AN OPTIMALLY WEIGHTED ECHO STATE NEURAL NETWORK FOR HIGHLY CHAOTIC TIME SERIES MODELLING

Anonymous authors
Paper under double-blind review

ABSTRACT

We demonstrate the development and implementation of a series of echo state neural networks in conjunction with optimal weighted averaging to produce robust, model-free predictions of highly chaotic time series. We deploy our model on simulated mass accretion data sets governing the formation of a protostar, accretion disk and gas envelope. Our methodology extends the parallel series approach of averaging all reservoir outputs by instead selecting an optimal set of weights for each realization representative of what we believe constructs the optimal output signal that best represents the data's underlying temporal dynamics. The method is demonstrated by modelling hydrodynamic and stellar evolution simulations that are representative of highly chaotic systems.

1 INTRODUCTION

Neural networks have the capability to approximate any continuous function and are often referred to as universal approximators (Csáji, 2001). This has given neural networks tremendous ability to extract and estimate the underlying dynamical process occurring for a time series. As such, there has been interest in utilizing neural networks for time series forecasting. Echo state neural networks (ESN) implement the so called reservoir computing methodology as opposed to the more traditional multi-layered architecture of traditional neural networks. This reservoir acts as a complex dynamical system that maintains a given internal state dependent on its history (Pathak et al., 2017; Lukoševičius and Jaeger, 2009). This concept replicates a form of memory that the network has similar to traditional recurrent networks. Deployments of ESNs on benchmark chaotic time series such as the Mackay-Glass set (Jaeger and Haas, 2004) have demonstrated excellent temporal dynamic extraction as well as good short term predictions of the Lorenz system (Jaeger and Haas, 2004). Of course, like any form of prediction based modelling on highly chaotic time series, local errors become amplified for longer term predictions which hinders and limits these prediction time. Given the recent success in the deployment of ESNs, they do however present some statistical challenges primarily involved with generating the input and reservoir matrices. The reservoir matrix $\mathbf{W}_{\mathrm{r}}$ is taken as a $N_r \times N_r$ sparse random matrix aimed at providing rich recurrent connections for the input. The randomness associated with this introduces a degree of stochastic behaviour that can sometimes propagate to the network's output. Although the general trend of the output signal is relatively unchanged, having even minor differences in any model's output with all else fixed can be cumbersome when deployed in practice. Several alternatives have been demonstrated to mitigate such challenges utilizing ensemble methods and parallel series architectures (Liu et al., 2018; Wu et al., 2018). Here we propose an extension to the parallel series method. Our approach uses historical data to independently train and validate a series of reservoirs and constructs a constraint linear optimization problem seeking a set of optimal weights aimed at extracting what we believe is a unique, optimal path that best represents the underlying dynamics of a given time series. The method is structured in a way to take a weighted average of realizations rather than the standard mean. This weighted average approach is expected to give preference to better performing realizations and suppress poor performing ones. We find significant improvement in model performance for certain changes in the network's hyperparameters. We deploy our approach on a series highly chaotic time series governed from hydrodynamical and stellar evolution simulations (Vorobyov et al., 2017). Our model

demonstrates accurate short term performance, but we additionally quantify long term performance by studying the time evolving RMSE.

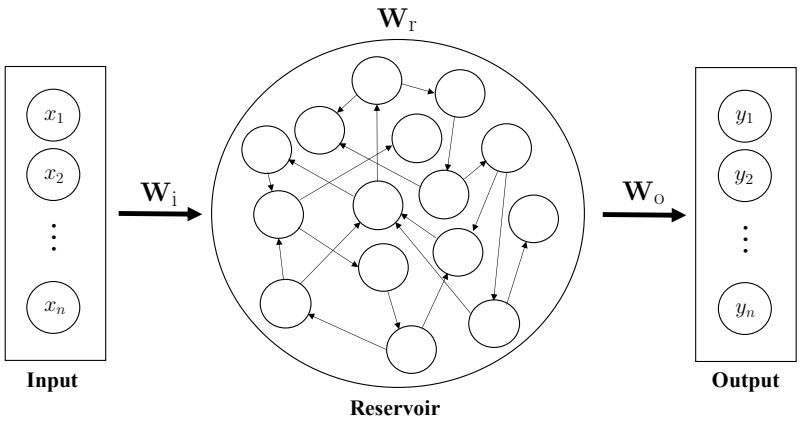

Figure 1: Standard architecture of a shallow ESN.

2 METHODS

2.1 SHALLOW ESN

The shallow ESN (single reservoir network) is composed of an input, reservoir and output layer as demonstrated in Figure 1 having weights $\mathbf{W}_\mathrm{i}$ and $\mathbf{W}_\mathrm{r}$ being randomly generated, sparse matrices. For an input time series $\mathbf{x}(t) = [x_1(t), x_2(t), \cdots, x_n(t)]^\top$, each quantity $x_i(t)$ is used to construct an $N_r \times 1$ reservoir state vector $\mathbf{v}_i$, initialized to $\mathbf{v}_1 = \mathbf{0}$ and updated according to

$$\mathbf{v}_{i+1} = (1-\alpha)\mathbf{v}_i + \alpha f_\mathrm{a}(\mathbf{W}_\mathrm{i} x_i(t) + \mathbf{W}_\mathrm{r}\mathbf{v}_i + \mathbf{W}_\mathrm{b}), \qquad 1 \leq i \leq n \tag{1}$$

where N_r is the reservoir size and $0 < \alpha < 1$ is the leaking rate. We additionally include a bias $\mathbf{W}_\mathrm{b}$ and adopt an activation function of the form $f_\mathrm{a}(x) = \tanh(x)$. We define a washout quantity $\omega < n$ as an initially discarded transient and for every $i > \omega$, the internal state $\mathbf{X}$ is contructed as

$$\mathbf{X} = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ x_i(t) & x_{i+1}(t) & \cdots & x_n(t) \\ \mathbf{v}_i & \mathbf{v}_{i+1} & \cdots & \mathbf{v}_n \end{bmatrix} \tag{2}$$

Furthermore, we compute the output matrix $\mathbf{W}_\mathrm{o}$ through a readout operation on the set $\overline{\mathbf{x}}(t) = [x_{\omega+1}(t), x_{\omega+2}(t), \cdots, x_n(t)]^\top$:

$$\mathbf{W}_\mathrm{o} = f_\mathrm{r}(\overline{\mathbf{x}}(t), \mathbf{X}) \tag{3}$$

where the readout is taken as a regression either in the form of the Tikhonov regularized regression (Ridge-Regression) or the Moore-Penrose inverse (Pseudo-Inverse):

$$f_\mathrm{r}(\overline{\mathbf{x}}, \mathbf{X}) = \overline{\mathbf{x}}^* \left(\mathbf{X}^*\mathbf{X}\right)^{-1} \mathbf{X}^* \qquad \text{(Pseudo-Inverse)}$$
$$f_\mathrm{r}(\overline{\mathbf{x}}, \mathbf{X}) = \overline{\mathbf{x}}^* \left(\mathbf{X}^*\mathbf{X} + \lambda\mathbf{I}\right)^{-1} \mathbf{X}^* \quad \text{(Ridge-Regression)}$$

where $\mathbf{I}$ is the identity matrix and $\lambda > 0$ acts as a regularization parameter. Once the output matrix is computed, the reservoir's output can be computed as:

$$\mathbf{y}(t) = \mathbf{W}_\mathrm{o}\mathbf{X} \tag{4}$$

Therefore, in order to produce an estimate to the quantity x_{n+1}, we construct the internal state using x_n and use $\mathbf{W}_\mathrm{o}$ to compute the output using equation 4.

