SUPPLEMENTARY INFORMATION

Adaptive Nonlinear Vector Autoregression: Robust Forecasting for Noisy Chaotic Time Series

Azimov Sherkhon, Susana López-Moreno, Eric Dolores-Cuenca, Sieun Lee, Sangil Kim

This document contains the following supplementary information:

Supplementary figures

Grid search for the standard NVAR with observation frequency or skip $s = 1$ (95/5 split)

In this section we provide the RMSE heatmaps showing the results of grid search for the standard NVAR when forecasting the chaotic Lorenz 63 time series with 0%, 5%, 10% and 15% noise. This process is crucial for the optimal performance of the standard NVAR method, since it is very sensitive to the following two parameters: the delay k and the ridge parameter γ. We use a two-phase grid search to find the optimal values for these two parameters in order to guarantee a fair comparison between the standard NVAR and the adaptive NVAR model.

These experiments contain 1600 training data points, 200 warm-up data points and 100 testing data points. We found that a 95/5 split gave the best RMSE results in all cases. This split separates the training data into two subsets: 5 percent (80 points) is used for validation, and the remaining 95 percent (1520 points) is used for model training. Next, we look for the combination of k and γ that produces the lowest validation RMSE. After determining the ideal hyperparameters, we retrain the NVAR model with the chosen k and γ on the entire 1600-point training set, and then assess its performance on the 100-point test set. We remark that the white cells on the heatmaps highlight the hyperparameter configurations where the learned readout matrix W_{out} produced unstable predictions that resulted in numerical overflow.

Each case took approximately 3 hours on an Intel Xeon CPU with 2 virtual CPUs (vCPUs).

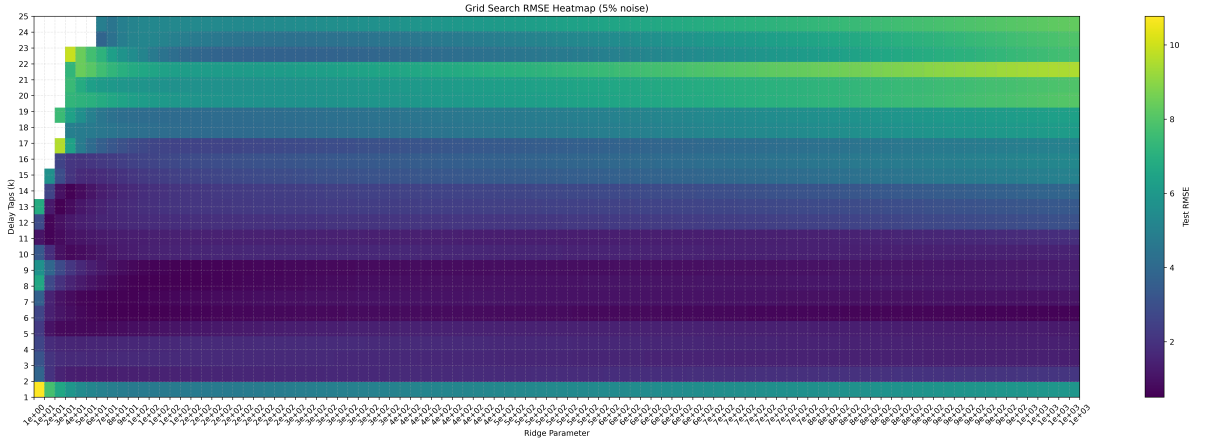

Figure 1: **Grid Search RMSE Heatmap 5% noise level (delay k vs ridge parameter γ).** During the initial stage of the grid search, we used 25 values to test delay tap values in the range $k = 1$ to 25 and ridge parameters γ from 1 to 10^3 using 100 values. The best performing configuration that was determined by this coarse grid search was $k = 8$ and $\gamma \approx 120.880$, yielding a validation RMSE of 0.503792.

