Robust DNN-based Decoder Model with an Embedded State-Space Model Layer

Pedram Rajaei
Department of Biomedical Engineering, University of Houston, TX, USA
prajaei@CougarNet.uh.edu

Pavan Kallam
Department of Biomedical Engineering, University of Houston, TX, USA
pkallam@.uh.edu

Benito Garcia
Department of Psychiatry & Behavioral Sciences, University of Minnesota, MN, USA
garc1054@umn.edu

Shreya Yadav
Department of Psychiatry & Behavioral Sciences, University of Minnesota, MN, USA
yadav094@umn.edu

Aaron McInnes
Department of Psychiatry & Behavioral Sciences, University of Minnesota, MN, USA
mcinn125@umn.edu

Blair Brown
Department of Psychiatry & Behavioral Sciences, University of Minnesota, MN, USA
brow6522@umn.edu

Miriam Freedman
Department of Psychiatry & Behavioral Sciences, University of Minnesota, MN, USA
mfreedma@umn.edu

Michael Bronstein
Department of Psychiatry & Behavioral Sciences, , Institute for Health Informatics, University of Minnesota, MN, USA
brons139@umn.edu

Melanie D Goodman Keiser
Department of Psychiatry & Behavioral Sciences, University of Minnesota, MN, USA
mgoodman@umn.edu

Alik S. Widge
Department of Psychiatry & Behavioral Sciences, University of Minnesota, MN, USA
awidge@umn.edu

Ali Yousefi
Department of Biomedical Engineering, University of Houston, TX, USA
aliyousefi@uh.edu

***Abstract*— Characterizing biobehavioral time series data recorded under different task conditions is a critical step in neuroscience data analysis. The inherent complexity and stochasticity present in these data pose significant modeling challenges. Deep Neural Network (DNN) models have been widely adopted for analyzing such data. Despite their success, DNNs have significant limitations: they are sensitive to noise in training data and require large training datasets. These issues limit the applicability of DNNs to neuroscience data – where sample size is often smaller and data are volatile and feature noise artifacts. The present manuscript introduces a novel framework that embeds a state-space model (SSM) in the classic DNN structure. We demonstrate that this framework, which we call SSM-DNN, can overcome the sample size and noise sensitivity issues that plague classic DNNs. For this purpose, we illustrate training and inference steps when SSM-DNN is applied to a dataset recorded during a Death Implicit Association Test (D-IAT) task. This task was designed to produce biobehavioral data that facilitate decoding of participant phenotypes (e.g., person with depression vs. psychologically healthy person). We show SSM-DNN performance reaches an accuracy of 78%, which is 20% higher than state-of-art DNN decoder models. Its area under curve (AUC) is 0.8, which reflects its high specificity and sensitivity. The SSM-DNN modeling framework is scalable to high-dimensional time-series data and it can be applied broadly to neuroscience data, providing a robust and accurate decoding performance.**

Keywords — Neural Decoder, Bayesian Inference, Adaptive Dimensionality Reduction, Supervised State Space Model, DNN

I. INTRODUCTION

Deep neural network models (DNN) and its variants such as CNN and RNN have revolutionized many domains of science, achieving state-of-the-art performance in many prediction and classification tasks [1]. In neuroscience, we can point to many applications of DNN models such as decoding EEG and fMRI data to infer underlying cognitive states or predicting categories present in the data [2, 3]. However, DNN performance is highly dependent on availability of large and high-quality labeled datasets, requiring significant resources and time. Neuroscience data are generally limited in size, and contaminated with different sources of noise and volatility. With these issues, DNN models often show limited and inconsistent performance accuracy [4]. Potential solutions to these challenges include data augmentation [5], advanced pre-processing techniques [6], and the use of generative models [7]. However, these remedies are not effective in all circumstances.

To address the aforementioned challenges, we added SSM to the classic DNN structure. SSMs are a powerful tool for adaptive noise reduction. They are also used to extract dynamical, low-dimensional representations of the data and facilitate manifold discovery, capturing the essential dynamics present in the data. SSMs are generative models capable of producing augmentation data that preserve the temporal and statistical properties of the original dataset, making them particularly useful for generating realistic synthetic samples. Examples of the use of SSMs for this

purpose in neuroscience data analysis can be found in previous studies, such as [8] and [9].