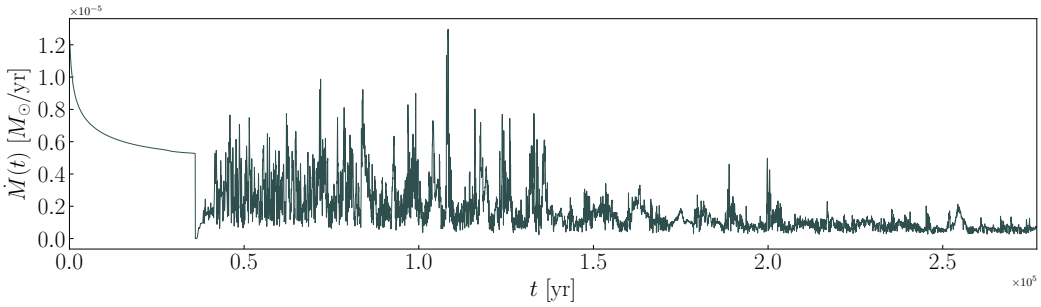

Figure 2: Mass accretion rate as a function of time for chaotic regime of model 29 per Vorobyov et al. (2017)

2.2 OWA-ESN

Liu et al. (2018) introduced a parallel series methodology to mitigate some of the statistical challenges of the shallow ESN. The parallel ESN effectively generates L independent input $\mathbf{W}_\mathrm{i}$ and reservoir $\mathbf{W}_\mathrm{r}$ matrices. The time series is trained and validated on each reservoir to form L output matrices $\mathbf{W}_\mathrm{o}$ each of which is used to output a unique signal. Finally, the model's final output signal $\widehat{\mathbf{y}}(t)$ is taken as the mean of all L realizations:

$$\widehat{\mathbf{y}}(t) = \frac{1}{L}\sum_j \mathbf{y}^{(j)}(t). \tag{5}$$

In our extension, we propose to take a weighted average approach that favors better performing outputs. The OWA-ESN assumes that the final output signal is a linear combination of all signals from all L reservoirs

$$\widehat{\mathbf{y}}(t;\hat{\boldsymbol{\beta}}) = \sum_j \hat{\beta}_j \mathbf{y}^{(j)}(t). \tag{6}$$

for a set of optimal weights $\hat{\beta}_j$ that minimizes the error of the final output. We treat this as an optimization problem of the form

$$\begin{aligned} \min_{\hat{\boldsymbol{\beta}}} \quad & \left\| \mathbf{x}_\mathrm{F}(t) - \hat{\mathbf{y}}(t;\hat{\boldsymbol{\beta}}) \right\|^2 \\ \text{subject to} \quad & \textstyle\sum_j \hat{\beta}_j = 1 \end{aligned} \tag{7}$$

on a set of the data used for optimization $\mathbf{x}_\mathrm{F}(t)$. Here, we are minimizing the sum of squared residuals, but other loss functions can be used as well. Since our approach aims to replicate a weighted average, we impose that $0 < \hat{\beta}_j < 1$. Furthermore, we require that the error be at least less than or equivalent to the smallest error of any single realization from the series.

$$\text{error}\left(\hat{\mathbf{y}}(t;\hat{\boldsymbol{\beta}})\right) \leq \text{error}\left(\mathbf{y}^{(j)}(t)\right) \quad \forall j \in (1, L) \tag{8}$$

The critical assumption with our method is that the set of weights that optimize the data segment $\mathbf{x}_\mathrm{F}(t)$ will additionally produce the best performing output outside of this regime. Therefore, in order to implement the OWA-ESN for out of sample predictions, we outline our methodology below:

1. We assume that the data available spans from t_0 up to some T. Anything beyond T is considered out of sample. Split the data into the respective training, validation and free running segments:
   (i) Training segment: $t_0 < t \leq t_\mathrm{T}$
   (ii) Validation segment: $t_\mathrm{T} < t \leq t_\mathrm{V}$
   (iii) Free running segment: $t_\mathrm{V} < t \leq T$
2. Generate L input and reservoir matrices independently.
3. Train (over the segment $t_0 < t \leq t_\mathrm{T}$) and validate (over the segment $t_\mathrm{T} < t \leq t_\mathrm{V}$) on each reservoir in the series according to the methodology discussed in subsection 2.1.

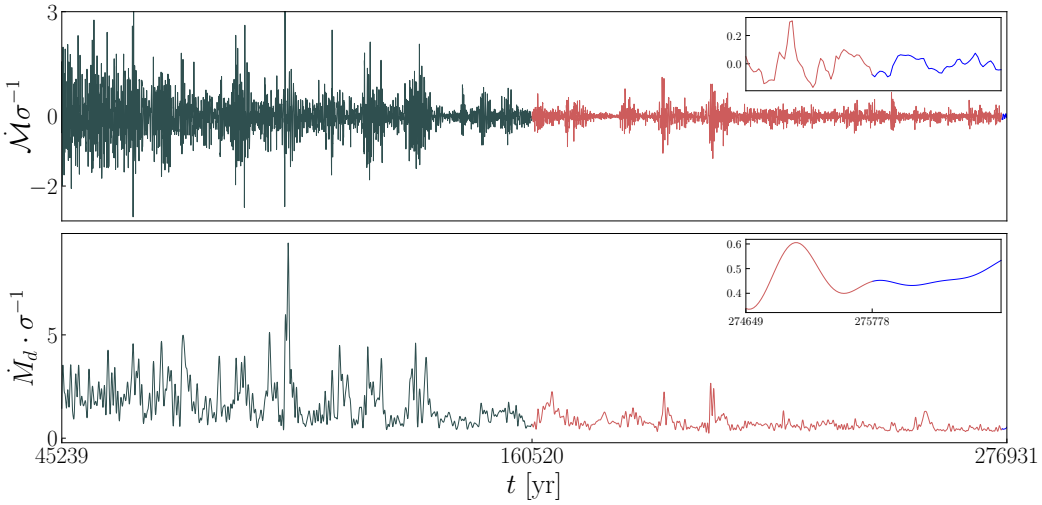

Figure 3: The model's standardized fluctuating component **(top)** and standardized deterministic component **(bottom)**. The time series demonstrates the respective training (grey), validation (red) and free running (blue) segments.

4. Simulate L independent free running signals from each reservoir up to some $t > T$ that is outside of the available data.
5. Perform the optimization on the free running segment of the data $t_{\text{V}} < t \leq T$ and compute the optimal weights $\hat{\beta}_j$.
6. Compute the weighted sum of all L free running signals using equation equation 6 to produce a forecast beyond time T.

Here, the term "free run" is representative of the network computing the quantity y_{i+1} using its previous output y_i. That is, after the network has been trained and validated, the input quantity at each time step will be the network's previous output.

2.3 Hydrodynamics & Stellar Evolution

We demonstrate the predictive capabilities of our OWA-ESN through its implementation on the time series obtained by calculating the rate of mass accretion onto a central protostar during the star formation process. An episodic accretion process has been demonstrated by Vorobyov and Basu (2006; 2010) by solving the following hydrodynamical equations:

$$\frac{\partial \Sigma}{\partial t} = -\nabla_{\text{p}} \cdot (\Sigma v_{\text{p}}) \tag{9}$$

$$\frac{\partial}{\partial t}(\Sigma v_{\text{p}}) + [\nabla \cdot (\Sigma v_{\text{p}} \otimes v_{\text{p}})]_{\text{p}} = -\nabla_{\text{p}} P + \Sigma g_{\text{p}} + (\nabla \cdot \Pi)_{\text{p}} \tag{10}$$

$$\frac{\partial e}{\partial t} + \nabla_{\text{p}} \cdot (e v_{\text{p}}) = -P(\nabla_{\text{p}} \cdot v_{\text{p}}) - \Omega + \Gamma + (\nabla \cdot v)_{\text{pp}'} : \Pi_{\text{pp}'} \tag{11}$$

which correspond to mass, momentum, and energy transport respectively. The equations model a disk geometry in a cylindrical coordinate system where physical variables are vertically integrated (z-direction) and have a dependence on the polar coordinates (r, ϕ) in the disk plane. Here Σ is the surface mass density, e is the internal energy per surface area, P is the vertically integrated pressure, $v_{\text{p}} = v_r \widehat{\boldsymbol{r}} + v_\phi \widehat{\boldsymbol{\phi}}$ is the velocity in the plane of the disk and $\nabla_{\text{p}} = \widehat{\boldsymbol{r}} \partial/\partial r + \widehat{\boldsymbol{\phi}} r^{-1} \partial/\partial \phi$ is the planar gradient operator (gradient taken along the plane of the disk). The gravitational acceleration in the disk plane g_{p} accounts for self gravity of the disk and the gravity of the protostar (Vorobyov and Basu, 2010). Lastly, Π is the viscous stress tensor, Ω is the radiative cooling

$$\Omega = F_c \sigma_b T_{\text{mp}}^4 \frac{\tau}{1 + \tau^2} \tag{12}$$

Table 1: Respective performance for each component over the free running segment

Low input scaling		
Readout	**MSE**	**Component**
Ridge	0.00069	$\dot{\mathcal{M}}(t) \cdot \sigma^{-1}$
Pseudo	0.00051	$\dot{\mathcal{M}}(t) \cdot \sigma^{-1}$
Ridge	0.00017	$\dot{M}_d(t) \cdot \sigma^{-1}$
Pseudo	5.31e$-$6	$\dot{M}_d(t) \cdot \sigma^{-1}$
Ridge	0.00089	$\dot{M}(t) \cdot \sigma^{-1}$
Pseudo	0.00049	$\dot{M}(t) \cdot \sigma^{-1}$

