The Alpha-Alternator: Dynamic Adaptation To Varying Noise Levels In Sequences Using The Vendi Score For Improved Robustness and Performance

Anonymous authors
Paper under double-blind review

Abstract

Current state-of-the-art dynamical models, such as Mamba, assume the same level of noisiness for all elements of a given sequence, which limits their performance on noisy temporal data. In this paper, we introduce the **α-Alternator**, a novel generative model for time-dependent data that dynamically adapts to the complexity introduced by varying noise levels in sequences. The α-Alternator leverages the Vendi Score (VS), a flexible similarity-based diversity metric, to adjust, at each time step t, the influence of the sequence element at time t and the latent representation of the dynamics up to that time step on the predicted future dynamics. This influence is captured by a parameter that is learned and shared across all sequences in a given dataset. The sign of this parameter determines the direction of influence. A negative value indicates a noisy dataset, where a sequence element that increases the VS is considered noisy, and the model relies more on the latent history when processing that element. Conversely, when the parameter is positive, a sequence element that increases the VS is considered informative, and the α-Alternator relies more on this new input than on the latent history when updating its predicted latent dynamics. The α-Alternator is trained using a combination of observation masking and Alternator loss minimization. Masking simulates varying noise levels in sequences, enabling the model to be more robust to these fluctuations and improving its performance in trajectory prediction, imputation, and forecasting. Our experimental results demonstrate that the α-Alternator outperforms both Alternators and state-of-the-art state-space models across neural decoding and time-series forecasting benchmarks.

1 Introduction

Time-dependent data is central to the natural sciences and engineering disciplines. Modeling such data accurately requires methods that can capture variability both across sequences and within individual sequences. State-space models, such as Mambas, have emerged as a popular framework for sequence modeling (Wang et al., 2025; Gu & Dao, 2023). They have demonstrated strong performance in various applications, including speech recognition (Zhang et al., 2024) and protein folding (Xu et al., 2024). However, Mambas rely on fixed state-space representations that assume smooth transitions across time steps and do not dynamically adjust to noise fluctuations. This is particularly limiting in applications where stochasticity plays a significant role, e.g. neural decoding.

More recently, Alternators have been introduced as an alternative modeling framework for time-dependent data (Rezaei & Dieng, 2024). Unlike Mambas, which use a structured state-space representation, Alternators explicitly modulate the influence of past and present observations through a gating mechanism, which offers them great flexibility and the ability to capture long-range dependencies. However, despite this flexibility, Alternators still rely on a fixed weighting scheme that does not explicitly account for varying noise levels in the sequence. As a result, they can also struggle in situations where sequence noise fluctuates significantly.

In this work, we introduce the α-Alternator, a novel Alternator model that dynamically adjusts its reliance on each sequence element based on its level of noisiness. This mechanism is based on the Vendi Score, a

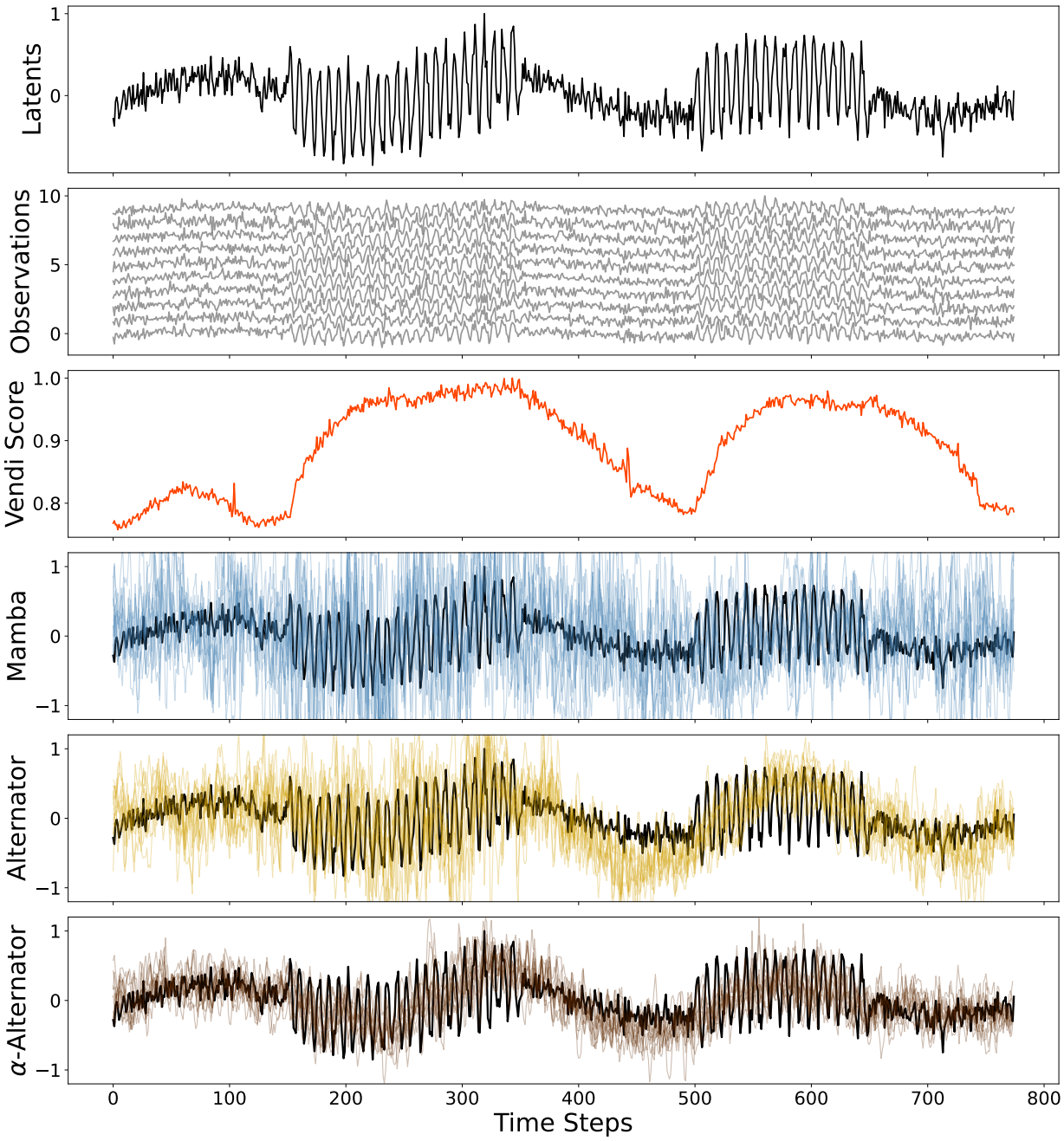

Figure 1: The α-Alternator is robust to varying noise levels compared to a Mamba and an Alternator. The Alternator is more robust to noise than the Mamba.

flexible similarity-based diversity metric (Friedman & Dieng, 2023). The α-Alternator learns the weight determining its reliance on a given sequence element by applying a sigmoid function to the output of a linear layer, which takes the Vendi Score—computed over two shifted versions of the sequence around that element—as input. The parameter of the linear layer is learned and shared across all sequences in a given dataset, with its sign indicating the direction of influence for each sequence element. When the parameter is negative, sequence elements with large VS values are treated as noisy inputs. As a result, the model places greater emphasis on the latent history when processing these elements. In contrast, when the parameter is positive, sequence elements with high VS are considered informative, and the model relies more on them to

update the predicted latent dynamics. This simple mechanism enables the α-Alternator to adapt to varying noise levels in sequences and generalize better. We illustrate this behavior in Fig. 1, where we show the stability of the α-Alternator when modeling sequences with varying noise levels across time steps.

