2D variant. The detailed experiments and results can be found below.
1) Contribution of Adversarial and Perceptual Losses: We first conducted an ablation study on the choice of loss functions. Three different configurations were compared, and results were provided in Table 4. A previous study suggested that adversarial training (adversarial and perceptual losses) can not only improve image quality in terms of high-frequency components but also will improve intensity-based metrics such as SSIM [28]. However, we noticed that our PTNet3D could be directly and efficiently trained with MSE and achieve very high performance in intensity-based metrics. Nonetheless, only using MSE will lead to resultant scans, which are over-smooth and lack high-frequency components. Incorporating adversarial loss and perceptual loss will yield the optimal FID score and best visual quality without scarifying intensity restoration accuracy too much.

2) Contribution of Pyramid Layer: As introduced before, we assume utilizing transformers directly at downsampled image will take advantage of the full-rank attention to avoid potential information loss caused by the performer. We conducted an ablation experiment on removing the pyramid layer, adding one or two pyramid layers to justify the effectiveness of the proposed pyramid layer. We only stacked pyramid layers up to two because of resolution limitation. Taking a $64^3$ block as the input, the lowest spatial resolution of main branch is $8^3$, while the third pyramid layer will run at a resolution of $4^3$. The results were concluded in Table 5. Compared to removing pyramid layers, adding one pyramid layer running at a $16^3$ resolution improved all metrics. This improvement was consistently observed when two pyramid layers were utilized. We believe these results suggest the pyramid layer benefits the PTNet3D by directly applying transformers at different scales, especially with respect to capturing global dependencies. After downsampling the image from $64^3$ to $8^3$, the local structural details are barely preserved in the input of the second pyramid layer, but it still improves the synthesis performance. The pyramid layer might provide such an improvement as long as it runs at a reasonable resolution.

3) 2D and 3D Comparison: Lastly, we conducted an ablation study between the proposed PTNet3D and our previous 2D variant [60]. The results were listed below. The PTNet3D taking 3D blocks as input outperformed the 2D variant. This finding is in alignment with several previous studies using CNN-based model [28], [29]. Similar to CNN-based model, PTNet3D also benefits from the 3D information within the input.

E. Visual Comparisons of Models’ Feature Maps

We further visualized and compared internal feature maps from pix2pixHD-global and PTNets. Fig. 9 a) panel showed feature maps from the decoder path of PTNet-2D with a matrix size of $56 \times 64$; b) panel showed stitched feature maps from the PTNet3D decoder with a matrix size of $56 \times 64$ (the original size is $16 \times 16$, and we stitched feature maps from 16 neighboring blocks to form this one); and c) panel displayed feature maps from pix2pixHD-global with a matrix size of $56 \times 64$. It is remarkable that the transformer-based network generated more structured activations given the same input. The transformer-based PTNet3D model is able to detect very fine structural details (e.g., edges, textures) and significantly enhances the feature richness compared with the CNN-based pix2pixHD. We also noticed that the PTNet3D almost provided rich and meaningful activation at each channel while the pix2pixHD generated zero-value activation at some channels. We speculate that such a remarkable difference in feature maps may account for the improved quality of synthesized MRI scans.

F. Application—Prevent Data Exclusion in Downstream Tasks by Synthesizing Corrupted Scans

To showcase the potential utility of our PTNet3D, we first quantitatively demonstrated how corrupted scans could affect downstream processing and analysis. We take the segmentation

### Table IV

| Loss Functions | SSIM (%) | pSNR (dB) | MAE (10^-3) | FID |
|----------------|----------|-----------|-------------|-----|
| MSE            | 97.77±0.56 | 32.45±1.11 | 8.00±1.73  | 21.79±0.45 |
| MSE+Adv        | 97.25±0.65 | 31.48±1.00 | 7.85±1.74  | 18.63±2.57 |
| MSE+Adv=Perc   | 97.25±0.66 | 31.46±1.06 | 8.86±1.83  | 17.51±2.29 |

Adv, Adversarial Loss; Perc, Perceptual Loss.