High input scaling (ϱ = 1)		
Readout	**MSE**	**Component**
Ridge	0.00048	$\dot{\mathcal{M}}(t) \cdot \sigma^{-1}$
Pseudo	0.00024	$\dot{\mathcal{M}}(t) \cdot \sigma^{-1}$
Ridge	1.75e$-$5	$\dot{M}_d(t) \cdot \sigma^{-1}$
Pseudo	4.77e$-$6	$\dot{M}_d(t) \cdot \sigma^{-1}$
Ridge	0.00049	$\dot{M}(t) \cdot \sigma^{-1}$
Pseudo	0.00025	$\dot{M}(t) \cdot \sigma^{-1}$

and Γ is the heating function:

$$\Gamma = F_c \sigma_b T_{\text{irr}}^4 \frac{\tau}{1+\tau^2} \tag{13}$$

where σ_b is the Stefan-Boltzmann constant, T_{mp} is the midplane temperature of gas, T_{irr} is the irradiation temperature from the protostar and τ is the optical depth (further details found in Vorobyov and Basu (2010); Vorobyov et al. (2017)). We use accretion data from 7 of the 35 models (labelled 26 to 32) calculated by Vorobyov et al. (2017). Our extensive analysis will only pertain to model 29, but we present the optimization outputs of all remaining models in the appendix. Furthermore, the respective mass accretion dynamics are given in Figure 2 (all model dynamics are demonstrated in Figure 7).

3 RESULTS

3.1 HYPERPARAMETER SEARCH

In order to optimally tune the model's hyperparameters, we deploy an exhaustive grid search on a prespecified subset of the hyperparameter space given in table 3 (appendix) where we additionally survey the random matrices $\mathbf{W}_{\text{i}}$ and $\mathbf{W}_{\text{r}}$ to be sampled either uniformly, normally or lognormally. The estimated optimal hyperparameters are given in the appendix noting that the demonstrated MSE is given in regards to the validation error.

3.2 OPTIMIZATION

In order to demonstrate the OWA-ESN, we place the network into a free running state. The selection of training, validation and testing segments aimed to capture as much of the chaotic portions of the model as possible without incorporating too much of the flat lining dynamics exhibited at later periods. Given that certain models exhibit longer episodic periods than others, the number of data points in the training and validation sets vary from one model to the next. We present our outputs in terms of each model's respective maximum Lyapunov exponent (given in the appendix). We adopt the methodology discussed in 2.2 where we solve the constrained optimization problem using the interior-point algorithm. We assume that the data is separable of the form:

$$\dot{M}(t) = \dot{M}_d(t) + \dot{\mathcal{M}}(t) \tag{14}$$

where $\dot{M}_d(t)$ and $\dot{\mathcal{M}}(t)$ are representative of the model's deterministic and fluctuating components. In order to extract the fluctuating component, we adopt an elliptic high pass filter having a passband frequency of 2 kHz and a sampling rate of 50 kHz. Given that the model's deterministic component is non-stationary, we induce stationarity through first order differencing. We additionally standardize the time series to each model's respective standard deviation. Model 29 had a standard deviation of $\sigma = 1.3948\text{e}{-6}$ and a chaotic regime containing roughly 10,050 data points; to which 5,000 were used for training, 5,000 for validating and 50 to optimize in the free running segment. This free run segment translates to roughly 1,153 years or 9.2 Lyapunov times. Figure 3 demonstrates the model's fluctuating component (top panel) and deterministic component (bottom panel). The grey portion of the time series is representative of the training segment for $t_0 < t \leq t_{\text{T}}$, the red portion is representative of the validation segment for $t_{\text{T}} < t \leq t_{\text{V}}$ and finally the blue portion is taken as the free running segment for $t_{\text{V}} < t \leq T$. The fluctuating and deterministic components

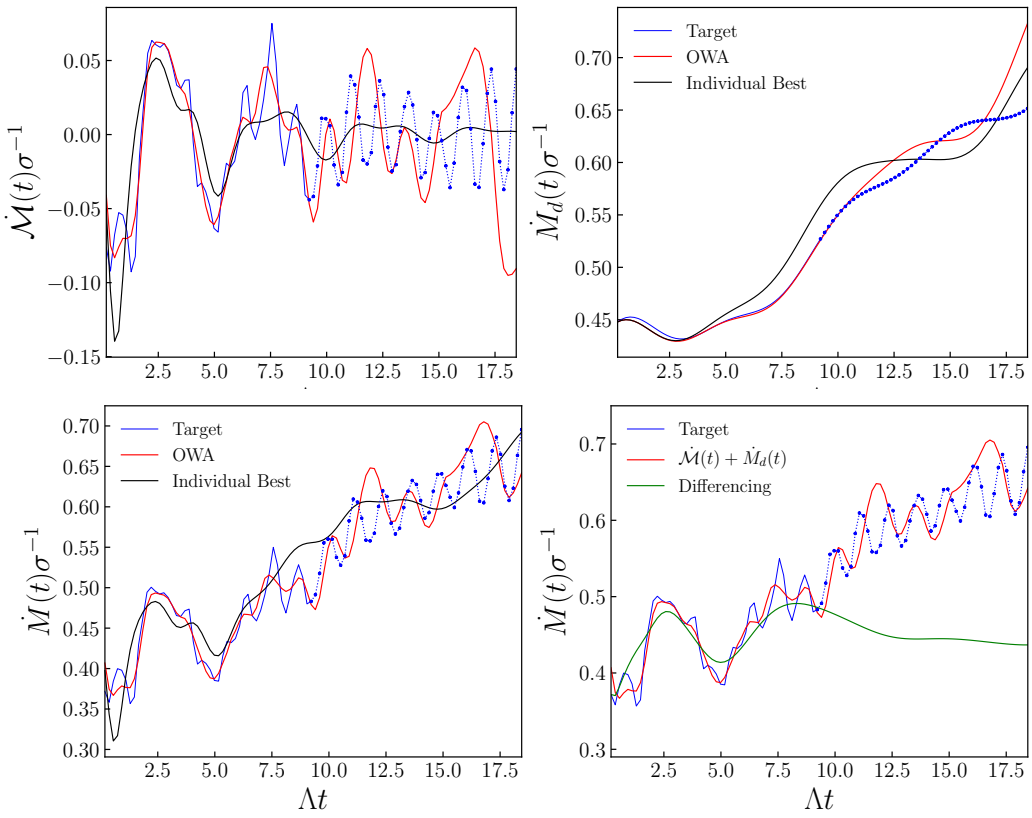

Figure 4: The OWA-ESN optimized output signal along with the best individual output. The solid blue line is representative of the free running segment of data used in the optimization, whereas the dotted dashed line is the remaining segment $t > T$ that is out of sample. OWA-ESN outputs for the fluctuating and deterministic components **(top left & right)** respectively and the final OWA-ESN output **(bottom left panel)** is found by summing the components $\dot{\mathcal{M}}(t) \cdot \sigma^{-1}$ with $\dot{M}_d(t) \cdot \sigma^{-1}$. Furthermore, in the bottom right panel, we compare the OWA-ESN output when the data is decomposed into $\dot{M}_d(t)$ and $\dot{\mathcal{M}}(t)$ (red line) with the OWA output when we difference the data directly (green line).

Table 2: OWA-ESN network hyperparameters

Component	N_r	ρ_s	ϱ	α	$\tilde{\mathrm{C}}$	λ	**Readout**	**Sampling**
Fluctuating	250	1.0	1.0	0.2	0.01	-	Pseudo	Normal
Deterministic	250	1.0	0.1	0.2	0.01	-	Pseudo	Normal

are trained and validated on two network's separately. Each network is then free run, optimized and summed together to produce the final output for $\dot{M}(t)$. We adopt the optimal hyperparameters from table 5 given in the appendix with $\omega = 100$ and implement a series of $L = 100$ reservoirs. The respective optimized free running signals for $\dot{\mathcal{M}}(t)$ and $\dot{M}_d(t)$ adopting the optimal hyperparameters are demonstrated in the left most plots for panels (a) and (b) in Figure 11 respectively. We find that in certain cases, inducing a highly non-linear response in the network's reservoir can produce better performing outputs for the optimization in the free running segment. This in turn generates more variability in each realization of the reservoir series and can sometimes allow the OWA-ESN to perform a more efficient optimization. This is done by increasing the respective input scaling to $\varrho = 1$ and the outputs are given in the right most plots of panels (a) and (b) in Figure 11. We do however caution this approach and find that it can sometimes lead to stability issues for long range outputs. Since high input scalings result in larger variability in the realizations, they can sometimes diverge from one another rapidly. We sum the individual outputs from $\dot{M}_d(t)$ and $\dot{\mathcal{M}}(t)$ to produce the final signal for $\dot{M}(t)$. The bottom panel of Figure 11 demonstrates the standardized final optimization output and we summarize the respective mean squared errors for all outputs in table 1.

