Statistical Analysis of Longitudinally and Conditionally Generated Neuroimaging Measures via Conditional Recurrent Flow

Seong Jae Hwang[1] Univ. of Pittsburgh
Zirui Tao Univ. of Wisconsin-Madison
Won Hwa Kim[2] Univ. of Texas at Arlington
Vikas Singh[2] Univ. of Wisconsin-Madison

Abstract

We develop a conditional generative model for longitudinal image datasets based on sequential invertible neural networks. Longitudinal image acquisitions are common in various scientific and biomedical studies where often each image sequence sample may also come together with various secondary (fixed or temporally dependent) measurements. The key goal is not only to estimate the parameters of a deep generative model for the given longitudinal data, but also to enable evaluation of how the temporal course of the generated longitudinal samples are influenced as a function of induced changes in the (secondary) temporal measurements (or events). Our proposed formulation incorporates recurrent subnetworks and temporal context gating, which provide a smooth transition in a temporal sequence of generated data that can be easily informed or modulated by secondary temporal conditioning variables. We show that the formulation works well despite the smaller sample sizes common in these applications. Our model is validated on two video datasets and a longitudinal Alzheimer's disease (AD) dataset for both quantitative and qualitative evaluations of the generated samples. Further, using our generated longitudinal image samples, we show that we can capture the pathological progressions in the brain that turn out to be consistent with the existing literature.

1. Introduction

Consider a dataset of *longitudinal sequences* of data samples $\{\mathbf{x}^t\}_{i=1}^N$ where each sample $\mathbf{x}_i$ comes with *sequential covariates* $\{\mathbf{y}^t\}_{i=1}^N$, one for each time point t. We assume that for each sequential sample i, $\mathbf{x}_i^1, \cdots, \mathbf{x}_i^T = \{\mathbf{x}^t\}_i$, the sequential covariates $\mathbf{y}_i^1, \cdots, \mathbf{y}_i^T = \{\mathbf{y}^t\}_i$ provide some pertinent auxiliary information associated with that sequential sample. Our goal is to design conditional generative models for such sequential data. In particular, we want a model which provides us a type of flexibility that is highly desirable in this setting. For instance, for a sample drawn from a distribution after a generative model has been estimated, we should be able to "adjust" the sequential covariates dynamically to influence the expected future predictions after some time t for *that* sample. Notice that when $t = 1$, this construction is similar to conditional generative models where the "covariate" or condition $\mathbf{y}$ may simply denote an attribute that we may want to adjust for a sample.

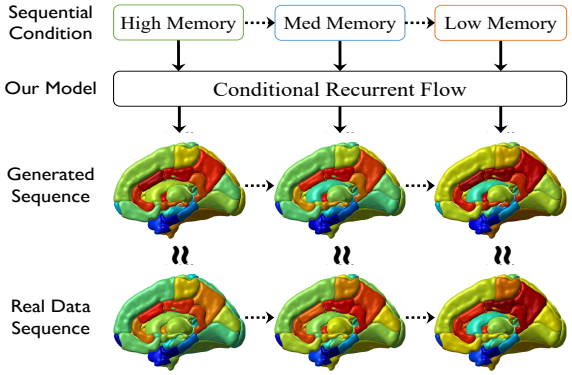


Figure 1: Conditional sequence generation illustration. **1) Given:** a decreasing memory test score sequence $\mathbf{y}_i^1 \rightarrow \mathbf{y}_i^2 \rightarrow \mathbf{y}_i^3$. **2) Model:** Conditional Recurrent Flow (CRow). **3) Generate:** a sequence of brain image progression $\mathbf{x}_i^1 \rightarrow \mathbf{x}_i^2 \rightarrow \mathbf{x}_i^3$ corresponding to the given memory progression (i.e., brain regions with high (red) and low (blue) disease pathology). The Generated sequence follows the trend of the Real Data Sequence.

In this regard, neural network based generative models for modeling complex and high dimensional real life data while allowing tractable Bayesian approaches became feasible recently: Monte Carlo drop-out [8, 15], tractable exponential families [20, 6], and sufficient statistics estimation [18, 19]. These approaches, however, make limited assumptions on the true distribution of the dataset which may not accurately reflect complex datasets we deal with.