### Table V

| Pyramid Layers | SSIM (%) | pSNR (dB) | MAE (10^-3) | FID |
|----------------|----------|-----------|-------------|-----|
| None           | 97.11±0.67 | 31.22±1.04 | 9.14±1.87  | 17.78±2.25 |
| 1              | 97.25±0.66 | 31.46±1.06 | 8.86±1.83  | 17.51±2.29 |
| 2              | 97.37±0.63 | 31.64±1.06 | 8.53±1.76  | 17.16±2.20 |

### Table VI

| Model        | SSIM (%) | pSNR (dB) | MAE (10^-3) | FID |
|--------------|----------|-----------|-------------|-----|
| 2D           | 96.69±0.78 | 30.52±1.04 | 9.85±1.97  | 18.50±2.47 |
| 3D           | 97.25±0.66 | 31.46±1.06 | 8.86±1.83  | 17.51±2.29 |

Fig. 9. Feature maps from the decoding path of different models. a) PTNet_2D, b) PTNet3D, c) pix2pixHD-Global. *The checkboard artifact in panel b. is caused by stitching and doesn’t indicate failures.*
TABLE VII
SEGMENTATION RESULTS UNDER DIFFERENT RATIOS OF CORRUPTED OR SYNTHESIZED SCANS

| Ratio | PTNet3D | Mean Dice | Mean ASD (mm) | Mean HD95 (mm) |
|-------|---------|-----------|---------------|----------------|
| 0% (Upper Bound) | ✗ | 0.92±0.01 | 0.13±0.03 | 0.72±0.13 |
| 5% | ✗ | 0.91±0.05 | 0.19±0.25 | 0.96±1.34 |
| | ✓ | 0.92±0.02 | 0.13±0.04 | 0.73±0.15 |
| 10% | ✗ | 0.90±0.07 | 0.23±0.52 | 1.15±2.38 |
| | ✓ | 0.92±0.02 | 0.14±0.04 | 0.74±0.16 |
| 15% | ✗ | 0.88±0.09 | 0.32±0.68 | 1.56±3.07 |
| | ✓ | 0.91±0.02 | 0.14±0.04 | 0.75±0.15 |
| 20% | ✗ | 0.87±0.11 | 0.30±0.47 | 1.50±2.08 |
| | ✓ | 0.91±0.02 | 0.15±0.04 | 0.75±0.16 |
| 25% | ✗ | 0.87±0.11 | 0.41±0.70 | 2.12±3.60 |
| | ✓ | 0.91±0.03 | 0.15±0.05 | 0.76±0.15 |

Italic indicates results from datasets containing different proportions of corrupted scans. Bold indicates results from datasets with PTNet-generated synthetic scans replacing the corrupted scans. Average results from 87 brain regions of 83 volumes were reported.

as an example, which is an important step towards both volume-based and surface-based analysis of brain regions.

Using the test dataset introduced in Section IV.C, we first randomly sampled 5%, 10%, 15%, 20%, and 25% of good quality T2 scans in the original test dataset. Next, we added random motion artifacts to these scans. Then we investigated how these corrupted scans might affect segmentation results and how synthesized ones using PTNet3D could attenuate the problem.

Random motion artifacts were introduced using approaches developed in previous studies [60], [61]. Details and examples of motion artifact injection were provided in Appendix G. These datasets which contain different ratios of corrupted scans were fed into the pre-trained segmentation model described in Section IV.C. The average dice score, mean surface distance, and mean 95% HD across 87 regions were reported in Table 7.

Table 7 shows that the segmentation performance will continue to deteriorate as more corrupted scans are included. Average Dice for 87 brain regions fell 5% when the ratio of corrupted scans rose to 25%. In the same scenario, we also noticed a 215% and 194% increase in the ASD and HD95, respectively. Contrary to this, when replacing corrupted scans with synthesized scans using our PTNet3D, the average Dice only dropped by 1%. Meanwhile, mean ASD and HD95 only increased by 15% and 6%, respectively.