The figure illustrates some latent state, simulated using a noisy sine function that incorporates both low-frequency and high-frequency components. The base signal, a sine wave at 2 Hz, represents the low-frequency component, while higher-frequency components at 60 Hz are added within two distinct time windows to introduce more complex dynamics. Gaussian noise is then applied to the modulated signal. Ten noisy sequences are drawn as observations by applying random scaling and adding multivariate Gaussian noise, resulting in diverse yet structurally related time series. To quantify the variability of the observations over time, we compute the Vendi Score of the noisy observations using a sliding window of 100 time steps. We then evaluate the performance of the Mamba, the Alternator, and the α-Alternator in recovering the latent signal from the ten observations. As shown in the figure, the Mamba struggles with handling highly noisy regions. The Alternator exhibited improved robustness to noise compared to the Mamba, but it still faced challenges in fully adapting to varying noise levels. In contrast, the α-Alternator maintains predictive stability even in sequence regions with large amounts of noise.

The performance of the α-Alternator was further assessed on neural decoding and time-series forecasting benchmarks. In neural decoding, the model outperformed state-of-the-art baselines in mapping cortical and hippocampal activity to behavioral states. We observed the same thing on time-series forecasting tasks where the α-Alternator surpassed Mambas and Alternators among other baselines.

To understand the contributions of the two key ingredients that make up the α-Alternator—the noise adaptation mechanism and the input masking during training—we conducted an ablation study. The findings confirmed that both ingredients are essential for achieving great performance.

2 Background

Time-dependent data often exhibits complex dynamics and varying levels of noise across time steps. To effectively model such data, frameworks are needed that can capture the underlying latent dynamics while adapting to input noise. This section outlines the foundations of the α-Alternator, described in the next section, which dynamically adjusts its dependency on the current time step or the latent history based on the temporal diversity of the sequence. We begin by describing Alternators, a probabilistic framework for sequence modeling, and then review the Vendi Score, a metric designed to flexibly and accurately quantify diversity.

2.1 Alternators

Consider a sequence $\boldsymbol{x}_{1:T}$. An Alternator models this sequence by coupling it with a sequence of latent variables, $\boldsymbol{z}_{0:T}$, within a joint probability distribution Rezaei & Dieng (2024),

$$p_{\theta,\phi}(\boldsymbol{x}_{1:T}, \boldsymbol{z}_{0:T}) = p(\boldsymbol{z}_0) \prod_{t=1}^{T} p_\theta(\boldsymbol{x}_t|\boldsymbol{z}_{t-1}) p_\phi(\boldsymbol{z}_t|\boldsymbol{z}_{t-1}, \boldsymbol{x}_t). \tag{1}$$

Here $p(\boldsymbol{z}_0) = \mathcal{N}(0, \boldsymbol{I})$ is a prior distribution over the initial latent variable $\boldsymbol{z}_0$, $p_\theta(\boldsymbol{x}_t|\boldsymbol{z}_{t-1})$ determines how to generate the sequence elements from the latent state $\boldsymbol{z}_{t-1}$ and $p_\phi(\boldsymbol{z}_t|\boldsymbol{z}_{t-1}, \boldsymbol{x}_t)$ models the evolution of that state over time. Here $\boldsymbol{z}_{t-1}$ acts as a *memory* summarizing the history of the sequence before time t. It is updated dynamically, at each time step, by accounting for both the current state of the memory and the sequence element at time t. This is achieved by defining the mean of $p_\phi(\boldsymbol{z}_t|\boldsymbol{z}_{t-1}, \boldsymbol{x}_t)$ using a gating mechanism,

$$p_\phi(\boldsymbol{z}_t|\boldsymbol{z}_{t-1}, \boldsymbol{x}_t) = \mathcal{N}\left(\boldsymbol{\mu}_{z_t}, \sigma_z^2\right), \text{ where } \boldsymbol{\mu}_{z_t} = \sqrt{\alpha_t} \cdot g_\phi(\boldsymbol{x}_t) + \sqrt{(1 - \alpha_t - \sigma_z^2)} \cdot \boldsymbol{z}_{t-1}.$$

The distribution $p_\theta(\boldsymbol{x}_t|\boldsymbol{z}_{t-1})$ is on the other hand defined as

$$p_\theta(\boldsymbol{x}_t|\boldsymbol{z}_{t-1}) = \mathcal{N}\left(\boldsymbol{\mu}_{x_t}, \sigma_x^2\right) \text{ where } \boldsymbol{\mu}_{x_t} = \sqrt{(1 - \sigma_x^2)} \cdot f_\theta(\boldsymbol{z}_{t-1}).$$

Here θ and ϕ are parameters of two neural networks and they are learned by minimizing the Alternator loss function

$$\mathcal{L}(\theta, \phi) = \mathbb{E}_{p(\boldsymbol{x}_{1:T}) p_{\theta,\phi}(\boldsymbol{z}_{0:T})} \left[\sum_{t=1}^{T} \|\boldsymbol{z}_t - \boldsymbol{\mu}_{z_t}\|_2^2 + \frac{D_z \sigma_z^2}{D_x \sigma_x^2} \|\boldsymbol{x}_t - \boldsymbol{\mu}_{x_t}\|_2^2 \right], \tag{2}$$

where $p(\boldsymbol{x}_{1:T})$ is the data distribution and $p_{\theta,\phi}(\boldsymbol{z}_{0:T})$ is the marginal distribution of the latent variables induced by the joint distribution in Eq. 1.

2.2 The Vendi Score

The Vendi Score (VS) was introduced by Friedman & Dieng (2023) and quantifies the diversity of a collection of elements. Consider a finite set of data points $\{\boldsymbol{r}_1, \ldots, \boldsymbol{r}_n\}$. Let $k(\cdot, \cdot)$ denote a positive semi-definite kernel function such that $k(\boldsymbol{r}_i, \boldsymbol{r}_i) = 1$ for all i. Let K be the corresponding similarity matrix, e.g. $K_{i,j} = k(\boldsymbol{r}_i, \boldsymbol{r}_j)$ for all i, j. The VS is defined as the exponential of the Shannon entropy of the normalized eigenvalues of K,

$$\mathrm{VS}(\{\boldsymbol{r}_1, \ldots, \boldsymbol{r}_n\}; k) = \exp\left(-\sum_{i=1}^{n} \overline{\lambda}_i \log \overline{\lambda}_i \right), \tag{3}$$

where $\overline{\lambda}_1, \ldots, \overline{\lambda}_n$ are the normalized eigenvalues of K. The VS is the *effective number* of distinct elements in the dataset and reaches its minimum value of 1 when all samples are identical (i.e. $k(\boldsymbol{r}_i, \boldsymbol{r}_j) = 1$ for all $i \neq j$), and its maximum value of n when all samples are distinct from each other (i.e. $k(\boldsymbol{r}_i, \boldsymbol{r}_j) = 0$ for all $i \neq j$).

The VS can be generalized to incorporate sensitivity to rare or common features by using the Renyi entropy of order $q \geq 0$ (Pasarkar & Dieng, 2024):

$$\mathrm{VS}_q(\{\boldsymbol{r}_1, \ldots, \boldsymbol{r}_n\}; k) = \exp\left(\frac{1}{1-q} \log\left(\sum_{i=1}^{n} (\overline{\lambda}_i)^q \right) \right), \tag{4}$$

The parameter q controls sensitivity to rarity, with $q = 1$ corresponding to the original Vendi Score. Lower values of q emphasize rare features, while higher values give higher weight to common ones.

The VS's ability to accurately quantify sample diversity without rigid distributional assumptions gives it great flexibility as evidenced by its various applications (Askari Hemmat et al., 2024; Kannen et al., 2024; Liu et al., 2024; Nguyen & Dieng, 2024; Pasarkar et al., 2023; Berns et al., 2023). In this paper, we use the VS to measure sequence diversity to define a noise adaptation mechanism for Alternators.