We want our formulation to provide a modified set of $\mathbf{x}^t$s adaptively, if we adjust one or more sequential covariates $\mathbf{y}^t$s for that sample. If we know some important clinical information at some point during the study (say, at $t = 5$), this information should influence the future generation $\mathbf{x}^{t>5}$ conditioned both on this sequential covariate or event $\mathbf{y}^5$ as well as the past sequence of this sample $\mathbf{x}^{t<5}$. This will require *conditioning* on the corresponding sequential covariates at *each* time point t by accurately capturing the posterior distribution $p(\mathbf{x}^t|\mathbf{y}^t)$. Such *conditional sequence generation* needs a generative model for *sequential* data which can dynamically incorporate time-specific *sequential*

[1]: Work done while SJH was at the University of Wisconsin-Madison
[2]: Shared senior authorship

covariates $\mathbf{y}^t$ of interest to adaptively modify sequences.

We seek a model which (i) efficiently generates high dimensional sequence samples of *variable lengths* (ii) with dynamic time-specific conditions reflecting upstream observations (iii) with fast posterior probability estimation. We tackle the foregoing issues by introducing an invertible recurrent neural network, **CRow**, that includes **recurrent subnetwork** and **temporal context gating**. *Invertibility* lets us precisely estimate the distribution of $p(\mathbf{x}^t|\mathbf{y}^t)$ in latent space. Introducing *recurrent subnetworks* and *temporal context gating* enables obtaining cues from previous time points $\mathbf{x}^{<t}$ to generate temporally sensible subsequent time points $\mathbf{x}^{\geq t}$. Our **contributions** are: **(A)** Our model generates conditional sequential samples $\{\mathbf{x}^t\}$ given sequential covariates $\{\mathbf{y}^t\}$ for $t = 1, \ldots, T$ time points where T can be arbitrarily long. Specifically, we allow this by posing the task as a *conditional sequence inverse problem* based on a conditional invertible neural network [1]. **(B)** Our model estimates the posterior probabilities $p(\mathbf{x}^t|\mathbf{y}^t)$ of the generated sequences at each time point for potential downstream analyses involving uncertainty. **(C)** We show that the generated longitudinal brain pathology trajectories (e.g., Fig. 1) can lead to identifying specific regions in the brain statistically associated with Alzheimer's disease (AD).

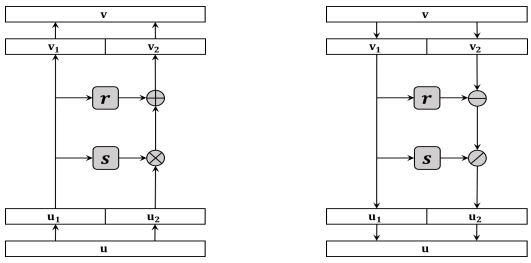
2. Preliminary: Invertible Neural Networks

We first describe an *invertible neural network* (INN) which inverts an output back to its input for solving inverse problems (i.e., $\mathbf{z} = f(\mathbf{x}) \Leftrightarrow \mathbf{x} = f^{-1}(\mathbf{z})$). A *normalizing flow* [22, 21] first learns a function $f(\cdot)$ which maps a sample $\mathbf{x}$ to a latent variable $\mathbf{z} = f(\mathbf{x})$ where $\mathbf{z}$ is from a standard normal distribution $\mathbf{Z}$. Then, with a change of variables formula, we estimate $p_{\mathbf{X}}(\mathbf{x}) = p_{\mathbf{Z}}(\mathbf{z})/|J_{\mathbf{X}}|$ and $|J_{\mathbf{X}}| = \left|\frac{\partial[\mathbf{x}=f^{-1}(\mathbf{z})]}{\partial \mathbf{z}}\right|$ where $|J_{\mathbf{X}}|$ is a Jacobian determinant. Thus, $f(\cdot)$ must be invertible, and to use a neural network as $f(\cdot)$, a *coupling layer* structure was introduced in RealNVP [4, 5] for an easy inversion and efficient $|J_{\mathbf{X}}|$.