One challenge in combining the SSM and DNN lies in their joint training, which will be different from training each separately. The SSM and DNN have significantly different numbers of free parameters and process input data at different temporal scales. As a result, developing a training solution that jointly and optimally adjusts the free parameters of both models is a challenging modeling step [10]. To overcome this challenge, we propose a novel training algorithm for the SSM-DNN that optimizes the free parameters of both the SSM and the DNN simultaneously. With this training approach, we demonstrate that the SSM-DNN inherits the aforementioned capabilities of SSMs while integrating the expressiveness and discriminative attributes of DNNs. We show its application using an example dataset (the D-IAT dataset) that showcases the model's ability to address DNN challenges related to data size and noise. Additionally, we show that its decoding performance surpasses state-of-the-art decoding solutions.

In the following sections, we first introduce the SSM-DNN model followed by its training and inference solutions. We then present our decoding result in the D-IAT task and compare it with other decoder models performance. Lastly, we discuss potential directions for future research on SSM-DNN combinations.

II. METHODS

A. Model Definition

Let us consider an experiment conducted over T trials ($t = 1, \dots, T$). During each trial, the observed data consist of biobehavioral signals, $Y^t = \{y_{1:K}^t\}$ which are recorded at K time points. The observed data has dimension N, $y_k^t \in R^N$, and label l^t is a categorical variable representing C categories, defined by $l^t \in \{1, \dots, C\}$. The label represents the experiment condition or stimulus. We assume the distribution of observed data and its label are defined by:

$$y_k^t | x_k^t \sim g_\phi(x_k^t) \quad (1.a)$$

$$l^t | x_{0:K}^t \sim h_\Lambda(x_{0:K}^t) \quad (1.b)$$

where, x_k^t is the latent process at time index k, and $g_\phi(x_k^t)$ and $h_\Lambda(x_{0:K}^t)$ are respectively the observation and discriminative (or classifier model) with ϕ and Λ free parameters. The specific case of the conditional distribution defined in **Eq. (1.a)** is $y_k^t | x_k^t \sim \mathcal{N}(g_\phi(x_k^t), Q)$, where the distribution follows a multivariate normal with additive Gaussian noise. An example of a discriminative model is a two-layer perceptron with an output layer containing a neuron that uses a sigmoid function, designed for a dataset with two labels.

In **equation set (1)**, x_k^t is a latent dynamical variable. It represents an underlying manifold that shapes the observed data and is the key feature used by the classifier. The evolution of the latent variable in time is defined by a state equation:

$$x_{k+1}^t | x_k^t \sim f_\psi(x_k^t) \quad (2)$$

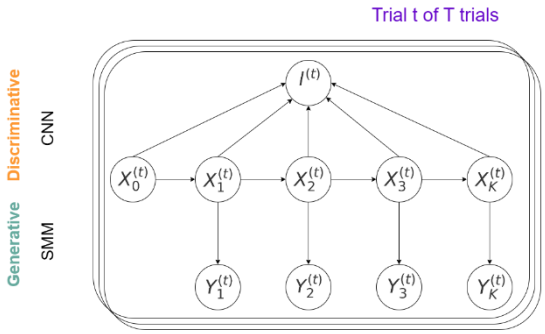

Fig. 1. SMM-DNN Model Architecture The model combines SSM and DNN which jointly characterizes biobehavioral signal ($y_{1:K}^t$) and predict its associated labels (l^t). x_k^t represents the state process which is generally a low-dimensional dynamical latent variable. It is passed to a DNN to predict the label and an observation process to generate biobehavioral signals. With this model structure, DNN and SSM work in parallel to jointly predict the label and generate data through a common latent process.

where, $f_\psi(x_k^t)$ defines the conditional distribution with ψ as distribution free parameters. The specific case defined by this state process equation is $x_{k+1}^t | x_k^t \sim N(f_\psi(x_k^t), R)$, where this distribution follows a multivariate normal with additive Gaussian noise. The latent variable has dimension D, $x_k^t \in R^D$, which is generally much smaller than N.

