

Bayesian Posterior Distribution Estimation of Kinetic Parameters in Dynamic Brain PET Using Generative Deep Learning Models

Yanis Djebra, *Member, IEEE*, Xiaofeng Liu, Thibault Marin, *Member, IEEE*, Amal Tiss, Maeva Dhaynaut, Nicolas Guehl, Keith Johnson, Georges El Fakhri, *Fellow, IEEE*, Chao Ma*, *Member, IEEE* and Jinsong Ouyang* *Senior Member, IEEE*

Abstract—Positron Emission Tomography (PET) is a valuable imaging method for studying molecular-level processes in the body, such as hyperphosphorylated tau (p-tau) protein aggregates, a hallmark of several neurodegenerative diseases including Alzheimer’s disease. P-tau density and cerebral perfusion can be quantified from dynamic PET images using tracer kinetic modeling techniques. However, noise in PET images leads to uncertainty in the estimated kinetic parameters, which can be quantified by estimating the posterior distribution of kinetic parameters using Bayesian inference (BI). Markov Chain Monte Carlo (MCMC) techniques are commonly used for posterior estimation but with significant computational needs. This work proposes an Improved Denoising Diffusion Probabilistic Model (iDDPM)-based method to estimate the posterior distribution of kinetic parameters in dynamic PET, leveraging the high computational efficiency of deep learning. The performance of the proposed method was evaluated on a [18F]MK6240 study and compared to a Conditional Variational Autoencoder with dual decoder (CVAE-DD)-based method and a Wasserstein GAN with gradient penalty (WGAN-GP)-based method. Posterior distributions inferred from Metropolis-Hastings MCMC were used as reference. Our approach consistently outperformed the CVAE-DD and WGAN-GP methods and offered significant reduction in computation time than the MCMC method (over 230 times faster), inferring accurate ($< 0.67\%$ mean error) and precise ($< 7.23\%$ standard deviation error) posterior distributions.

Index Terms—Kinetic modeling, Dynamic PET imaging, Posterior distribution, Deep learning, Diffusion models, Conditional Variational Autoencoder

I. INTRODUCTION

Positron Emission Tomography (PET) is a powerful non-invasive imaging modality that can be used for diagnosis and monitoring of various diseases. One emerging application of PET imaging is the detection of hyperphosphorylated tau (p-tau) protein in the brain, a hallmark of neurodegenerative diseases such as Alzheimer’s disease (AD), progressive supranuclear palsy cases (PSP), or frontotemporal dementia (FTD) [1]. Using specific p-tau-binding radiotracers, e.g., [18F]-MK6240, PET images can provide a better understanding of the progression and distribution of p-tau pathology in the brain, which not only can aid in the early diagnosis and monitoring of AD, but could also support research into potential treatments before reaching late stages of the disease. The density of p-tau tangles in the brain can be quantified using tracer kinetic modeling techniques [2]–[6]. Among these techniques, reference-tissue techniques alleviate the need for arterial input function by choosing a region without specific radiotracer binding to estimate kinetic parameters from target regions. Radiotracer-specific kinetic parameters such as the Distribution Volume Ratio (DVR) and relative delivery rate constant R_1 can then be derived using the dynamic PET time-activity curves (TACs) to quantitatively estimate the p-tau burden and cerebral perfusion with the injected radiotracer.

Conventional kinetic modeling methods estimate kinetic parameters that best fit the noisy measured data with a given model. The resultant estimation is typically non-statistical, in the sense that it lacks information regarding uncertainties. From a frequentist perspective, one can predict the variation of the estimated kinetic parameters caused by measurement noise using a Cramér–Rao lower bound (CRLB)-based analysis. However, this technique does not take into consideration prior knowledge of the kinetic parameters. From a Bayesian perspective, this problem can be addressed by estimating the posterior distribution $p(\mathbf{x}|\mathbf{y})$ of the kinetic parameters \mathbf{x} given the measured TACs \mathbf{y} and prior knowledge of \mathbf{x} , i.e., $p(\mathbf{x})$. Commonly used approaches to estimate $p(\mathbf{x}|\mathbf{y})$ include techniques such as Approximate Bayesian Computation (ABC,

This work was supported in part by the National Institutes of Health under the grants P41EB022544, R01EB033582, R01EB035093, R21EB034911, R01AG076153, R21AG070714, R01AG085561, and P01AG036694.

* indicates corresponding authors.

Y. Djebra, X. Liu, T. Marin, M. Dhaynaut, N. Guehl and G. El Fakhri are with the Yale Biomedical Imaging Institute, Department of Radiology & Biomedical Imaging, Yale School of Medicine, Yale University, New Haven, CT USA.

A. Tiss and K. Johnson are with the Department of Radiology, Massachusetts General Hospital, Harvard Medical School, Boston, MA, USA.

Part of this work was done while Y. Djebra, X. Liu, T. Marin, M. Dhaynaut, N. Guehl, G. El Fakhri, C. Ma, and J. Ouyang were with the Department of Radiology, Massachusetts General Hospital, Harvard Medical School, Boston, MA, USA.

C. Ma and J. Ouyang are with the Yale Biomedical Imaging Institute, Department of Radiology & Biomedical Imaging, Yale School of Medicine, Yale University, New Haven, CT USA (email: [chao.ma.cm2943, jinsong.ouyang]@yale.edu).

likelihood-free techniques [7] and Markov Chain Monte Carlo (MCMC) methods [8]. While ABC methods can be particularly useful when the likelihood is intractable, they suffer from calibration difficulties and introduce bias in the estimated posterior [7]. MCMC methods can produce asymptotically unbiased estimation of the posterior distribution, given enough samples of the parameter space. Unfortunately, inferring the posterior distribution using MCMC methods is often impractical because of its inherent low sampling efficiency with high-dimensional data [9]. Emerging generative deep learning (DL) techniques (Conditional Variational Autoencoders (CVAEs) [10], Generative Adversarial Networks (GANs) [11], Neural Ordinary Differential Equations (Neural ODE) [12]) offer new opportunities to address Bayesian inversion problems and efficiently infer posterior distributions. For instance, frameworks utilizing CVAEs [13], [14] or GANs [15] showed promising results for posterior distribution inference in the field of medical imaging.

In this work, we propose a DL-based method to predict the joint posterior distributions of kinetic parameters from multiple regions or voxels. A flowchart of the proposed method can be found in Fig 1. The proposed method leverages the representation power of a conditional improved Denoising Diffusion Probabilistic Model (iDDPM) [16] to learn the mean and variance of the reverse noise diffusion process. The kinetic parameters' prior distribution $p(\mathbf{x})$ was obtained from an *in vivo* [18F]-MK6240 dynamic PET study with a total of 50 subjects. The kinetic parameters' posterior distribution was then inferred using the target TACs \mathbf{y} . The posterior distributions inferred by the proposed method were then compared to posterior distributions from a CVAE with dual decoder (CVAE-DD)-based method [13] and a Wasserstein GAN with gradient penalty (WGAN-GP)-based method [17], and unbiased MCMC was used as reference for validation purposes. Region-of-interest (ROI) based brain kinetic parameters' maps representing the mean and standard deviation (SD) of the posterior distribution were then produced from *in vivo* data.

II. BACKGROUND

In this section, we provide a concise overview of diffusion models, along with background knowledge on radiotracer kinetic modeling.

A. Improved Denoising Diffusion Probabilistic Models (iDDPM)

Diffusion models belong to a category of probabilistic models that establish a forward Markovian diffusion process denoted as q . This process involves incrementally introducing Gaussian noise, represented by ϵ , to an initial input \mathbf{x} . The added noise follows a specific noise schedule denoted by β_t (noise variance) for a timestep parameter t .

Given a measurement \mathbf{y} conditioning the random variable \mathbf{x} , the forward diffusion process is expressed as:

$$\begin{aligned} q(\mathbf{x}_{1:T}|\mathbf{x}_0, \mathbf{y}) &= \prod_{t=1}^T q(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{y}), \\ q(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{y}) &= \mathcal{N}\left(\mathbf{x}_t|\mathbf{y}; \sqrt{1-\beta_t}\mathbf{x}_{t-1}|\mathbf{y}, \beta_t\mathbf{I}\right) \quad (1) \end{aligned}$$

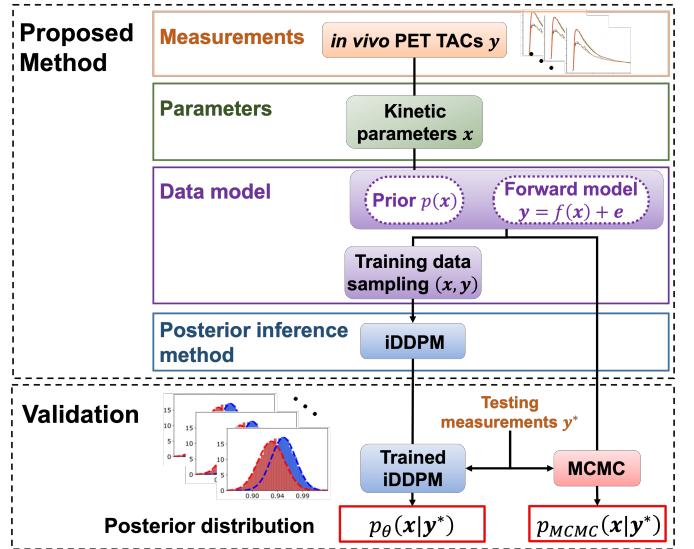


Fig. 1: Flow chart of the proposed method and its validation. The multivariate prior distribution of the multi-ROI kinetic parameters \mathbf{x} and the forward model f with noise e are defined. An improved diffusion model is then trained on data sampled from the prior to estimate the posterior distribution of kinetic parameters. The posterior distributions are validated on testing sets by comparing them with distributions from unbiased MCMC.

where $\mathcal{N}(\mathbf{x}_t|\mathbf{y}; \boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t)$ describes the random variable $\mathbf{x}_t|\mathbf{y}$ following a Gaussian distribution of mean $\boldsymbol{\mu}_t$ and covariance $\boldsymbol{\Sigma}_t$, T is the total number of timestep and \mathbf{x}_0 is the original non-corrupted input.

The key idea in diffusion models is then to approximate the reverse diffusion process p using neural networks:

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{y}) = \mathcal{N}(\mathbf{x}_{t-1}|\mathbf{y}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t, \mathbf{y}), \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t, \mathbf{y})) \quad (2)$$

Training the neural network can be performed by optimizing the variational lower bound (VLB):

$$L_{\text{vib}} = L_0 + L_1 + \dots + L_{T-1} + L_T \quad (3)$$

$$L_0 = -\log p_\theta(\mathbf{x}_0|\mathbf{x}_1, \mathbf{y}) \quad (4)$$

$$L_{t-1} = D_{KL}(q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0, \mathbf{y}) \| p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{y})) \quad (5)$$

$$L_T = D_{KL}(q(\mathbf{x}_T|\mathbf{x}_0, \mathbf{y}) \| p(\mathbf{x}_T, \mathbf{y})) \quad (6)$$

where D_{KL} is the Kullback–Leibler (KL) divergence.

In practice, training directly on the VLB can prove difficult [18]. A simplified loss L_{simple} was therefore introduced by Ho et al. where $\boldsymbol{\mu}_\theta(\mathbf{x}_t, t, \mathbf{y})$ is reparametrized with $\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t, \mathbf{y})$ (i.e., predicting the added noise at each timestep) and where the terms in L_{vib} are reweighted for computational simplicity:

$$L_{\text{simple}} = \mathbb{E}_{t, \mathbf{x}_0, \boldsymbol{\epsilon}, \mathbf{y}} \|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t, \mathbf{y})\|^2 \quad (7)$$

where $\bar{\alpha}_t = \prod_{s=1}^t (1 - \beta_s)$.