Fig. 10 shows an example of how PTNet3D can improve segmentation. From the error map, it is evident that the corrupted T2w scan impairs the segmentation, especially at the boundary between white matter and gray matter. These errors could be further propagated if surface-based analysis is performed on such masks. These corrupted scans are typically excluded to prevent inaccurate surface analysis, which will reduce the data availability. From Table 7, results of segmenting synthesized scans were significantly better than those of segmenting corrupted scans. It was also very close to segmenting good-quality scans. Therefore, we conclude that our proposed PTNet3D can be used to synthesize MRI scans to surrogate the corrupted ones. This will avoid data exclusion in downstream tasks and allow for larger sample size in infant MRI brain studies.

V. DISCUSSIONS

In this work, we introduced a novel 3D MRI synthesis framework – PTNet3D – specifically for infants and toddlers. The convolution-free PTNet3D first introduces performer and transformer together into brain MRI synthesis task. We compared its performance with other state-of-the-art CNN-based models on two independent and large-scale infant MRI datasets. The results of extensive experiments show that PTNet3D consistently outperforms other models. More importantly, PTNet3D is able to tackle the challenging task of synthesizing longitudinal infant scans, which have different tissue contrast and appearances at each age. It performs consistently well on the longitudinal BCP dataset. In contrast, other CNN-based models fail to synthesize good quality scans for infants under 6 months. We also found that our PTNet3D could extract more structured and richer features than CNN-based models. This may partially explain its superior performance.

Our PTNet3D solves the intensive computation requirement for the attention-based transformer block by incorporating a novel performer block that approximates the full-rank attention mechanism by FAVOR+. This allows the processing of high-resolution image blocks. PTNet3D is also equipped with a pyramid layer. This allows it to avoid information loss caused by performer blocks by taking advantage of the pyramid representation of images, which has been proven to benefit vision tasks in the past. To provide insights into the early stages of application of transformer in MRI synthesis tasks, we conducted extensive ablation studies on the loss functions, number of pyramid layers, as well as 2D/3D comparisons. We found that unlike CNN-based models introduced in previous studies, PTNet3D could achieve exceptional performance in intensity-based metrics by using only L2 loss. After incorporating adversarial loss and perceptual loss – two important components in CNN-based GAN model training, PTNet3D avoids over-smooth results and yields more realistic scans with rich high-frequency components. In addition, in agreement with previous studies, we found that pyramid layer and 3D input effectively improve the performance of PTNet3D.

In addition to the direct comparison using quantitative metrics, we conducted indirect comparison among PTNet3D and other methods by comparing segmentation results on synthesized scans. The results indicate that synthesized scans from PTNet3D are more reliable than CNN-based models.

Moreover, we demonstrated an important use for PTNet3D – avoiding data exclusion by replacing corrupted scans with synthesized ones. PTNet3D will offer developmental researchers a valuable tool to investigate the developing brain. Longitudinal MRI studies that investigate brain development from infancy to toddlerhood may suffer from substantial data loss, with
Fig. 10. Using PTNet3D in real world application. Two concatenated inputs (good-quality T1w + corrupted T2w, good-quality T1w + synthesized T2w) are fed into a dual-channel 3D UNet. The bottom panel visualizes the segmentation maps from different inputs. From left to right: segmentation from corrupted scans, ground truth released by dHCP study, and segmentation from synthesized scans.

Fig. 11. Examples of data inclusion and exclusion. Scans (a-d) were included during model development, and (e-h) were excluded. Scan (a) has the best quality while (b) and (c) are slightly worse than a. And scan (d) has minor artifacts (circled region) and was not excluded since it is acceptable. Scans (e-h) were excluded because of their unacceptable qualities.

APPENDIX

A. Data Preprocessing and Exclusion

To remove outliers with extremely high intensities, each volume was normalized to [0,1] by its minimum intensity value and 99.95 percentile maximum intensity value. The dHCP scans have a matrix size of $290 \times 290 \times 203$ and was cropped to $224 \times 256 \times 202$ by removing background. The original matrix size of BCP is $208 \times 300 \times 213$. The BCP scans underwent the same preprocessing as dHCP scans did, including cropping, padding, and normalization. The resultant matrix size is $224 \times 256 \times 210$.

The quality of skull-stripping and co-registration was assessed by a senior MRI technician. We excluded paired scans with motion/scanner artifacts, poor skull stripping, or co-registration problem. In total, we included 416 paired T1w and T2w scans from dHCP and 231 paired T1w and T2w scans from age of 2 months to 34 months from BCP. We provide some example scans in Fig. 11.