Eqs. (1) and **(2)** define the SSM-DNN model. The model parameters are $\omega = \{\phi, \Lambda, \psi\}$, which need to be estimated given training data. We will discuss model training in the next section. **Fig. 1** depicts the model structure. The latent variable changes from one time point to the next, and the biobehavioral signals evolve on the same time scale as the latent process. The whole trajectory of the latent process enters into the DNN model, as input features, which can be used to generate the label or category. Note that when the model parameters, ω, are known, we can leverage the generative nature of the model to create novel label and biobehavioral data.

B. Model Training

The model training objective corresponds to maximizing the evidence of the labels and observed data over all trials of the experiment. We assume each trial is independent from all others; thus, the evidence is defined by:

$$P_\omega(\mathbf{Y}, L) = \int_{X^T} \dots \int_{X^0} \prod_{t=1}^T P_\omega(Y^t, l^t, X^t)\, dX^0 \dots dX^T \quad (3)$$

where, $\mathbf{Y} = \{Y^t; t = 1, \dots, T\}$, $L = \{l^t; t = 1, \dots, T\}$ and $\mathbf{X} = \{X^t; t = 1, \dots, T\}$ with $X^t = \{x_{0:K}^t\}$. $P_\omega(Y^t, l^t, X^t)$ is the full joint likelihood of the observed data, label, and latent variable in trial t, and it is defined by:

$$P_\omega(Y^t, l^t, X^t) = P(X^t) \times P(Y^t | X^t) \times P(l^t | X^t)$$

$$= P(x_0^t) \times \prod_{k=1}^K P(x_k^t | x_{k-1}^t)\, P(y_k^t | x_k^t) \times P(l^t | x_{0:K}^t) \quad (4)$$

where, the conditional independence in time reflects the model structure depicted in **Fig. 1**.

Table 1 MCMC Process for State Estimation in SSM-DNN for Trial t. The superscript t is being dropped from Y^t, l^t, X^t for the sake of clarity. The process is going to be the same for trials t from 1 to T.

- Initialization:

a. For $u = 1 \cdots U$ particles, draw samples of the initial state $\boldsymbol{x_0^{(u)}}$ from the initial density $p(\boldsymbol{x_0})$, and set $\widetilde{w}_0^{(u)} = U^{-1}$ for all u.

b. Define the proposal density, $\pi_k(x_k)^*$ from which we draw particles.

For $k = 1$ to K repeat following steps

- Importance Sampling Step: At each time step k, for $u = 1, \cdots, U$, draw sample from $\pi_k(x_k)$ and set $\hat{x}_{0:k}^{(u)} = (x_{0:k-1}^{(u)}, \hat{x}_k^{(u)})$.

Compute the importance weights

if $k < K$:

$$w_k^{(u)} = \widetilde{w}_{k-1}^{(u)} \frac{p(y_k|\hat{x}_k^{(u)})p(\hat{x}_k^{(u)}|x_{k-1}^{(u)})}{\pi_k(x_k)}$$

else:

$$w_k^{(u)} = \widetilde{w}_{k-1}^{(u)} \frac{p(y_k|\hat{\boldsymbol{x}}_{\boldsymbol{k}}^{(\boldsymbol{u})})p(\hat{\boldsymbol{x}}_{\boldsymbol{k}}^{(\boldsymbol{u})}|\boldsymbol{x}_{\boldsymbol{k-1}}^{(\boldsymbol{u})})}{\pi_k(x_k)} \times h_\Lambda(\hat{x}_{0:k}^{(u)})$$

and normalize the importance weights

$$\widehat{w}_k^{(u)} = w_k^{(u)} \left[\sum_{u=1}^{U} w_k^{(u)}\right]^{-1}$$

- Resampling step: Resample with replacement U particles $x_{0:k}^{(u)}$, $u = 1 \cdots U$ from the set $\left(\hat{x}_{0:k}^{(u)}, u = 1 \cdots U\right)$ with probabilities determined by the normalized importance weights $\widehat{w}_k^{(u)}$. Reset the weights to $\widetilde{w}_k^{(u)} = U^{-1}$.