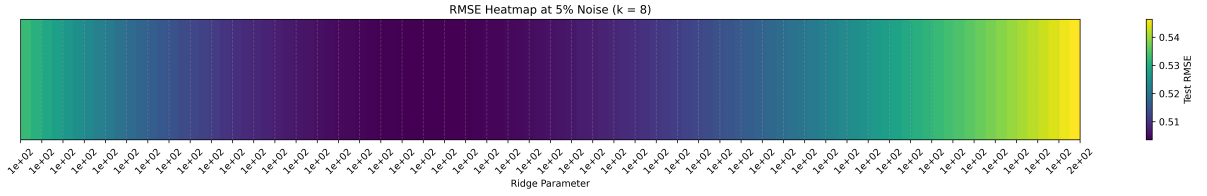

Figure 2: **RMSE Heatmap 5% noise level at $k = 8$.** We conducted a second phase of the search to fine-tune it, limiting the delay tap to $k = 8$ and sweeping γ from 100 to 200. As a result, RMSE improved marginally to 0.503605 with $\gamma \approx 119$.

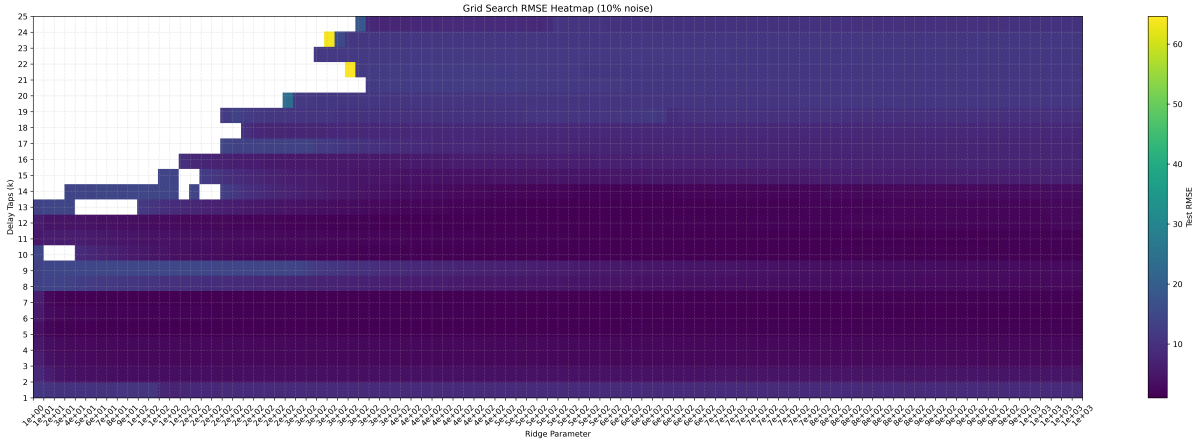

Figure 3: **Grid Search RMSE Heatmap 10% noise level (delay k vs ridge parameter γ).** During the initial stage of the grid search for the 10% noise case, we tested values in the range $k = 1$ to 25 and ridge parameters γ from 1 to 10^3. The best performing configuration determined by this coarse grid search was $k = 7$ and $\gamma \approx 160.840$, yielding a validation RMSE of 1.022701.

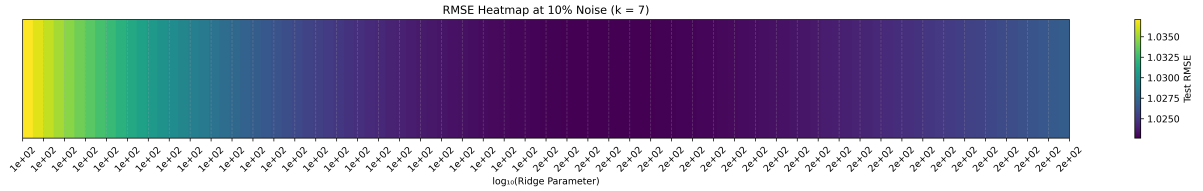

Figure 4: **RMSE Heatmap 10% noise level at $k = 7$.** We conducted a second phase of the search to fine-tune the coarse grid search, limiting the delay to $k = 7$ and more precisely sweeping γ across 100 values from 100 to 200. As a result, RMSE improved marginally to 1.022640 with $\gamma \approx 156$.