For the BCP dataset, 156 scans were used for training; 29 scans were used for validation, and the remaining 46 scans were used for testing by partially random sampling. The data split was randomly generated based on the following rules: 1) The ratio of training, validation, and testing datasets is 7:1:2; 2) All scans were randomly sampled to three partitions based on the ratio; 3) At least one scan per age was guaranteed to be distributed to each partition; 4) No significant distribution difference across three groups. The exact age distribution can be found in Fig. 12.

In addition, to make sure the age distribution is balanced in the training set, we applied data augmentation for the training scans while the distribution of validation and testing scans remain unchanged. For the data augmentation, a 3D rotation along the x-y axis was applied. Given the maximum frequency of scans at 24 months of age ($n = 25$), for other ages with $M$ scans, we randomly sampled one scan from that age and applied rotation by $\frac{2\pi}{n-M} \times X$, $X = 1, 2, \ldots, n - M$. After
TABLE VIII  
ALL MODELS’ DETAILS

| Models       | Input Size | Maximum Batch Size | Parameters Number (M) | Training Time (s/step) |
|--------------|------------|--------------------|-----------------------|------------------------|
| pix2pix      | 224x256    | 347                | 16.65                 | 0.01 1.34              |
| pix2pixHD    | 224x256    | 60                 | 182.43                | 0.06 1.08              |
| pix2pixHD -Local | 224x256 | 32                 | 504.20                | 0.10 0.74              |
| StarGAN      | 224x256    | 42                 | 44.77                 | 0.04 1.04              |
| PTNet3D      | 64x64x64   | 4                  | 8.01                  | 0.21 1.06              |

Min Batch indicated minimum batch size (1) was used. Max Batch indicated maximum batch size was used. The number of workers was set to 0 to calculate the training time. All comparisons were conducted on an NVIDIA Titan RTX GPU with 24 GB memory.

data augmentation, each age had 25 scans in the training set, resulting in 625 scans.

B. Training Configurations

We provided comparisons of computational cost in Table 8. For a fair comparison, all models were first trained at a fixed learning rate $2e^{-4}$ for 5 epochs then another 5 epochs at a linearly decreasing learning rate (to 0) on the dHCP dataset. Similarly, on the BCP dataset, all models were trained for 3 epochs at a fixed learning rate and another 3 epochs at a linearly decreasing learning rate because there were more scans. After training, the model with the highest Structural Similarity Index Measure (SSIM) on the validation dataset was selected for comparison on the testing dataset. All experiments were trained on an GeForce RTX 2080 TI with 11GB memory and an NVIDIA Titan RTX GPU with 24 GB memory. The entire framework was implemented in PyTorch, and publically available on GitHub (https://github.com/XuzheZ/PTNet3D).

C. Synthetic Results at Different Time Points on BCP Dataset

We provided the distributions of pSNR for each model and age group separately in Fig. 13. Our PTNet3D consistently shows the best performance.

D. Introduction to the Evaluation Metrics

We employed four metrics to evaluate synthetic results from different perspectives. The MAE and pSNR assess the image quality from the accuracy of tissue intensity restoration between the synthetic and real scans. SSIM is correlated to the quality perception of the human visual system in the perspective of distortion and degradation of structural information. Although MAE, pSNR, and SSIM are widely used in previous image synthesis tasks, they might ignore the high-frequency components that also play a critical role in visual perception. FID evaluates the performance of GANs in terms of visual similarity and is more consistent with human judgment than the Inception Score. The detailed mathematical justification can be found in the original studies [56]–[58].