3.3 Forecasting

Once the network has been optimized and the weights $\hat{\beta}_j$ have been computed, the OWA-ESN can be used to run the signal beyond the free running segment. This is done by taking the weighted sum of all realization for $t > T$ using the weights $\hat{\beta}_j$ computed over the free running segment. We run the out of sample forecast for an additional 9.2 Lyapunov times. The output is given in Figure 4 where the solid blue line is representive of the free running segment used for the optimization $t_{\mathrm{V}} < t \leq T$ and the dashed dotted line is for $t > T$. For comparison, we include the network's top performing individual signal (black line). That is, of the 100 individual paths given by the reservoir series, the top performing output is taken as the one with the lowest MSE in the free run segment. The hyperparameters used are given in table 2. In Figure 4, the MSE of the OWA signal is 0.00085 and that of the best individual signal is 0.00114. We can gauge longer term performance by running the forecast up to a total of 27.6 Lyapunov times. We demonstrate the time evolving MSE with respect to three changes in the input scaling of the fluctuating component for $\varrho = [0.1, 0.3, 1.0]$ which is representative of low, moderate and high $\varrho-$values. The respective MSE at $\Lambda t = 27.6$ are 0.0068, 0.0074 and 0.0081 respectively for $\varrho = [0.1, 0.3, 1.0]$. This is shown in Figure 5. In generating these forecasts, the random reservoir initialization is seeded for reproducibility with an initial seed value of $s_0 = 503489$ and iterated throughout the series in accordance to the LCG algorithm:

$$s_{n+1} = (as_n + b) \bmod c, \tag{15}$$

for $a = 2^{16} + 3$, $b = 0$ and $c = 2^{31}$. Here, the recurrence is run for a total $n = 1, 2, \cdots, L$ where the same sequence is used for the deterministic and fluctuating components. We note that any recurrence can be used to seed the OWA-ESN so long as each reservoir is seeded with a different value. We note however, that selecting a seed should be done so prior to any hyperparameter tuning.

4 Discussion & Conclusion

Our methodology is shown to quantitatively output high performing signals and demonstrates how the OWA-ESN can resolve intrinsic underlying temporal features of a highly chaotic time series system in a model-free fashion. We find that separating out the fluctuations in the data from the overall trend and summing the individual component outputs greatly improves the network's overall performance compared to differencing the time series directly. In the bottom right panel of Figure

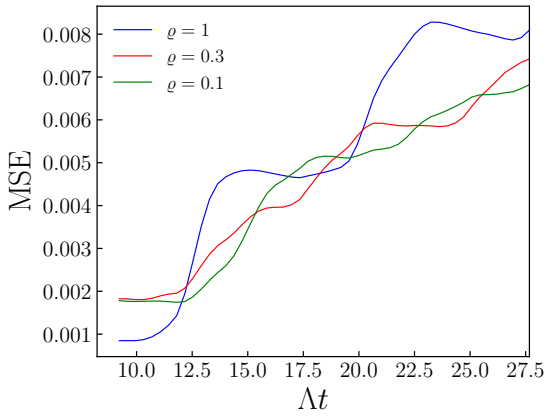

Figure 5: The time evolving MSE for the forecast run outside the free run segment implementing the OWA-ESN under three scenarios. Each scenario varies the fluctuating component's input scaling.

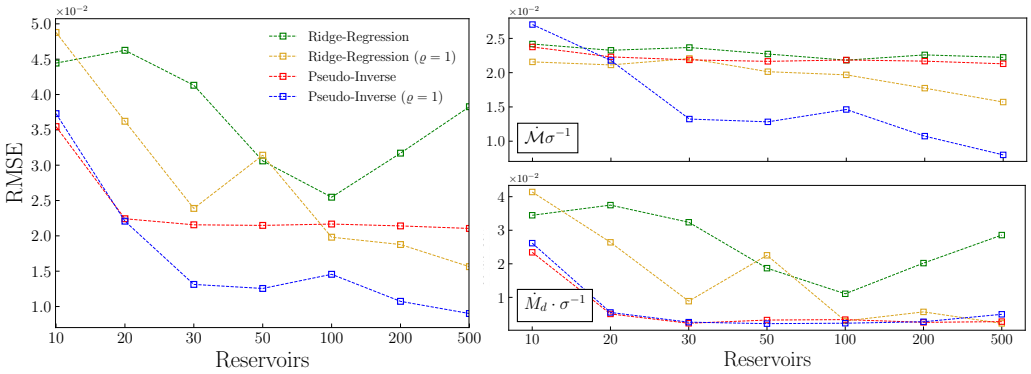

Figure 6: The RMSE demonstrated as a function of the number of reservoirs in the series over the free running segment used in the optimization. The right panels demonstrate the same respective RMSE for the individual fluctuating **(top)** and deterministic **(bottom)** components.

4 we compare the outputs of the OWA-ESN when the data has been separated into fluctuating and deterministic components with the outputs achieved from just performing first order differencing. We achieve an MSE of 0.0138 for the output using a first order difference compared to an MSE of 0.00085 when the decomposition is applied. Although we used an elliptic filter for the data decomposition, we achieve comparable results implementing a butterworth filter. Furthermore, our implementation utilizes a series of 100 reservoirs, but we do see somewhat of an improvement in model performance with increasing L. Figure 6 demonstrates the RMSE as a function of the number of reservoirs. Upon visual inspection, we see significant improvement up to and including $L = 30$ reservoirs with a potential plateau beyond this point. Whether there is truly a plateau or not raises the question on scaling and whether there is additional improvement at any point beyond $L > 500$ reservoirs. For large enough L computational efficiency can become significantly compromised, but given that training and validation at each reservoir are independent, it becomes natural to deploy the OWA-ESN in parallel. The OWA-ESN aimed to exploit some of the statistical challenges of the shallow ESN. The random generation of the input and reservoir matrices in shallow ESNs can introduce stochastic nature to the outputs which can be cumbersome in practice. The weighted average approach aims to suppress some of the lower performing outputs while emphasizing the better performing ones within a given set of constraints. Over the free running segment, the optimized signal is indeed found to significantly mitigate the randomness in outputs. However, outside of the free run segment, the sensitivity to the network's hyperparameters becomes more important. Particularly, models with high input scaling parameter introduce higher variability in the output. These models should be seeded for reproducibility and should be done so prior to any hyperparameter tuning. This will ensure that any given network is optimized relative to it's given seed. However, we find this became less important when the input scaling is low. Overall, we implemented the OWA-ESN which

had demonstrated forecasts maintaining relatively high performance and low MSE. In future work, we can aim to reduce the amount of data to be partitioned by performing the optimization either during training or during validation. This alternative can potentially achieve similar performance, but completely removes the need for a free running segment.

REFERENCES

Csáji, B. C. (2001). Approximation with artificial neural networks. *Faculty of Sciences, Etvs Lornd University, Hungary*, 24:7.

Jaeger, H. and Haas, H. (2004). *Science*, pages 78–80.

Liu, X., Chen, M., Yin, C., and Saad, W. (2018). *Analysis of memory capacity for deep echo state networks*.

Lukoševičius, M. and Jaeger, H. (2009). *Computer Science Review*, pages 127–149.

Pathak, J., Lu, Z., Hunt, B. R., Girvan, M., and Ott, E. (2017). *Chaos: An Interdisciplinary Journal of Nonlinear Science*, page 121102.

Vorobyov, E. and Basu, S. (2006). *The Astrophysical Journal*, page 956.

Vorobyov, E. I. and Basu, S. (2010). *The Astrophysical Journal*, page 1896.

Vorobyov, E. I., Elbakyan, V., Hosokawa, T., Sakurai, Y., Guedel, M., and Yorke, H. (2017). *Astronomy & Astrophysics*, page A77.

Wu, Q., Fokoue, E., and Kudithipudi, D. (2018). *arXiv preprint arXiv:1802.07369*.