It is worth noting that only the mean of the reverse diffusion process is learned with the simplified loss L_{simple} . Using DDPMs, the variance of the reverse diffusion process $\boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t, \mathbf{y})$ cannot be learned and is fixed. While Ho et al. demonstrated DDPMs can produce high-quality samples in terms of image metrics, they did not manage to produce competitive negative log-likelihood (NLL), a metric conveying how well generative models capture the modes of the data

distribution [16], [19]. More recently, Nichol et al. [16] explored the reasons why DDPMs performed poorly in terms of NLL: while fixing the variance of the reverse diffusion process may be beneficial for sample quality, it is detrimental for the log-likelihood, indicating a potential poor mode coverage. To achieve both high-sample quality and better NLL, they propose an improved DDPM (iDDPM) where the variance of the reverse diffusion process $\Sigma_\theta(\mathbf{x}_t, t, \mathbf{y})$ is learned, and introduce a hybrid loss incorporating the VLB term:

$$L_{\text{hybrid}} = L_{\text{simple}} + \lambda_{\text{vlb}} L_{\text{vlb}} \quad (8)$$

where λ_{vlb} is a parameter scaling the contribution between L_{simple} (i.e., learning the mean $\mu_\theta(\mathbf{x}_t, t, \mathbf{y})$) and L_{vlb} (i.e., learning the variance $\Sigma_\theta(\mathbf{x}_t, t, \mathbf{y})$). L_{vlb} (3) is typically hard to optimize, notably due to noisy gradients. To avoid L_{vlb} from overwhelming L_{simple} , λ_{vlb} is then typically kept small.

B. Radiotracer kinetic modeling

1) *Logan plot model*: Introduced by Logan et al. [2], the Logan plot model is a widely employed technique in kinetic modeling providing a simple graphical approach to estimate the binding potential BP (or equivalently DVR = BP+1). This graphical model was later modified to estimate DVR without blood sampling, using a reference region instead [20], i.e., a region without specific binding from the radiotracer.

The Logan plot involves the linearization of the integral equation that links the concentration of the radiotracer in a target region $C_T(t)$ to that of a reference region $C_R(t)$ for a time t sufficiently large ($t > t^*_{\text{LOGAN}}$):

$$\frac{\int_0^t C_T(\tau) d\tau}{C_T(t)} = \text{DVR} \frac{\int_0^t C_R(\tau) d\tau + C_R(t)/k'_2}{C_T(t)} + \gamma \quad (9)$$

where k'_2 is the reference region clearance constant, γ is a constant, and t^*_{LOGAN} is a time determined based on the characteristics of the radiotracer used for the study.

By plotting $\int_0^t C_T(\tau) d\tau/C_T(t)$ against $(\int_0^t C_R(\tau) d\tau + C_R(t)/k'_2)/C_T(t)$, the resultant graph displays an affine line with its slope equal to DVR, if $C_R(t)/C_T(t)$ is sufficiently constant over time. This simplification facilitates the estimation of DVR, making this model an effective tool in quantitative analysis of PET data [21].

2) *Simplified Reference Tissue Model*: The Simplified Reference Tissue Model (SRTM) [3] was introduced as another alternative to kinetic modeling methods requiring arterial cannulation and blood sampling. The SRTM is a 3-parameter model: it estimates the Distribution Volume Ratio (DVR), the relative delivery rate of the radiotracer (R_1), and the reference region clearance constant (k'_2) from PET TACs. Its operational equation is given by:

$$C_T(t) = R_1 C_R(t) + R_1 [k'_2 - k_2] C_R(t) \otimes e^{-k_2 t} \quad (10)$$

where \otimes is the convolution operator, k_2 is the target region clearance constant, and $\text{DVR} = R_1 k'_2 / k_2$. Similarly to II-B.1, data fitting with (10) is performed for $t > t^*_{\text{SRTM}}$.

SRTM was later improved in [5] with SRTM2, a 2-step and 2-parameter SRTM for improved robustness and noise

reduction. In SRTM2, the reference region clearance constant k'_2 does not vary and is fixed for all the voxels. As such, DVR and R_1 are estimated using (10) with k'_2 fixed. The k'_2 value can be fixed patient-wise using SRTM a first time for all voxels and then set it to the average of the obtained values. Another approach is to precalculate the reference region clearance constant for a population and set k'_2 to the average of the resulting values. The latter approach was used in this study.

III. PROPOSED METHODS

Let \mathbf{x} denote a vector consisting of DVRs and R_1 s values from multiple regions or voxels, and \mathbf{y} the corresponding PET TACs. This work aims to estimate the posterior distribution $p(\mathbf{x}|\mathbf{y})$, given the measured TACs \mathbf{y} , and prior knowledge on the kinetic parameters $p(\mathbf{x})$. We propose to do so by approximating the true posterior with the variational distribution $p_\theta(\mathbf{x}|\mathbf{y})$ of an iDDPM [16] trained on data drawn from the prior $p(\mathbf{x})$ and conditioned by the TACs \mathbf{y} . Samples \mathbf{x}^* are then drawn from the posterior distribution using the proposed method, the compared methods, and the MCMC reference.

A. Data model \mathcal{D}

In the context of this study, relevant kinetic parameters \mathbf{x} are obtained from dynamic PET images \mathbf{y} by fitting an appropriate kinetic model to the measured TACs from 50 subjects. The estimated kinetic parameters from these subjects are in turn used to define the prior distribution $p(\mathbf{x})$. A forward model f with noise, defined as $\mathbf{y} = f(\mathbf{x}) + \mathbf{e}$, where \mathbf{e} is the TAC measurement noise, is then used to generate training and validation data.

1) *Kinetic parameters' estimation from *in vivo* study*: In this work, kinetic parameters' fitting was performed on a dynamic PET [¹⁸F]-MK6240 *in vivo* study involving a total of 50 subjects: 39 control subjects (Cognitively Normal, CN), 5 AD subjects and 6 MCI subjects. Each subject was scanned for up to $T_{\text{acq}} = 120$ min and a structural MRI (MPRAGE) was performed for anatomical reference. PET images were reconstructed with a matrix-size of $256 \times 256 \times 89$ and $T_{\text{TAC}} = 54$ time frames with the following time partition: 6×10 s, 8×15 s, 6×30 s, 8×60 s, 8×120 s, and 18×300 s. MCFLIRT registration from FSL [22] was applied on PET images for inter-frame motion correction. Anatomical ROIs were generated from the MR images using FreeSurfer parcellation [23] and registered to the PET image space to extract the regional TACs.

The PET time-activity curves were then processed as in [24]: the Logan plot fitting [2] was used to obtain DVR values, and the SRTM2 model [5] to obtain R_1 values. This choice is motivated by recent studies [25], [26] demonstrating that the Logan plot method provides robust DVR estimates, while SRTM/SRTM2 can provide robust R_1 estimates [24] but significant underestimation of DVR for the quantification of tau load. For both models, the cerebellar cortex was used as reference region and k'_2 was fixed in each scan using a precalculated value from a cohort of patients. Kinetic parameter analysis was performed on ROIs for decreased susceptibility to noise and artifacts. t^*_{LOGAN} was set to 40 mins

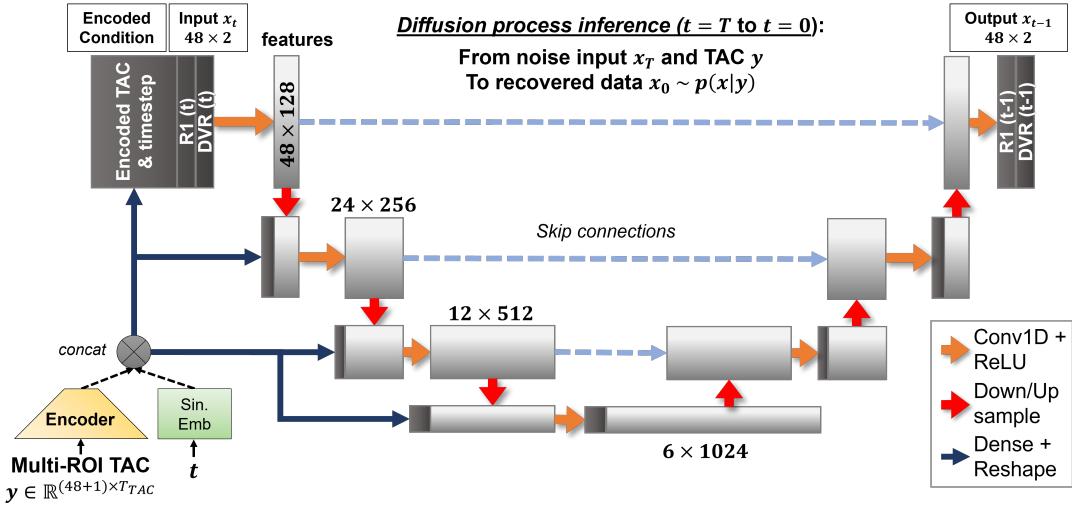


Fig. 2: Architecture of the 1D U-net proposed for the posterior estimation of the kinetic parameters \boldsymbol{x} for the $n_{\text{ROI}} = 48$ regions, given the TAC \boldsymbol{y} from the n_{ROI} regions and the reference region. The TAC is encoded using a dense layer-based encoder and the timestep using a sinusoidal embedding (Sin. Emb.). The encoded outputs are concatenated at every stage of the U-net along the channel dimension. For visual clarity, the arrows representing the concatenation of the TACs and timesteps in the decoder stages have been omitted. During inference, a given TAC \boldsymbol{y}^* is supplied and the input \boldsymbol{x}_T is set to random noise. The network is then iteratively applied T times on \boldsymbol{x}_t to recover $\boldsymbol{x}_0 \sim p(\boldsymbol{x}|\boldsymbol{y}^*)$.

for DVR. For R_1 , t_{SRTM}^* was set to 0 mins and fitting was performed on the 5 first minutes of data only, as in [24]. For simplicity, in the rest of this paper, \boldsymbol{y} will denote the ROI TACs of the n_{ROI} target regions and the reference regions, and \boldsymbol{x} the corresponding fitted kinetic parameters DVR and R_1 .

2) Prior model $p(\boldsymbol{x})$: The prior distribution p for the kinetic parameters \boldsymbol{x} was represented as multivariate Gaussian distributions. The parameters of $p(\boldsymbol{x})$ were determined as follows: for every subject, DVRs and R_1 s values from the n_{ROI} target ROIs were obtained from an *in vivo* PET study, as described in Section III-A.1, to form $\boldsymbol{x}^{\text{prior}} = [\boldsymbol{x}_{\text{DVR}}^{\text{prior}}, \boldsymbol{x}_{R_1}^{\text{prior}}]$. The mean and covariance matrix characterizing the prior knowledge of each kinetic parameter were then estimated by calculating the mean value of each ROI (i.e., μ_{DVR} and μ_{R_1}) and the covariance matrices describing the variation and correlation of the DVRs and R_1 s within and between the ROIs (i.e., Σ_{DVR} and Σ_{R_1}). Note that cross-correlation between kinetic parameters DVR and R_1 is assumed to be zero. The prior distribution $p(\boldsymbol{x})$ is then characterized by its mean $\boldsymbol{\mu} = [\mu_{\text{DVR}}, \mu_{R_1}]$ and its block-wise diagonal covariance matrix Σ :

$$\Sigma = \begin{bmatrix} \Sigma_{\text{DVR}} & \mathbf{0} \\ \mathbf{0} & \Sigma_{R_1} \end{bmatrix} \quad (11)$$

New samples drawn from the above prior distribution $\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma)$ can therefore be obtained for neural network training, validation, and testing. For convenience, DVR and R_1 values from an *in vivo* cohort were used to estimate the parameters of $p(\boldsymbol{x})$ (i.e., mean and covariance matrix). Alternatively, these parameters could be derived from a literature review. It is important to note that the actual DVR and R_1 values from the *in vivo* study, and the corresponding TACs, were not used for training/testing until the fine-tuning step, as described in Section III-B.3.