Forward map (Fig. 2a). We use an input $\mathbf{u} \in \mathbb{R}^d$ and an output $\mathbf{v} \in \mathbb{R}^d$ (i.e., $\mathbf{u} \rightarrow \mathbf{v}$). We split $\mathbf{u}$ into $\mathbf{u}_1 \in \mathbb{R}^{d_1}$ and $\mathbf{u}_2 \in \mathbb{R}^{d_2}$ where $d = d_1 + d_2$. Then, we forward map $\mathbf{u}_1$ and $\mathbf{u}_2$ to $\mathbf{v}_1$ and $\mathbf{v}_2$ respectively such that $\mathbf{v}_1 = \mathbf{u}_1$ and $\mathbf{v}_2 = \mathbf{u}_2 \otimes \exp(s(\mathbf{u}_1)) + r(\mathbf{u}_1)$ where s and r are independent functions (i.e., subnetworks), and $\otimes$ and $+$ are element-wise product and addition respectively. Then, $\mathbf{v}_1$ and $\mathbf{v}_2$ construct $\mathbf{v}$ (e.g., $[\mathbf{v}_1, \mathbf{v}_2] \rightarrow \mathbf{v}$).

Inverse map (Fig. 2b). A straightforward arithmetic allows an exact inverse from $\mathbf{v}$ to $\mathbf{u}$ (i.e., $\mathbf{v} \rightarrow \mathbf{u}$): $\mathbf{u}_1 = \mathbf{v}_1$ and $\mathbf{u}_2 = (\mathbf{v}_2 - r(\mathbf{v}_1)) \oslash \exp(s(\mathbf{v}_1))$ where the subnetworks s and r are *identical* to those in the forward map. $\oslash$ and $-$ are element-wise division and subtraction respectively.

To transform the "bypassed" split $\mathbf{u}_1$ (since $\mathbf{u}_1 = \mathbf{v}_1$), a *coupling block* consisting of two complementary coupling layers is constructed to transform both $\mathbf{u}_1$ and $\mathbf{u}_2$: $\mathbf{v}_1 = \mathbf{u}_1 \otimes \exp(s_2(\mathbf{u}_2)) + r_2(\mathbf{u}_2)$ and $\mathbf{v}_2 = \mathbf{u}_2 \otimes \exp(s_1(\mathbf{v}_1)) + r_1(\mathbf{v}_1)$, and its inverse in a similar manner. In the context of normalizing flow, we consider $\mathbf{u}$ and $\mathbf{v}$ to be $\mathbf{x}$ and $\mathbf{z}$.



(a) Forward map (b) Inverse map

Figure 2: Coupling layer in normalizing flow. Note the change of operation orders: $\mathbf{u} \rightarrow \mathbf{v}$ in forward and $\mathbf{v} \rightarrow \mathbf{u}$ in inverse.

3. Model Setup: Conditional Recurrent Flow

Estimating the *conditional probability* $p(\mathbf{x}|\mathbf{y})$ is desirable since it represents an underlying phenomenon of the input $\mathbf{x} \in \mathbb{R}^d$ and covariate $\mathbf{y} \in \mathbb{R}^k$ (e.g., the probability of a specific brain imaging measure $\mathbf{x}$ of interest given a diagnosis $\mathbf{y}$). In terms of normalizing flow, the goal is to construct an invertible network $f(\cdot)$ mapping a given input $\mathbf{x} \in \mathbb{R}^d$ to its corresponding covariate/label $\mathbf{y} \in \mathbb{R}^k$ and its latent variable $\mathbf{z} \in \mathbb{R}^m$ such that $[\mathbf{y}, \mathbf{z}] = f(\mathbf{x})$. The mapping has an inverse for $\mathbf{x} = f^{-1}([\mathbf{y}, \mathbf{z}])$ to be recovered.

Specifically, when a flow-based model jointly encodes label and latent information (i.e., $[\mathbf{y}, \mathbf{z}] = \mathbf{v} = f(\mathbf{x})$) while ensuring that $p(\mathbf{y})$ and $p(\mathbf{z})$ are independent, then the network becomes *conditionally* invertible (i.e., $\mathbf{x} = f^{-1}([\mathbf{y}, \mathbf{z}])$ conditioned on given $\mathbf{y}$). Such a network can be theoretically constructed through a bidirectional-type training [1], and this allows a conditional sampling $\mathbf{x} = f^{-1}([\mathbf{y}, \mathbf{z}])$ and the posterior estimation $p(\mathbf{x}|\mathbf{y})$.