3 Method

The α-Alternator extends the original Alternator by introducing a mechanism for dynamically adjusting the weighting parameter α_t. Consider given n sequences $\boldsymbol{x}_{1:T}^{(1)}, \ldots, \boldsymbol{x}_{1:T}^{(n)}$. The α-Alternator first applies a binary mask to these sequences,

$$m_t^{(i)} \sim \text{Bernoulli}(p_{\text{mask}}) \text{ for all } t \in \{1, \ldots, T\} \text{ and for } i \in \{1, \ldots, n\} \tag{5}$$

$$\tilde{\boldsymbol{x}}_t^{(i)} = m_t^{(i)} \cdot \boldsymbol{x}_t^{(i)} + (1 - m_t^{(i)}) \cdot \mathbf{0} \text{ for all } t \in \{1, \ldots, T\} \text{ and for } i \in \{1, \ldots, n\} \tag{6}$$

where $0 \leq p_{\text{mask}} \leq 1$ is a given masking rate and $\mathbf{0}$ denotes the null vector. At each time step t, the α-Alternator then computes the *noisiness* of the element $\tilde{\boldsymbol{x}}_t^{(i)}$ at that time step using the VS. More specifically, the noisiness of $\tilde{\boldsymbol{x}}_t^{(i)}$, which we denote by $\mathrm{VS}_t^{(i)}$, is defined as the VS of two shifted versions of $\tilde{\boldsymbol{x}}_{1:T}^{(i)}$,

$$\mathrm{VS}_t^{(i)} = \mathrm{VS}\left(\left\{ \tilde{\boldsymbol{x}}_{t-L:t}^{(i)}, \tilde{\boldsymbol{x}}_{t-L+1:t+1}^{(i)} \right\}; k \right) \tag{7}$$

Algorithm 1: Sequence modeling with the α-Alternator

Inputs: Data $\boldsymbol{x}_{1:T}^{(1:n)}$, batch size B, variances σ_x^2 and σ_z^2, and masking rate p_{mask}
Initialize model parameters θ, ϕ, w, and b
while *not converged* **do**
  for $b = 1, \ldots, B$ **do**
    Draw initial latent $\boldsymbol{z}_0^{(b)} \sim \mathcal{N}(0, I_{D_z})$
    for $t = 1, \ldots, T$ **do**
      Compute $\boldsymbol{\mu}_{x_t}^{(b)} = \sqrt{(1-\sigma_x^2)} \cdot f_\theta(\boldsymbol{z}_{t-1}^{(b)})$
      Sample binary mask $m_t \sim \text{Bernoulli}(p_{mask})$
      Apply random masking $\tilde{\mathbf{x}}_t^{(b)} = m_t \cdot \mathbf{x}_t^{(b)} + (1 - m_t) \cdot \mathbf{0}$
      Compute adaptive weight $\alpha_t^{(b)} = \sigma\left(w \cdot \text{VS}_t^{(b)} + b\right) \cdot (1 - \sigma_z^2 - \epsilon_0)$
      Compute $\boldsymbol{\mu}_{z_t}^{(b)} = \sqrt{\alpha_t^{(b)}} \cdot g_\phi(\tilde{\boldsymbol{x}}_t^{(b)}) + \sqrt{(1 - \alpha_t^{(b)} - \sigma_z^2)} \cdot \boldsymbol{z}_{t-1}^{(b)}$
      Sample latent $\boldsymbol{z}_t^{(b)} \sim \mathcal{N}\left(\boldsymbol{\mu}_{z_t}^{(b)}, \sigma_z^2\right)$
    end
  end
  Compute loss $\mathcal{L}(\theta, \phi, w, b)$ in Eq. 9
  Backpropagate and update parameters θ, ϕ, w, and b
end

where $k(\cdot, \cdot)$ is a given positive semi-definite kernel and L is a given window length. The influence of $\tilde{\boldsymbol{x}}_t^{(i)}$ is then determined by

$$\alpha_t^{(i)} = \sigma\left(w \cdot \text{VS}_t^{(i)} + b\right) \cdot (1 - \sigma_z^2 - \epsilon_0), \tag{8}$$

where w and b are unknown scalar parameters, σ_z^2 denotes the variance of the distribution $p_\phi(\boldsymbol{z}_t | \boldsymbol{z}_{t-1}, \boldsymbol{x}_t)$ as described in Section 2, and ϵ_0 represents a small constant added to ensure numerical stability. This definition of α_t abides by the constraint $0 \leq \alpha_t < 1 - \sigma_z^2$ that it should satisfy (Rezaei & Dieng, 2024).

The α-Alternator also modifies the original Alternator loss function described in Eq. 2, using the adaptive $\alpha_t^{(i)}$ defined above,

$$\mathcal{L}(\theta, \phi, w, b) = \frac{1}{n} \sum_{i=1}^{n} \mathbb{E}_{p_{\theta,\phi}(\mathbf{z}_{0:T})} \left[\sum_{t=1}^{T} \left\| \mathbf{z}_t^{(i)} - \mu_{z_t}^{(i)} \right\|_2^2 + \alpha_t^{(i)} \frac{D_z \sigma_z^2}{D_x \sigma_x^2} \left\| \mathbf{x}_t^{(i)} - \mu_{x_t}^{(i)} \right\|_2^2 \right]. \tag{9}$$

where $\boldsymbol{x}_t^{(i)}$ is the element at time t of the i^{th} sequence, before any masking is applied, and $\mathbf{z}_t^{(i)}$ and $\mu_{z_t}^{(i)}$ are defined as

$$\boldsymbol{\mu}_{z_t}^{(i)} = \sqrt{\alpha_t^{(i)}} \cdot g_\phi(\tilde{\boldsymbol{x}}_t^{(i)}) + \sqrt{(1 - \alpha_t^{(i)} - \sigma_z^2)} \cdot \boldsymbol{z}_{t-1}^{(i)} \text{ and } \boldsymbol{z}_t^{(i)} \sim \mathcal{N}\left(\boldsymbol{\mu}_{z_t}^{(i)}, \sigma_z^2\right).$$

The role of $\alpha_t^{(i)}$ in determining the influence of the current observation $\boldsymbol{x}_t^{(i)}$ when predicting the latent dynamics is apparent. When $\alpha_t^{(i)}$ is high, $\boldsymbol{x}_t^{(i)}$ has a strong influence on the prediction of the dynamics. On the other hand, when $\alpha_t^{(i)}$ is low, $\boldsymbol{x}_t^{(i)}$ has a low influence on the prediction of the latent dynamics and the model relies more on the latent history $\boldsymbol{z}_{t-1}^{(i)}$, which ensures smooth transitions and temporal consistency. In terms of the loss function $\mathcal{L}(\theta, \phi, w, b)$, $\alpha_t^{(i)}$ affects the reconstruction error terms for both $\boldsymbol{z}_t^{(i)}$ and $\boldsymbol{x}_t^{(i)}$. Since $\alpha_t^{(i)}$ scales the contribution of $\boldsymbol{x}_t^{(i)}$ in the loss, it dynamically adjusts the importance of the observation-based error term relative to the latent transition error term. This enables the model to adaptively shift between short-term reactivity and long-term memory, making it well-suited for handling diverse temporal structures in sequence modeling. Algorithm 1 describes the complete training procedure.

Once trained, sampling new sequences from the α-Alternator is simple, and Algorithm 2 describes the procedure.

Algorithm 2: Sampling sequences using the α-Alternator

Inputs: Variances σ_x^2, σ_z^2, constant ϵ_0, and learned parameters θ, ϕ, w, b
Draw initial latent $\boldsymbol{z}_0 \sim \mathcal{N}(0, I_{D_z})$
for $t = 1, \ldots, T$ **do**
  Draw noise variables $\boldsymbol{\epsilon}_{xt} \sim \mathcal{N}(0, I_{D_x})$ and $\boldsymbol{\epsilon}_{zt} \sim \mathcal{N}(0, I_{D_z})$
  Draw $\boldsymbol{x}_t = \sqrt{(1-\sigma_x^2)} \cdot f_\theta(\boldsymbol{z}_{t-1}) + \sigma_x \cdot \boldsymbol{\epsilon}_{xt}$
  Compute adaptive weight $\alpha_t = \sigma\left(w \cdot \mathrm{VS}_t + b\right) \cdot (1 - \sigma_z^2 - \epsilon_0)$
  Draw $\boldsymbol{z}_t = \sqrt{\alpha_t} \cdot g_\phi(\boldsymbol{x}_t) + \sqrt{(1-\alpha_t-\sigma_z^2)} \cdot \boldsymbol{z}_{t-1} + \sigma_z \cdot \boldsymbol{\epsilon}_{zt}$
end

4 Experiments

In this section, we test the α-Altenator against strong baselines on neural decoding and time-series forecasting.