3) Forward model: Let $\boldsymbol{x}^{\text{sampled}}$ be DVR and R_1 samples drawn from the prior defined in Section III-A.2. A kinetic

model forward operation denoted by f can be defined to obtain the corresponding target TACs, i.e., $\boldsymbol{y}_{\text{noiseless}} = f(\boldsymbol{x}, \boldsymbol{y}_{\text{ref}})$. In this work, f was defined as the SRTM2 forward model in (10). Reference region-based kinetic models need a reference region TAC $\boldsymbol{y}_{\text{ref}}$ to estimate target TACs. In the same spirit as Section III-A.2, and for better generalizability of the proposed method, reference TACs were resampled from the estimated *in vivo* reference TACs using a multivariate Gaussian distribution model. This way, every multidimensional target TAC sampled were produced with different (and known) reference TACs.

In regard to the noise model, while it can be well approximated by a Poisson distribution in the sinogram domain, such a model is not adequate in the image domain, (i.e., after image reconstruction) [27]–[29]. In this work, ROI-wise TACs are used, i.e., image voxel values are averaged across ROIs. Assuming large enough ROIs, and small noise correlations, the noise model in the ROI-wise TACs can be approximated by an empirical noise model using additive Gaussian noise on the TACs [5], [30], [31] where the noise standard deviation is correlated to the square root of the signal intensity at a given acquisition time t . In addition, different ROIs with different sizes are formed, which means the standard deviation of the noise is also scaled differently in every ROI. Denoting \boldsymbol{y}_r the TAC of the r -th ROI, and \boldsymbol{x}_r the corresponding kinetic parameters, the forward model is then given by:

$$\boldsymbol{y}_r = f(\boldsymbol{x}_r, \boldsymbol{y}_{\text{ref}}) + S_r \sqrt{\frac{f(\boldsymbol{x}_r, \boldsymbol{y}_{\text{ref}})}{\Delta t e^{-\lambda t}}} \mathcal{N}(\mathbf{0}, \mathbf{I}) \quad (12)$$

where Δt is the time frame duration, λ is the tracer radioactive decay constant, and S_r is a constant scaling the noise standard deviation in each ROI. In this study, the SRTM operational equation (10) was used as forward model, and S_r , $r = [1, \dots, n_{\text{ROI}}]$ was estimated from *in vivo* PET data. In detail, S_r was estimated using (12) with the noisy *in vivo* TACs for \boldsymbol{y}_r , and approximating the fitted curve from SRTM2 $f(\boldsymbol{x}_r, \boldsymbol{y}_{\text{ref}})$ for the noise-free TAC. Then, under the model

assumption, the quantity $\frac{y_r - f(\mathbf{x}_r, \mathbf{y}_{\text{ref}})}{\sqrt{f(\mathbf{x}_r, \mathbf{y}_{\text{ref}})}} \times \sqrt{\Delta t \cdot e^{-\lambda t}}$ follows a gaussian distribution with standard deviation equal to S_r . Note that, while this equation is valid for all t in theory, in practice the term $\frac{y_r - f(\mathbf{x}_r, \mathbf{y}_{\text{ref}})}{\sqrt{f(\mathbf{x}_r, \mathbf{y}_{\text{ref}})}}$ can be unstable in the early frames (e.g., due to very small $f(\mathbf{x}_r, \mathbf{y}_{\text{ref}})$ values, or too short Δt) and lead to exploding values in the ratio, contaminating S_r approximation. To avoid such an issue, the mean across time was taken to estimate the standard deviation S_r across samples.

B. Implementation details

1) *Network architecture*: The building block of the proposed conditional iDDPM was a 1D U-net (Fig. 2). Note that, in this work, ROI-wise processing was used, yielding 1 DVR and R_1 value per ROI. The input is therefore composed of two 1-dimensional vectors, hence the use of 1D convolutions. The convolution kernels slide on the ROI dimension, representing and extracting the spatial information and spatial correlation between ROIs, as each ROI represents a set of spatial locations in the image domain. The input $\mathbf{x} \in \mathbb{R}^{\text{batch} \times n_{\text{ROI}} \times 2}$ consisted of the DVR and R_1 vectors concatenated along the channel dimension. The network was conditioned by the corresponding n_{ROI} target TACs and the reference TAC, \mathbf{y} . The reverse diffusion process starts with a random noise input \mathbf{x}_T at timestep $t = T$. Noise is then iteratively removed from $t = T$ to $t = 0$ to recover the denoised input data $\mathbf{x}_0 \sim p(\mathbf{x}|\mathbf{y})$ (i.e., the predicted DVR and R_1 values). Residual connections were used after convolution blocks. The architecture consisted of four encoder/decoder stages, where each stage consisted of a 6×1 convolutional layer, ReLU, and 2×1 max pooling (2×1 upsampling for decoder stages). The first convolution layer had 128 filters, doubled after every encoder stage, and halved after every decoder stage. The diffusion timestep was embedded using sinusoidal embedding [32] of dimension 64, and the condition \mathbf{y} went through a dense layer-based encoder (256, 128, and 64 nodes, latent space of dimension 32). The encoded TACs and embedded timesteps were used to condition the neural network by concatenating them at each stage of the U-net. To ensure dimensions agreed during concatenation, a dense layer (dark blue arrow) adjusted the dimension of the conditions to fit the dimension of each encoder/decoder stage input. A cosine noise scheduler [16] was used for training, and a total of $T = 1000$ timesteps was used for training and inference of the diffusion model. In the scope of this paper, $n_{\text{ROI}} = 48$ ROIs were used, covering most of the brain.

2) *Network training*: The neural network is trained on data \mathbf{x} drawn from the prior distribution (Section III-A.2) and conditioned by the corresponding TACs \mathbf{y} . $N_{\text{tr}} = 100000$ new datapoints $\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ were sampled, divided in 90% training and 10% validation. The corresponding target TACs were generated with the SRTM2 forward model using resampled reference TACs, as described in Section III-A.3. The noise level S_r , $r = [1, \dots, n_{\text{ROI}}]$ for each ROI was estimated from the in vivo PET study. The network was trained for 500 epochs using a batch size of 64. An

exponentially decreasing learning rate (from 2×10^{-4} to 5×10^{-5}) using the Adam optimizer was used, and λ_{vlb} was set to 0.1 for L_{vlb} and L_{simple} scaling. Training was performed on an NVIDIA A100-SXM4 GPU using the TensorFlow framework and took approximately 2.6 hours in the case $n_{\text{ROI}} = 48$. Code implementing the proposed method can be found on GitHub (https://github.com/yanisdjebra/PET_posterior_distribution).

3) *Fine-tuning on in vivo data*: Once the neural network was trained, tested, and validated with MCMC; fine-tuning was performed using data from an *in vivo* study (see Section III-A.1). 42 subjects were used for training and 8 subjects (2 AD, 2 MCI, and 4 control) were used for testing. ROI-wise kinetic model analysis on the dataset was performed as detailed in Section III-A.1 (Logan plot for DVR, SRTM2 for R_1) for the same 48 ROIs. No explicit prior was defined here: the new 42 training data pairs (\mathbf{x}, \mathbf{y}) from the *in vivo* training set define an implicit prior from which the previously trained iDDPM network will adapt by fine-tuning with low learning rate. More specifically, the previously trained iDDPM network was fine-tuned for 200 epochs with a batch size of 6 and a low learning rate (from 1×10^{-6} to 5×10^{-6}) using the Adam optimizer.

C. Performance validation

1) *Testing set sampling*: A testing set \mathbf{x}^{test} with N_{test} measurements was produced similarly to the training set (Section III-B.2) to validate the proposed method. For quantitative evaluation purposes, DVR and R_1 were sampled from the bulk of the distribution $p(\mathbf{x})$, avoiding the tails of the distribution. This reduced the probability of sampling outliers in the testing set, avoiding unfeasible DVR and R_1 values in the biological sense (i.e., extremely high or extremely low) values. This was enforced using the Mahalanobis distance $d_M(\mathbf{x}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sqrt{(\mathbf{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu})}$: the larger d_M , the more uncommon the sample (i.e., potential outlier) [33]. In such case, the square of the Mahalanobis distance d_M^2 is distributed as a χ^2 random variable with n_{ROI} degree of freedom. Upon defining a percentile threshold P , outliers can be identified as samples with a Mahalanobis distance larger than the value of the P^{th} -quantile of the χ^2 distribution, denoted by Q . Mathematically, this translates to testing samples accepted if they satisfy:

$$d_M(\mathbf{x}^{\text{test}}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) < Q \quad (13)$$

The Mahalanobis distance has advantages compared to the standard Euclidean distance used in [34] as it normalizes the differences between the variables \mathbf{x} and $\boldsymbol{\mu}$ by their respective standard deviations. This normalization ensures that each dimension contributes equally to the overall distance calculation, allowing better sensitivity to the shape of the distribution. In this work, $P = 0.8$ was used to sample testing datapoints from the bulk of the prior distribution.

2) *Compared method: CVAE with dual decoder*: The posterior distributions inferred by the proposed method were compared to the posterior distributions from a trained CVAE-DD [13] (Fig. 3a). The input \mathbf{x} consisted of DVR and

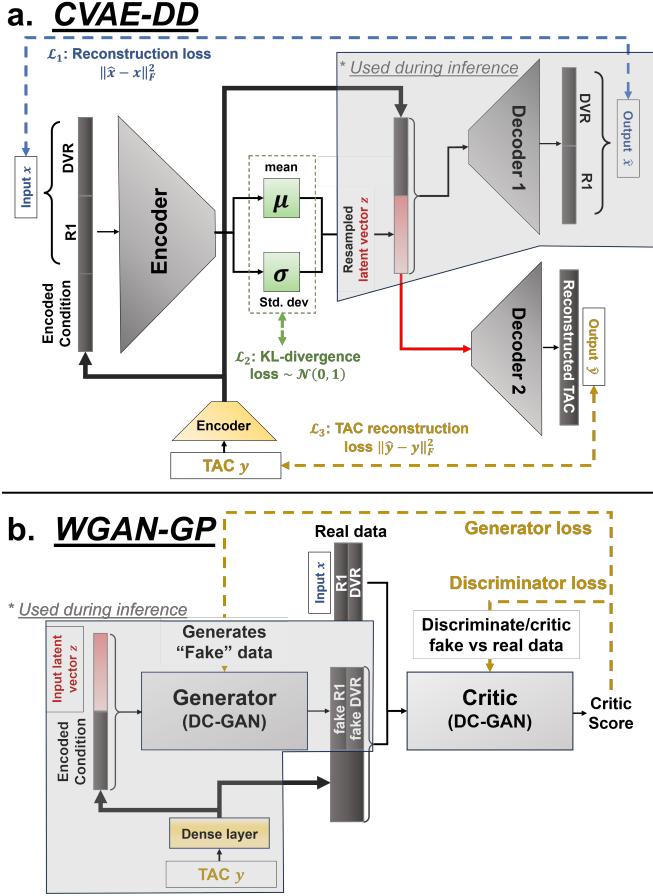


Fig. 3: Architecture of the compared methods (see Section III-C.2 and III-C.3 for details). x corresponds to the kinetic parameters for the $n_{ROI} = 48$ target regions, and y to the TAC from the n_{ROI} target regions and the reference region. Gray-shaded boxes indicate parts used during inference. **a.** CVAE with dual decoder (CVAE-DD). The encoder and decoder are dense-layer based. The latent vector z concatenated to the encoded TAC goes through a 1st decoder producing the approximated posterior samples $\hat{x} \sim p(\hat{x}|y)$. The second decoder uses the latent vector z as input to reconstruct the TAC \hat{y} . **b.** Wasserstein GAN with gradient penalty (WGAN-GP). The generator generates "fake" data from a latent vector z conditioned by an encoded TAC condition y , approximating the posterior distribution $p(x|y)$. The critic evaluates the quality of the "fake" data compared to the real ones and provides feedback (WGAN loss) to the generator.