3.1. Conditional Recurrent Flow (CRow)

We introduce Conditional Recurrent Flow (CRow) model for conditional sequence generation. Given a sequence of input/output pairs $\{\mathbf{u}^t, \mathbf{v}^t\}$ for $t = 1, \ldots, T$ time points, modeling the relationship between the variables across time needs to also account for the temporal characteristic of the sequence. We are specifically interested in an *invertible network which is also recurrent* such that given a *sequence* of inputs $\{\mathbf{u}^t\}$ (i.e., features $\{\mathbf{x}^t\}$) and their *sequence* of outputs $\{\mathbf{v}^t\}$ (i.e., covariates/labels and latent information $\{\mathbf{y}^t, \mathbf{z}^t\}$), we can model the invertible relationship between those sequences for posterior estimation and conditional sequence generation as illustrated in Fig. 1.

We use the coupling block to setup a normalizing flow type invertible model. Then, we construct a *recurrent* subnetwork q which also contains a recurrent network (e.g., GRU) internally. This allows q to take the previous hidden state $\mathbf{h}^{t-1}$ and output the next hidden state $\mathbf{h}^t$ as $[\mathbf{q}, \mathbf{h}^t] = q(\mathbf{u}, \mathbf{h}^{t-1})$ where $\mathbf{q}$ is an element-wise transformation vector derived from $\mathbf{u}$ analogous to the output of a subnetwork $s(\mathbf{u})$. In previous coupling layers, two transfor-

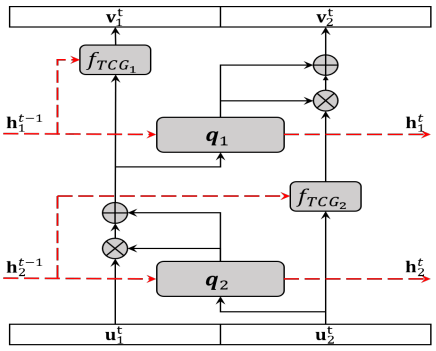


Figure 3: The CRow model. Only the forward map of a single block (two coupling layers) is shown for brevity. The inverse map involves a similar order of operations (analogous to Fig. 2a and Fig. 2b)

mation vectors $\mathbf{s} = s(\cdot)$ and $\mathbf{r} = r(\cdot)$ were explicitly computed from two subnetworks for each layer. For CRow, we follow the structure of Glow [14] which computes a single vector $\mathbf{q} = q(\cdot)$ and split it as $[\mathbf{s}, \mathbf{r}] = \mathbf{q}$. This allows us to use a single hidden state while concurrently learning $[\mathbf{s}, \mathbf{r}]$ which we denote as $\mathbf{s} = q_s(\cdot)$ and $\mathbf{r} = q_r(\cdot)$ to indicate the individual vectors. Thus, at each t with given $[\mathbf{u}_1^t, \mathbf{u}_2^t] = \mathbf{u}^t$ and $[\mathbf{v}_1^t, \mathbf{v}_2^t] = \mathbf{v}^t$,

$$\begin{aligned} \mathbf{v}_1^t &= \mathbf{u}_1^t \otimes \exp(q_{s_2}(\mathbf{u}_2^t, \mathbf{h}_2^{t-1})) + q_{r_2}(\mathbf{u}_2^t, \mathbf{h}_2^{t-1}) \\ \mathbf{v}_2^t &= \mathbf{u}_2^t \otimes \exp(q_{s_1}(\mathbf{v}_1^t, \mathbf{h}_1^{t-1})) + q_{r_1}(\mathbf{v}_1^t, \mathbf{h}_1^{t-1}) \end{aligned} \tag{1}$$

and the inverse is computed similarly. Note that the hidden states $\mathbf{h}_1^t$ and $\mathbf{h}_2^t$ generated from the recurrent network of the subnetworks are implicitly used within the subnetwork architecture and also passed to their corresponding recurrent network in the next time point as in Fig. 3.

