A DIMENSIONAL R2 REGRESSION METRIC

Anonymous authors
Paper under double-blind review

ABSTRACT

Evaluation metrics are the primary guide in modeling. For regression tasks, the R2 score is the gold standard, offering a magnitude-agnostic measure of accuracy that captures variance. However, R2 has three key limitations: it is limited to at most two dimensional inputs, it reduces the score to a single scalar that hides rich patterns of prediction accuracy, and it is sensitive to low-variance noise channels which can yield large, uninterpretable negative values. We introduce the Dimensional R2 score (Dim-R2), a simple extension of R2 that accepts data of arbitrary dimensionality, provides a multidimensional view of accuracy, and reduces sensitivity to noise. We demonstrate its advantages on both synthetic sinusoidal data and data-constrained recurrent neural networks trained to simulate mouse neural activity during a skilled motor task. Dim-R2 offers an interpretable and flexible metric that illuminates patterns in regression accuracy, guiding regression modeling.

1 INTRODUCTION

Evaluation metrics are the lighthouse of modeling. They quantify how well the model's predictions match the target data and guides decisions such as model tuning and data cleaning (Jordan & Mitchell, 2015). For regression tasks, the R2 score is considered the gold standard compared to metrics like mean absolute error (MAE) or mean squared error (MSE) (Chicco et al., 2021). MAE shows a simple magnitude based error but cannot differentiate between biased predictions and lazy mean predictions. MSE penalizes variance and resolves the lazy mean prediction issue, but its value ranges per data domain making it hard to interpret. In contrast, R2 is a normalized, data domain-independent metric that reflects variance explained by the model normalized by the variance of the data (Eq. 1). It ranges from 1 (perfect prediction) to $-\infty$, with 0 indicating performance equal to the lazy mean prediction. Due to its ability to capture variance and its normalized score, R2 is widely used in regression evaluation (Chicco et al., 2021; Sykes, 1993; Ash & Shwartz, 1999; Sauerbrei et al., 2020; Pedregosa et al., 2011).

However, R2 has three key limitations. First, R2 is defined for 1D data which is averaged across channels for 2D data, and it cannot be directly applied to higher-dimensional regression data. Second, the R2 reduces model performance to a single scalar, offering no insight into how accuracy varies across data dimensions (Fig. 1). A multidimensional view of regression accuracy could reveal structure that could help modelers target specific features of their data and model for improvement. Third, R2 is highly sensitive to low-variance noise channels in multi-channel (2D) regression tasks. It can yield large negative values when the true data has little variation, as is the case for noisy channels. When these R2 scores are averaged across channels for multi-channel data, the mean R2 can be largely negative which obscures the presence of high accuracy channels (Fig. 2).

To address these limitations, we introduce the Dimensional R2 score (Dim-R2). Dim-R2 simply flattens selected dimensions into independent observations and computes the standard R2, while retaining the shape of the remaining dimensions. It accepts regression data of arbitrary dimensionality, overcoming the 2D limitation of conventional R2 by flexibly flattening any dimensions. As Dim-R2 can flatten and keep any dimensions, it also enables a multidimensional view of prediction accuracy (Fig. 1). For example, in data shaped (Trials, Time, Channels) (Perich et al., 2020; Yoo et al., 2022), Dim-R2 can reveal data regions that are predictable in certain trials, specific time periods, or certain channels, highlighting noisy trials, temporally localized features, or task-relevant channels. This multidimensional score helps modelers identify patterns in both their data and models. Finally, by flattening selected dimensions into independent observations, high-variance (informative) channels

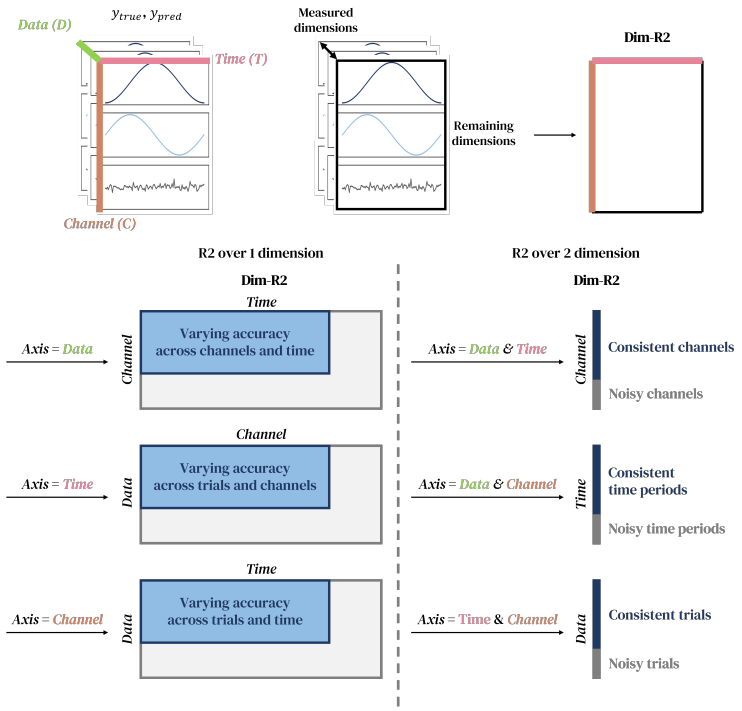


Figure 1: Dim-R^2 presents rich patterns of prediction accuracy across designated dimensions (Axis).

outweigh low-variance (noisy) ones, yielding a more robust score than the mean of per-channel R2 scores. This better highlights the presence of high accuracy channels (Fig. 2).