4.1 Neural Decoding

Neural decoding is a fundamental challenge in neuroscience, essential for understanding the mechanisms linking brain function and behavior. In neural decoding, neural data are translated into information about variables such as movement, decision-making, perception, or cognitive functions (Donner et al., 2009; Lin et al., 2022; Rezaei et al., 2018; 2023).

We use the α-Alternator to decode neural activities from three distinct experiments, each targeting a different brain region with specialized functional roles.

In the first experiment, we recorded the 2D velocity of a monkey as it controlled a cursor on a screen, alongside a 21-minute recording from the Motor Cortex (MC), capturing activity from 164 neurons. The motor cortex is responsible for planning and executing voluntary movements, making it a critical region for decoding motion-related neural signals.

The second experiment involved the same monkey performing a similar cursor control task; however, instead of the motor cortex, neural recordings were obtained from the Somatosensory Cortex (SS). This 51-minute recording included 52 neurons. The somatosensory cortex processes sensory inputs such as touch, proprioception, and movement-related feedback, allowing us to explore how sensory-driven neural activity contributes to movement execution and adaptation.

Finally, in the third experiment, we recorded the 2D positions of a rat as it navigated a platform in search of rewards. This session lasted 75 minutes and captured activity from 46 neurons in the hippocampus, a brain region essential for spatial memory and navigation. The hippocampus contains "place cells" that encode location-specific information, providing insights into how neural representations guide movement in a learned environment.

Our objective in using the α-Alternator on these varied datasets is to demonstrate its effectiveness across brain regions responsible for different cognitive and behavioral roles, such as motor control, sensory integration, and spatial memory/navigation. For more details on datasets, we refer the reader to Glaser et al. (2020; 2018). The time horizons for these experiments were divided into 1-second windows for decoding, with a time resolution of 5 milliseconds. We used the first 70% of each recording for training and the remaining 30% as the test set. In this experiment, we define the features as the velocity/position and the observations as the neural activity data.

4.1.1 Empirical setup

For the two networks with parameters θ and ϕ of the α-Alternator, we used attention-based models with two layers, each followed by a hidden layer containing 10 units. We set $\sigma_z = 0.1$ and $\sigma_x = 0.2$. We used a window length $L = 10$ and set $q = 0.2$ when computing the VS with an RBF kernel.

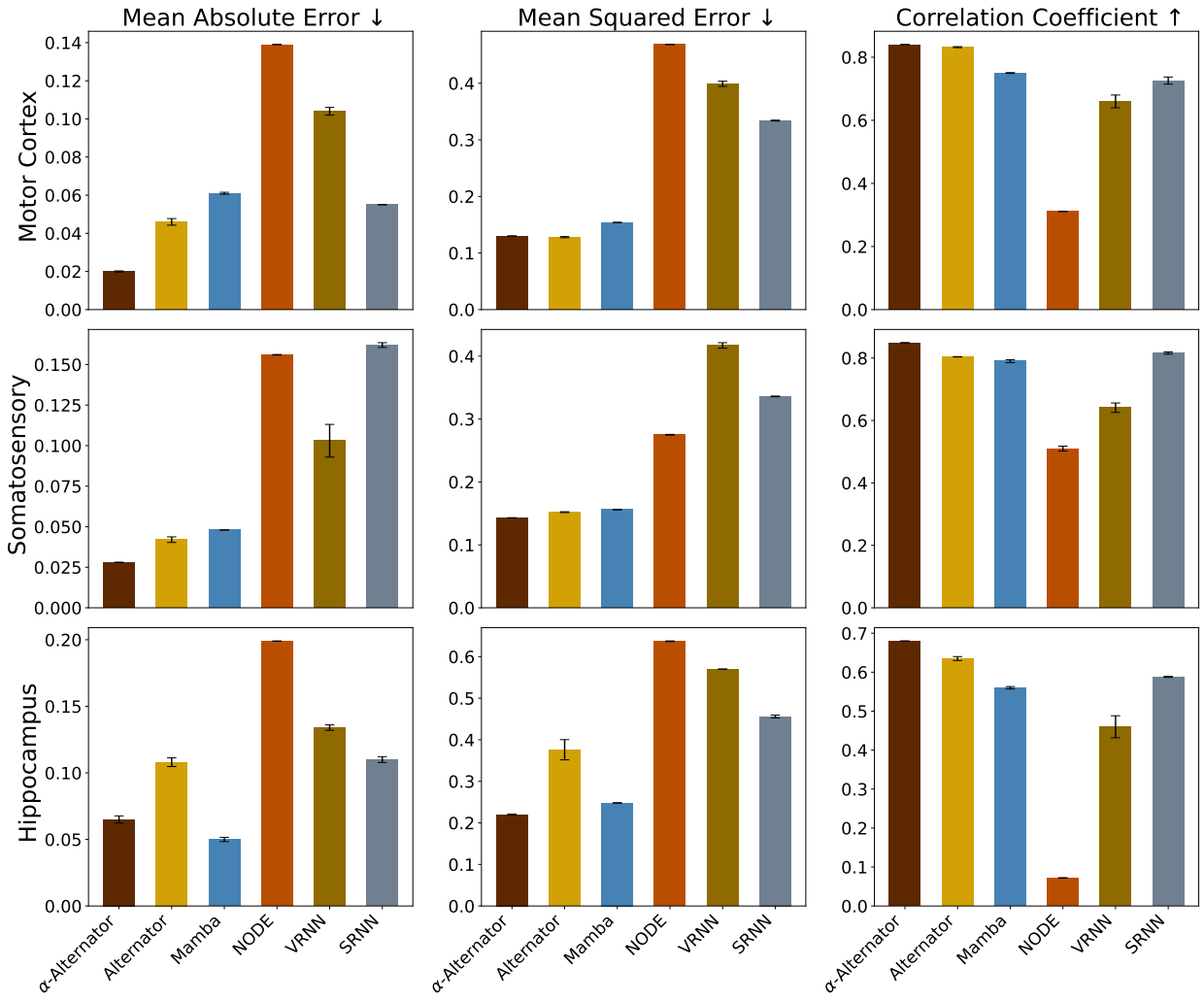

Figure 2: The α-Alternator outperforms other models on trajectory prediction in the neural decoding task on all three datasets in terms of MSE and CC. In terms of MAE, the α-Alternator outperforms the baselines on all datasets except the Hippocampus dataset, which has lower temporal diversity as shown in Figure 3.

The model was trained for 500 epochs on three different datasets: Motor Cortex, Somatosensory, and Hippocampus. We used the Adam optimizer with an initial learning rate of 0.01. Additionally, a cosine annealing learning rate scheduler was applied, with a minimum learning rate of $1e^{-3}$ and 5 warm-up epochs to stabilize the early training phase. We benchmarked the α-Alternator on its ability to accurately predict velocity/position given neural activity against the Alternator, the Mamba, VRNN (Chung et al., 2015), SRNN (Fraccaro et al., 2016), and Neural ODE or NODE (Chen et al., 2018).

4.1.2 Results

As Figure 2 shows, the α-Alternator achieves superior performance across all three neural datasets, showing particular strength in handling complex neural decoding tasks. In the Motor Cortex dataset, the model achieves notably lower MAE compared to all baselines, including Mamba, NODE, VRNN, and SRNN, while maintaining the highest CC of approximately 80%. This improvement is especially significant given the Motor Cortex's intricate temporal patterns, where the α-Alternator's adaptive mechanism appears to capture motion-related neural dynamics more effectively than traditional approaches.