R_1 , and the condition y consisted of the corresponding n_{ROI} target TACs and reference TAC. Similarly to the proposed method, the TACs y went first through an encoder (512, 256, 128, 64 nodes, and 16 nodes for latent space) before being concatenated to the input and the latent space. This first encoder is used for a better expression of the TAC through concatenation. x went through a dense layer-based encoder (512, 256, 128, 64 nodes, latent space of dimension 32 for mean and standard deviation at the bottleneck). The latent vector z concatenated to the encoded TAC went through a dense layer-based decoder (64, 128, 256, 512 nodes) producing the approximated posterior samples $\hat{x} \sim p(\hat{x}|y)$. The second dense layer-based decoder (64, 128, 256, 512 nodes) used the latent vector z as input to reconstruct the TAC \hat{y} . The idea behind using a second decoder for y is that the latent space should represent both x and y and that $p(z|y) \neq p(z)$. For more details on CVAE-DD, refer

to [13]. The encoder used a ReLU activation function, and both decoders used a leaky ReLU activation function. The CVAE-DD was trained on the same data as the proposed method. Training was performed for 200 epochs using the RMSProp optimizer with an exponentially decreasing learning rate (from 2×10^{-6} to 1×10^{-6}) with a batch-size of 256. β_{KL} , the term balancing the contribution of the reconstruction losses and the KL-divergence, was set to 10. During testing, the latent space is randomly sampled, producing a latent vector z . This latent vector goes through the 1st decoder to obtain samples from the approximated posterior distribution.

3) Compared method: Wasserstein GAN with gradient penalty: The posterior distributions inferred by the proposed method were also compared with a trained conditional WGAN-GP [17] (Fig. 3b). For this method, the generator generated "fake" data from a latent vector z (and encoded TAC condition y), approximating the posterior distribution $p(x|y)$. A critic (generally referred to as discriminator in regular GANs) then attributes a score to the fake samples, indicating the quality of the generated samples relative to the real data x and guiding the generator to produce more realistic samples. In this work, the architecture for the generator and the critic was inspired by deep convolutional GANs (DCGANs) [35], offering more stable training and better performance [17], [35], [36]. The latent space for the generator was of size 48 (i.e., dimension of z) and its architecture was similar to a U-Net. More specifically, the generator consisted of four encoder/decoder stages, where each consisted of a 6×1 convolutional layer, ReLU, another 6×1 convolutional layer, ReLU, and a 2×1 downsampling/upsampling step performed using strided convolutions. The first stage had 64 filters, doubled (resp. halved) after every encoder (resp. decoder) stage. The output corresponds to the approximated posterior samples, with size 48×2 (DVR and R_1 concatenated along the channel dimension). The discriminator had a similar structure, replacing ReLU with Leaky ReLU for activation function, and with an additional final layer (dense layer with one node), to obtain the 1-dimensional critic score for each sample in the batch. The WGAN-GP was trained on the same data as the proposed method. Training was performed for 150 epochs using the RMSProp optimizer for both generator and discriminator. The learning rate was set to 1×10^{-5} and batch size to 64. A two-sided gradient penalty was used [17] with λ_{gp} set to 1. During training, the critic was updated 3 times before each generator update to ensure higher-quality feedback from the critic to the generator. During testing, the latent space is randomly sampled, producing a latent vector z and using the generator to produce approximated posterior samples.

4) MCMC reference: Given a measurement TAC y^* from the testing set (Section III-C.1), a Metropolis-Hastings (MH) based MCMC algorithm (Pymc, [37], [38]) was utilized to produce asymptotically unbiased estimation of $p(x|y^*)$ to validate the proposed method. To palliate the low MCMC sampling efficiency in high dimensions, a large number of burn-in steps and samples was used. Specifically, for

each TAC measurement in the testing set, $N_{\text{sample}}^{\text{MCMC}} = 20000$ samples \mathbf{x}^* were generated using 4 Markov chains, after 40000 burn-in steps. Convergence was assessed examining \hat{R} [39], a chain convergence diagnostic metric, and through visual examination of the obtained distributions. For more details on MCMC, the burn-in steps and convergence properties, see [9], [39]. The typical posterior distribution inference time for a TAC measurement was about 3 hours 50 mins for MCMC, approximately 1 min for the proposed method (using $T = 1000$ timesteps), and less than a second for CVAE-DD and WGAN-GP.

5) Evaluation metrics: The performance of the proposed method was then characterized using the ROI-wise mean absolute percentage error (MAPE) on DVR and R_1 :

$$\bar{\delta}_\mu = \frac{1}{N_{\text{test}}} \sum_m^{N_{\text{test}}} \frac{|\boldsymbol{\mu}_{:,m}^{\text{MCMC}} - \boldsymbol{\mu}_{:,m}^{\text{DL}}|}{\boldsymbol{\mu}_{:,m}^{\text{MCMC}}} \quad (14)$$

$$\bar{\delta}_\sigma = \frac{1}{N_{\text{test}}} \sum_m^{N_{\text{test}}} \frac{|\boldsymbol{\sigma}_{:,m}^{\text{MCMC}} - \boldsymbol{\sigma}_{:,m}^{\text{DL}}|}{\boldsymbol{\sigma}_{:,m}^{\text{MCMC}}}$$

where $\boldsymbol{\mu}_{:,m}^{\text{DL}}$ (resp. $\boldsymbol{\sigma}_{:,m}^{\text{DL}}$) is the mean (resp. standard deviation) of the posterior distribution of the measurement m (from the testing set) for all ROIs estimated using a DL-based method. Note that the division bar represents element-wise division.

In addition to the MAPE, the effective sampling size (ESS) of the estimated distributions was computed. The ESS is a metric used to quantify the efficiency of an independent and identically distributed (i.i.d) sampling process [39], [40]. It represents the number of i.i.d samples that would convey the same amount of information as the actual correlated samples generated. To define the ESS, first consider a stationary sequence of possibly correlated random variables $[\mathbf{x}_1^*, \dots, \mathbf{x}_{N_{\text{sample}}}^*]$ obtained from a sampling method. The autocorrelation of samples for a time lag k is defined as:

$$R_k = \frac{\text{cov}(\mathbf{x}_1^*, \mathbf{x}_{1+k}^*)}{\text{var}(\mathbf{x}_1^*)} \quad (15)$$

The ESS can then be defined as:

$$\text{ESS} = \frac{N_{\text{sample}}}{1 + 2 \sum_{k=1}^{\infty} R_k} \quad (16)$$

In general, the ESS is lower than the actual number of generated samples N_{sample} . The ESS value is influenced by the autocorrelation within a chain, as indicated by (16): if the chain exhibits high autocorrelation and redundancy, fewer effectively independent samples are obtained, leading to an ESS lower than the number of samples drawn N_{sample} . This metric is valuable for assessing the reliability and precision of the estimated distribution. When sampling with multiple chains, one can sum the ESS of those chains, or compute a cross-chain ESS [39]. Summing the ESS across multiple chains often results in an overestimation, which is why, in this study, ESS will refer to the cross-chain ESS for MCMC to better reflect the actual effective sampling size.

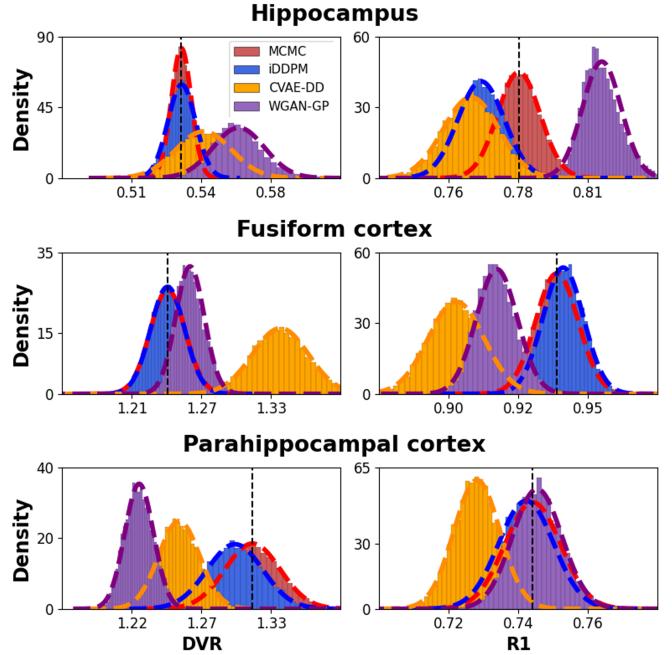


Fig. 4: Posterior distributions of kinetic parameters DVR (left) and R_1 (right) estimated using MCMC, CVAE-DD, WGAN-GP, and the proposed method for a few representative regions. The distributions plotted are for the hippocampus (ROI #0) in the first row, the fusiform cortex (ROI #2) in the second row, and the parahippocampal cortex (ROI #4) in the third row. The vertical black line represents the posterior true mean calculated from the converged MCMC. The colored dashed lines represent a fitted Gaussian distribution for MCMC (red), CVAE-DD (yellow), WGAN-GP (purple), and the proposed method (blue).

IV. RESULTS

TAC measurements \mathbf{y}^* from the testing set (see Section III-C.1) were utilized to generate $N_{\text{sample}}^{\text{DL}} = 10000$ DVR and R_1 posterior samples (i.e., $\mathbf{x}^* \sim p_\theta(\mathbf{x}|\mathbf{y}^*)$) with the proposed and the compared methods. The obtained distributions were then compared to the distributions estimated with MCMC. The proposed method was then fine-tuned on TACs measurements from an *in vivo* PET study to produce DVR and R_1 mean and SD maps.

A. Validation on testing set drawn from prior

The posterior distribution was first estimated on a TAC measurement from the testing set, obtained as described in Section III-C.1. The resulting kinetic parameter distributions are shown in Fig. 4 for the hippocampus, the fusiform cortex, and the parahippocampal cortex. The estimated posterior distribution is displayed in red for the MCMC method, in blue for the proposed method, yellow for the CVAE-DD, and purple for the WGAN-GP. As can be seen, for the ROIs shown in Fig. 4 the distribution inferred by the proposed method is consistent with the MCMC method, while the compared methods display more bias, both in terms of mean and standard deviation.