In our study, we normalized the ground truth and synthesized volumes to the same intensity range [0,1]. SSIM and PSNR were calculated using Eq. (22) and Eq. (23), to be
noted that, in Eq. (22) and Eq. (23), $X$ and $Y$ represented volume instead of slice; $\mu$ indicated mean intensity; $\sigma$ was standard deviation; $\sigma_{XY}$ was the covariance between $X$ and $Y$; positive constant $C$ was used to prevent division by zero. The MAE was also calculated on a volume basis. FID can only be calculated on 2D slices, so we calculated subject-wise FID by taking the average FID scores from three orientations (i.e., sagittal, coronal, and axial).

$$PSNR (X, Y) = 10 \log_{10} \left( \frac{1}{MSE (X, Y)} \right)$$ (22)$$

$$SSIM (X, Y) = \frac{2\mu_X\mu_Y + C_1}{\mu_X^2 + \mu_Y^2 + C_1} \frac{2\sigma_X\sigma_Y + C_2}{\sigma_X^2 + \sigma_Y^2 + C_2} \frac{\sigma_{XY} + C_3}{\sigma_X\sigma_Y + C_3}$$ (23)

E. Detail Configurations of 3D Dual-Channel UNet for Whole Brain Segmentation

We used vanilla 3D UNet with the following parameters:
1. Encoding Conv block: \{GroupNormalization (num_groups = 8), Conv3D layer (input_channel, output_channel), ReLU\} + \{GroupNormalization (num_groups = 8), Conv3D layer (output_channel, output_channel), ReLU\}
2. Layers of Encoding Conv block: 4
3. Pooling layer used in the encoding path: 3D max pooling with stride = 2, kernel size = 2
4. Input channels for each encoding block: 2, 32, 64, 128
5. Output channels for each encoding block: 32, 64, 128, 256
6. Bottleneck: N/A
7. Decoding Conv block: \{GroupNormalization (num_groups = 8), Conv3D layer (input_channel, output_channel), ReLU\} + \{GroupNormalization (num_groups = 8), Conv3D layer (output_channel, output_channel), ReLU\}
8. Layers of Encoding Conv block: 3
9. Upsampling layer: 3D nearest interpolation upsampling
10. Input channels for each decoding block: 384, 192, 96
11. Output channels for each decoding block: 128, 64, 32
12. Final Conv block: Conv3D (32, 8)
13. Input 3D patch size: $128 \times 128 \times 128$
F. Dice of 87 Segmented Regions on Synthesized Scans

We reported regions with significant higher Dice score while using PTNet3D to synthesize the T2w MRI rather than using pix2pixHD-Local in Fig. 14. Reader can find the corresponding brain region of each tag number in the publication of dHCP study [7].

G. Generation of Motion-Artifacted Scans

Following the previous studies [60], [61], we generated random motion-corrupted scans with these parameters: random rotation (−7.5, 7.5) degrees, random translation (−7.5, 7.5) mm, and the number of transformations: 6. The randomization follows a uniform distribution. We randomly chose 4 subjects and provided visualizations of injecting motion artifacts in three orientations (i.e., sagittal, coronal, and axial) (Fig. 15).

REFERENCES

[1] S. J. Short et al., “Associations between white matter microstructure and infants’ working memory,” *NeuroImage*, vol. 64, pp. 156–166, Jan. 2013, doi: 10.1016/j.neuroimage.2012.09.021.

[2] M. N. Spann, R. Bansal, T. S. Rosen, and B. S. Peterson, “ Morphological features of the neonatal brain support development of subsequent cognitive, language, and motor abilities,” *Hum. Brain Mapping*, vol. 35, no. 9, pp. 4459–4474, Sep. 2014, doi: 10.1002/hbm.22487.

[3] M. Fati et al., “Multimodal imaging of the self-regulating developing brain,” *Proc. Nat. Acad. Sci. USA*, vol. 109, no. 48, pp. 19620–19625, Nov. 2012, doi: 10.1073/pnas.1208243109.

[4] J. O’Muircheartaigh et al., “White matter development and early cognition in babies and toddlers,” *Hum. Brain Mapping*, vol. 35, no. 9, pp. 4475–4487, Sep. 2014, doi: 10.1002/hbm.22488.

[5] M. Prastawa, J. H. Gilmore, W. Lin, and G. Gerig, “Automatic segmentation of MR images of the developing newborn brain,” *Med. Image Anal.*, vol. 9, no. 5, pp. 457–466, Oct. 2005, doi: 10.1016/j.media.2005.05.007.

[6] J. H. Gilmore, R. C. Knickmeyer, and W. Gao, “Imaging structural and functional brain development in early childhood,” *Nature Rev. Neurosci.*, vol. 19, no. 3, pp. 123–137, 2018, doi: 10.1038/NRN.2018.1.