* For the model introduced in the application section, the $\pi_k(x_k^u)$ is defined by $\pi_k(x_k) = p(x_k \mid x_{k-1})$

Given the evidence definition, the model training corresponds to finding a parameters set that maximize it, defined by:

$$\widehat{\omega} = \underset{\omega}{\operatorname{argmax}}\, P_\omega(\mathbf{Y}, L) \tag{5}$$

The evidence term, $P_\omega(\mathbf{Y}, L)$, when viewed as a function of ω is a likelihood function. As a result, in **Eq. (5)**, we find the maximum likelihood estimate of ω given the data and corresponding labels over T trials.

To find this maximum likelihood estimate, we first need to take the integral over the latent variable as described in **Eq. (3)**. Solutions to this class of problems include Expectation-Maximization (EM) and Variational Inference approaches [11, 12]. These solutions provide a lower bound for the evidence, and can be combined with sampling techniques to solve the necessary integration calculations. Our training model uses the EM approach, as we were able to derive an approximate posterior distribution of the state. EM has two steps: expectation (E-step) and maximization (M-step). For E-step, we first find the posterior distribution of the latent processes given the observed data and labels. We use an MCMC approach to approximate this posterior, and draw samples from the posterior trajectory to approximate the expectation of the logarithm of the full likelihood [13]. **Table 1** describes the MCMC solution that finds the state posterior distribution for SSM-DNN. For the M-step, we use stochastic gradient ascent to update the model parameters [14]. **Table 2** summarizes the training process. Note, E- and M-steps are iteratively called until convergence to a local maximum is achieved. Details of implementation can be found on GitHub at: https://github.com/Pedram-Rajaei/SSM-DNN

Table 2 SSM-DNN Training Process. We use stochastic gradient descent combined with E-M for the model training.

- Initialization: Select an initial estimate for the model parameters, $\omega^{(0)}$.

- E-step: Using U particles of inferred state trajectories, described in **Table 1**, the approximate conditional expectation of log-likelihood function for T trials are defined by:

$$Q(\omega|\omega^{(\mathrm{m})}) = \sum_{t=1}^{T}\sum_{u=1}^{U} logP_\omega(Y^t, l^t, X^{t(u)})$$

$$= \sum_{t=1}^{T}\sum_{u=1}^{U}\sum_{k=1}^{K} \log P_\omega\left(y_k^{t(u)}|x_k^{t(u)}\right) + \sum_{t=1}^{T}\sum_{u=1}^{U} \log P_\omega\left(l^t|X^{t(u)}\right)$$

$$+ \sum_{t=1}^{T}\sum_{u=1}^{U}\sum_{k=1}^{K} \log P_\omega\left(x_k^{t(u)}|x_{k-1}^{t(u)}\right) + \sum_{t=1}^{T}\sum_{u=1}^{U} \log P_\omega\left(x_0^{t(u)}\right)$$

The likelihood function was defined in Eq. (4).

- M-step: Find a new set of parameters that maximizes Q function:

$$\omega^{(\mathrm{m}+1)} = \arg\max_{\omega} Q(\omega|\omega^{(\mathrm{m})})$$

During the optimization process, we use:

a. Stochastic gradient descent to update the SSM model parameters ϕ,ψ^*

b. ADAM optimizer to update the DNN model parameter.

* Details of gradient descent step can be found in https://github.com/Pedram-Rajaei/SSM-DNN

C. Decoding Process

We define the decoding process as predicting probability of the label per trial given the model parameters and observed biobehavioral data. The decoding process starts by finding the posterior distribution of the state given the observed data, $P(X^z \mid Y^z)$, where z represents a sample trial index from the test trials. With the posterior distribution, we then move to predict the probability of different classes given trajectories of the inferred state, $P\left(l^z|x_{0:K}^{z(u)}\right) \sim h_\Lambda\left(x_{0:K}^{z(u)}\right)$.

By computing the distribution for U trajectories, $u = 1, \ldots, U$, we can build a sample distribution of different classes given the data. Note that in contrast to the training process, the state inference in the decoding step is defined using the biobehavioral signals without the labels. Otherwise, the state inference will be the same as explained in **Table 1**, except we set $h_\Lambda\left(x_{0:K}^{z(u)}\right) = 1$. We argue that the decoding process is effective because it relies on the learning step, where the model parameters, learned during training, direct the state and observation parameters into a domain that enables state inference to align with the corresponding category or label. Simply put, the inference acts as an adaptive feature extraction process, which is then passed to the DNN. In the next section, we present an application of the framework to the D-IAT [15] dataset; collected by our research group at the University of Minnesota.