The paper is organized as follows. We introduce Dim-R2 as a simple extension of the conventional R2 metric, designed to support three key improvements. We then present the dimensional view of regression accuracy using Dim-R2, while emphasizing the effects of Dim-R2 arguments on the result. Next, we evaluate the resilience of Dim-R2 by comparing it to conventional Mean-R2. The input data dimensionality is not compared between Dim-R2 and R2 as it is the metric definition rather than a property of the resulting scores. Both the dimensional-view and resilience features are demonstrated using a toy sinusoidal data and a data-constrained recurrent neural network (DC-RNN) (Perich et al., 2020; Perich & Rajan, 2020) trained to simulate mouse neural activity during a reach-to-grab task. The DC-RNN example is a valuable in-silico model for neuroscience, with applications in both theory and medicine (Barh et al., 2020; Sourmpis et al., 2025; Tuladhar et al., 2021). Throughout this paper, the terms dimensions and axes, data and trials, and channels and neurons are used interchangeably.

2 Materials and Methods

2.1 Regression Metrics: Conventional and Dimensional R2

The conventional R2 score is a negative squared error normalized by the variance of y_{true}, defined for a single channel data (1D) only (Eq. 1). The value of R2 ranges from $[-\infty, 1]$, where 1 indicates perfect prediction, 0 corresponds to predicting $\bar{y}_{true}$, the mean of y_{true}, and negative values indicate worse performance than predicting $\bar{y}_{true}$.

$$R2 = 1 - \frac{RSS}{TSS} = 1 - \frac{\sum_i (y_{true,i} - y_{pred,i})^2}{\sum_i (y_{true,i} - \bar{y}_{true})^2} \tag{1}$$

where i, RSS, TSS refers to observation index, residual sum of squares, and total sum of squares, respectively. In the multi-channel case (2D), R2 score is measured per channel and averaged to yield

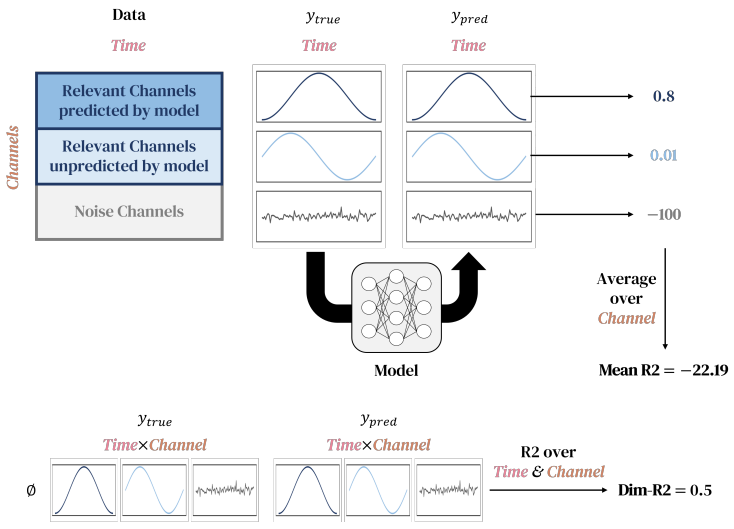


Figure 2: Dim-R^2 is resilient to noise channels than conventional mean R^2, and presents the existence of high accuracy channels among noise channels.

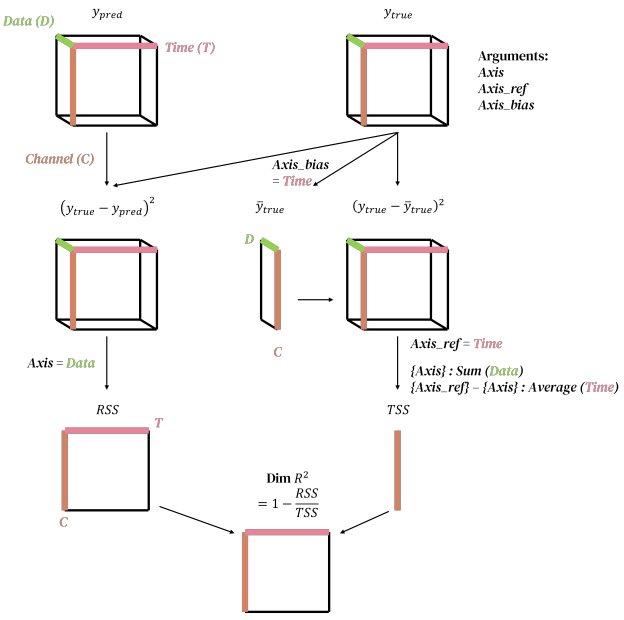


Figure 3: Schematic of Dim-R^2 calculation. There is one argument (Axis) and two optional arguments (Axis_ref, Axis_bias).

the mean R2 (Eq. 2). As a result, the conventional R2 score supports 1D data, or at most 2D data when computing mean R2.

$$Mean\ R2 = \frac{1}{C}\sum_{c}^{C}[1 - \frac{\sum_i (y_{true,c,i} - y_{pred,c,i})^2}{\sum_i (y_{true,c,i} - \overline{y_{true,c}})^2}] \tag{2}$$

where C, c refers to the number of channels and channel index, respectively.

Dim-R2 follows the concept of Eq. 1 but computes RSS and TSS in multidimensional form (Fig. 3, Eq. 3-6):

$$RSS = \sum_{i \in Axis}^{N_{Axis}} (y_{true,i} - y_{pred,i})^2 \tag{3}$$

$$\bar{y}_{true} = \frac{1}{N_{Axis\_bias}} \sum_{i \in Axis\_bias}^{N_{Axis\_bias}} y_{true,i} \tag{4}$$

$$TSS = \frac{1}{N_{Axis\_ref-Axis}} \sum_{j \in Axis\_ref-Axis}^{N_{Axis\_ref-Axis}} \sum_{i \in Axis}^{N_{Axis}} (y_{true,i,j} - \bar{y}_{true})^2 \tag{5}$$