[7] A. Makropoulos et al., “The developing human connectome project: A minimal processing pipeline for neonatal cortical surface reconstruction,” *NeuroImage*, vol. 173, pp. 88–112, Jun. 2018, doi: 10.1016/j.neuroimage.2018.01.054.

[8] L. Zöllei, J. E. Iglesias, Y. Ou, P. E. Grant, and B. Fischl, “Infant FreeSurfer: An automated segmentation and surface extraction pipeline for T1-weighted neuroimaging data of infants 0–2 years,” *NeuroImage*, vol. 218, Sep. 2020, Art. no. 116946, doi:10.1016/j.neuroimage.2020.116946.

[9] L. Wang et al., “Volume-based analysis of 6-month-old infant brain MRI for autism biomarker identification and early diagnosis,” in *Proc. Int. Conf. Med. Image Comput. Assist. Intervent. (MICCAI)*, vol. 11072, Sep. 2018, pp. 411–419, doi: 10.1007/978-3-030-00931-1_47.

[10] J. Dolz, C. Desrosiers, L. Wang, J. Yuan, D. Shen, and I. B. Ayed, “Deep CNN ensembles and suggestive annotations for infant brain MRI segmentation,” *Computerized Med. Imag. Graph.*, vol. 79, Jan. 2020, Art. no. 101660, doi: 10.1016/j.compmedimag.2019.101660.

[11] Y. Sun et al., “Multi-site infant brain segmentation algorithms: The iSeg-2019 challenge,” *IEEE Trans. Med. Imag.*, vol. 40, no. 5, pp. 1363–1376, May 2021, doi: 10.1109/TMI.2021.3055428.

[12] S. H. Dar, M. Yurt, L. Karacan, A. Erdem, E. Erdem, and T. Çukur, “Image synthesis in multi-contrast MRI with conditional generative adversarial networks,” *IEEE Trans. Med. Imag.*, vol. 38, no. 10, pp. 2375–2388, Oct. 2019, doi: 10.1109/TMI.2019.2901750.

[13] A. Jog, A. Carass, D. L. Pham, and J. L. Prince, “Random forest flair reconstruction from T1j, T2j, and PD-weighted MRI,” in *Proc. IEEE Int. Symp. Biomed. Imag.*, Apr. 2014, pp. 1079–1082, doi: 10.1109/ISBI.2014.6868061.

[14] J. E. Iglesias, E. Konukoglu, D. Zikic, B. Glocker, K. Van Leemput, and B. Fischl, “Is synthesizing MRI contrast useful for inter-modality analysis?” in *Proc. Int. Conf. Med. Image Comput. Assist. Intervent.*, vol. 16, no. 1, 2013, pp. 631–638, doi: 10.1007/978-3-642-40811-3_79.
Z. Wang, Z. Cui, and Y. Zhu, “Multi-modal medical image fusion by E. H. Adelson, C. H. Anderson, J. R. Bergen, P. J. Burt, and E. Denton, S. Chintala, A. Szlam, and R. Fergus, “Deep generative image translation using cycle-consistent adversarial networks,” in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Oct. 2017, pp. 2242–2251, doi: 10.1109/ICCV.2017.244.

T. R. Shaham, T. Dekel, and T. Michaeli, “SinGAN: Learning a generative model from a single natural image,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2019, pp. 4570–4580.

Z. Liu et al., “Swin transformer: Hierarchical vision transformer using shifted Windows,” 2021, arXiv:2103.14030.

D. A. Hudson and C. L. Zitnick, “Generative adversarial transformers,” 2021, arXiv:2103.01209.

N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, “End-to-end object detection with transformers,” in Proc. Eur. Conf. Comput. Vis., Cham, Switzerland: Springer, 2020, pp. 213–229.

P. Esser, R. Rombach, and B. Ommer, “Taming transformers for high-resolution image synthesis,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2021, pp. 12873–12883.

Y. Jiang, S. Chang, and Z. Wang, “TransGAN: Two pure transformers can make one strong GAN, and that can scale up,” in Proc. Adv. Neural Inf. Process. Syst., vol. 34, 2021, pp. 1–13.