III. BRIEF-DIAT TASK AND DATASET

The dataset used in this study features behavioral data collected during a modified version of the classic D-IAT task. This Brief DIAT (B-DIAT) [16] was used to investigate potential behavioral and neural differences between

individuals with Major Depressive Disorder (MDD) and healthy controls (CTL). The data collection protocol was IRB approved (Advarra IRB, approved on 05/31/2023). 23 participants' data were retained. The remaining 4 participants' data were excluded due to behavior that suggested inattention or failure to follow task instructions (missing responses to over 240 trials, having accuracy below chance level). The 23 participants in the final dataset included 11 participants with an MDD diagnosis and 12 CTL participants.

The experimental B-DIAT task required participants to match a stimulus word with one of two categories displayed as headers on the screen: 'life + me' or 'death + me.' Stimulus words were presented sequentially, and participants were instructed to press an arrow key depending on whether the stimulus semantically matched one of the category headers. To counterbalance response mappings, participants with odd-numbered subject IDs were instructed to press the right arrow key when a stimulus matched one of the categories, whereas those with even-numbered subject IDs pressed the left arrow key for a semantic match. In both cases, the opposite key was used when the stimulus did not match the categories. Stimulus words, such as 'I,' 'they,' 'die,' 'live,' or 'survive,' were carefully selected to align or contrast with the header categories and reflect semantic associations. For example, under the header 'life + me,' the word 'die' does not match semantically, so participants should press the 'left' arrow key. Conversely, the word 'I' matches 'me,' requiring participants to press the 'right' key for a correct response. The task is designed to elicit cognitive conflict via discrepancies between a participant's self-concept and the task directions. For example, the header 'death + me' may introduce implicit conflict for a CTL participant who does not associate themselves with death, potentially altering their behavior. Participants had 2.0 seconds to respond to each trial, with the header categories alternating every 20 trials to reduce response bias and introduce variability. During the experiment, behavioral data and neural activity were recorded. For the present demonstration, we focus on behavioral data from this task. Neural data will be explored in future research as an input to the SSM-DNN model. Each participant completed 360 trials, and the reaction time (RT) time series across these trials was used as input to the SSM-DNN, where the participant group served as the trial label, to predict whether the participant belonged to the MDD or CTL group.

IV. MDD VS. CTL DECODER MODEL

For the decoder model, we assume the latent process is a random walk model, defined by:

$$x_{k+1}^t = x_k^t + \epsilon_k \qquad \epsilon_k \sim \mathcal{N}(0, \sigma_\epsilon^2) \qquad (6.a)$$

$$x_0^t \sim \mathcal{N}(0, \sigma_0^2) \qquad (6.b)$$

The observation and label model are defined by:

$$\ln(y_k^t) = x_k^t + v_k^t \qquad v_k^t \sim \mathcal{N}(0, \sigma_v^2) \qquad (7.a)$$

$$l^t | x_{0:K}^t \sim h_\Lambda(x_{0:K}^t) \qquad (7.b)$$

For the model, we use a linear SSM with the log of RT as input [17]. We expect the SSM to regress noise from the RT data and create a smoothed RT to be used as input into the DNN. Note that this assumption does not limit the generality of our framework, as we can build a non-linear SSM to capture more complex patterns in the observed data, which will be the focus of future research. The neural network used here is a one-dimensional spectral CNN with two convolutional layers and an output neuron with a sigmoidal function. We use the spectral CNN because we are interested in longer-term patterns in the data. With this model setting, we applied our training process, explained in **Table 2**, to adjust parameters for the SSM (e.g., observation and state noise variance) as well as the CNN.