$$Dim\text{-}R2 = 1 - \frac{RSS}{TSS} \tag{6}$$

where N_{Axis}, N_{Axis_bias}, and $N_{Axis\_ref-Axis}$ are the number of observations along the specified axes, used to compute the mean. Dim-R2 takes one argument (Axis) and two optional arguments (Axis_ref, Axis_bias). Axis is the dimension to collapse and is used to sum the error (RSS) and the y_{true} variance for normalizing (TSS). Axis_ref is the dimension which the reference variance TSS is aggregated additionally. When measuring TSS, averaging is applied to the axes in the set difference of Axis and Axis_ref (Axis-Axis_ref), to keep the magnitude of TSS consistent. Axis_bias specifies the dimension used to compute $\bar{y}_{true}$. It must be a subset of Axis_ref ($Axis\_bias \subset Axis\_ref$) since the reference variance is measured as deviation from this mean. When summing RSS and TSS, the specified dimensions (Axis, Axis_ref) are treated as independent observations. Then Dim-R2 is computed by broadcasting the shapes of RSS and TSS. This allows Dim-R2 to accept data of arbitrary dimensionality. The Dim-R2 is implemented in Python using NumPy (Harris et al., 2020) and follows the scikit-learn syntax (Pedregosa et al., 2011).

2.2 Datasets

2.2.1 Synthetic Sinusoidal Dataset

To illustrate the dimensional view of regression accuracy (Section 3.1), we generated a waveform of shape (1000, 100, 5), corresponding to trials, time, and neurons, respectively (Fig. 4). Neuron N0 to N3 share the same sine wave while N4 is pure Gaussian noise adjusted to have the same variance with clean signal N0. From N0 to N3, the noise conditions were: no noise (N0), linearly increasing noise over time (N1), linearly decreasing noise over time (N2), and constant noise over time (N3). The y_{true} and y_{pred} share the same underlying waveform but different independent random noise, to show the gradual change in accuracy. To illustrate how Dim-R2 arguments affect the dimensional view, two data conditions were used: one with no added bias (No Bias), and one with a time bias varying from 0 to 4 across neurons N0 to N4, randomly assigned across trials (Varying neural bias) (Fig. 4, 5). The No Bias condition contains no trial variability and minimal cross-neuron variability, while the Varying Neural Bias condition contains both trial and cross-neuron variability.

To demonstrate how Dim-R2 better reflects the presence of high-accuracy channels compared to mean R2 (Section 3.2), multichannel sinusoidal data of shape (100 timesteps, 100 channels) was generated for y_{true} and y_{pred} across 100 random repetitions. A fixed ratio of channels was replaced with Gaussian noise of set variance (Fig. 8). The noise channel variances for y_{true} and y_{pred} were set independently as they affect the R2 calculation.

2.2.2 Data-constrained recurrent neural network predictions on mouse motor cortex neuropixel recordings

2.2.3 Data

A total of 22 sessions were collected from four mice performing a reach-to-grab task (Sauerbrei et al., 2020; Guo et al., 2021; Levy et al., 2020). Neural activity (Spiketrains) was recorded simultaneously from the deep cerebellar nuclei (DCN), primary motor cortex (M1), Striatum, Thalamus using Neuropixels probes (Steinmetz et al., 2021). Spiketrains were recorded at 500Hz (2ms bin) where each value indicates the presence (1) or absence (0) of a spike. For each brain region, this results in a binary array of shape (time, neurons).

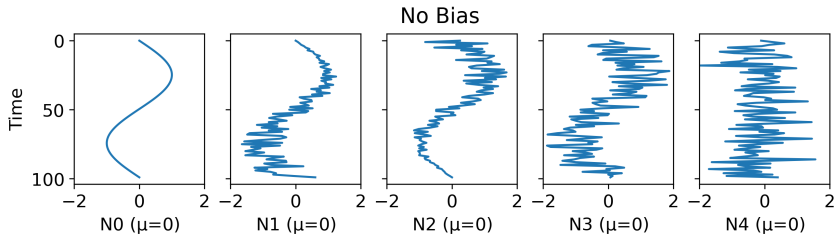


Figure 4: Example sinusoidal waveforms with time-varying noise, used for both y_{true} and y_{pred} in showing dimensional accuracy. Different biases per neuron and trial were added to other data – Consistent Neural Bias and Varying Neural Bias.

Hand kinematics were recorded using a camera at sampling rate of 500Hz, synchronized with the Neuropixels. The 3D hand coordinates (Units: mm) were extracted using Animal Part Tracker (Lee et al., 2020). Movement onset was defined as the time point when the hand position exited a predefined square region representing the resting state.

2.2.4 PREPROCESSING

Preprocessing of spiketrains involved 5 steps: slicing it to the region of interest (100ms before to 400ms after movement onset), Gaussian filtering (σ=50ms), reordering neuron indices based on peak activity for interpretability (does not affect training), normalizing activity range to match the range of DC-RNN activation function, and stacking the neurons across brain regions. The train set metadata was used to process the validation and test set for neuron reordering and activity normalization stages.

2.2.5 EXPERIMENT SETUP

The DC-RNNs were trained with 5 by 3 fold nested k-fold cross validation with 3 different random seeds, resulting in $5 \times 3 \times 3 = 45$ experiments per condition. Each train, validation, test split were stratified-split with respect to the number of reach success trials, to balance the data characteristics. The random seeds affected model weight initialization and cross validation splits.

The DC-RNN was a vanilla RNN where the number of hidden neurons matched the number of recorded neurons in the spiketrain data. The activation function was tanh. The Adam optimizer (Kingma & Ba, 2014) was used to train the DC-RNN (Werbos, 1990) with learning rate of 1e-3 and batch size of 32. Early stopping stopped training by measuring Dim-R2 every 100 updates with patience value of 100.

The y_{true} and y_{pred} were aggregated (Yoo et al., 2025) from predictions of different DC-RNN sweeps under the same experiment conditions, to evaluate the experiment condition. Resulting y_{true} and y_{pred} shapes were (Random seeds, Validation folds, Test batch, Time, Channels).