While Fig. 4 displays the distribution for 3 representative ROIs, the proposed and compared methods actually yield distributions for all 48 regions. To visualize these multidimensional distributions, we display the mean and SD per ROI in

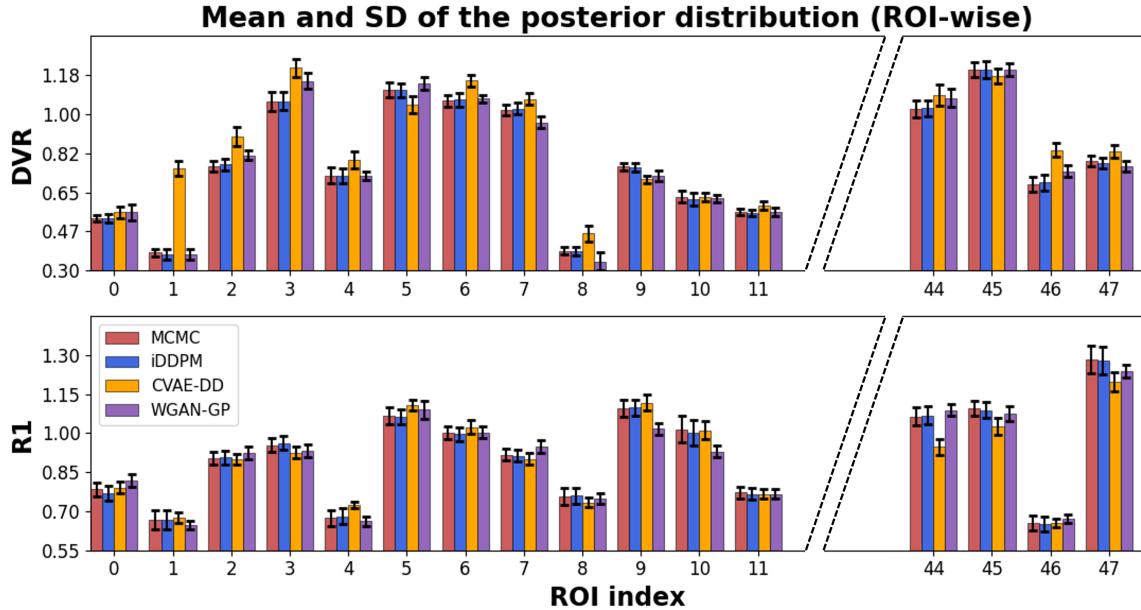


Fig. 5: Bar plot representing the posterior distribution of kinetic parameters DVR (top) and R_1 (bottom) per ROI for a testing measurement y^* . Bars represent the mean of the posterior distribution for MCMC (red), the proposed method (blue), CVAE-DD (yellow), and WGAN-GP (purple). The black error-bars represent the corresponding standard deviation.

a bar plot in Fig 5. Due to space constraints, and for visual clarity, 16 out of 48 regions are displayed. This figure shows the posterior distribution of kinetic parameters DVR (top) and R_1 (bottom) per ROI for a given TAC y^* from the testing set. The mean of the estimated posterior distribution using the MCMC method, the proposed method, the CVAE-DD method and the WGAN-GP method are respectively represented by the red, blue, yellow, and purple bars. The standard deviations of the corresponding posterior distribution are represented by the black error-bars. We can notice that the distribution predicted by the proposed method is consistent with the MCMC method, both in terms of mean and standard deviation. On the other hand, while the estimated mean from the compared methods is not too far from the MCMC method, the SD is biased for many ROIs.

To validate the performance of the proposed method, the above process was repeated over the $N_{\text{test}} = 20$ TACs from the testing set. Mean, SD values and Effective sampling size (16) were estimated from the resulting posterior distributions. The ESS value averaged over the $N_{\text{test}} = 20$ estimated posterior distributions is displayed in Table I for MCMC, the proposed method, and compared method. For visual clarity, 16 regions out of 48 are displayed in the table. In terms of sampling, we can notice that MCMC is about half as efficient as the proposed method, with an average ESS across ROIs of 4202 as opposed to 9846 for the proposed method, 9835 for CVAE-DD, and 9879 for WGAN-GP. The low ESS value for MCMC is expected due to its low sampling efficiency in high dimensional spaces. This further justifies the need for a high number of samples required with MCMC to be able to correctly span a multidimensional distribution. Notably, in this instance, it took drawings from four chains with $N_{\text{sample}}^{\text{MCMC}} = 20000$ posterior samples to reach an ESS of 4202. On the other hand, the proposed method and compared

ROI #	ESS _{MCMC}	ESS _{iDDPM}	ESS _{CVAE-DD}	ESS _{WGAN-GP}
(0) Hippocampus	3888	9878	9870	9876
(1) Cortex entorhinal	4631	9878	9868	9910
(2) Cortex fusiform	2497	9858	9801	9915
(3) Cortex inferiortemporal	1587	9873	9804	9861
(4) Cortex parahippocampal	3086	9833	9794	9885
(5) Cortex posteriорcingulate	7115	9881	9863	9804
(6) Cortex superiorfrontal	1908	9887	9915	9936
(7) Cortex superiortemporal	4423	9887	9840	9873
(8) Amygdala	2215	9898	9931	9888
(9) Thalamus	3881	9884	9939	9862
(10) Accumbens-area	2740	9835	9901	9916
(11) Brain-Stem	7265	9857	9708	9890
Mean	4202	9846	9835	9879

TABLE I: Average ESS of the samples drawn from the posterior distribution on $N_{\text{test}} = 20$ testing TACs for MCMC, the proposed method (iDDPM) and the compared methods (CVAE-DD, WGAN-GP). The ESS displayed are averaged for DVR and R_1 . The last row corresponds to the mean ESS across ROIs.

methods exhibit high ESS values: their ESS is close to the number of samples drawn ($N_{\text{sample}}^{\text{DL}} = 10000$ samples). This is expected in the sense that samples drawn from the approximated posterior $p_{\theta}(\mathbf{x}|y^*)$ are inherently uncorrelated.

Using the mean and SD of the $N_{\text{test}} = 20$ obtained posterior distributions; ROI-wise mean absolute percentage errors (14) for mean and SD were then quantified using MCMC as reference. The results are presented in Table II. For visual clarity, 16 regions out of 48 are displayed in the table. While the CVAE-DD method approximates decently the mean of the posterior distribution (about 8% and 5% error in terms of mean), it exhibits large errors in terms of SD for DVR and R_1 (about 39% error in average for DVR, and 20% error for

R_1). The WGAN-GP method achieves better performance in terms of mean posterior error (about 4% error), and overall moderate advancement in performance in terms of posterior standard deviation approximation (about 31% error in average for DVR, and 23% error for R_1). In comparison, the posterior distribution predicted by the proposed diffusion model closely matches the posterior distribution estimated with MCMC, as reflected by the relatively low errors in terms of mean and SD. More specifically, the prediction of the proposed method exhibits substantially high accuracy (< 0.7% average error in terms of mean) and high degree of precision (about 7% average error in terms of SD).

B. Performance comparison across training epochs: DDPM, iDDPM and compared methods

The results in Table II outline the performance of the proposed method for a given amount of training epochs (see Section III-B.2, III-C.2 and III-C.3). Additionally, another interesting point is the stability of the proposed method during training. Training the diffusion model to estimate the posterior was quite stable: the model converged and yielded good results with a wide choice of hyperparameters, with low overfitting, and displayed increasing performances with more training epochs. Moreover, learning the variance of the noise diffusion process allowed for better maximization of the log-likelihood, and therefore better mode coverage. In the proposed framework, mode coverage is crucial as the whole posterior distribution is of interest, hence the choice of using iDDPM over regular DDPM.

Fig. 6 displays the average error across ROIs $\bar{\delta}$ (i.e., last row of Table II) plotted against the number of training epochs for the proposed method, the compared methods (CVAE-DD and WGAN-GP) and regular DDPM. The first row shows the average error in terms of posterior mean for DVR (left) and for R_1 (right); and the second row shows the average error in terms of posterior standard deviation for DVR (left) and for R_1 (right). For the error in terms of posterior mean, all methods exhibit a downward trend, with decreasing errors as training progresses. Although the error for the proposed method and DDPM are much lower than the compared methods, all methods seem to yield satisfactory results. However, an important aspect of this work is also to accurately predict the shape and standard deviation of the posterior distribution. For the CVAE-DD, the lowest error in terms of standard deviation is reached around 200 epochs, with an upward trend with more epochs. This increase in posterior standard deviation error was due to the inferred distributions shrinking in width around its mean (i.e., standard deviation smaller and smaller with training epochs, while mean becomes more accurate), demonstrating challenging training for this method due to overfitting. For the WGAN-GP, the standard deviation error begins at a low level, reaches its best performance around 150 epochs, and increases afterward. The DDPM method prevails over the compared methods with optimal performances between 250 and 350 epochs, reaching approximately 10-15% error for DVR and R_1 , and increasing as well afterward. In comparison, the error in terms of standard deviation of

proposed methods (iDDPM) steadily and smoothly decreases with training epochs, reaching approximately 7% error around 400-500 epochs, and outperforming DDPM. Note that both DDPM and iDDPM were trained using the exact same network architecture and hyperparameters (i.e., same U-Net, optimizer, learning rate, etc.) at the exception of λ_{vlb} which does not apply for regular DDPM.

C. Effect of domain shift on testing data

Results from Section IV-B (Table II and Fig. 6) demonstrated the accurate posterior distribution estimation using the proposed method on testing data drawn from same the distribution (Section III-A.2) than the training data with which the model was trained. However, in practice, the distribution of the training data may differ to some extent from that of the testing data — a phenomenon known as domain shift, which may negatively impact the model’s accuracy and generalization. In this section, we evaluated the effect of domain shift on the proposed method. More specifically, the proposed method was trained on data drawn from the prior distribution defined in III-A.2, and inferred on testing data drawn from 4 different distributions. In addition to the testing set defined in III-C.1, 3 additional testing sets were drawn from distributions $\tilde{p}^{(n)}(\mathbf{x})$ with mean values equal to μ shifted by a multiple n of the square root of the prior distribution covariance matrix diagonal terms $\Sigma_{r,r}$ (11). The corresponding TACs \mathbf{y} were then generated using the noisy forward model (12), as described in Section III-A.3. Formally, the distribution $\tilde{p}^{(n)}(\mathbf{x})$ can be written as:

$$\begin{aligned}\tilde{\mu}_r^{(n)} &= \mu_r + n \sqrt{\Sigma_{r,r}}, \quad r \in [1, \dots, n_{ROI}] \\ \tilde{p}^{(n)}(\mathbf{x}) &\sim \mathcal{N}(\tilde{\mu}^{(n)}, \Sigma)\end{aligned}\quad (17)$$

The testing sets with $N_{test} = 20$ TACs measurements \mathbf{y} for $n = [0, 1, 2, 4]$ were produced as in Section III-C.1. MCMC was performed for each set as in Section III-C.4 to obtain reference posterior distributions, and the proposed method was used to infer the approximated posterior distributions. For both DVR and R_1 , the mean and standard deviation were computed from these distributions, resulting in $N_{test} \times n_{ROI}$ values per metric for each testing set. The results are displayed in a scatter plot in Figure 7, where each point represents the computed metric (mean or standard deviation of DVR or R_1) for a specific testing sample and ROI. As observed, the posterior distributions estimated with the proposed method for $n = 0$ (no shift, orange stars) and $n = 1$ (green squares) closely match the identity line for all metrics, demonstrating good agreement with the MCMC reference. For $n = 2$ (blue diamonds), one can see that points are more spread out and further away from the identity line, especially for σ_{DVR} σ_{R_1} , while still remaining fairly close to the identity line. Finally, the case $n = 4$ (red circles) demonstrates the largest bias among the 4 testing sets. The mean absolute percentage error (MAPE) across testing measurements and ROIs — computed similarly to the last row of Table II — can be also be evaluated for each metric and testing set. As expected, the highest error was observed for the testing set with $n = 4$, yielding a MAPE