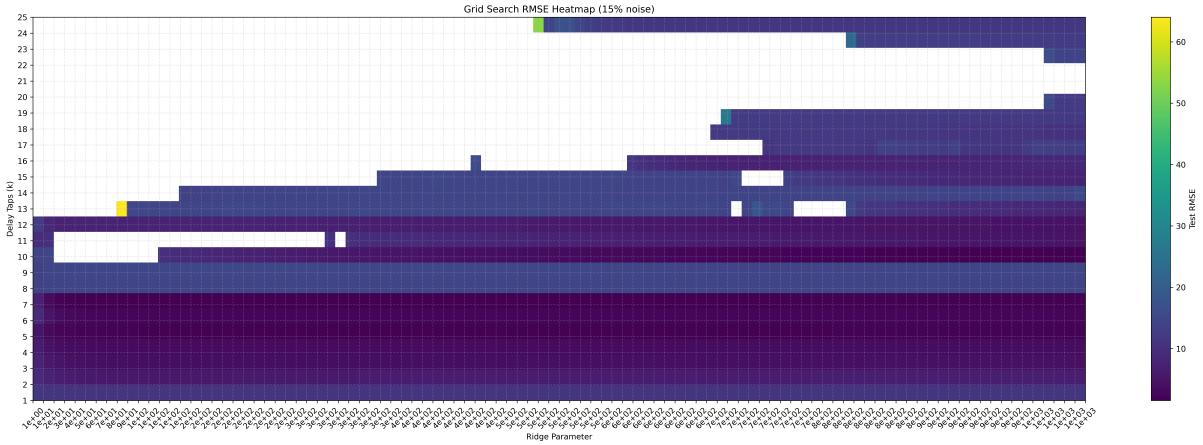

Figure 5: **Grid Search RMSE Heatmap 15% noise level (delay k vs ridge parameter γ).** For the first phase of the grid search for the 15% noise, case we tested values in the range $k = 1$ to 25 with the ridge parameters γ going from 1 to 10^3. The best configuration found was $k = 7$ and $\gamma = 120.880$, yielding a validation RMSE of 1.552449.

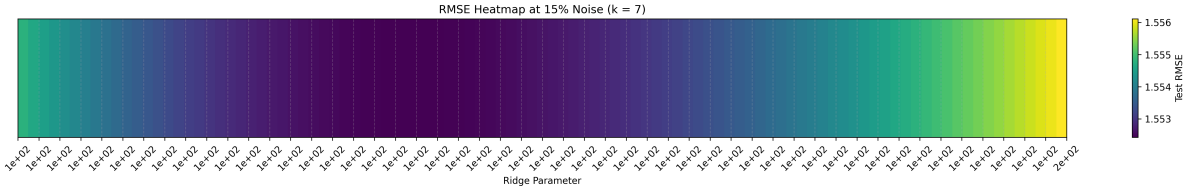

Figure 6: **RMSE Heatmap 15% noise level at $k = 7$.** The best configuration of parameters found on the second phase of grid search were $k = 7$ and ridge parameter $\gamma = 118.500$.