A APPENDIX

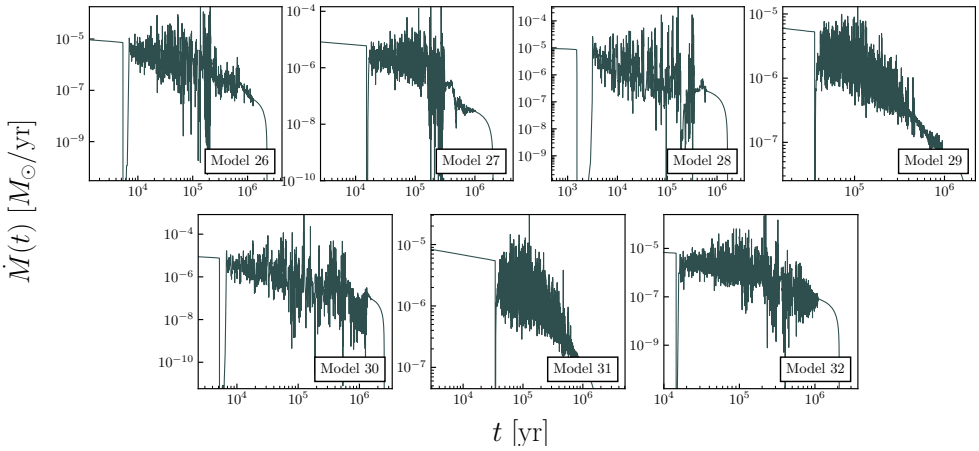

Figure 7: Mass accretion rate from Vorobyov et al. (2017) as a function of time for models 26-32.

Table 3: Hyperparameter search space.

Parameters	**Search Space**
Reservoir Size, N_r	$(250, 1000)$ incremented by 250
Spectral Radius, ρ_s	$(0.1, 1)$ incremented by 0.1
Input Scaling, ϱ	$(0.1, 1)$ incremented by 0.1
Leaking Rate, α	$(0.1, 1)$ incremented by 0.1
Connectivity, $\tilde{\mathrm{C}}$	$[0.01, 0.05, 0.1]$
Regularization, λ	$[1, 10, 100, 1000] \times 10^{-8}$
Random Distribution	Uniform, Normal, Lognormal

Table 4: $\Lambda_{\mathbf{max}}$ for each model.

Model Number	$\mathbf{\Lambda_{max}}$
26	0.20821945
27	0.2014386
28	0.20973413
29	0.18445392
30	0.16244096
31	0.19877397
32	0.1922906

Table 5: Hyperparameter search output

Model	N_r	ρ_s	ϱ	α	$\tilde{\mathrm{C}}$	λ	**Readout**	**Sampling**	**MSE**
Model 26	250	1.0	0.5	1.0	0.05	-	Pseudo	Uniform	0.0078
	250	0.9	0.1	0.1	0.01	10^{-6}	Ridge	Uniform	0.0028
Model 27	250	1.0	1.0	1.0	0.05	-	Pseudo	Uniform	0.0111
	250	0.7	0.1	0.2	0.01	10^{-5}	Ridge	Uniform	0.0019
Model 28	250	1.0	1.0	1.0	0.05	-	Pseudo	Uniform	0.1693
	250	0.2	0.1	0.1	0.05	10^{-5}	Ridge	Uniform	0.1527
Model 29	250	1.0	0.1	0.2	0.01	-	Pseudo	Normal	0.0033
	500	0.9	0.2	0.4	0.05	10^{-7}	Ridge	Uniform	0.0028
Model 30	250	1.0	1.0	1.0	0.05	-	Pseudo	Uniform	0.0046
	250	1.0	0.1	0.3	0.1	10^{-5}	Ridge	Uniform	2.52×10^{-4}
Model 31	250	0.8	0.1	0.4	0.01	-	Pseudo	Uniform	0.0072
	500	0.9	0.1	0.3	0.05	10^{-7}	Ridge	Uniform	0.0068
Model 32	500	0.1	0.1	0.2	0.01	-	Pseudo	Uniform	3.32×10^{-4}
	250	1.0	0.1	0.4	0.1	10^{-7}	Ridge	Uniform	2.67×10^{-4}

Figures 8 to 14 demonstrate the OWA-ESN output over the free running segment for all models demonstrated in Figure 7. We note that there are differences in the training, validation and testing segments since each model exhibits different chaotic periods that occupy different lengths of time. However, we still see some strong optimizations for each model. Finally, each model utilized 50 data points for the free run segment adopting the hyperparameters from table 5 still experimenting with high input scaling.

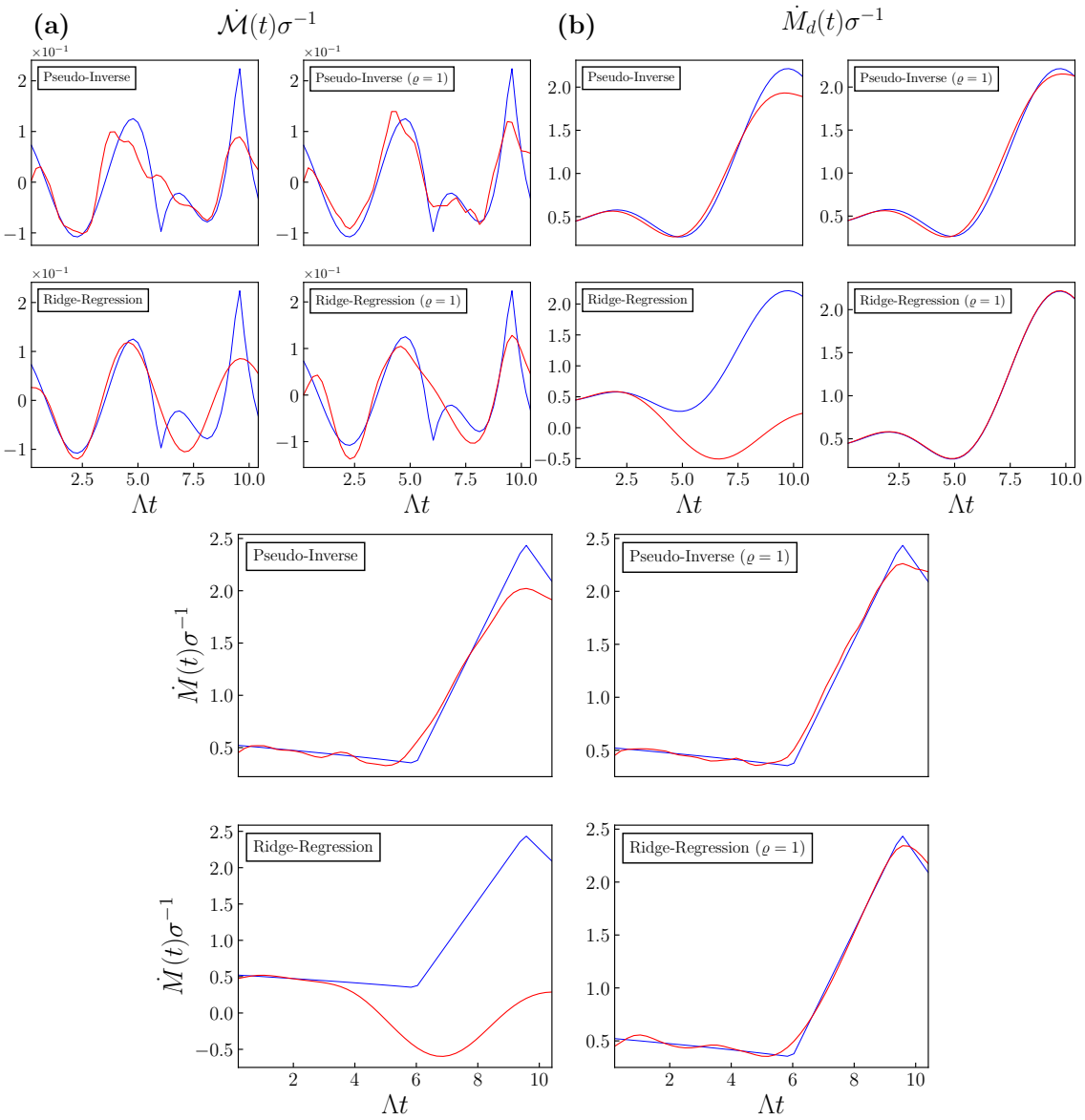

Figure 8: The optimized free running signal (red) for model 26 overlaid with the free running segments (blue) of the **(a)** standardized fluctuating, **(b)** standardized deterministic and **(bottom)** final component $\dot{M}(t)$ given in terms of $\Lambda \equiv \Lambda_{\text{max}} = 0.20821945$. Model 26 had a standard deviation of $\sigma = 2.0231\text{e}{-5}$ and utilized 6450 data points where 3200 were used for training and 3200 for validating.