3 RESULTS

3.1 DIM-R2 PROVIDES A DIMENSIONAL VIEW OF REGRESSION ACCURACY

To demonstrate how Dim-R2 provides a dimensional view of regression accuracy, sinusoidal waveforms with time varying noise was used as y_{true} and y_{pred} (Fig. 4). No bias data, and varying time bias across trials and neurons, were added to show different characteristics of Dim-R2 (Fig. 5, see Section 2.2.1). Mean R2 was not measured as it is not defined for data over two dimensions.

Dim-R2 captures different types of variability depending on the Axis_ref argument (Fig. 3). If not specified, Axis_ref defaults to the same dimension as Axis, measuring reference variance along the collapsed dimension. For example, setting Axis_ref = Trial causes Dim-R2 to set y_{true} trial variability as its reference (Fig. 6a, c, e, g). When trial variability exists in both y_{true} and y_{pred}, capturing it is a meaningful regression goal. Thus, Dim-R2 yields high scores across all neurons including the noise neuron 4 (Fig. 6a). While lazy mean prediction $\bar{y}_{true}$ across the Axis should

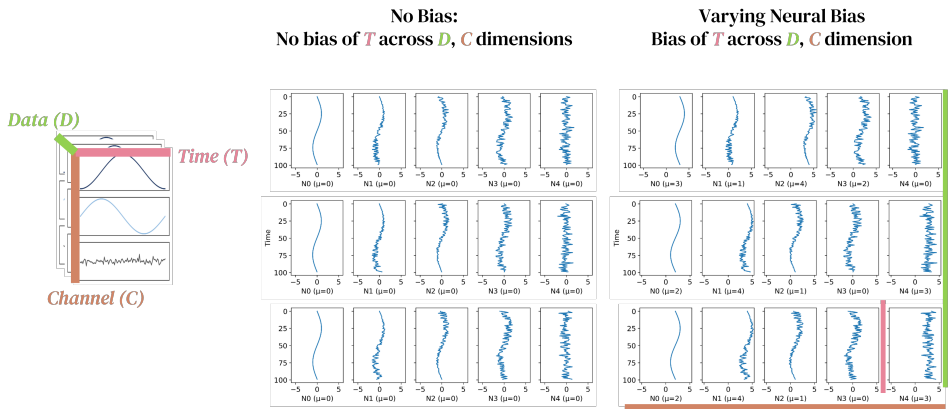


Figure 5: Three sinusoidal datasets with time-varying noise and different biases along the Time dimension, used for both y_{true} and y_{pred}. Different biases are added to each data to show the characteristics of Dim-R2. Color-coded lines in the bottom right indicate each dimension type.

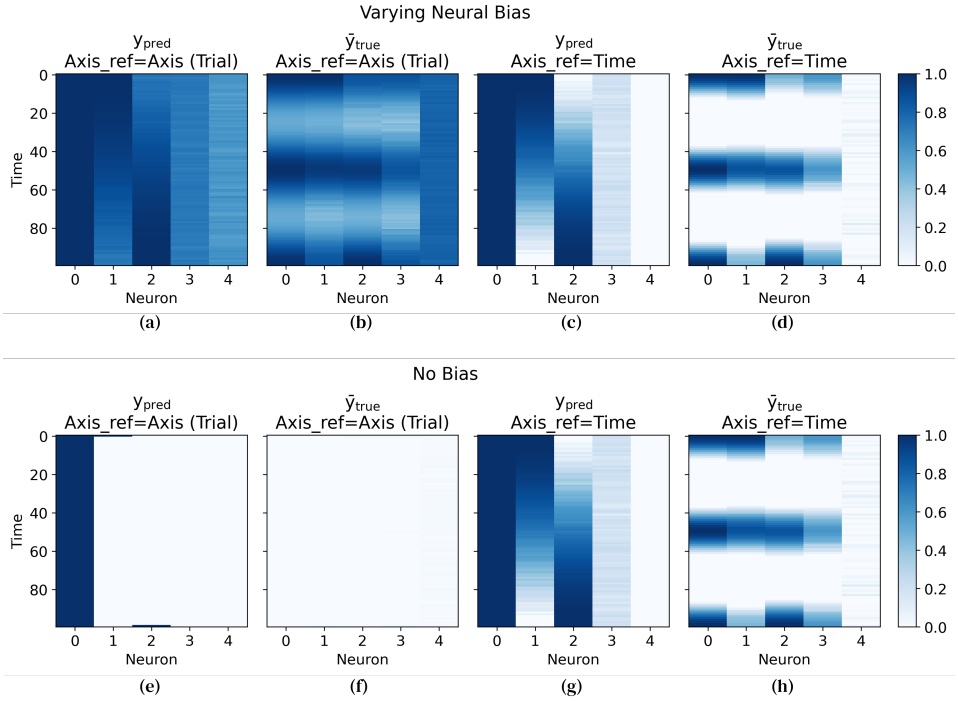


Figure 6: Dim-R2 shows dimensional view of regression accuracy for synthetic sinusoidal dataset. Each heatmap shows Dim-R2 computed between y_{true} and the specified prediction, measured across trials to reveal R2 scores across the Time and Neuron dimensions. (a) & (e) Dim-R2 on y_{pred} with Axis_ref=Axis (Trial). (b) & (f) Dim-R2 on $\bar{y}_{true}$ with Axis_ref=Axis (Trial). (c) & (g) Dim-R2 on y_{pred} with Axis_ref=Time. (d) & (h) Dim-R2 on $\bar{y}_{true}$ with Axis_ref=Time. (a)-(d) Data with varying neural bias across trials. (e)-(h) Data with no bias across trials.

yield an R2 of zero by definition, time-averaged $\bar{y}_{true}$ can still capture trial variability since the signal is time-averaged time while Axis_ref is set across trials. As a result, Dim-R2 yields a high score when trial variability exists (Fig. 6b), and a near-zero score when it does not (Fig. 6f). When y_{true} has very small variance (TSS), R2 amplifies small prediction errors (RSS), producing large negative scores. The same occurs in Dim-R2 when trial variability in y_{true} is negligible, where setting Axis_ref=Trial causes Dim-R2 to yield largely negative values (Fig. 6e).