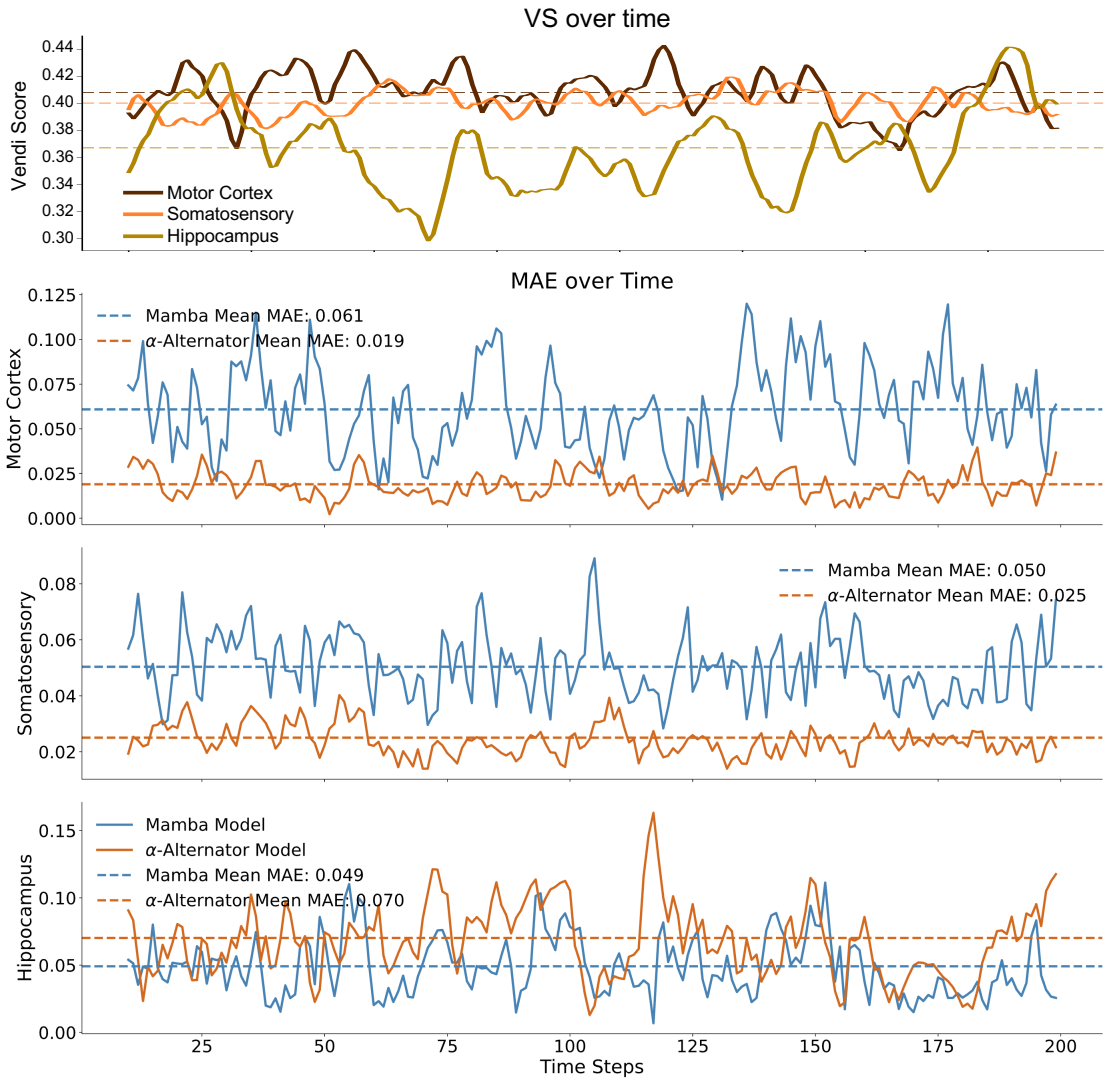

Figure 3: VS over time for the Motor Cortex, Hippocampus, and Somatosensory Cortex datasets. Lower VS values in the Hippocampus indicate less diverse observations across time steps, leading to a diminished effect of the adaptive mechanism in the α-Alternator compared to the Mamba. In contrast, for the Motor Cortex and the Somatosensory datasets, the α-Alternator effectively leverages VS-based adaptation, outperforming the Mamba in handling varying noise levels.

The superiority of the α-Alternator extends to the Somatosensory dataset, where it maintains consistently lower error rates across both MAE and MSE metrics. While the baselines, particularly the Mamba and the NODE, show competitive performance in terms of CC, the α-Alternator achieves better overall accuracy. This suggests enhanced capability in processing complex somatosensory inputs, where precise temporal relationships are crucial for accurate decoding.

In the Hippocampus dataset, the α-Alternator outperforms most of the baselines across multiple performance metrics. It achieves significantly lower MSE values while maintaining the highest CC among all tested models. However, the α-Alternator does not surpass the Mamba in terms of MAE on this dataset, despite its advantages in MSE and CC. As shown in Fig. 3, the average temporal diversity (as measured by VS) in the Hippocampus dataset is lower compared to the Motor Cortex and Somatosensory datasets. This

Table 1: Ablation study. The MAE, MSE, and CC between the true and predicted trajectories in the neural decoding task on three different datasets are better when using the two ingredients that make up the α-Alternator, the noise adaptation mechanism for α_t and the observation masking for training.

Dataset	Adaptive α_t?	Masking?	MAE↓	MSE↓	CC↑
Motor Cortex	✗	✗	0.041	0.130	0.837
	✗	✓	0.046	**0.128**	0.832
	✓	✗	0.057	0.158	0.796
	✓	✓	**0.023**	0.131	**0.841**
Somatosensory	✗	✗	0.042	0.152	0.804
	✗	✓	0.038	0.147	0.825
	✓	✗	0.042	0.179	0.771
	✓	✓	**0.028**	**0.143**	**0.849**
Hippocampus	✗	✗	0.108	0.376	0.635
	✗	✓	0.081	0.332	0.651
	✓	✗	0.067	0.225	0.671
	✓	✓	**0.065**	**0.222**	**0.681**

suggests that the observations in the Hippocampus dataset are less diverse over time, which may reduce the effectiveness of the α-Alternator's adaptive weighting mechanism.

4.1.3 Ablation study

Our ablation study demonstrates the significant benefits of combining the noise adaptation mechanism, i.e. adaptive α_t, with masking across three neural datasets. The experimental results, shown in Table 1, reveal consistent performance improvements when both components are utilized together.

In the Motor Cortex dataset, the combination of adaptive α_t and masking achieved the best performance with an MAE of 0.023 and CC of 0.841, representing a 43.9% reduction in MAE compared to the baseline model (no adaptive α_t, no masking). While using masking alone showed modest improvements in MSE, the full model's superior performance in MAE and CC highlights the synergistic effect of combining both ingredients.

The Somatosensory dataset exhibited similar trends, with the complete model achieving optimal results across all metrics. The improvement is particularly noteworthy compared to using either component in isolation, where masking alone or adaptive α_t alone showed limited benefits. The full model demonstrated a 33.3% reduction in MAE from the baseline configuration.

Most notably, the Hippocampus dataset showcased the strongest complementary effects between adaptive α_t and masking. The complete model achieved the best performance across all metrics, representing a substantial 39.8% improvement in MAE compared to the baseline configuration. Interestingly, both adaptive α_t and masking showed individual benefits on this dataset, but their combination led to better results.

These results consistently demonstrate that while each component offers certain advantages independently, their combination produces the most robust and accurate predictions across different neural regions.