3.2. Temporal Context Gating (TCG)

A standard (single) coupling layer transforms only a part of the input by design which results in the determinant of a triangular Jacobian matrix $J_{\mathbf{v}}$. CRow incorporates a *temporal context gating* $f_{\text{TCG}}(\alpha^t, \mathbf{h}^{t-1})$ using the temporal information $\mathbf{h}^{t-1}$ on a given input α^t at t as $f_{\text{TCG}}(\alpha^t, \mathbf{h}^{t-1}) = \alpha^t \otimes cgate(\mathbf{h}^{t-1})$ where $cgate(\mathbf{h}^{t-1})$ can be any learnable function/network with a sigmoid function at the end. This is analogous to the context gating [17] in video analysis. Note that this still preserves the Jacobian structure:

$$|J_{\mathbf{v}}| = \begin{vmatrix} \frac{\partial \mathbf{v}_1}{\partial \mathbf{u}_1} & \frac{\partial \mathbf{v}_1}{\partial \mathbf{u}_2} \\ \frac{\partial \mathbf{v}_2}{\partial \mathbf{u}_1} & \frac{\partial \mathbf{v}_2}{\partial \mathbf{u}_2} \end{vmatrix} = \begin{vmatrix} diag(cgate(\mathbf{h}^{t-1})) & 0 \\ \frac{\partial \mathbf{v}_2}{\partial \mathbf{u}_1} & diag(\exp s(\mathbf{u}_1)) \end{vmatrix}$$

where $|J_{\mathbf{v}}| = [\prod_j cgate(\mathbf{h}^{t-1})_j] * [\exp(\sum_i (s(\mathbf{u}_1))_i)]$.

4. Experiments

In this section, we show various experimental results on a video data as well as real longitudinal brain imaging measures to demonstrate and validate our model.

4.1. Conditional Moving MNIST Generation

Moving Digit MNIST: On a controlled Moving Digit MNIST dataset [23] of image sequences showing a handwritten digit from 0 to 9 moving in a path and bouncing off the boundary, we specifically chose two digits (e.g., 0 and 1) to construct ∼13K controlled sequences of frame length $T = 6$ where each frame of a sequence is an image of size 20 by 20 and has a one-hot vector $\mathbf{y}^t \in \mathbb{R}^2$ of digit label at t indicating one of the two possible digits.

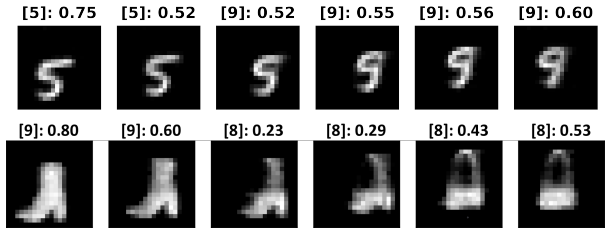


Figure 4: Examples of generated sequences using CRow given the changing conditions. Top: 5→9, Bottom: boot (9) → bag (8). (top of each frame: **density for labels**).

Training. Our model consists of three coupling blocks (q has one GRU cell, three residual fully connected (FC) networks, and ReLU activation). For each TCG, $cgate(\cdot)$ is a single FC with sigmoid activation. Each input frame $\mathbf{u}^t = \mathbf{x}^t$ is split into two halves $\mathbf{u}_1$ and $\mathbf{u}_2$. Each training sequence has a digit label sequence $\{\mathbf{y}^t\}$ for $t = 1, \ldots, 6$ where all $\mathbf{y}^t$ are "identical" in each sequence since the the same digit is shown throughout the sequence.

Generation. We first specified sequential conditions (i.e., digit label) that change midway through the sequence (e.g., $\{\mathbf{y}^t\}$ sequence indicating digit labels 5→5→9→9→9→9). Then, we generated the corresponding sequences $\{\mathbf{x}^t\}$ and visually check if the changes across the frames look natural. Note that we trained *only* the image sequences with consistent digit labels. One demonstrative result is shown in Fig. 4 where we compare the generated image sequences with condition (i.e., digit label) changing from 5 to 9. Our model quantifies its output confidence in the form of density (i.e., likelihood) shown at the top of each generated images with lower density at the frame showing the most drastic transformation which were not observed in the training.

Moving Fashion MNIST: We also tested our model on a more challenging Moving Fashion MNIST [25] of moving apparel image sequences in a very similar manner (Fig. 4).

4.2. Longitudinal Neuroimaging Analysis

In this experiment, we evaluate if our conditionally generated samples actually exhibit statistically robust and clinically sound characteristics when trained with a longitudinal Alzheimer's disease (AD) brain imaging dataset.

Dataset. We use a longitudinal neuroimaging dataset called The Alzheimer's Disease Prediction of Longitudinal Evolution (TADPOLE) [16] (N=276 and $T = 3$).