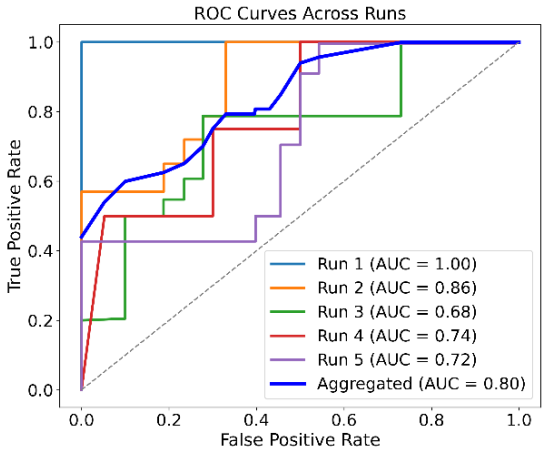

Fig. 2. Performance of the SSM-DNN Decoder on the B-DIAT Task Across 5-Fold Cross-Validation. The plot demonstrates the consistency and generalization capability of the model, highlighting high AUCs across all runs.

V. DECODING RESULTS

To assess the model decoding accuracy, we partitioned the dataset into training and testing sets. Specifically, we allocated 6 participants (3 MDD and 3 CTL) to the test set, while the remaining 16 participants (8 MDD and 8 CTL) were used for training. We dropped one participant's data to have a balanced number of samples in each group, better ensuring that model training is unbiased. We created these partitions randomly, and repeated the training and test steps 5 times. We measure performance by averaging decoding accuracy over the test sets.

We examined both accuracy and AUC as performance metrics. **Fig 2.** shows all 5 ROC curves and the aggregated one. The average AUC for the SSM-DNN is 0.8. Average model prediction accuracy is 78%. In parallel, we trained a ResNet [18] classifier with log of RT as its input, which we trained using a transfer learning approach. The ResNet prediction accuracy was only 58%, and its AUC was significantly lower than the SSM-DNN's. The higher AUC in SSM-DNN suggests that we built a more robust classifier, and thus, we can get a higher sensitivity and specificity in distinguishing between MDD and CTL participants.

VI. DISCUSSION

This manuscript introduces a novel decoder framework, the SSM-DNN. As we demonstrated through our modeling results, the SSM-DNN framework effectively addresses sensitivity to noise in DNNs, as its SSM module efficiently de-noises the data. We also demonstrated the generative nature of our pipeline in creating sufficient data for training the DNN. However, this aspect of SSM-DNN requires further exploration with higher-dimensional input. Our proposed solution reaches an AUC of 0.8, which is significantly higher than the ResNet model. We believe our novel framework can be expanded to multivariate time-series data, such as EEG with multiple channels of recording. In this case, the latent process would infer a low dimensional manifold present in data, which can also encode the condition or label of the data – e.g., neural features [19, 20]. An advantage of this approach would be that the dimension reduction, feature extraction, and DNN training can be done in tandem, rather than sequentially [21]. A second advantage of applying the SSM-DNN approach to neural data is that we explicitly learn the latent process shaping the observed data. Newfound knowledge of this process may provide insight into underlying neural mechanisms driving the observed behaviors.

VII. FUTURE WORK

Future work will expand the latent process to combine behavioral and EEG data for predicting participant labels. While a simple linear state equation was used, the framework is scalable to higher-dimensional and more complex state equations, enabling analysis of multivariate time-series data and capturing additional temporal dependencies. These extensions will enhance the framework's utility in advanced data preprocessing and modeling.

ACKNOWLEDGMENT

This project is sponsored by the Defense Advanced Research Projects Agency (DARPA) under cooperative agreement No. N660012324016. The content of the information does not necessarily reflect the position or the policy of the Government, and no official endorsement should be inferred.

REFERENCES

[1] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep Learning. Nature, 521(7553), 436–444.

[2] Lawhern, V. J., Solon, A. J., Waytowich, N. R., Gordon, S. M., Hung, C. P., & Lance, B. J. (2018). EEGNet: A compact convolutional neural network for EEG-based brain–computer interfaces. Journal of Neural Engineering, 15(5), 056013.

[3] Y. Gao, Y. Zhang, Z. Cao, X. Guo and J. Zhang, "Decoding Brain States From fMRI Signals by Using Unsupervised Domain Adaptation," in IEEE Journal of Biomedical and Health Informatics, vol. 24, no. 6, pp.