4.1.4 Missing value imputation

As shown in Figure 4, the α-Alternator demonstrates strong robustness in handling missing values, consistently outperforming other models in imputation across neural datasets, even under extreme missing rates ranging from 10% to 95%. The α-Alternator achieves a lower MAE of approximately 0.06 for the Motor Cortex dataset, surpassing the Mamba model (MAE $\approx$ 0.10) and showing a particularly notable improvement over NODE and VRNN, both of which have MAE values exceeding 0.20. The model also excels in MSE performance, maintaining consistently lower values around 0.35, whereas baseline models, including the Mamba, exhibit higher variability and error rates exceeding 0.5. Moreover, the α-Alternator sustains

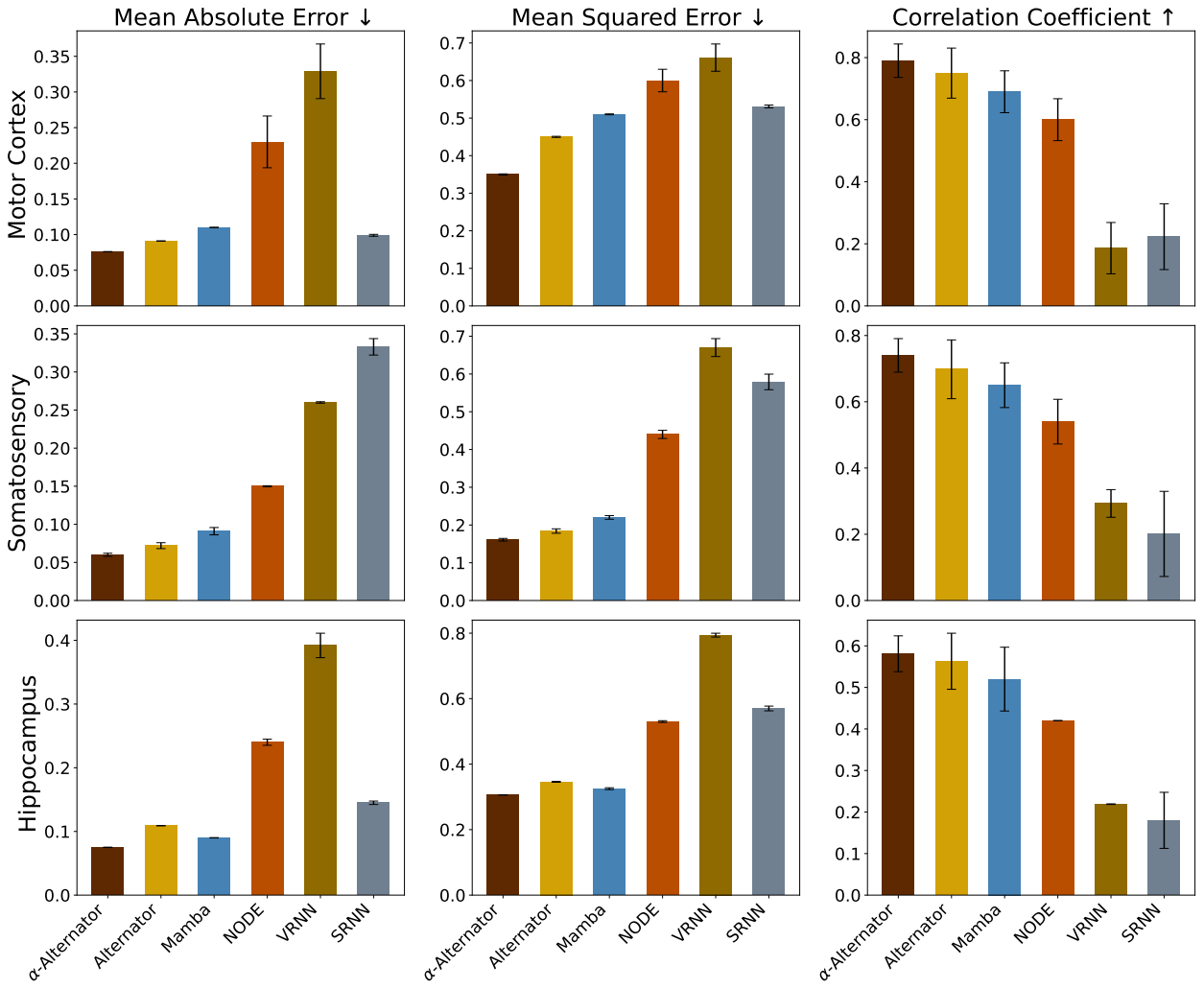

Figure 4: Comparison of performance on neural imputation across different brain regions. The α-Alternator consistently outperforms the baselines in imputing missing values across Motor Cortex, Somatosensory, and Hippocampus datasets. Results are averaged across missing value rates ranging from 10% to 95%, with performance measured using MAE, MSE, and CC. Vertical bars indicate standard errors across different missing value rates. The α-Alternator achieves notably lower errors and higher CCs across all three neural regions, with particularly strong performance in the complex Hippocampus dataset.

a high CC of approximately 0.78, substantially outperforming other models even under challenging missing value conditions. Similar trends are observed in the other datasets.

For the Somatosensory dataset, the α-Alternator demonstrates even greater advantages in imputation performance. The model consistently achieves the lowest MAE (approximately 0.06). Furthermore, its improvements in CC are especially notable, reaching values close to 0.75, while competing models struggle to maintain reliable correlations under high missing value rates, with the Mamba, for instance, achieving a CC of only around 0.65.

The Hippocampus dataset poses the most challenging imputation scenario due to its complex spatiotemporal dependencies, yet the α-Alternator exhibits good imputation performance on this dataset. The model consistently achieves a lower MSE of approximately 0.07, outperforming the Mamba (MSE = 0.1) and the other baselines. The narrow standard error bars of the α-Alternator across all metrics indicate stable predictive performance across varying missing value rates, suggesting that the model's adaptive mechanism

Table 2: Forecasting results for the α-Alternator and several strong baselines on the Electricity, Exchange, Weather, and Solar-Energy datasets. The lookback length L is set to 96 and the forecast length T is set to 96, 192, 336, 720. **blue** indicates the best performance while **orange** indicates second-best performance.