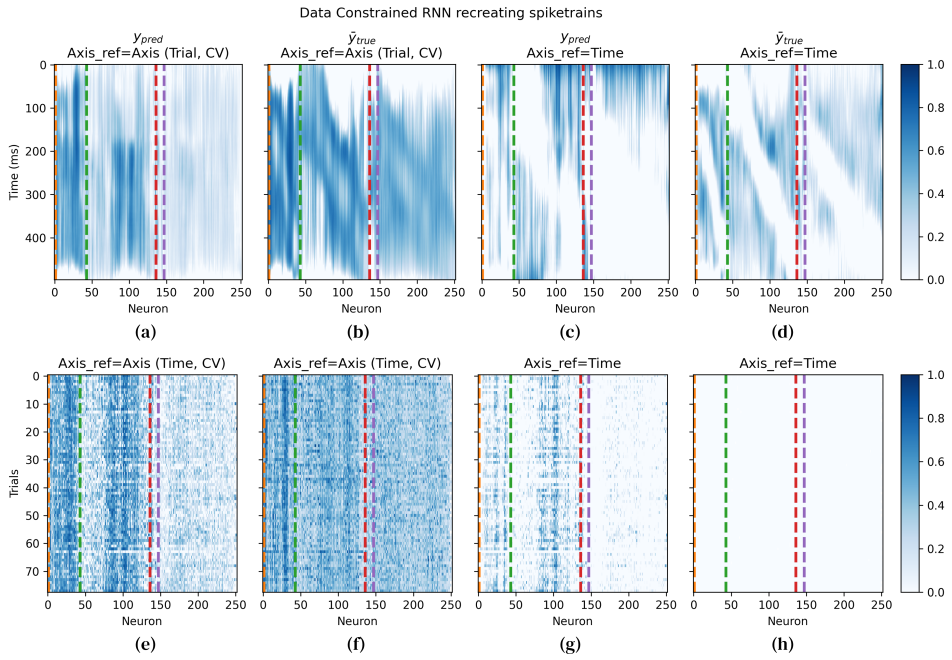


Figure 7: Dim-R2 shows dimensional view of regression accuracy for data-constrained recurrent neural networks trained to recreate neural activity. Each heatmap shows Dim-R2 computed between y_{true} and the specified prediction, measured across different dimensions to reveal R2 scores across other remaining dimensions. Dashed lines separate neurons by brain region (from left): DCN (orange), M1 (green), Striatum (red), Thalamus (purple). (a) & (e) Dim-R2 on y_{pred} with Axis_ref=Axis. (b) & (f) Dim-R2 on $\bar{y}_{true}$ with Axis_ref=Axis. (c) & (g) Dim-R2 on y_{pred} with Axis_ref=Time. (d) & (h) Dim-R2 on $\bar{y}_{true}$ with Axis_ref=Time. (a)-(d) Dim-R2 with Axis=(Random seed, Cross validation folds, Trials). (e)-(h) Dim-R2 with Axis=(Random seed, Cross validation folds, Time).

While capturing trial-variability is important, it is also meaningful to assess how well each neuron is predicted across time, regardless of trial variability. By setting Axis_ref=Time and measuring Dim-R2 across trials, the time-varying accuracy of each neuron is revealed (Fig. 6c, g). This shifts the reference variance to the time dimension not trials, resulting in identical Dim-R2 values for datasets with and without trial biases–reflecting only the time-varying noise in y_{true} and y_{pred} (Fig. 4).

Note that when the Axis (Trials) and the averaging dimension of $\bar{y}_{true}$ (Time) differ, Dim-R2 measured with $\bar{y}_{true}$ does not yield a value of 0. This is because R2 equals 0 when the prediction is the mean across the given Axis. The Dim-R2 for $\bar{y}_{true}$ with Axis_ref=Time reflects the time-varying error patterns of the sine wave, showing increased scores at the start, middle, and end for Neurons 0 to 3 (Fig. 6d, h). Therefore, interpreting Dim-R2 correctly under different Axis and Axis_ref settings, along with selecting a proper domain-specific negative control such as $\bar{y}_{true}$, is critical for meaningful analysis with Dim-R2.

To demonstrate a use case of Dim-R2, DC-RNNs were trained to reproduce neural spiketrains from wild-type mouse motor circuits during a reach-to-grab task (Fig. 13). Dim-R2 with Axis=(Trials, cross validation folds) reveals accuracy across Time and Neuron dimensions (Fig. 7a), where high values correspond to periods of strong trial variability (Fig. 6a,b,e,f). The DC-RNN captures trial variability well in DCN throughout most of the trial and in M1 after lift onset (100ms). In the ground truth y_{true}, trial variability peaks at each neuron's maximal activation (Fig. 7b). Since neurons are sorted by peak activation time within each brain region, this appears as a diagonal band of high scores per region. This trial variability is also reflected in Dim-R2 with Axis=(Time, cross validation folds) (Fig. 7e, f), where the neurons with higher trial variability show vertical bands with higher scores for both y_{pred} and $\bar{y}_{true}$.