ROI #	iDDPM				CVAE-DD				WGAN-GP			
	$\bar{\delta}_{\mu}^{\text{DVR}}$	$\bar{\delta}_{\mu}^{R_1}$	$\bar{\delta}_{\sigma}^{\text{DVR}}$	$\bar{\delta}_{\sigma}^{R_1}$	$\bar{\delta}_{\mu}^{\text{DVR}}$	$\bar{\delta}_{\mu}^{R_1}$	$\bar{\delta}_{\sigma}^{\text{DVR}}$	$\bar{\delta}_{\sigma}^{R_1}$	$\bar{\delta}_{\mu}^{\text{DVR}}$	$\bar{\delta}_{\mu}^{R_1}$	$\bar{\delta}_{\sigma}^{\text{DVR}}$	$\bar{\delta}_{\sigma}^{R_1}$
(0)	0.46	0.59	9.55	5.37	8.69	2.69	28.95	24.81	4.09	3.84	41.47	13.75
(1)	1.25	0.8	8.03	4.02	19.02	4.22	71.3	15.3	4.03	4.29	53.84	20.21
(2)	0.66	0.37	6.03	8.7	10.41	4.06	53.66	13.91	3.4	3.28	33.61	26.46
(3)	0.52	0.51	4.03	10.52	8.7	4.41	46.66	11.48	3.58	3.06	32.32	19.39
(4)	0.63	0.69	3.3	5.72	6.68	4.8	47.98	24.12	6.5	2.63	53.27	21.12
(5)	0.33	0.41	6.94	9.34	6.05	3.81	46.27	14.15	3.59	2.12	19.86	20.66
(6)	0.31	0.48	11.17	8.05	9.04	2.19	29.34	13.6	3.31	2.56	27.81	18.97
(7)	0.38	0.32	6.74	3.65	6.37	3.71	19.64	16.48	3.79	2.65	21.21	25.25
(8)	0.65	0.55	5.49	5.24	13.87	5.77	63.83	18.1	7.95	4.45	47.99	18.07
(9)	0.46	0.29	8.33	4.85	7.61	4.44	16.82	28.89	5.48	5.03	32.3	17.04
(10)	0.77	0.66	5.85	4.13	7.32	5.38	40.02	14.79	4.63	5.39	39	51.54
(11)	0.63	0.31	10.8	3.69	7.63	3.44	27.61	23.94	5.5	2.65	39.89	16.05
...
(44)	0.67	0.54	5.1	3.75	12.22	6.28	51.42	14.8	0.75	0.8	6.56	9.14
(45)	0.55	0.48	9.72	7.47	5.76	5.08	37.96	12.3	0.91	0.61	20.37	13.75
(46)	1.07	1.01	8.92	6.51	9.48	3.44	53.47	25.77	0.99	0.33	14.58	9.7
(47)	0.98	0.78	5.86	4.37	6.29	7.71	19.04	27.31	0.51	0.47	10.16	4.13
Mean	0.67	0.57	7.23	6.09	8.13	5.10	38.74	19.85	4.73	4.01	31.44	23.41

Values expressed in percent (%)

TABLE II: Mean absolute percentage error of the mean and standard deviation of the estimated posterior $p_{\theta}(\mathbf{x}|\mathbf{y}^*)$. The MCMC metrics were used as reference to calculate the errors, as in (14). Errors were averaged on results from $N_{\text{test}} = 20$ testing TACs. The last row corresponds to the mean error across ROIs ($\bar{\delta} = 1/n_{\text{ROI}} \sum_r \bar{\delta}_r$).

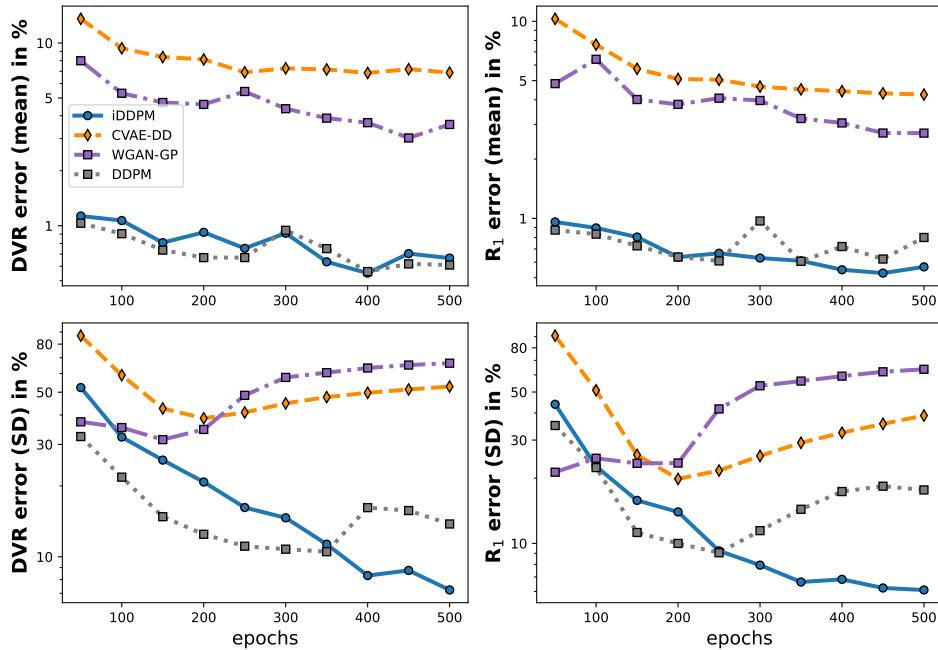


Fig. 6: Mean absolute percentage errors per epoch trained for the proposed method (iDDPM, blue), CVAE-DD (orange), WGAN-GP (purple), and DDPM (gray). The MCMC metrics were used as reference to calculate the errors, as in (14). Note that DDPM was trained with the same architecture and hyperparameters as iDDPM. Percentage error (14) in terms of mean (top) and standard deviation (bottom) are shown for DVR (left) and R₁ (right) over 500 training epochs. The errors displayed were averaged on results from $N_{\text{test}} = 20$ testing TACs. These errors reflect the average bias in estimating the mean of the posterior distribution (top) and the average bias in estimating the standard deviation (bottom) of the posterior distribution for different methods. As can be seen, the proposed method outperforms the compared methods and the DDPM with steadily decreasing error with training epochs.

of $\bar{\delta}_{\mu}^{\text{DVR}} = 5.48\%$ and $\bar{\delta}_{\sigma}^{\text{DVR}} = 14.44\%$ for the mean and standard deviation of DVR, respectively; and a MAPE of $\bar{\delta}_{\mu}^{R_1} = 1.57\%$ and $\bar{\delta}_{\sigma}^{R_1} = 15.58\%$ for the mean and standard deviation of R₁. While these errors are higher than those reported for $n = 0$ (no shift) in Table II (last row), they remain relatively low, highlighting the robustness and adaptability of

the proposed method under domain shift.

D. in vivo results

Once the proposed method was validated on testing data drawn from the prior, the network was fine-tuned with a

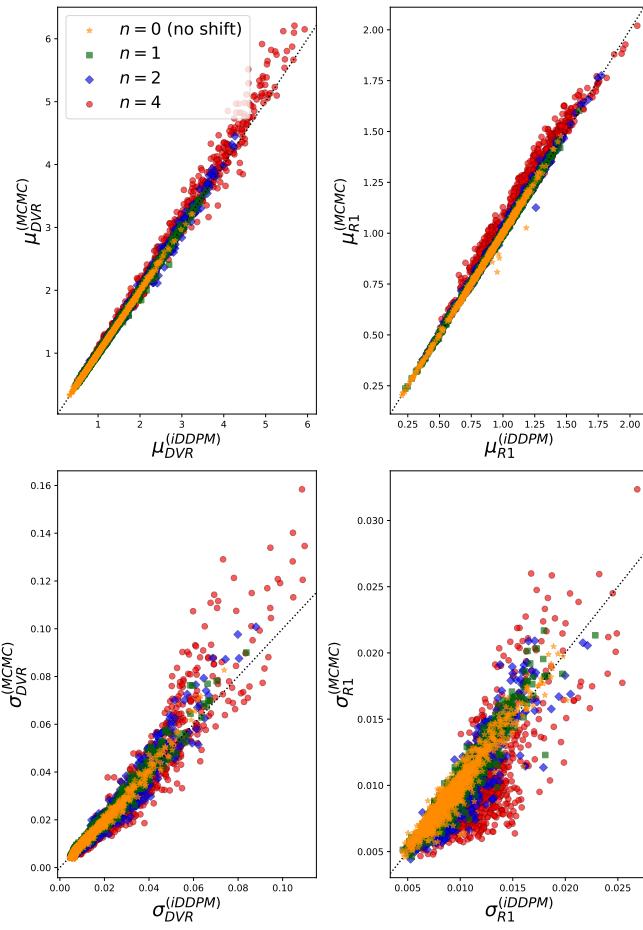


Fig. 7: Scatter plot of the mean (top) and standard deviation (bottom) of the estimated posterior distributions with the proposed method (x -axis) vs. the MCMC reference (y -axis). The dotted line represents the identity line $y = x$. The proposed method was trained on data drawn from the prior distribution detailed in III-A.2. Posterior distributions were then inferred on 4 testing sets designed to model domain shift. The 4 testing sets were drawn from distributions with shifted mean values compared to $p(\mathbf{x})$, i.e., $\tilde{p}^{(n)}(\mathbf{x}) \sim \mathcal{N}(\mu + n \text{ diag}(\Sigma), \Sigma)$, with n respectively equal to 0 (i.e., no shift, orange stars), 1 (green squares), 2 (blue diamonds), and 4 (red circles).

low learning rate on *in vivo* data and applied to the *in vivo* PET TACs from the testing set (see Section III-B.3). Fig. 8 shows the results of the posterior distribution obtained using the proposed method on a subject with Alzheimer’s disease (left column) and on a control subject (right column). More specifically, the first row displays the structural MRI (MPRAGE) used for ROI analysis, rows 2 and 3 display the posterior distribution mean and SD for DVR, and rows 4 and 5 display the posterior distribution mean and SD for R_1 . As observed for the AD subject, certain brain regions such as the inferiortemporal and the inferiorparietal cortex exhibit high mean DVR values, suggesting a presence of p-tau burden in those areas. Beyond the mean DVR, the standard deviation values provide additional insight and help to express the measurement uncertainty. This information can prove to be valuable in clinical settings, enabling margins and corrections of estimated DVR (e.g., using credible intervals). On the other hand, for the control subject, the mean DVR values appear considerably lower. The highest DVR value

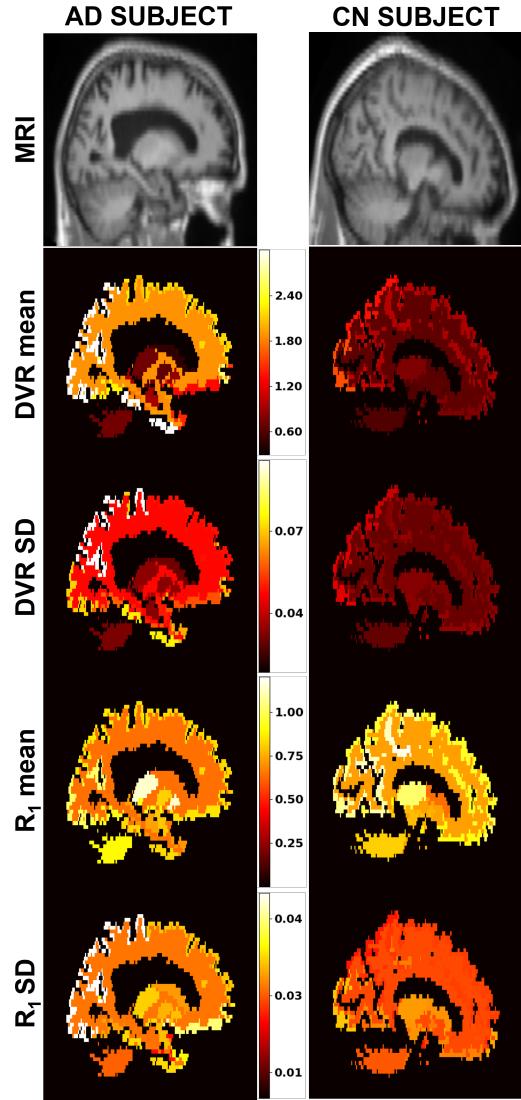


Fig. 8: Mean and standard deviation of the posterior distribution for the kinetic parameters DVR (rows 2 and 3) and R_1 (rows 4 and 5) for an AD subject (left column) and a cognitively normal (CN) subject (right column). First row displays the structural MRI used for anatomical reference and ROI masks. The posterior distribution was obtained with the proposed method using the TACs from the *in vivo* testing set. High DVR values in regions such as the inferiortemporal and inferiorparietal cortex suggest p-tau protein burden for the AD subject, while the standard deviation indicates the level of confidence in these values.

observed is around 1.5, which confirms the presence of a healthy subject. In addition, the standard deviation values here are also much lower (approximately 5 times lower), indicating a high confidence in the inferred DVR values.