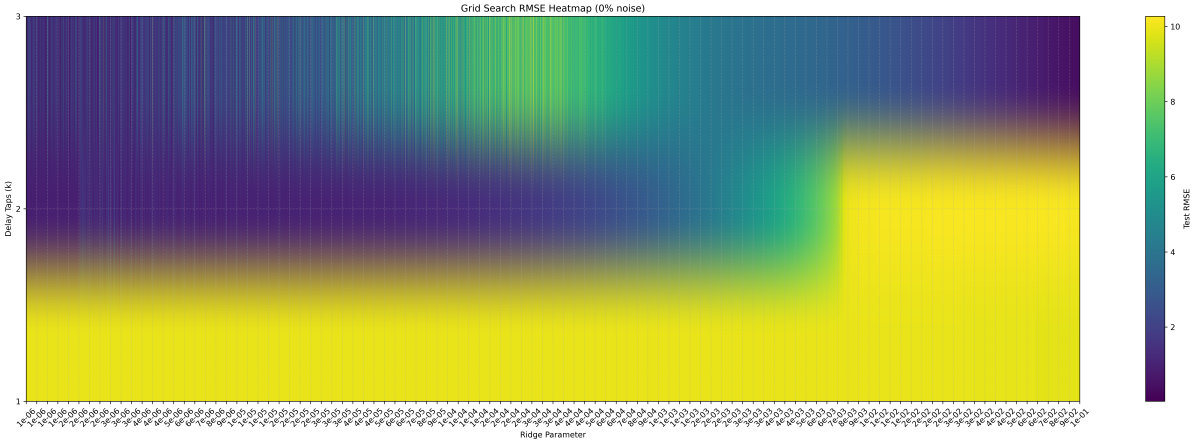

Figure 7: **RMSE Heatmap for noise-free case at $k = 2$.** For the noise-free case, $k = 2$ and ridge $\gamma = 2 \times 10^{-6}$ produced the best result with a validation RMSE of 0.025845.

Grid search for the standard NVAR for different observation frequencies or skip values s (95/5 split)

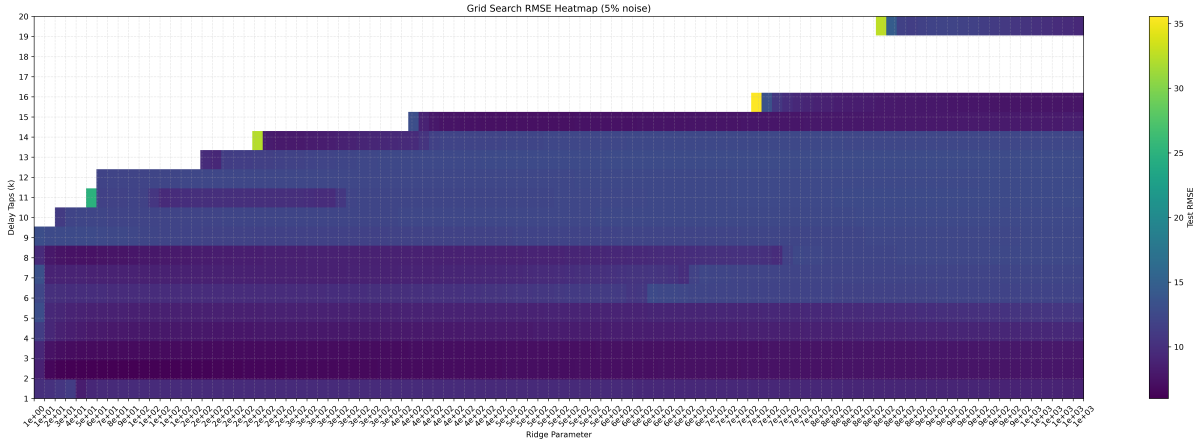

Figure 8: **Grid Search RMSE Heatmap 10% noise level for observation frequency $s = 2$ (delay k vs ridge parameter γ).** For the first phase of the grid search for the 10% noise case, we tested values in the range $k = 1$ to 20 with the ridge parameters γ going from 1 to 10^3. The best configuration found was $k = 2$ and $\gamma = 50.950$, yielding a validation RMSE of 6.002254.

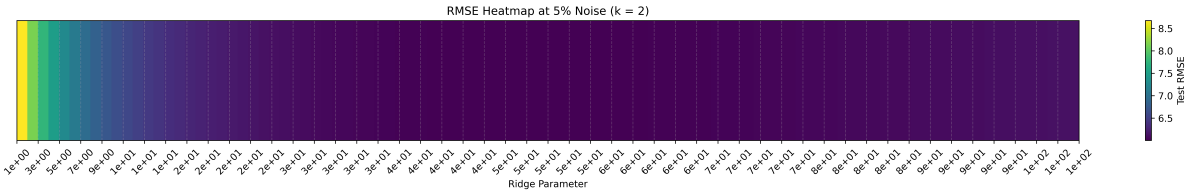

Figure 9: **RMSE Heatmap 10% noise level at $k = 2$ for observation frequency $s = 2$.** The best configuration of parameters found on the second phase of grid search were $k = 2$ and ridge parameter $\gamma = 48.520$.