Models		$\alpha-$Alternator		S-Mamba		iTransformer		**Alternator**		Crossformer		TiDE		TimesNet		DLinear		FEDformer		Autoformer	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Electricity	96	**0.142**	**0.238**	**0.139**	**0.235**	0.148	0.240	0.223	0.318	0.219	0.314	0.237	0.329	0.168	0.272	0.197	0.282	0.193	0.308	0.201	0.317
	192	**0.157**	**0.252**	**0.159**	0.255	0.162	**0.253**	**0.162**	0.256	0.231	0.322	0.236	0.330	0.184	0.289	0.196	0.285	0.201	0.315	0.222	0.334
	336	**0.169**	**0.266**	0.176	0.272	0.178	**0.269**	**0.175**	0.270	0.246	0.337	0.249	0.344	0.198	0.300	0.209	0.301	0.214	0.329	0.231	0.338
	720	**0.193**	**0.289**	**0.204**	0.298	0.225	0.317	0.208	**0.297**	0.280	0.363	0.284	0.373	0.220	0.320	0.245	0.333	0.246	0.355	0.254	0.361
	Avg	**0.165**	**0.259**	**0.170**	**0.265**	0.178	0.270	0.192	0.285	0.244	0.334	0.251	0.344	0.192	0.295	0.212	0.300	0.214	0.327	0.227	0.338
Exchange	96	**0.086**	**0.207**	**0.086**	**0.206**	**0.086**	**0.206**	0.093	0.216	0.256	0.367	0.094	0.218	0.107	0.234	**0.088**	0.218	0.148	0.278	0.197	0.323
	192	0.178	**0.300**	0.182	0.304	**0.177**	**0.299**	0.183	0.306	0.470	0.509	0.184	0.307	0.226	0.344	**0.176**	0.315	0.271	0.315	0.300	0.369
	336	0.332	**0.417**	**0.328**	**0.415**	0.331	**0.417**	0.336	0.420	1.268	0.883	0.349	0.431	0.367	0.448	**0.313**	0.427	0.460	0.427	0.509	0.524
	720	**0.836**	**0.689**	0.867	0.703	0.847	**0.691**	0.855	0.698	1.767	1.068	0.852	0.698	0.964	0.746	**0.839**	0.695	1.195	0.695	1.447	0.941
	Avg	**0.358**	**0.403**	0.367	**0.408**	0.360	**0.403**	0.366	0.410	0.940	0.707	0.370	0.413	0.416	0.443	**0.354**	0.414	0.519	0.429	0.613	0.539
Weather	96	**0.165**	**0.207**	**0.165**	**0.210**	0.174	0.214	0.175	0.215	**0.158**	0.230	0.202	0.261	0.172	0.220	0.196	0.255	0.217	0.296	0.266	0.336
	192	0.228	0.260	**0.214**	**0.252**	0.221	**0.254**	0.222	0.258	**0.206**	0.277	0.242	0.298	0.219	0.261	0.237	0.296	0.276	0.336	0.307	0.367
	336	**0.272**	**0.295**	**0.274**	0.297	0.278	**0.296**	0.284	0.301	**0.272**	0.335	0.287	0.335	0.280	0.306	0.283	0.335	0.339	0.380	0.359	0.395
	720	0.351	0.348	**0.350**	**0.345**	0.358	**0.347**	0.362	0.353	0.398	0.418	0.351	0.386	0.365	0.359	**0.345**	0.381	0.403	0.428	0.419	0.428
	Avg	**0.254**	**0.278**	**0.251**	**0.276**	0.258	**0.278**	0.262	0.281	0.259	0.315	0.271	0.320	0.259	0.287	0.265	0.317	0.309	0.360	0.338	0.382
Solar-Energy	96	**0.202**	0.242	**0.205**	0.244	**0.203**	**0.237**	**0.205**	**0.238**	0.310	0.331	0.312	0.399	0.250	0.292	0.290	0.378	0.242	0.342	0.884	0.711
	192	**0.234**	**0.261**	0.237	0.270	**0.233**	**0.261**	0.239	**0.264**	0.0	0.725	0.339	0.416	0.296	0.318	0.320	0.398	0.285	0.380	0.834	0.692
	336	**0.248**	**0.276**	0.258	0.288	**0.248**	**0.273**	**0.250**	**0.276**	0.750	0.735	0.368	0.430	0.319	0.330	0.353	0.415	0.282	0.376	0.941	0.723
	720	**0.250**	**0.277**	0.260	0.288	**0.249**	**0.275**	0.253	0.279	0.769	0.765	0.370	0.425	0.338	0.337	0.356	0.413	0.357	0.427	0.882	0.717
	Avg	**0.234**	**0.264**	0.240	0.273	**0.233**	**0.262**	0.236	**0.264**	0.641	0.639	0.347	0.417	0.301	0.319	0.330	0.401	0.291	0.381	0.885	0.711

effectively captures the intricate patterns of hippocampal activity, even under substantial missing data settings.

The results of the missing value imputation task highlight the robust imputation capabilities of the α-Alternator, which excels even in challenging scenarios with high proportions of missing values.

4.2 Time-series forecasting

We evaluated the effectiveness of the α-Alternator across four time-series forecasting benchmarks, each presenting unique challenges. The forecasting performance of the α-Alternator and the baselines is measured using MAE and MSE across four different lookback lengths L. Table 2 summarizes the results. The best and second-best models are highlighted in blue and orange, respectively.

The **Electricity** dataset, which records hourly consumption patterns of 321 customers from 2012 to 2014, showcases the α-Alternator's superior performance in handling multivariate periodic data. Notably, the α-Alternator achieved the best performance with an average MSE of 0.165 and MAE of 0.259, outperforming both the Alternator and other state-of-the-art models. Compared to S-Mamba (MSE: 0.170, MAE: 0.265), the α-Alternator demonstrated a notable improvement across all forecasting horizons, particularly excelling in longer forecasting windows. In the challenging 720-hour forecast length, the α-Alternator maintained a lower MSE (0.193) and MAE (0.289) compared to S-Mamba (MSE: 0.204, MAE: 0.298), confirming its robustness in long-term forecasting.

The **Solar-Energy** dataset comprises 10-minute interval data from 137 photovoltaic plants. While iTransformer showed slightly better performance in terms of average metrics (MSE: 0.233, MAE: 0.262), the α-Alternator achieved similar results (MSE: 0.234, MAE: 0.264) and outperformed other models including S-Mamba with an average MSE of 0.240 and MAE of 0.273.

In the **Exchange** dataset, which presents the complex challenge of forecasting aperiodic daily exchange rates across eight countries from 1990 to 2016, the α-Alternator also outperformed the strongest baselines. The model achieved the best MAE of 0.403 and second-best MSE of 0.358 in average performance, showing particular strength in long-term forecasting where it secured the best performance in the 720-day horizon (MSE: 0.836, MAE: 0.689) setting, surpassing S-Mamba (MSE: 0.867, MAE: 0.703), highlighting its effectiveness in handling complex and volatile financial sequences.

For the **Weather** dataset, the α-Alternator achieved overall strong performance (MSE: 0.254, MAE: 0.278), closely following S-Mamba (MSE: 0.251, MAE: 0.276).

Overall, the α-Alternator emerges as the top-performing model for these challenging time-series forecasting benchmarks, ranking first or second in most scenarios.

5 Related Work

5.1 State-Space Models

State-space models (SSMs) have emerged as a popular framework for modeling time-dependent data across various domains (Gu & Dao, 2023; Rezaei et al., 2022; 2021; Auger-Méthé et al., 2021; Rangapuram et al., 2018). Recent advancements include the Mamba architecture (Gu & Dao, 2023), which employs a selective state space mechanism defined by

$$\mathbf{h}_t = \text{SSM}(\boldsymbol{x}_t, \mathbf{h}_{t-1}), \quad \boldsymbol{y}_t = \text{Linear}(\mathbf{h}_t)$$

where $\mathbf{h}_t$ represents the hidden state, $\boldsymbol{x}_t$ is the input, and $\boldsymbol{y}_t$ is the output at time t. In contrast, the α-Alternator employs a dynamic state transition, see Algorithm 2, where $\mathbf{z}_t$ serves an analogous role to Mamba's $\mathbf{h}_t$ but with explicit control over state transitions through α_t. While Mamba has demonstrated success in applications from speech recognition (Zhang et al., 2024) to protein folding (Xu et al., 2024), its architecture requires high-dimensional hidden states $\mathbf{h}_t \in \mathbb{R}^d$ that have the same dimensionality as the data. The α-Alternator addresses this limitation by operating in a lower-dimensional latent space $\mathbf{z}_t \in \mathbb{R}^{d_z}$ where $d_z \ll d$, while incorporating the adaptive weighting mechanism to balance between observation influence and state persistence. The lower dimensional state of the α-Alternator ($d_z \ll d$) yields reduced computational complexity while the adaptive weighting mechanism is particularly beneficial for stochastic processes like neural recordings where noise characteristics vary significantly over time.

5.2 Alternators

The Alternator framework (Rezaei & Dieng, 2024) represents a significant departure from traditional SSMs by introducing a dual-network architecture that alternates between producing observations and low-dimensional latent variables over time. The parameters of these two networks are learned by minimizing a cross entropy criterion over the resulting trajectories (Rezaei & Dieng, 2024). This approach has demonstrated superior performance compared to established methods such as Neural ODEs (Chen et al., 2018), dynamical VAEs such as VRNNs (Gregor et al., 2014), and diffusion models (Dutordoir et al., 2022; Lin et al., 2023) across various sequence modeling tasks. However, the Alternator uses a fixed weighting parameter α when defining the mean of the latent states, which is limiting. The α-Alternator extends this framework by letting α vary across time steps using the Vendi Score to automatically adjust its reliance on observations versus latent history. The α-Alternator maintains the computational efficiency of the original Alternator while providing greater robustness to temporal variations in sequence noise. Furthermore, the α-Alternator's masking strategy during training strengthens its ability to handle missing or corrupted data, a common challenge in real-world applications such as neural decoding and time-series forecasting.