Input. For the longitudinal sequence $\{\mathbf{x}^t\}$, we chose Florbetapir (AV45) Positron Emission Tomography (PET) scan measuring the level of *amyloid-beta* in brain which has been a known pathology of AD [24, 12]. The AV45 images were registered to a common brain template (MNI152) to derive the gray matter regions of interests (82 Desikan ROIs [3]). Thus, each of the 82 ROIs ($\mathbf{x}^t \in \mathbb{R}^{82}$) holds an average

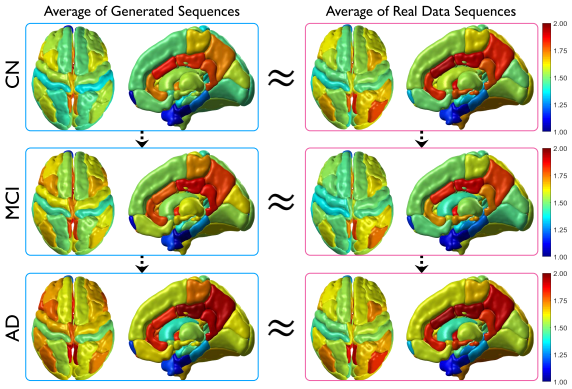


Figure 5: Generated sequences vs. real data sequences comparison for CN (top)→MCI (middle)→AD (bottom). Each blue/pink frame has top, side (interior of right hemisphere), and front views. **Left:** The average of the 100 generated sequences conditioned on CN→MCI→AD. **Right:** The average of the real samples with CN→MCI→AD in the dataset. Red/blue indicate high/low AV45. ROIs are expected to turn more red as CN→MCI→AD. The generated samples show magnitudes and sequential patterns similar (≈) to those of the real samples from the training data.

Standard Uptake Value Ratio (SUVR) measure of AV45.

Condition. For the longitudinal condition $\{\mathbf{y}^t\}$, we chose three covariates known to be tied to AD progression ([normal → impaired]): **(1)** *Diagnosis*: Control (CN), Mild Cognitive Impairment (MCI), and Alzheimer's Disease (AD) [CN→MCI→AD]. **(2)** *ADAS13*: AD Assessment Scale [0→85]. **(3)** *MMSE:* Mini Mental State Exam [0→30]. These assessments impose disease *progression*.

Analysis. We performed a statistical group analysis on each condition $\{\mathbf{y}^t\}$ independently with the following pipeline: *(1) Training:* We trained our model (the same subnetwork as Sec. 4.1) using the sequences of SUVR in 82 ROIs for $\{\mathbf{x}^t\}$ and the covariate ('label') sequences for $\{\mathbf{y}^t\}$. *(2) Conditional longitudinal sample generation:* We generated longitudinal samples $\{\hat{\mathbf{x}}^t\}$ conditioned on two distinct longitudinal conditions: *Control* (healthy covariate sequence) versus *Progression* (worsening covariate sequence). Specifically, for each condition (e.g., Diagnosis), we generate N_1 samples of Control (e.g., $\{\hat{\mathbf{x}}_1^t\}_{i=1}^{N_1}$ conditioned on $\{\mathbf{y}^t\}_=$ CN→CN→CN) and N_2 samples of Progression ($\{\hat{\mathbf{x}}_2^t\}_{i=1}^{N_2}$ conditioned on $\{\mathbf{y}^t\}$ = CN→MCI→AD). Then, we perform a two sample t-test at $t = 3$ for each of 82 ROIs between $\{\hat{\mathbf{x}}_1^3\}_{i=1}^{N_1}$ and $\{\hat{\mathbf{x}}_2^3\}_{i=1}^{N_2}$ groups, and derive p-values to tell whether the pathology levels between the groups significantly differ in those ROIs.

	# of Statistically Significant ROIs		
Covariates	Diagnosis	ADAS13	MMSE
Control	CN→CN→CN	10→10→10	30→30→30
Progression	CN→MCI→AD	10→20→30	30→26→22
cINN	11 (4)	5 (2)	5 (0)
Ours	25 (11)	24 (12)	19 (**2**)
Ours + TCG	**28** (**12**)	**32** (**14**)	**31** (**2**)
Control	CN→CN→CN	10→10→10	30→30→30
Early-progress.	CN→MCI→MCI	10→13→16	30→28→26
cINN	2 (0)	2 (2)	2 (0)
Ours	**6** (2)	6 (4)	11 (**4**)
Ours + TCG	**6** (**4**)	**8** (**5**)	**12** (**4**)

Table 1: Number of ROIs identified by group analysis using the generated measures with respect to various covariates associated with AD with the significance level of $\alpha = 0.01$ (type-I error controlled result shown in parenthesis). Each column represents sequences of disease progression represented by diagnosis or test scores. *CRow* considers the progression sequences. *CRow* + TCG yields the most number of significant ROIs.