A. Vaswani et al., “Attention is all you need,” 2017, arXiv:1706.03762.

X. Huang, Y. Li, O. Poursaeed, J. Hopcroft, and S. Belongie, “Stacked generative adversarial networks,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jul. 2017, pp. 1866–1875, doi: 10.1109/CVPR.2017.202.

P. J. Burt and E. H. Adelson, “The Laplacian pyramid as a compact image code,” IEEE Trans. Commun., vol. COM-31, no. 4, pp. 532–540, Apr. 1983, doi: 10.1109/TCOM.1983.1095851.

M. Brown and D. G. Lowe, “Recognising panoramas,” in Proc. 9th IEEE Int. Conf. Comput. Vis., vol. 2, Oct. 2003, pp. 1218–1225, doi:10.1109/ICCV.2003.1238630.

E. Denton, S. Chintada, A. Szlam, and R. Fergus, “Deep generative image models using a Laplacian pyramid of adversarial networks,” presented at the 28th Int. Conf. Neural Inf. Process. Syst., vol. 1, Montreal, QC, Canada, 2015.

E. H. Adelson, C. H. Anderson, J. R. Bergen, P. J. Burt, and J. M. O’Gden, “Pyramid methods in image processing,” RCA Eng., vol. 29, no. 6, pp. 33–41, 1984.

Z. Wang, Z. Cui, and Y. Zhu, “Multi-modal medical image fusion by Laplacian pyramid and adaptive sparse representation,” Comput. Biol. Med., vol. 123, Aug. 2020, Art. no. 103823.

Z. Lei, L. Qi, Y. Wei, and Y. Zhou, “Infant brain MRI segmentation with dilated convolution pyramid downsampling and self-attention,” 2019, arXiv:1912.12570.

J. Johnson, A. Alahi, and L. Fei-Fei, “Perceptual losses for real-time style transfer and super-resolution,” in Proc. Eur. Conf. Comput. Vis., 2016, pp. 694–711.

D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun, and M. Paluri, “A closer look at spatiotemporal convolutions for action recognition,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Jun. 2018, pp. 6450–6459.

A. Hore and D. Ziou, “Image quality metrics: PSNR vs. SSIM,” in Proc. 20th Int. Conf. Pattern Recognit., Aug. 2010, pp. 2366–2369, doi: 10.1109/ICPR.2010.579.

Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, “Image quality assessment: From error visibility to structural similarity,” IEEE Trans. Image Process., vol. 13, no. 4, pp. 600–612, Apr. 2004, doi: 10.1109/TIP.2003.819861.

M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, “GANs trained by a two time-scale update rule converge to a local nash equilibrium,” in Proc. Adv. Neural Inf. Process. Syst., vol. 30, 2017, pp. 1–60.

Ö. Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox, and O. Ronneberger, “3D U-Net: Learning dense, volumetric segmentation from sparse annotation,” in Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent., Cham, Switzerland: Springer, 2016, pp. 424–432.

X. Zhang et al., “PINet: A high-resolution infant MRI synthesizer based on transformer,” 2021, arXiv:2105.13993.

R. Shaw, C. Sudre, S. Ourselin, and M. J. Cardoso, “MRI k-space motion artefact augmentation: Model robustness and task-specific uncertainty,” presented at the 2nd Int. Conf. Med. Imag. With Deep Learn., 2019. [Online]. Available: https://proceedings.mlr.press/v102/shaw19a.html

F. Pérez-García, R. Sparks, and S. Ourselin, “TorchDO: A Python library for efficient loading, preprocessing, augmentation and patch-based sampling of medical images in deep learning,” Comput. Methods Programs Biomed., vol. 208, Sep. 2021, Art. no. 106236, doi: 10.1016/j.cmpb.2021.106236.

C. J. Jordan, S. R. B. Weiss, K. D. Howlett, and M. P. Freund, “Introduction to the special issue on ‘informing longitudinal studies on the effects of maternal stress and substance use on child development: Planning for the HEALthy brain and child development (HBCD) study,’” Adversity Resilience Sci., vol. 1, no. 4, pp. 217–221, Dec. 2020, doi: 10.1007/s42844-020-00022-6.