The obtained kinetic parameters were also compared with the kinetic models used in the forward model. More precisely, the Logan equation (9) was used on the *in vivo* TACs to plot the first row in Fig. 9 (green lines) for an AD subject (left) and a control subject (right). Using the proposed method, we obtained 10000 DVR values drawn from the posterior distribution, and plotted the Logan equation with the posterior mean DVR for slope (blue dashed line). An error range (red area) is represented using the minimum and maximum DVR values as slopes in (9). Both the posterior mean and the error

range are in accordance with the DVR value estimated from the Logan fitting, which also gives an idea of uncertainty on the slope fitting from a Bayesian perspective.

The second row of Fig. 9 displays the *in vivo* TACs (green lines) of an AD subject for the first 10 minutes. The black solid line corresponds to the synthesized TAC using the SRTM2 forward model with the posterior mean DVR and R_1 . The red area corresponds to the maximum error range of the synthesized TACs from the 10000 DVR and R_1 values drawn from the posterior distribution using the proposed method. The black error bars represent the standard deviation of the 10000 produced TACs at each time point. Once again, the proposed method yields conclusive results: the synthesized TAC from the posterior mean is reasonably close to the *in vivo* TAC, while the maximum error range indicates a range of plausible TACs from the posterior distribution kinetic parameters.

V. DISCUSSIONS

The main goal of this study was to propose an alternative to MCMC methods to obtain a posterior distribution based on measured TACs. This work presented a case with 48 ROIs, showcasing the speed and accuracy of the proposed method. Posterior distributions were estimated over 230 times faster than MCMC using the proposed method, with less than 0.7% error in terms of mean and less than 7.3 % error in terms of standard deviation. The method presented in this article is also versatile: it can be trained and used with any number of regions. In [34], we demonstrated the usefulness and performance of the proposed method with 8 ROIs with 30 times faster posterior distribution inference. This versatility can prove useful, giving the option to only process regions that are of interest for the study (e.g., if memory available on a GPU is limited), or the whole brain. The findings presented here extend the aforementioned work in several ways: we used a greater number of ROIs covering most of the brain, took into account ROI-wise correlation of DVR and R_1 assuming multidimensional prior and posterior distribution (thus excluding the possibility of decomposition into n_{ROI} 1-dimensional problems), incorporated a more intricate data and noise model, and compared results with two promising generative DL-based methods (CVAE-DD, WGAN-GP). In addition, we provided insight on training stability and performance of the proposed and the compared methods, and included a comparison with regular DDPM [18] (Fig. 6). This highlighted the superior performance in terms of posterior distribution approximation when the variance of the reverse diffusion process is learned with an iDDPM [16].

Note that the modifications and improvements made in the proposed method greatly increased the complexity of the data structure and of the posterior distribution, explaining the reduced performance of the CVAE-DD compared to [13]. While the added loss term (i.e., TAC reconstruction loss) helped during training compared to regular CVAE, it remained more challenging than the proposed method to train and the performances were sensitive to the optimizer and the hyperparameters choice (such as β_{KL} , or the number of training epochs). Without good parameter choice, the training

could then become unstable, notably due to KL vanishing and/or overfitting. A potential cause for this instability could be the fully connected architecture used in the encoder-decoder network, as implemented in [13], which may pose challenges when dealing with higher-dimensional data as is the case for GANs [35]. Investigating alternative architectures for the CVAE-DD falls outside the scope of this study.

As anticipated, MCMC inference time greatly increased as the number of regions increased, rendering it impractical with high dimensional data. In this work, we demonstrate a significant acceleration in posterior distribution inference time. Such performance highlights the increasing appeal of leveraging DL-based methods for posterior inference in the context of high-dimensional data. In addition to needing more burn-in steps to reach convergence, MCMC also required more tuning and more samples to correctly span the parameter space of high-dimensional data, as indicated by the low ESS in Table I. A considerable advantage of the proposed method over MCMC is that the samples are by design independent, i.e., the number of samples N produced by the diffusion model are roughly equal to the effective sampling size. This suggests that to attain a similar parameter space coverage as achieved by the proposed method, a considerable number of MCMC samples with multiple chains would be necessary ($N_{\text{sample}}^{\text{MCMC}} \gg N_{\text{sample}}^{\text{DL}}$). As a side note, Liu *et al.* reported an acceleration factor of 40 between the CVAE-based methods and MCMC in [13]. In the proposed work, the higher complexity of the model and higher dimensionality of the data made it more challenging for MCMC to converge, needing much more time to infer posterior samples. In contrast, deep learning methods benefit from greater efficiency in inference time, aided by GPU parallelization, and do not suffer from the same dimensionality challenges as MCMC. While the proposed method offers an inference speedup of 230 times, it is worth noting that such speedup would be higher with CVAE-DD or WGAN-GP, although at the cost of significantly higher error in terms of posterior distribution standard deviation.

Regarding the posterior distribution of the *in vivo* measurements, one can notice differences in terms of standard deviation between the AD and CN cohorts (Fig. 8). This can be explained using Bayesian statistic principles: the posterior distribution is proportional to the prior distribution and the data likelihood. Different standard deviations can then be caused by either a deviation from the prior mean, or a significant amount of noise (likelihood), or both. Because a limited number of studies were available, the prior in the proposed study was defined using data from CN, MCI, and AD subjects, with CN subjects dominating because they are the majority. If we assume the noise level is the same for all subjects, the high prior density around the kinetic parameters typical of CN subjects means that combining the likelihood with this prior yields a more concentrated (narrow) posterior distribution for the CN subject. Conversely, the prior is weaker for the AD subject, so the corresponding posterior mass has to spread out more. In deep learning, the training (and testing) data pairs (\mathbf{x}, \mathbf{y}) implicitly define a data distribution. This work defined an explicit prior distribution obtained from an *in vivo* study to train a neural network, i.e., to adapt to the data distribution,

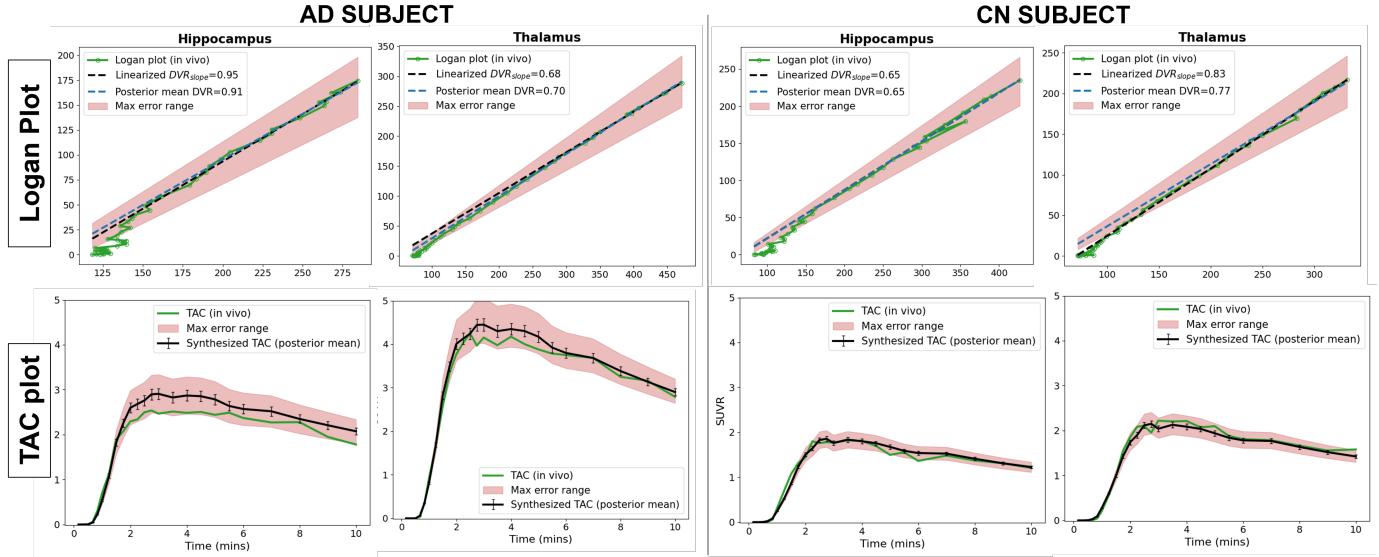


Fig. 9: Logan plot (green lines, 1st row) and TAC plot (green lines, 2nd row) for an AD subject (left) and a control subject (right) for the Hippocampus (left column) and the Thalamus (right column). In the first row, the black dashed line represents the estimated linearized curve using Logan's plot equation (9) ($y_{\text{Logan}} = \text{DVR}_{\text{slope}} x_{\text{Logan}} + \gamma$) while the blue dashed lines use the posterior mean DVR for slope. The red area represents the maximum error range, replacing $\text{DVR}_{\text{slope}}$ with the minimum and maximum DVR values from $p_{\theta}(\mathbf{x}|\mathbf{y})$. In the second row, the black solid line corresponds to the synthesized TAC using the SRTM2 forward model with the posterior mean DVR and R_1 values. The red area corresponds to the maximum error range of the synthesized TACs from the 10000 DVR and R_1 values drawn from $p_{\theta}(\mathbf{x}|\mathbf{y})$, and the black error bars to the standard deviation at each time point.

and infer posterior distributions. Eventually, fine-tuning is performed using the implicit prior from the *in vivo* dataset, obtaining posterior distributions for the kinetic parameters of an *in vivo* study. One of the main limitations of this work is the limited amount of *in vivo* data available for training and testing. With more data available, prior distributions could be obtained from different cohorts (i.e., CN, MCI, and AD) and better fine-tuning could be applied (more data available means better estimation of data distribution).

The current study validated the method with ROI-wise TACs. Yet, the proposed methodology for posterior inference does not rely on the ROI-wise estimation and could be applied voxel-wise, i.e., estimating the posterior distributions on 2D or 3D PET images voxel-wise. In such case, in addition to the input TAC measurements, one could also leverage the anatomical MRI as input of the neural network, and consider it as prior information. Voxel-wise posterior distribution estimation rather than ROI-wise also implies new challenges such as: what prior model to use, how to estimate the noise level in the forward model on a voxel-wise basis, and more importantly how to validate the obtained posterior distribution from a DL-based method. The computational feasibility of the proposed method was internally validated on 128×128 images, requiring approximately 3 minutes to produce 200 posterior image samples. However, validation with MCMC proves to be challenging in this scenario due to its high computational demand. For a 128×128 image, the posterior distribution would need to be estimated on $128^2 = 16384$ parameters, which is roughly 341 times larger than the 48 ROIs proposed in this work. Assuming MCMC could converge in such a high-dimensional space at the same speed as in this work (a challenging task due to the curse of dimensionality and reduced sampling efficiency), generating posterior samples

would take around 57 days, making it impractical. In this scenario, likelihood-free techniques such as ABC can prove to be useful, reducing the computational needs at the cost of approximate posterior distributions used as reference. Research on voxel-wise posterior is ongoing research and will be shared in future publications.