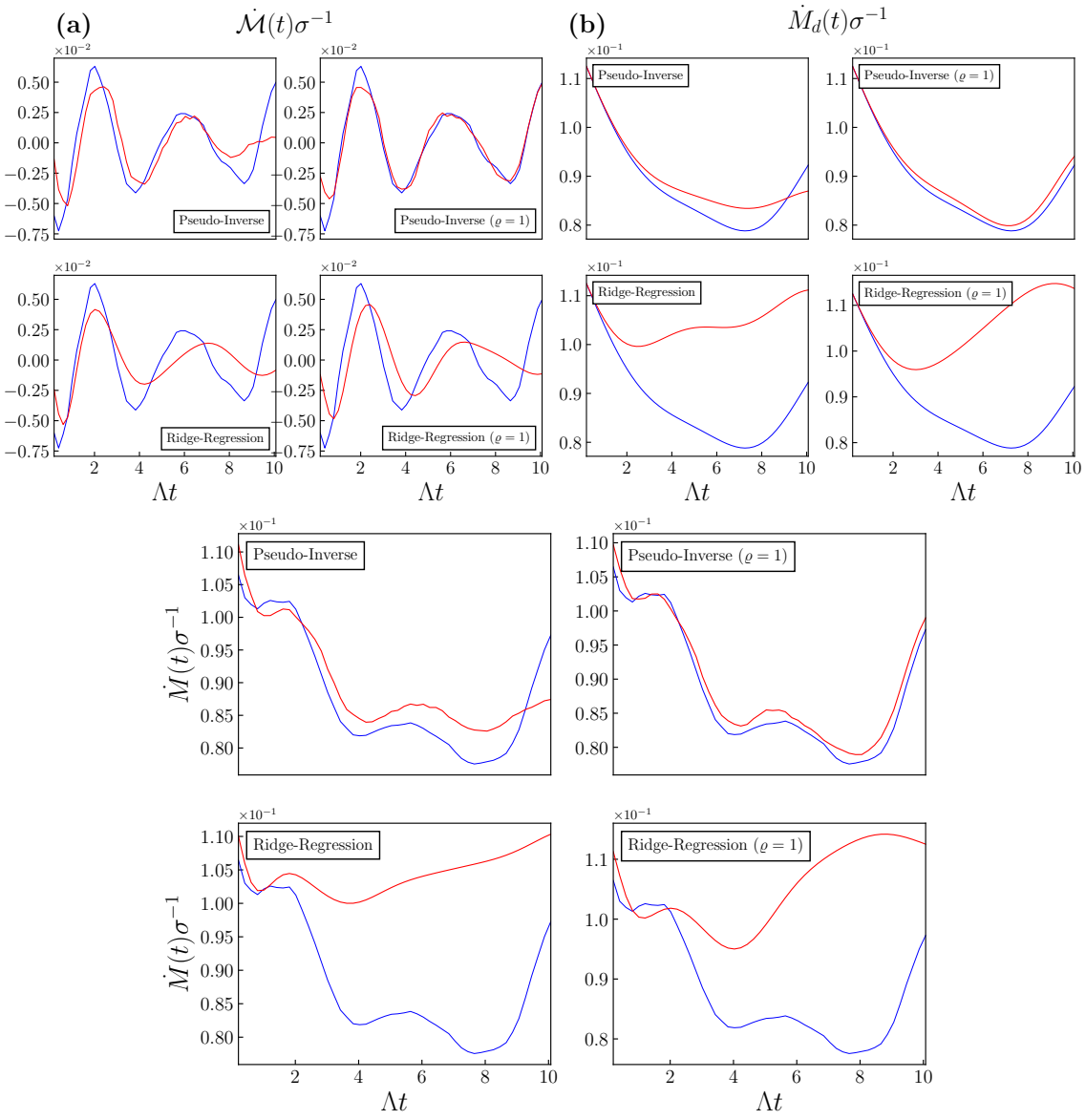

Figure 9: The optimized free running signal (red) for model 27 overlaid with the free running segments (blue) of the **(a)** standardized fluctuating, **(b)** standardized deterministic and **(bottom)** final component $\dot{M}(t)$ given in terms of $\Lambda \equiv \Lambda_{\text{max}} = 0.2014386$. Model 27 had a standard deviation of $\sigma = 1.4145\text{e}{-5}$ and utilized 10050 data points where 5000 were used for training and 5000 for validating.

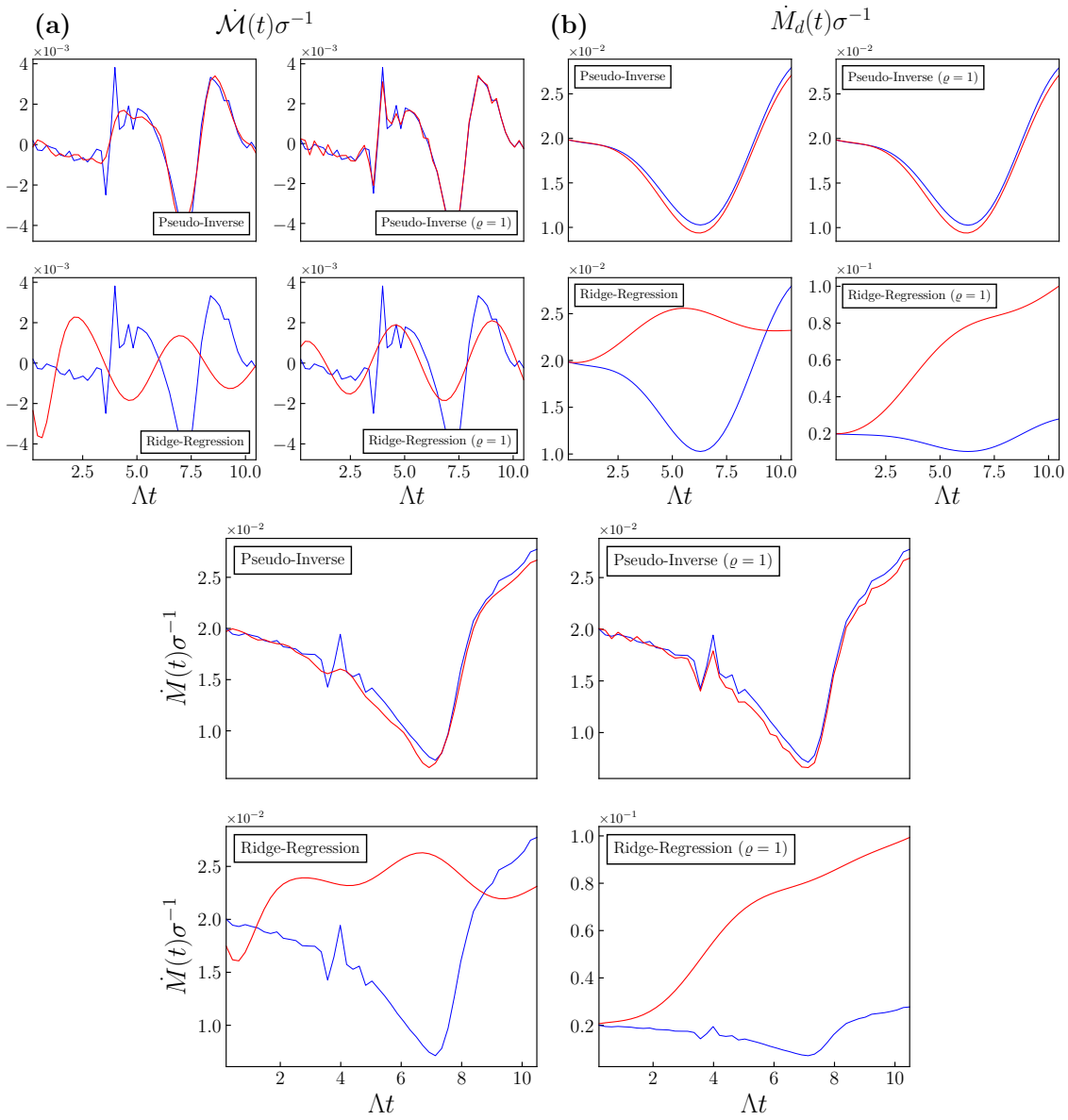

Figure 10: The optimized free running signal (red) for model 28 overlaid with the free running segments (blue) of the **(a)** standardized fluctuating, **(b)** standardized deterministic and **(bottom)** final component $\dot{M}(t)$ given in terms of $\Lambda \equiv \Lambda_{\text{max}} = 0.20973413$. Model 28 had a standard deviation of $\sigma = 1.3449\text{e}{-5}$ and utilized 4050 data points where 2000 were used for training and 2000 for validating.