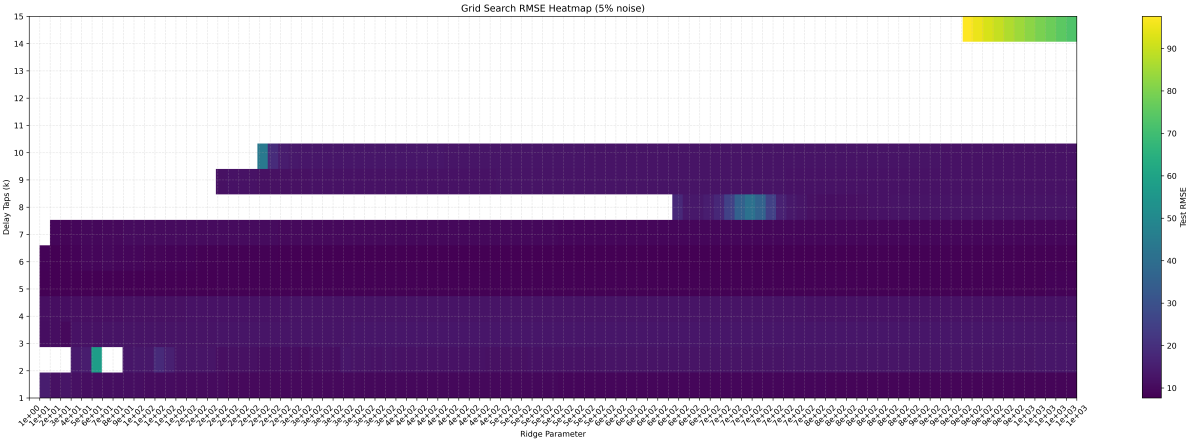

Figure 10: **Grid Search RMSE Heatmap 10% noise level for observation frequency $s = 4$ (delay k vs ridge parameter γ).** For the first phase of the grid search for the 10% noise case, we tested values in the range $k = 1$ to 20 with the ridge parameters γ going from 1 to 10^3. The best configuration found was $k = 5$ and $\gamma = 50.950$, yielding a validation RMSE of 7.653768.

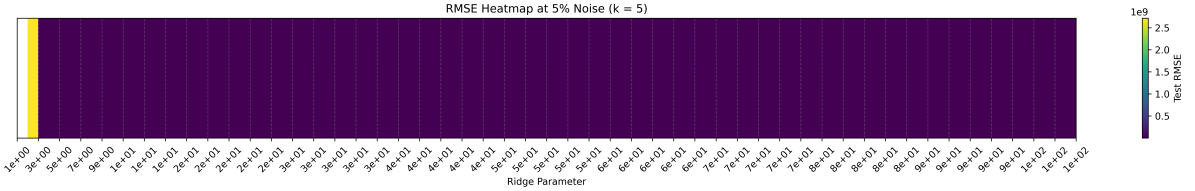

Figure 11: **RMSE Heatmap 10% noise level at $k = 5$ for observation frequency $s = 4$.** The best configuration of parameters found on the second phase of grid search were $k = 5$ and ridge parameter $\gamma = 53.470$.

Supplementary example

Example 1. We illustrate our adaptive NVAR model using a simple 3-dimensional time series with delay embedding and joint feature learning.

(i) Dimension of the state vector: $d = 3$.
(ii) Number of delays: $k = 2$.
(iii) Neural network output dimension: $m = dk(dk + 1)/2$, to match the number of fixed nonlinear features in NVAR. Alternatively, m can be treated as a tunable hyperparameter.
(iv) Input dimension: $dk + m = 6 = 21 = 27$.
(v) Number of training points: $n = 5$ ($I_{\text{train}} = \{2, 3, 4, 5, 6\}$ are valid training indices if $k = 2$).
(vi) For each $i \in I_{\text{train}}$, we use $H_{\text{lin},i} = X_i \oplus X_{i-1} \in \mathbb{R}^6$.