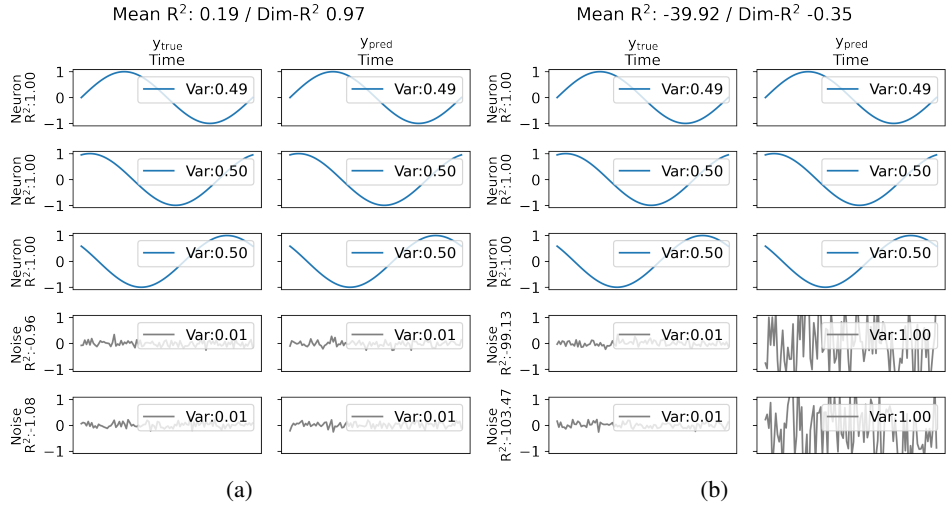


Figure 8: Example waveforms of y_{true} and y_{pred} with their corresponding mean R^2 and Dim-R^2 scores. (a) Noise channel variance: $y_{true} = 0.01$, $y_{pred} = 0.01$, (b) $y_{true} = 0.01$, $y_{pred} = 1.00$. Full variance combinations of y_{true} and y_{pred} noise channels are presented in Fig. 11.

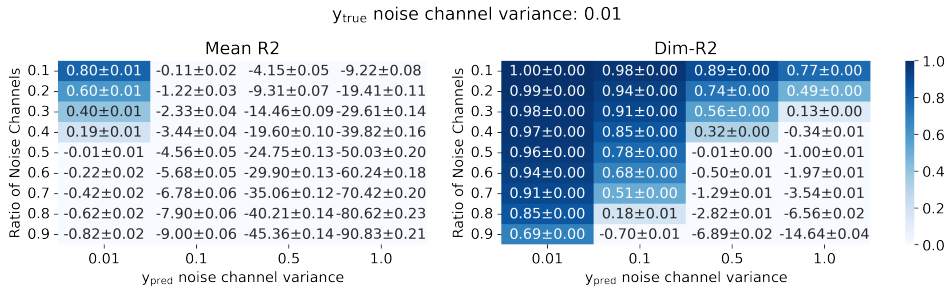
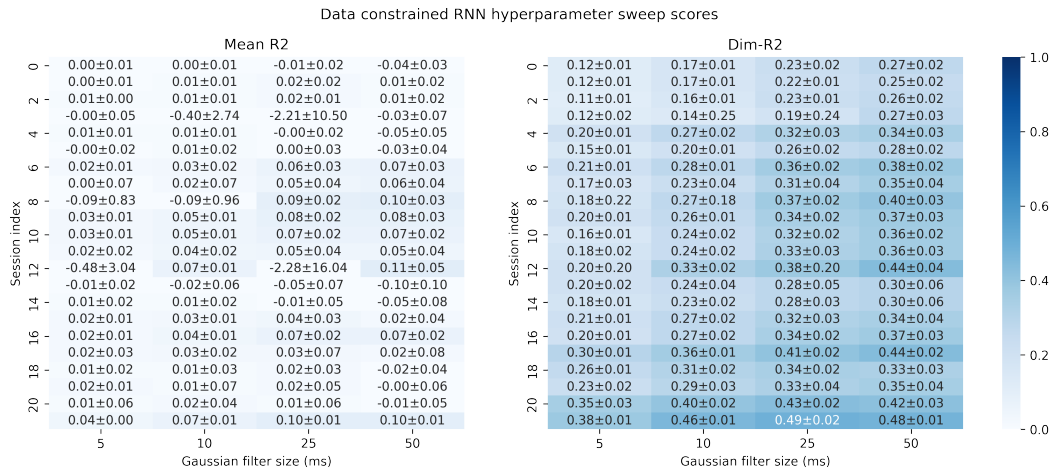


Figure 9: Dim-R2 highlights the presence of channels with high predictive accuracy in the presence of noisy channels. Scores were measured on simulated sinusoidal data (Fig. 8) across hyperparameter sweeps. Each entry shows the mean±standard deviation measured across 100 random repetitions.

Setting Axis_ref=Time evaluates each neuron's prediction accuracy across time within individual trials. Dim-R2 reveals that some neurons in the DCN and late-onset neurons in M1 are well predicted by the DC-RNN (Fig. 7c, g), despite widespread trial variability in most neurons (Fig. 7b, f). Comparing Fig 7g and h shows that DC-RNN recreates activity in specific DCN and M1 neurons. Some noisy trials can also be identified in Fig. 7g, appearing as horizontal bands with low scores. Fig. 7h correctly yields a Dim-R2 of 0 when Axis=Time as $\bar{y}_{true}$ are time averaged signals. Fig. 7d indirectly shows the underlying structure of the spiketrains, similar to Fig. 6d,g. These dimensional views reveal both the structure of the data y_{true} and the prediction patterns of the DC-RNN, guiding the modeling process.

3.2 Dim-R2 better reflects high-accuracy channels than Mean R2 in the presence of noise channels

When y_{true} has low variance, the normalizing term of R2 (TSS) becomes small which amplifies modest prediction errors into large negative values. This drags mean R2 to a negative value when multichannel y_{true} contains noise channels. In contrast, Dim-R2 with Axis set to all dimensions treats the collapsed dimensions as independent observations, allowing high variance informative



(a)

Figure 10: Dim-R2 highlights the presence of channels with high predictive accuracy in the presence of noisy channels. Scores were measured on data-constrained recurrent neural network predictions on neural spiketrains across hyperparameter sweeps. Each entry shows the mean±standard deviation.

channels to dominate the score. This reduces the effect of low variance noise channels, and better highlights the presence of high accuracy informative channels compared to mean R2.