[4] Vieira, S., Pinaya, W. H. L., & Mechelli, A. (2017). Using deep learning to investigate the neuroimaging correlates of psychiatric and neurological disorders: Methods and applications. Neuroscience & Biobehavioral Reviews, 74, 58–75

[5] Ullah, H., Anwar, S. M., & Bilal, M. (2021). Data Augmentation for Improving Performance of Deep Neural Networks in EEG-Based Emotion Recognition. Frontiers in Computational Neuroscience, 15, 723843.

[6] Rajaei, P., Jahanian, K. H., Beheshti, A., Band, S. S., Dehzangi, A., & Alinejad-Rokny, H. (2021). VIRMOTIF: A user-friendly tool for viral sequence analysis. Genes, 12(2), 186.

[7] Ruthotto, L., & Haber, E. (2021). An introduction to deep generative modeling. arXiv preprint arXiv:2103.05180.

[8] Yousefi, A., Basu, I., Paulk, A.C., Peled, N., Eskandar, E.N., Dougherty, D.D., Cash, S.S., Widge, A.S. and Eden, U.T., 2019. Decoding hidden cognitive states from behavior and physiology using a Bayesian approach. Neural computation, 31(9), pp.1751-1788.

[9] Krishnan, R. G., Shalit, U., & Sontag, D. (2017). Deep Kalman Filters. arXiv preprint arXiv:1511.05121.

[10] Rezaei, M. R., Hadjinicolaou, A. E., Cash, S. S., Eden, U. T., & Yousefi, A. (2022). Direct discriminative decoder models for analysis of high-dimensional dynamical neural data. Neural Computation, 34(5), 1100-1135.

[11] Smith, A. C., & Brown, E. N. (2003). Estimating a state-space model from point process observations. Neural Computation, 15(5), 965-991.

[12] Blei, D. M., Kucukelbir, A., & McAuliffe, J. D. (2017). Variational Inference: A Review for Statisticians. Journal of the American Statistical Association, 112(518), 859-877.

[13] Gilks, W. R., Richardson, S., & Spiegelhalter, D. J. (1995). Markov Chain Monte Carlo in Practice. Chapman and Hall/CRC.

[14] Robbins, H., & Monro, S. (1951). A Stochastic Approximation Method. The Annals of Mathematical Statistics, 22(3), 400–407.

[15] Nock, M. K., & Banaji, M. R. (2007). Assessment of self-injurious thoughts using a behavioral test. American Journal of Psychiatry, 164(5), 820–823.

[16] Millner, A. J., Coppersmith, D. D., Teachman, B. A., & Nock, M. K. (2018). The Brief Death Implicit Association Test: Scoring recommendations, reliability, validity, and comparisons with the Death Implicit Association Test. Psychological Assessment, 30(10), 1356.

[17] Yousefi, A., Paulk, A.C., Basu, I., Mirsky, J.L., Dougherty, D.D., Eskandar, E.N., Eden, U.T. and Widge, A.S., 2019. COMPASS: an open-source, general-purpose software toolkit for computational psychiatry. Frontiers in neuroscience, 12, p.957.

[18] H e, K., Zhang, X., Ren, S., & Sun, J. (2016). "Deep Residual Learning for Image Recognition." In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (pp. 770–778).

[19] Ziaei, N., Nazari, B., Eden, U. T., Widge, A., & Yousefi, A. (2024). A bayesian gaussian process-based latent discriminative generative decoder (ldgd) model for high-dimensional data. IEEE Access.

[20] Degenhart, A. D., Bishop, W. E., Oby, E. R., Tyler-Kabara, E. C., Chase, S. M., Batista, A. P., & Yu, B. M. (2020). Stabilization of a brain–computer interface via the alignment of low-dimensional spaces of neural activity. Nature biomedical engineering, 4(7), 672-685.

[21] Basu, I., Yousefi, A., Crocker, B., Zelmann, R., Paulk, A.C., Peled, N., Ellard, K.K., Weisholtz, D.S., Cosgrove, G.R., Deckersbach, T. and Eden, U.T., 2023. Closed-loop enhancement and neural decoding of cognitive control in humans. Nature biomedical engineering, 7(4), pp.576-588..