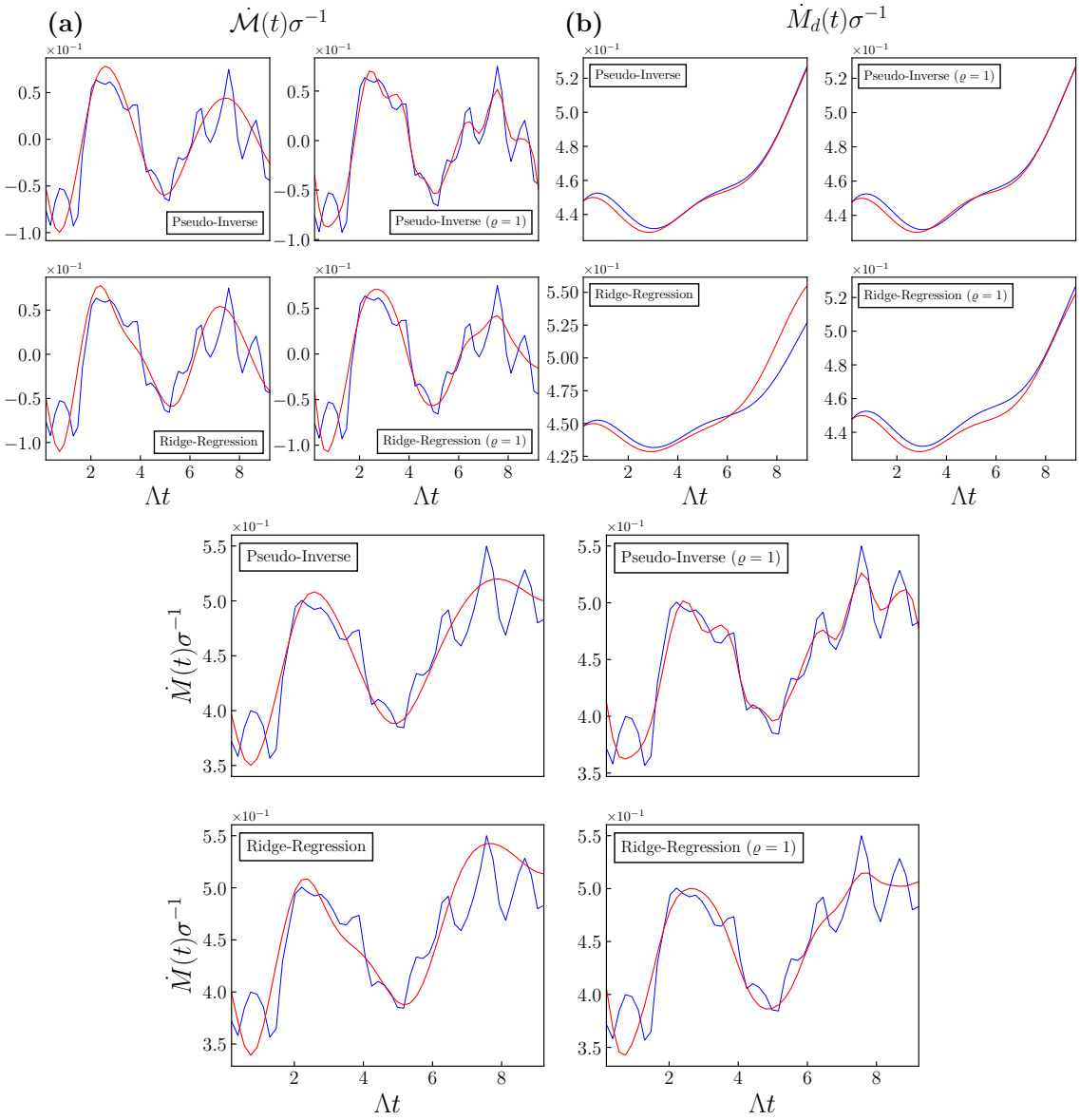

Figure 11: The optimized free running signal (red) for model 29 overlaid with the free running segments (blue) of the **(a)** standardized fluctuating, **(b)** standardized deterministic and **(bottom)** final component $\dot{M}(t)$ given in terms of $\Lambda \equiv \Lambda_{\text{max}} = 0.184453923$. Model 29 had a standard deviation of $\sigma = 1.3948\text{e}{-6}$ and utilized 10050 data points where 5000 were used for training and 5000 for validating.

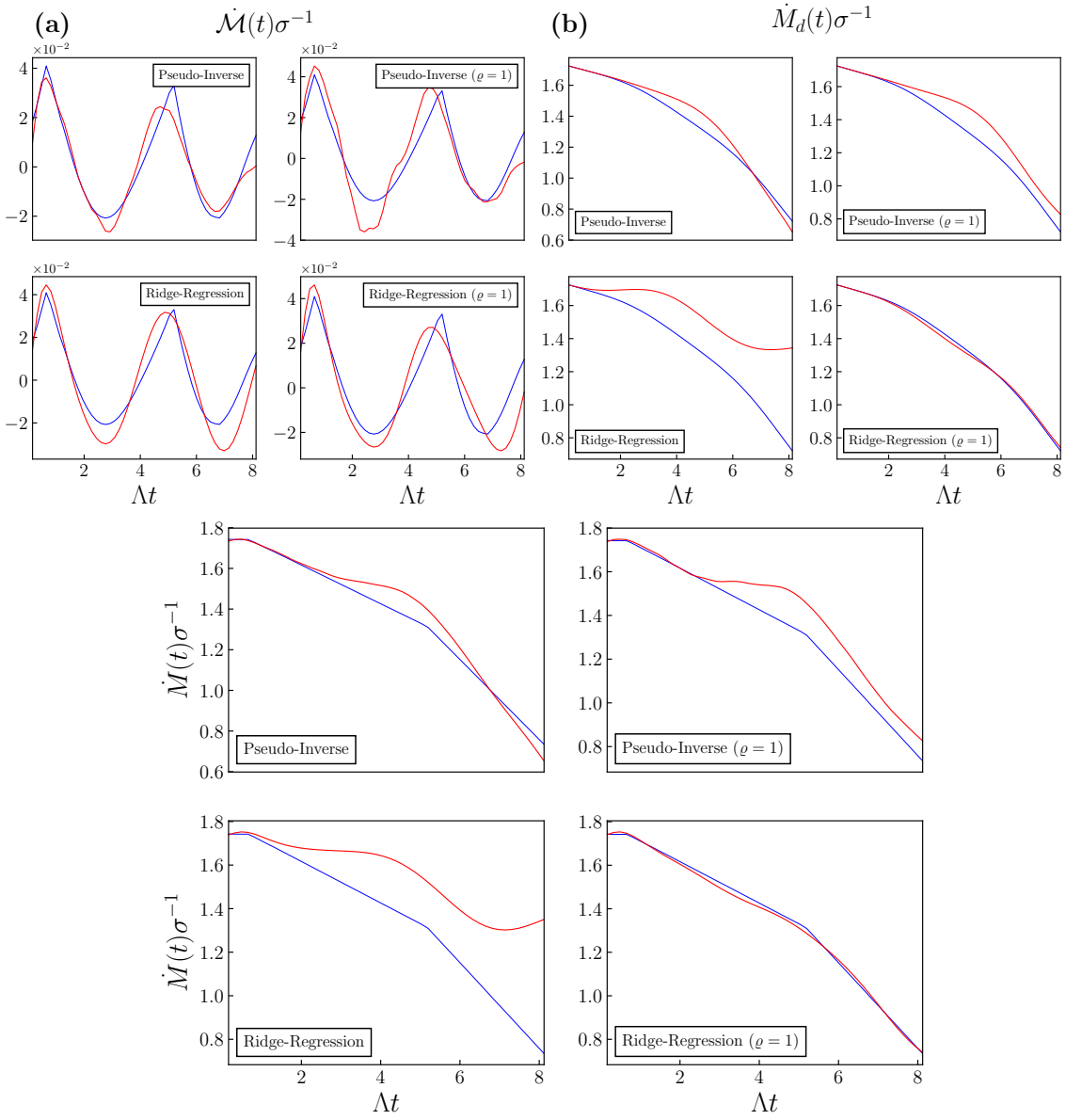

Figure 12: The optimized free running running signal (red) for model 30 overlaid with the testing segments (blue) of the **(a)** standardized fluctuating, **(b)** standardized deterministic and **(bottom)** final component $\dot{M}(t)$ given in terms of $\Lambda \equiv \Lambda_{\text{max}} = 0.16244096$. Model 30 had a standard deviation of $\sigma = 1.0350\text{e}{-4}$ and utilized 7850 data points where 3900 were used for training and 3900 for validating.