To demonstrate how Dim-R2 better reflects the presence of high-accuracy channels, two-dimensional sinusoidal 2D data was generated for y_{true} and y_{pred} (Fig. 8, 11. See Section 2.2.1). When the y_{true} noise channel variance is low (0.01) compared to the signal channels (0.5), Dim-R2 yields significantly higher scores than mean R2 that highlight the presence of high accuracy channels (Fig. 9). This advantage decreases as the y_{true} noise channel variance increases and eventually exceeds the signal channel variance (Fig. 12). Both mean R2 and Dim-R2 decrease as y_{pred} noise channel variance increases.

To demonstrate the noise channel resilience use case of Dim-R2, Dim-R2 and mean R2 were measured on DC-RNN predictions of neural spiketrains across different recording sessions and Gaussian filter sizes used in preprocessing (Fig. 10). The scores gradually increase with larger filter sizes, as both y_{true} and y_{pred} become smoother. While this trend is visible in Dim-R2, it is less apparent in Mean R2 due to the low-variance noise channels that pull the score down. For instance, Sessions 3 and 12 at a 25ms filter size yield a moderately large negative mean R2 than other filter sizes, which is an outlier effect not observed in Dim-R2. This noise resilience can also be seen in each y_{true} and y_{pred} pairs from both trial-averaged and single trial neural activity (Fig. 13, Table 1). Thus, Dim-R2 allows users to identify the effects of hyperparameters on RNN performance without the confounding influence of noise.

4 Conclusion

We introduced Dim-R2, a regression metric that accepts data of arbitrary dimensionality, enables dimensional evaluation of regression accuracy, and highlights the presence of high-accuracy channels in the presence of noisy channels. When using Dim-R2 to assess accuracy across dimensions, careful consideration of Axis and Axis_ref is essential, especially when they differ. In such cases, a Dim-R2 score of 0 does not imply equivalence to mean prediction but may instead reflect the data structure (Fig. 6d, h; 7d). To interpret these dimensional scores meaningfully, it is important to compare against domain-specific controls, such as time-averaged $\bar{y}_{true}$. Dim-R2 offers a bird's-eye view of regression accuracy and a noise-resilient score for reliable hyperparameter exploration, offering guidance for model evaluation in fields such as artificial intelligence and neuroscience (Huang et al., 2024; Yoo et al., 2024; Badrulhisham et al., 2024; Zador et al., 2023; Yang & Wang, 2020; Lin et al., 2023).

REFERENCES

Arlene Ash and Michael Shwartz. R2: a useful measure of model performance when predicting a dichotomous outcome. *Statistics in medicine*, 18(4):375–384, 1999.

Fakhirah Badrulhisham, Esther Pogatzki-Zahn, Daniel Segelcke, Tamas Spisak, and Jan Vollert. Machine learning and artificial intelligence in neuroscience: A primer for researchers. *Brain, Behavior, and Immunity*, 115:470–479, 2024. ISSN 0889-1591. doi: https://doi.org/10.1016/j.bbi.2023.11.005. URL https://www.sciencedirect.com/science/article/pii/S0889159123003380.

Debmalya Barh, Eugenia Ch. Yiannakopoulou, Emmanuel O. Salawu, Atanu Bhattacharjee, Sudhir Chowbina, Joseph J. Nalluri, Preetam Ghosh, and Vasco Azevedo. Chapter 22 - in silico disease model: from simple networks to complex diseases. In Ashish S. Verma and Anchal Singh (eds.), *Animal Biotechnology (Second Edition)*, pp. 441–460. Academic Press, Boston, second edition edition, 2020. ISBN 978-0-12-811710-1. doi: https://doi.org/10.1016/B978-0-12-811710-1.00020-3. URL https://www.sciencedirect.com/science/article/pii/B9780128117101000203.

Davide Chicco, Matthijs J Warrens, and Giuseppe Jurman. The coefficient of determination r-squared is more informative than smape, mae, mape, mse and rmse in regression analysis evaluation. *Peerj computer science*, 7:e623, 2021.

Jian-Zhong Guo, Britton A Sauerbrei, Jeremy D Cohen, Matteo Mischiati, Austin R Graves, Ferruccio Pisanello, Kristin M Branson, and Adam W Hantman. Disrupting cortico-cerebellar communication impairs dexterity. *eLife*, 10:e65906, jul 2021. ISSN 2050-084X. doi: 10.7554/eLife.65906. URL https://doi.org/10.7554/eLife.65906.

Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, September 2020. doi: 10.1038/s41586-020-2649-2. URL https://doi.org/10.1038/s41586-020-2649-2.

Ann Huang, Satpreet H Singh, and Kanaka Rajan. Measuring and controlling solution degeneracy across task-trained recurrent neural networks. *arXiv preprint arXiv:2410.03972*, 2024.

M. I. Jordan and T. M. Mitchell. Machine learning: Trends, perspectives, and prospects. *Science*, 349(6245):255–260, 2015. doi: 10.1126/science.aaa8415. URL https://www.science.org/doi/abs/10.1126/science.aaa8415.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Allen Lee, Mayank Kabra, Alice Robie, Stephen Huston, Felipe Rodriguez, Roian Egnor, Austin Edwards, and Kristin Branson. APT: The Animal Part Tracker. https://github.com/kristinbranson/APT, 2020. Howard Hughes Medical Institute, Janelia Research Campus. GNU GPLv3 license.

Shahar Levy, Maria Lavzin, Hadas Benisty, Amir Ghanayim, Uri Dubin, Shay Achvat, Zohar Brosh, Fadi Aeed, Brett D. Mensh, Yitzhak Schiller, Ron Meir, Omri Barak, Ronen Talmon, Adam W. Hantman, and Jackie Schiller. Cell-type-specific outcome representation in the primary motor cortex. *Neuron*, 107(5):954–971.e9, 2020. ISSN 0896-6273. doi: https://doi.org/10.1016/j.neuron.2020.06.006. URL https://www.sciencedirect.com/science/article/pii/S0896627320304359.