Let $X_i \in \mathbb{R}^d$ for $i = \{1, \ldots, 10\}$ be our toy series dataset, where $d = 3$. The following table lists the components of X_i.

i	1	2	3	4	5	6	7	8	9	10
$x_{1,i}$	0.1	0.2	0.3	0.5	0.4	0.7	0.6	0.8	0.9	0.5
$x_{2,i}$	0.3	0.1	0.6	0.2	0.5	0.1	0.4	0.6	0.5	0.2
$x_{3,i}$	0.2	0.4	0.1	0.3	0.6	0.2	0.3	0.2	0.9	0.2

Table 1: Values of $x_{j,i}$ for $j = 1, 2, 3$ and $i = 1, \ldots, 10$.

We now construct the linear feature vectors and their corresponding targets for training:

i	$H_{\text{lin},i}$	Target X_{i+1}
2	$X_2 \oplus X_1$	X_3
3	$X_3 \oplus X_2$	X_4
4	$X_4 \oplus X_3$	X_5
5	$X_5 \oplus X_4$	X_6
6	$X_6 \oplus X_5$	X_7

Table 2: Linear combination inputs $H_{\text{lin},i}$ and their target outputs X_{i+1}.

The feature vectors are then in the following vector spaces:

1. The linear feature vector $H_{\text{lin},i} \in \mathbb{R}^{dk} = \mathbb{R}^6$.
2. The neural network (nonlinear) feature vector $H_{\mathcal{NN},i} = \mathcal{NN}(H_{\text{lin},i}; \theta) \in \mathbb{R}^m = \mathbb{R}^{21}$.
3. The total feature vector, defined by concatenating the linear and nonlinear features $H_{\text{total},i} = H_{\text{lin},i} \oplus H_{\mathcal{NN},i} \in \mathbb{R}^{dk+m} = \mathbb{R}^{27}$.

The neural network $\mathcal{NN}(H_{\text{lin},i}; \theta)$ is defined as a simple MLP with one hidden layer, a tanh activation function and a dropout layer in case of noisy data. Let the hidden layer have $h = 24$ units and assume our data X_i is not noisy. The MLP transformation is given by:

$$H_{\mathcal{NN},i} = \mathcal{NN}(H_{\text{lin},i}; \theta) = W \cdot \tanh(W_{\text{in}} H_{\text{lin},i}),$$

where $W_{\text{in}} \in \mathbb{R}^{h \times (dk)} = \mathbb{R}^{24 \times 6}$ and $W \in \mathbb{R}^{m \times h} = \mathbb{R}^{21 \times 24}$.

After concatenating $H_{\text{lin},i}$ and $H_{\mathcal{NN},i}$ into $H_{\text{total},i}$, we perform the readout layer as follows:

$$\hat{Y} = W_{\text{out}} H_{\text{total},i},$$

where $\hat{Y}_{i+1} = \hat{X}_{i+1} - \hat{X}_i$. Thus, the parameter set θ is $\theta = \{W_{\text{in}}, W, W_{\text{out}}\}$.

The mean squared error loss

$$L(\theta, W_{\text{out}}) = \frac{1}{5} \sum_{i=2}^{5} \|W_{\text{out}} H_{\text{total},i} - Y_{i+1}\|_2^2 = \frac{1}{5} \sum_{i=2}^{5} \|W_{\text{out}} [H_{\text{lin},i} \oplus \mathcal{NN}(H_{\text{lin},i}; \theta)] - Y_{i+1}\|_2^2$$

is used to compute the gradients with respect to both θ and W_{out}.

After completing training, we evaluate the model using the remaining points, referred to as test points. For each prediction, we use the trained W_{out} and $H_{\text{total},i}$.

$$\hat{Y}_8 = W_{\text{out}} H_{\text{total},7}, \text{ where } H_{\text{total},7} = H_{\text{lin},7} \oplus H_{\mathcal{NN},7}$$
$$\hat{Y}_9 = W_{\text{out}} H_{\text{total},8}, \text{ where } H_{\text{total},8} = H_{\text{lin},8} \oplus H_{\mathcal{NN},8}$$

and so on.