VI. CONCLUSIONS

We proposed a diffusion model-based approach to estimate the posterior distribution of kinetic parameters using dynamic PET brain data. The proposed method was evaluated and validated on a [18F]MK6240 study. Posterior distributions inferred using the proposed method were obtained over 230 times faster than the conventional method, with substantially high accuracy and precision, and only requiring target and reference time-activity curves as inputs. This approach could prove valuable in *in vivo* studies related to neurodegeneration, enabling the quantification of p-tau burden along with its associated uncertainty.

REFERENCES

- [1] A. Leuzy, K. Chiotis, L. Lemoine, P.-G. Gillberg, O. Almkvist, E. Rodriguez-Vieitez, and A. Nordberg, "Tau PET imaging in neurodegenerative tauopathies—still a challenge," *Molecular Psychiatry*, vol. 24, no. 8, pp. 1112–1134, aug 2019.
- [2] J. Logan, J. S. Fowler, N. D. Volkow, A. P. Wolf, S. L. Dewey, D. J. Schlyer, R. R. MacGregor, R. Hitzemann, B. Bendriem, S. J. Gatley, and D. R. Christman, "Graphical Analysis of Reversible Radioligand Binding from Time-Activity Measurements Applied to [N-11C-Methyl]-(-)-Cocaine PET Studies in Human Subjects," *Journal of Cerebral Blood Flow and Metabolism*, vol. 10, no. 5, pp. 740–747, 1990.
- [3] A. A. Lammertsma and S. P. Hume, "Simplified Reference Tissue Model for PET Receptor Studies," *NeuroImage*, vol. 4, no. 3, pp. 153–158, dec 1996.
- [4] R. N. Gunn, A. A. Lammertsma, S. P. Hume, and V. J. Cunningham, "Parametric imaging of ligand-receptor binding in PET using a simplified reference region model," *NeuroImage*, vol. 6, no. 4, pp. 279–287, nov 1997.

- [5] Y. Wu and R. E. Carson, "Noise reduction in the simplified reference tissue model for neuroreceptor functional imaging," *Journal of Cerebral Blood Flow and Metabolism*, vol. 22, no. 12, pp. 1440–1452, 2002.
- [6] M. Ichise, J. S. Liow, J. Q. Lu, A. Takano, K. Model, H. Toyama, T. Suhara, K. Suzuki, R. B. Innis, and R. E. Carson, "Linearized reference tissue parametric imaging methods: Application to [11C]DASB positron emission tomography studies of the serotonin transporter in human brain," *Journal of Cerebral Blood Flow and Metabolism*, vol. 23, no. 9, pp. 1096–1112, 2003.
- [7] J.-M. Marin, P. Pudlo, C. P. Robert, and R. J. Ryder, "Approximate Bayesian computational methods," *Statistics and Computing*, vol. 22, no. 6, pp. 1167–1180, 2012.
- [8] S. Chib, "Markov Chain Monte Carlo Methods: Computation and Inference," in *Handbook of Econometrics*. Elsevier, jan 2001, vol. 5, pp. 3569–3649.
- [9] J. S. Speagle, "A Conceptual Introduction to Markov Chain Monte Carlo Methods," 2019.
- [10] D. P. Kingma and M. Welling, "Auto-Encoding Variational Bayes," in *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.
- [11] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Advances in Neural Information Processing Systems*, vol. 27, 2014.
- [12] R. T. Q. Chen, Y. Rubanova, J. Bettencourt, and D. K. Duvenaud, "Neural ordinary differential equations," in *Advances in Neural Information Processing Systems*, vol. 31, 2018.
- [13] X. Liu, T. Marin, T. Amal, J. Woo, G. E. Fakhri, and J. Ouyang, "Posterior estimation using deep learning: a simulation study of compartmental modeling in dynamic positron emission tomography," *Medical Physics*, vol. 50, no. 3, pp. 1539–1548, mar 2023.
- [14] Y. Zhao, Q. Gu, A. Reader, Y. Fan, and S. Meikle, "Posterior Parameter Estimation in Compartmental Models from Dynamic PET: a Simulation Study of Feedback-enhanced CVAE," in *2024 IEEE Medical Imaging Conference (MIC)*, 2024, pp. 1–1.
- [15] J. Adler and O. Öktem, "Deep Bayesian Inversion," 2018.
- [16] A. Nichol and P. Dhariwal, "Improved Denoising Diffusion Probabilistic Models," *Proceedings of Machine Learning Research*, vol. 139, pp. 8162–8171, 2021.
- [17] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville, "Improved training of wasserstein GANs," *Adv. Neural Inf. Process. Syst.*, vol. 2017–December, pp. 5768–5778, 2017.
- [18] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *Advances in Neural Information Processing Systems*, vol. 2020–Decem, no. NeurIPS 2020, pp. 1–25, 2020.
- [19] A. Razavi, A. van den Oord, and O. Vinyals, "Generating diverse high-fidelity images with VQ-VAE-2," *Advances in Neural Information Processing Systems*, vol. 32, no. NeurIPS 2019, pp. 1–11, 2019.
- [20] J. Logan, J. S. Fowler, N. D. Volkow, G.-j. Wang, Y.-s. Ding, and D. L. Alexoff, "Distribution Volume Ratios without Blood Sampling from Graphical Analysis of PET Data," *Journal of Cerebral Blood Flow and Metabolism*, vol. 16, no. 5, pp. 834–840, sep 1996.
- [21] F. Zanderigo, R. T. Ogden, and R. V. Parsey, "Reference Region Approaches in PET: a Comparative Study on Multiple Radioligands," *Journal of Cerebral Blood Flow and Metabolism*, vol. 33, no. 6, pp. 888–897, jun 2013.
- [22] M. Jenkinson, C. F. Beckmann, T. E. Behrens, M. W. Woolrich, and S. M. Smith, "FSL," *NeuroImage*, vol. 62, no. 2, pp. 782–790, aug 2012.
- [23] B. Fischl, A. Van Der Kouwe, C. Destrieux, E. Halgren, F. Ségonne, D. H. Salat, E. Busa, L. J. Seidman, J. Goldstein, D. Kennedy, V. Caviness, N. Makris, B. Rosen, and A. M. Dale, "Automatically Parcellating the Human Cerebral Cortex," *Cerebral Cortex*, vol. 14, no. 1, pp. 11–22, jan 2004.
- [24] N. J. Guehl, M. Dhaynaut, B. J. Hanseeuw, S. H. Moon, C. Lois, E. Thibault, J. F. Fu, J. C. Price, K. A. Johnson, G. E. Fakhri, and M. D. Normandin, "Measurement of Cerebral Perfusion Indices from the Early Phase of [18F]MK6240 Dynamic Tau PET Imaging," *Journal of Nuclear Medicine*, vol. 64, no. 6, pp. 968–975, 2023.
- [25] N. J. Guehl, D. W. Wooten, D. L. Yokell, S. H. Moon, M. Dhaynaut, S. Katz, K. A. Moody, C. Gharagouzloo, A. Kas, K. A. Johnson, G. El Fakhri, and M. D. Normandin, "Evaluation of pharmacokinetic modeling strategies for in-vivo quantification of tau with the radiotracer [18F]MK6240 in human subjects," *European Journal of Nuclear Medicine and Molecular Imaging*, vol. 46, no. 10, pp. 2099–2111, 2019.
- [26] T. J. Betthauser, K. A. Cody, M. D. Zammit, D. Murali, A. K. Converse, T. E. Barnhart, C. K. Stone, H. A. Rowley, S. C. Johnson, and B. T. Christian, "In vivo characterization and quantification of neurofibrillary tau pet radioligand 18f-mk-6240 in humans from alzheimer disease dementia to young controls," *Journal of Nuclear Medicine*, vol. 60, no. 1, pp. 93–99, 2019.
- [27] H. Barrett, D. Wilson, and B. Tsui, "Noise properties of the EM algorithm. I. Theory," *Physics in Medicine & Biology*, vol. 39, no. 5, p. 833, may 1994.
- [28] E. Soares, C. Byrne, and S. Glick, "Noise characterization of block-iterative reconstruction algorithms. I. Theory," *IEEE Transactions on Medical Imaging*, vol. 19, no. 4, pp. 261–270, 2000.
- [29] E. Soares, S. Glick, and J. Hoppin, "Noise characterization of block-iterative reconstruction algorithms: II. Monte Carlo simulations," *IEEE Transactions on Medical Imaging*, vol. 24, no. 1, pp. 112–121, 2005.
- [30] J. Varga and Z. Szabo, "Modified regression model for the Logan plot," *Journal of Cerebral Blood Flow and Metabolism*, vol. 22, no. 2, pp. 240–244, 2002.
- [31] E. J. Li, B. A. Spencer, J. P. Schmall, Y. Abdelhafez, R. D. Badawi, G. Wang, and S. R. Cherry, "Efficient Delay Correction for Total-Body PET Kinetic Modeling Using Pulse Timing Methods," *Journal of nuclear medicine : official publication, Society of Nuclear Medicine*, vol. 63, no. 8, pp. 1266–1273, 2022.
- [32] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in Neural Information Processing Systems*, vol. 2017–Decem, no. Nips, pp. 5999–6009, 2017.
- [33] R. De Maesschalck, D. Jouan-Rimbaud, and D. L. Massart, "The Mahalanobis distance," *Chemometrics and Intelligent Laboratory Systems*, vol. 50, no. 1, pp. 1–18, jan 2000.
- [34] Y. Djebra, X. Liu, T. Marin, A. Tiss, M. Dhaynaut, N. Guehl, K. Johnson, G. El Fakhri, C. Ma, and J. Ouyang, "Diffusion model-based posterior distribution prediction for kinetic parameter estimation in dynamic PET (accepted)," in *2024 IEEE 21th International Symposium on Biomedical Imaging (ISBI)*, 2024.
- [35] A. Radford, L. Metz, and S. Chintala, "Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks," *4th Int. Conf. Learn. Represent. ICLR 2016 - Conf. Track Proc.*, pp. 1–16, nov 2015.
- [36] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein GAN," *Icml*, vol. 1, pp. 1–44, 2017.
- [37] J. Salvatier, T. V. Wiecki, and C. Fonnesbeck, "Probabilistic programming in Python using PyMC3," *PeerJ Computer Science*, vol. 2016, no. 4, pp. 1–20, 2016.
- [38] O. Abril-Pla, V. Andreani, C. Carroll, L. Dong, C. J. Fonnesbeck, M. Kochurov, R. Kumar, J. Lao, C. C. Luhmann, O. A. Martin, M. Osthege, R. Vieira, T. Wiecki, and R. Zinkov, "PyMC: a modern, and comprehensive probabilistic programming framework in Python," *PeerJ Computer Science*, vol. 9, pp. 1–35, 2023.
- [39] A. Vehtari, A. Gelman, D. Simpson, B. Carpenter, and P. C. Burkner, "Rank-Normalization, Folding, and Localization: An Improved (Formula presented) for Assessing Convergence of MCMC (with Discussion)*†," *Bayesian Analysis*, vol. 16, no. 2, pp. 667–718, 2021.
- [40] C. J. Geyer, "Practical markov chain monte carlo," *Statistical Science*, vol. 7, no. 4, pp. 473–483, 1992.