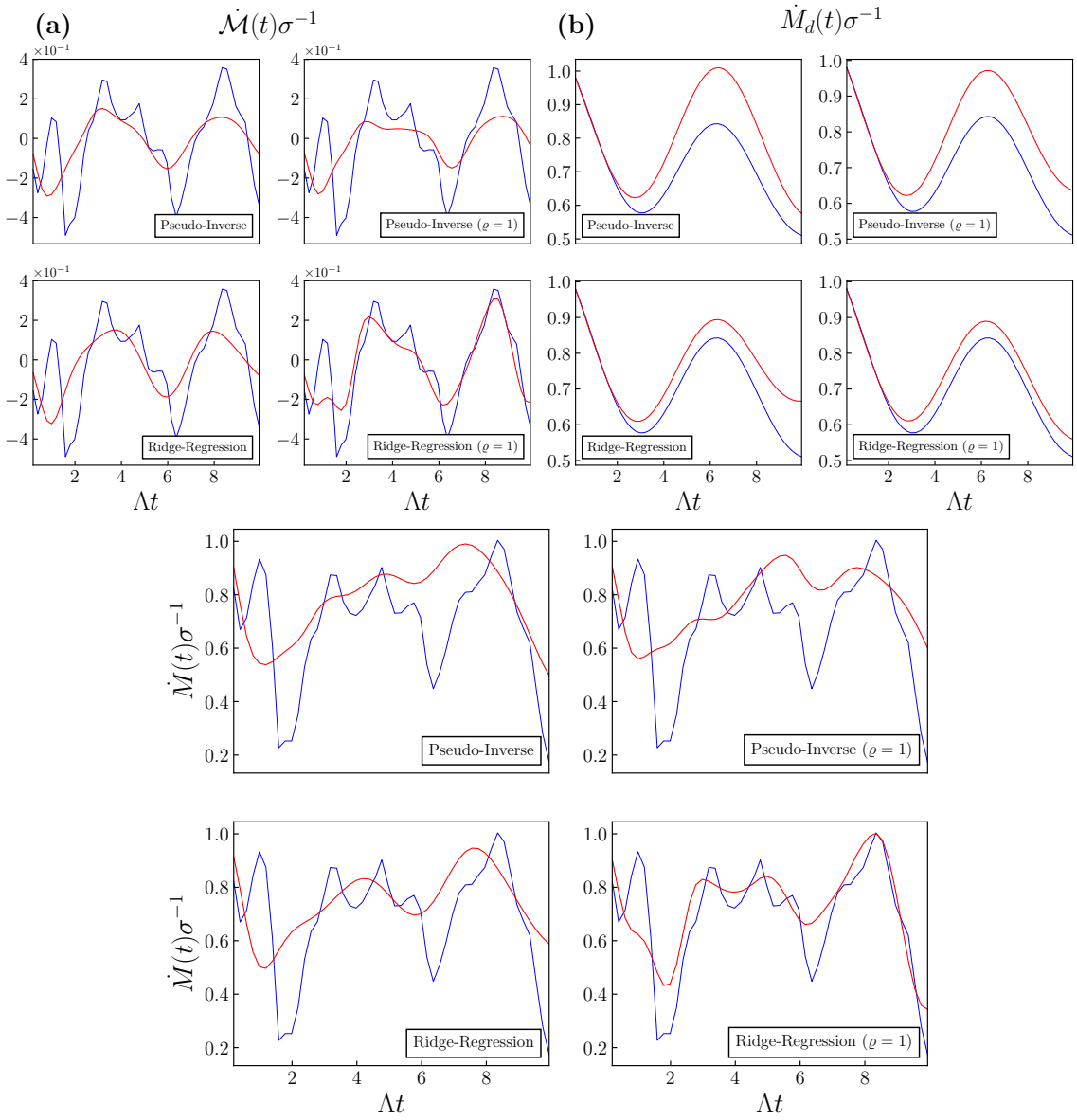

Figure 13: The optimized free running running signal (red) for model 31 overlaid with the testing segments (blue) of the **(a)** standardized fluctuating, **(b)** standardized deterministic and **(bottom)** final component $\dot{M}(t)$ given in terms of $\Lambda \equiv \Lambda_{\text{max}} = 0.19877397$. Model 31 had a standard deviation of $\sigma = 1.9617\text{e}{-6}$ and utilized 9450 data points where 4700 were used for training and 4700 for validating.

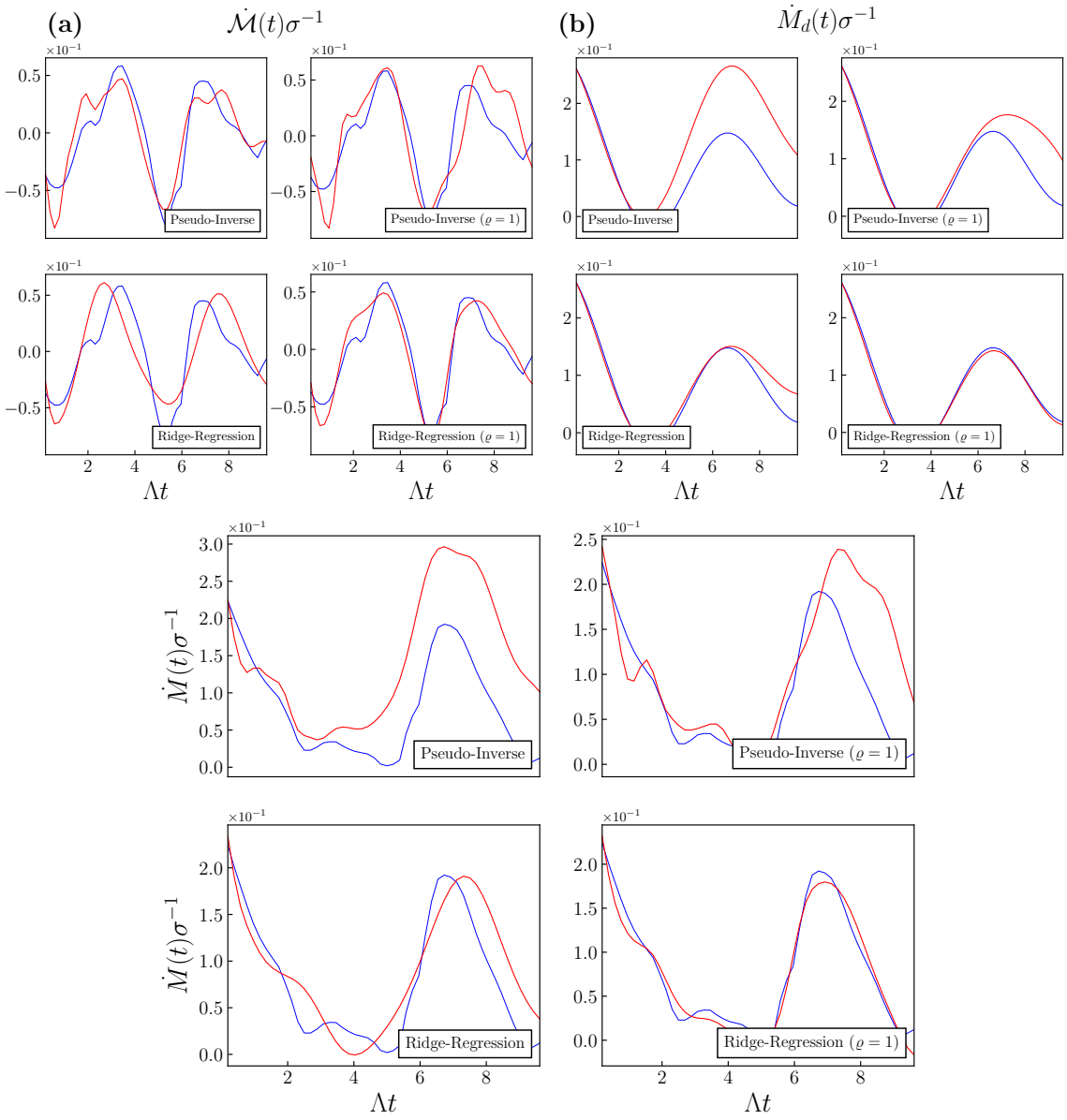

Figure 14: The optimized free running running signal (red) for model 32 overlaid with the testing segments (blue) of the **(a)** standardized fluctuating, **(b)** standardized deterministic and **(bottom)** final component $\dot{M}(t)$ given in terms of $\Lambda \equiv \Lambda_{\text{max}} = 0.1922906$. Model 32 had a standard deviation of $\sigma = 2.9740\text{e}{-5}$ and utilized 4950 data points where 2450 were used for training and 2450 for validating.