	Gen. vs. Real of Progressions			Gen. vs. Real of Early-progress.		
Covariates	Diagnosis	ADAS13	MMSE	Diagnosis	ADAS13	MMSE
cINN	1.2551	1.5968	1.1498	1.0656	1.4985	0.9482
Ours	0.4193	0.5562	0.3485	0.3591	0.5612	0.2953
Ours + TCG	**0.2828**	**0.3915**	**0.1679**	**0.2341**	**0.5248**	**0.0902**

Table 2: Difference between the generated sequences and the real sequences at $t = 3$. Lower the effect size (Cohen's d), smaller the difference between the comparing distributions.

Result 1: Control vs. Progression (Table 1, Top row block). We set the longitudinal conditions for each covariate based on its associated to healthy progression and disease progression. We generated $N_1 = 100$ and $N_2 = 100$ samples for each group respectively, then performed the above statistical group difference analysis under 4 setups: **(1)** Raw training data, **(2)** cINN [1], **(3)** Our model, and **(4)** Our model + TCG. With the raw data, the sample sizes of the desirable longitudinal conditions were extremely small for all setups, so no statistical significance was found after type-I error control. With cINN, we occasionally found few significant ROIs, but the non-sequential samples with only $t = 3$ could not generate realistic samples. CRow + TCG detected the most number of ROIs which include many AD-specific regions reported in the aging literature [10, 12].

Result 2: Control vs. Early-progression (Table 1, Bottom row block). We setup a more challenging task where we generate samples which resemble the subjects that show slower progression of the disease. Such case is especially important in AD when early detection leads to effective prevention. With $N_1 = 100$ and $N_2 = 100$ samples, no significant ROIs were found in all models. To improve the sensitivity, we generated $N_1 = 150$ and $N_2 = 150$ samples in all models and found several significant ROIs *only* with CRow related to an *early* AD progression [7, 11, 13, 9].

Generation assessments. In Fig. 5, we see the generated samples (Left) through CN→MCI→AD in two views of the ROIs and compare them to the real training samples (Right). We see that the generated samples have similar AV45 loads through the ROIs, and the progression pattern across the ROIs follows that of the real sequence. We also quantified the similarities between the generated and real data sequences by computing effect size (Cohen's d [2]) measuring the difference between the distributions (Table 2).

5. Conclusion

We developed a novel invertible neural network model that integrates recurrent subnetworks and temporal context gating to pass information within a sequence generation, enable the model to "learn" the conditional distribution of training data in a latent space, and generate sequential samples with progressive behavior according to the given conditions. We show experimental results using three datasets.

References

[1] L. Ardizzone, J. Kruse, S. Wirkert, et al. Analyzing Inverse Problems with Invertible Neural Networks. In *ICLR*, 2019. 2, 4

[2] J. Cohen. *Statistical power analysis for the behavioral sciences*. Routledge, 2013. 4

[3] R. S. Desikan, F. Ségonne, B. Fischl, et al. An automated labeling system for subdividing the human cerebral cortex on mri scans into gyral based regions of interest. *NeuroImage*, 31(3):968–980, 2006. 3

[4] L. Dinh, D. Krueger, and Y. Bengio. NICE: Nonlinear independent components estimation. *arXiv preprint arXiv:1410.8516*, 2014. 2

[5] L. Dinh, J. Sohl-Dickstein, and S. Bengio. Density estimation using Real NVP. *arXiv preprint arXiv:1605.08803*, 2016. 2

[6] M. Fortunato, C. Blundell, and O. Vinyals. Bayesian recurrent neural networks. *arXiv preprint arXiv:1704.02798*, 2017. 1

[7] N. Fox, E. Warrington, P. Freeborough, P. Hartikainen, A. Kennedy, J. Stevens, and M. N. Rossor. Presymptomatic hippocampal atrophy in Alzheimer's disease: A longitudinal MRI study. *Brain*, 119(6):2001–2007, 1996. 4