6 Conclusion

In this work, we introduced the α-Alternator, a novel sequence model designed to overcome the limitations of Alternators and existing state-space models by dynamically adapting to varying noise levels in sequences. The α-Alternator leverages the Vendi Score to determine the influence of sequence elements on the prediction of the latent dynamics through a gating mechanism. This same influence score is used to weigh the data reconstruction term in the Alternator loss. The model is trained by masking sequence elements at random during training to simulate varying noise levels. We demonstrate the effectiveness of the α-Alternator through an extensive empirical study on neural decoding and time-series forecasting tasks, where we show that it consistently outperforms several state-of-the-art sequence models, including Mambas and Alternators.

References

Reyhane Askari Hemmat, Melissa Hall, Alicia Sun, Candace Ross, Michal Drozdzal, and Adriana Romero-Soriano. Improving geo-diversity of generated images with contextualized vendi score guidance. In *European Conference on Computer Vision*, pp. 213–229. Springer, 2024.

Marie Auger-Méthé, Ken Newman, Diana Cole, Fanny Empacher, Rowenna Gryba, Aaron A King, Vianey Leos-Barajas, Joanna Mills Flemming, Anders Nielsen, Giovanni Petris, et al. A guide to state–space modeling of ecological time series. *Ecological Monographs*, 91(4):e01470, 2021.

Sebastian Berns, Simon Colton, and Christian Guckelsberger. Towards Mode Balancing of Generative Models via Diversity Weights. *arXiv preprint*, 2023. arXiv:2304.11961 [cs.LG].

Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. *Advances in neural information processing systems*, 31, 2018.

Junyoung Chung, Kyle Kastner, Laurent Dinh, Kratarth Goel, Aaron C Courville, and Yoshua Bengio. A recurrent latent variable model for sequential data. *Advances in neural information processing systems*, 28, 2015.

Tobias H Donner, Markus Siegel, Pascal Fries, and Andreas K Engel. Buildup of choice-predictive activity in human motor cortex during perceptual decision making. *Current Biology*, 19(18):1581–1585, 2009.

Vincent Dutordoir, Alan Saul, Zoubin Ghahramani, and Fergus Simpson. Neural diffusion processes. *arXiv preprint arXiv:2206.03992*, 2022.

Marco Fraccaro, Søren Kaae Sønderby, Ulrich Paquet, and Ole Winther. Sequential neural models with stochastic layers. *Advances in neural information processing systems*, 29, 2016.

Dan Friedman and Adji Bousso Dieng. The Vendi Score: A Diversity Evaluation Metric for Machine Learning. *Transactions on Machine Learning Research*, 2023.

Joshua I Glaser, Matthew G Perich, Pavan Ramkumar, Lee E Miller, and Konrad P Kording. Population coding of conditional probability distributions in dorsal premotor cortex. *Nature communications*, 9(1): 1788, 2018.

Joshua I Glaser, Ari S Benjamin, Raeed H Chowdhury, Matthew G Perich, Lee E Miller, and Konrad P Kording. Machine learning for neural decoding. *Eneuro*, 7(4), 2020.

Karol Gregor, Ivo Danihelka, Andriy Mnih, Charles Blundell, and Daan Wierstra. Deep autoregressive networks. In *International Conference on Machine Learning*, pp. 1242–1250. PMLR, 2014.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023.

Nithish Kannen, Arif Ahmad, Marco Andreetto, Vinodkumar Prabhakaran, Utsav Prabhu, Adji Bousso Dieng, Pushpak Bhattacharyya, and Shachi Dave. Beyond aesthetics: Cultural competence in text-to-image models. *arXiv preprint arXiv:2407.06863*, 2024.

Baihan Lin, Djallel Bouneffouf, and Guillermo Cecchi. Predicting human decision making in psychological tasks with recurrent neural networks. *PloS one*, 17(5):e0267907, 2022.

Lequan Lin, Zhengkun Li, Ruikun Li, Xuliang Li, and Junbin Gao. Diffusion models for time series applications: A survey. *arXiv preprint arXiv:2305.00624*, 2023.

Tsung-Wei Liu, Quan Nguyen, Adji Bousso Dieng, and Diego A Gómez-Gualdrón. Diversity-driven, efficient exploration of a mof design space to optimize mof properties. *Chemical Science*, 15(45):18903–18919, 2024.

Quan Nguyen and Adji Bousso Dieng. Quality-weighted vendi scores and their application to diverse experimental design. *arXiv preprint arXiv:2405.02449*, 2024.

Amey P. Pasarkar and Adji Bousso Dieng. Cousins Of The Vendi Score: A Family Of Similarity-Based Diversity Metrics For Science And Machine Learning. In *International Conference on Artificial Intelligence and Statistics*, pp. 3808–3816. PMLR, 2024.

Amey P Pasarkar, Gianluca M Bencomo, Simon Olsson, and Adji Bousso Dieng. Vendi Sampling For Molecular Simulations: Diversity As A Force For Faster Convergence And Better Exploration. *The Journal of Chemical Physics*, 159(14), 2023.

Syama Sundar Rangapuram, Matthias W Seeger, Jan Gasthaus, Lorenzo Stella, Yuyang Wang, and Tim Januschowski. Deep state space models for time series forecasting. *Advances in neural information processing systems*, 31, 2018.

Mohammad R Rezaei, Anna K Gillespie, Jennifer A Guidera, Behzad Nazari, Saeid Sadri, Loren M Frank, Uri T Eden, and Ali Yousefi. A comparison study of point-process filter and deep learning performance in estimating rat position using an ensemble of place cells. In *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pp. 4732–4735. IEEE, 2018.

Mohammad R Rezaei, Alex E Hadjinicolaou, Sydney S Cash, Uri T Eden, and Ali Yousefi. Direct discriminative decoder models for analysis of high-dimensional dynamical neural data. *Neural Computation*, 34 (5):1100–1135, 2022.

Mohammad R Rezaei, Haseul Jeoung, Ayda Gharamani, Utpal Saha, Venkat Bhat, Milos R Popovic, Ali Yousefi, Robert Chen, and Milad Lankarany. Inferring cognitive state underlying conflict choices in verbal stroop task using heterogeneous input discriminative-generative decoder model. *Journal of Neural Engineering*, 20(5):056016, 2023.

Mohammad Reza Rezaei and Adji Bousso Dieng. Alternators for sequence modeling. *arXiv preprint arXiv:2405.11848*, 2024.

Mohammad Reza Rezaei, Kensuke Arai, Loren M Frank, Uri T Eden, and Ali Yousefi. Real-time point process filter for multidimensional decoding problems using mixture models. *Journal of neuroscience methods*, 348:109006, 2021.

Zihan Wang, Fanheng Kong, Shi Feng, Ming Wang, Xiaocui Yang, Han Zhao, Daling Wang, and Yifei Zhang. Is mamba effective for time series forecasting? *Neurocomputing*, 619:129178, 2025.

Bohao Xu, Yingzhou Lu, Yoshitaka Inoue, Namkyeong Lee, Tianfan Fu, and Jintai Chen. Protein-mamba: Biological mamba models for protein function prediction. *arXiv preprint arXiv:2409.14617*, 2024.

Xiangyu Zhang, Qiquan Zhang, Hexin Liu, Tianyi Xiao, Xinyuan Qian, Beena Ahmed, Eliathamby Ambikairajah, Haizhou Li, and Julien Epps. Mamba in speech: Towards an alternative to self-attention. *arXiv preprint arXiv:2405.12609*, 2024.