[8] Y. Gal and Z. Ghahramani. Bayesian convolutional neural networks with Bernoulli approximate variational inference. *arXiv preprint arXiv:1506.02158*, 2015. 1

[9] S. J. Hwang, N. Adluru, W. H. Kim, S. C. Johnson, B. B. Bendlin, and V. Singh. Associations Between Positron Emission Tomography Amyloid Pathology and Diffusion Tensor Imaging Brain Connectivity in Pre-Clinical Alzheimer's Disease. *Brain connectivity*, 9. 4

[10] K. Jin, A. L. Peel, X. O. Mao, L. Xie, B. A. Cottrell, D. C. Henshall, and D. A. Greenberg. Increased hippocampal neurogenesis in Alzheimer's disease. *Proceedings of the National Academy of Sciences*, 101(1):343–347, 2004. 4

[11] S. C. Johnson, B. T. Christian, O. C. Okonkwo, J. M. Oh, S. Harding, G. Xu, A. T. Hillmer, D. W. Wooten, D. Murali, T. E. Barnhart, L. T. Hall, A. M. Racine, W. E. Klunk, C. A. Mathis, H. A. Rowley, B. P. Hermann, N. M. Dowling, S. Asthana, and M. A. Sager. Amyloid burden and neural function in people at risk for Alzheimer's disease. *Neurobiology of aging*, 35(3):576–584, 2014. 4

[12] A. D. Joshi, M. J. Pontecorvo, C. M. Clark, et al. Performance characteristics of amyloid PET with florbetapir F 18 in patients with Alzheimer's disease and cognitively normal subjects. *Journal of Nuclear Medicine*, 53(3):378–384, 2012. 3, 4

[13] W. H. Kim, A. M. Racine, N. Adluru, S. J. Hwang, K. Blennow, H. Zetterberg, C. M. Carlsson, S. Asthana, R. L. Koscik, S. C. Johnson, et al. Cerebrospinal fluid biomarkers of neurofibrillary tangles and synaptic dysfunction are associated with longitudinal decline in white matter connectivity: A multi-resolution graph analysis. *NeuroImage: Clinical*, 21, 2019. 4

[14] D. P. Kingma and P. Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. *arXiv preprint arXiv:1807.03039*, 2018. 3

[15] D. P. Kingma, T. Salimans, and M. Welling. Variational dropout and the local reparameterization trick. In *NIPS*, 2015. 1

[16] R. V. Marinescu, N. P. Oxtoby, A. L. Young, et al. TADPOLE Challenge: Prediction of Longitudinal Evolution in Alzheimer's Disease. *arXiv preprint arXiv:1805.03909*, 2018. 3

[17] A. Miech, I. Laptev, and J. Sivic. Learnable pooling with Context Gating for video classification. *arXiv preprint arXiv:1706.06905*, 2017. 3

[18] S. Narayanaswamy, T. B. Paige, J.-W. Van de Meent, A. Desmaison, N. Goodman, P. Kohli, F. Wood, and P. Torr. Learning disentangled representations with semi-supervised deep generative models. In *NIPS*, pages 5925–5935, 2017. 1

[19] G. Papamakarios and I. Murray. Fast ε-free inference of simulation models with Bayesian conditional density estimation. In *NIPS*, 2016. 1

[20] R. Ranganath, L. Tang, L. Charlin, and D. Blei. Deep exponential families. In *AISTATS*, 2015. 1

[21] D. J. Rezende and S. Mohamed. Variational inference with normalizing flows. *arXiv preprint arXiv:1505.05770*, 2015. 2

[22] O. Rippel and R. P. Adams. High-dimensional probability estimation with deep density models. *arXiv preprint arXiv:1302.5125*, 2013. 2

[23] N. Srivastava, E. Mansimov, and R. Salakhudinov. Unsupervised learning of video representations using LSTMs. In *ICML*, 2015. 3

[24] D. F. Wong, P. B. Rosenberg, Y. Zhou, et al. In vivo imaging of Amyloid deposition in Alzheimer's disease using the novel radioligand [18F] AV-45 (Florbetapir F 18). *Journal of nuclear medicine*, 51(6):913, 2010. 3

[25] H. Xiao, K. Rasul, and R. Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017. 3