Dongyan Lin, Ann Zixiang Huang, and Blake Aaron Richards. Temporal encoding in deep reinforcement learning agents. *Scientific Reports*, 13(1):22335, 2023.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017. URL https://openreview.net/forum?id=BJJsrmfCZ. Workshop Paper.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Matthew G Perich and Kanaka Rajan. Rethinking brain-wide interactions through multi-region 'network of networks' models. *Current Opinion in Neurobiology*, 65:146–151, 2020. ISSN 0959-4388. doi: https://doi.org/10.1016/j.conb.2020.11.003. URL https://www.sciencedirect.com/science/article/pii/S0959438820301707. Whole-brain interactions between neural circuits.

Matthew G Perich, Charlotte Arlt, Sofia Soares, Megan E Young, Clayton P Mosher, Juri Minxha, Eugene Carter, Ueli Rutishauser, Peter H Rudebeck, Christopher D Harvey, et al. Inferring brain-wide interactions using data-constrained recurrent neural network models. *BioRxiv*, pp. 2020–12, 2020.

Britton A Sauerbrei, Jian-Zhong Guo, Jeremy D Cohen, Matteo Mischiati, Wendy Guo, Mayank Kabra, Nakul Verma, Brett Mensh, Kristin Branson, and Adam W Hantman. Cortical pattern generation during dexterous movement is input-driven. *Nature*, 577(7790):386–391, 2020.

Christos Sourmpis, Carl C.H. Petersen, Wulfram Gerstner, and Guillaume Bellec. Biologically informed cortical models predict optogenetic perturbations. *bioRxiv*, 2025. doi: 10.1101/2024.09.27.615361. URL https://www.biorxiv.org/content/early/2025/03/15/2024.09.27.615361.1.

Nicholas A. Steinmetz, Cagatay Aydin, Anna Lebedeva, Michael Okun, Marius Pachitariu, Marius Bauza, Maxime Beau, Jai Bhagat, Claudia Böhm, Martijn Broux, Susu Chen, Jennifer Colonell, Richard J. Gardner, Bill Karsh, Fabian Kloosterman, Dimitar Kostadinov, Carolina Mora-Lopez, John O'Callaghan, Junchol Park, Jan Putzeys, Britton Sauerbrei, Rik J. J. van Daal, Abraham Z. Vollan, Shiwei Wang, Marleen Welkenhuysen, Zhiwen Ye, Joshua T. Dudman, Barundeb Dutta, Adam W. Hantman, Kenneth D. Harris, Albert K. Lee, Edvard I. Moser, John O'Keefe, Alfonso Renart, Karel Svoboda, Michael Häusser, Sebastian Haesler, Matteo Carandini, and Timothy D. Harris. Neuropixels 2.0: A miniaturized high-density probe for stable, long-term brain recordings. *Science*, 372(6539):eabf4588, 2021. doi: 10.1126/science.abf4588. URL https://www.science.org/doi/abs/10.1126/science.abf4588.

Alan O Sykes. An introduction to regression analysis. 1993.

Anup Tuladhar, Jasmine A Moore, Zahinoor Ismail, and Nils D Forkert. Modeling neurodegeneration in silico with deep learning. *Frontiers in Neuroinformatics*, 15:748370, 2021.

Paul J Werbos. Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, 78(10):1550–1560, 1990.

Guangyu Robert Yang and Xiao-Jing Wang. Artificial neural networks for neuroscientists: a primer. *Neuron*, 107(6):1048–1070, 2020.

Jaesung Yoo, Ilhan Yoo, Ina Youn, Sung-Min Kim, Ri Yu, Kwangsoo Kim, Keewon Kim, and Seung-Bo Lee. Residual one-dimensional convolutional neural network for neuromuscular disorder classification from needle electromyography signals with explainability. *Computer Methods and Programs in Biomedicine*, 226:107079, 2022. ISSN 0169-2607. doi: https://doi.org/10.1016/j.cmpb.2022.107079. URL https://www.sciencedirect.com/science/article/pii/S0169260722004606.

Jaesung Yoo, Fernanda de la Torre, and Guangyu Robert Yang. Dual policy as self-model for planning. *Journal of Korean Institute of Intelligent Systems*, 34(1):15–24, 2024. doi: https://doi.org/10.5391/JKIIS.2024.34.1.15.

Jaesung Yoo, Sunghyuk Choi, Ye Seul Yang, Suhyeon Kim, Jieun Choi, Dongkyeong Lim, Yaeji Lim, Hyung Joon Joo, Dae Jung Kim, Rae Woong Park, Hyung-Jin Yoon, and Kwangsoo Kim. Review learning: Real world validation of privacy preserving continual learning across medical institutions. *Computers in Biology and Medicine*, 192:110239, 2025. ISSN 0010-4825. doi: https://doi.org/10.1016/j.compbiomed.2025.110239. URL https://www.sciencedirect.com/science/article/pii/S0010482525005906.

Anthony Zador, Sean Escola, Blake Richards, Bence Ölveczky, Yoshua Bengio, Kwabena Boahen, Matthew Botvinick, Dmitri Chklovskii, Anne Churchland, Claudia Clopath, et al. Catalyzing next-generation artificial intelligence through neuroai. *Nature communications*, 14(1):1597, 2023.

A APPENDIX

A.1 METHODS DETAILS

DC-RNN weights were initialized using He initialization, a uniform distribution $\mathcal{U}(-\sqrt{\frac{1}{hidden\_size}}, \sqrt{\frac{1}{hidden\_size}}$, which is the default in PyTorch (Paszke et al., 2017). The DC-RNN was implemented with PyTorch (Paszke et al., 2017). The learning rate and batch size were selected after initial parameter sweep with learning rate of 1e-2, 1e-3, and batch size of 16, 32.

A.2 THE USE OF LARGE LANGUAGE MODELS (LLMS)

The Large Language Model ChatGPT (https://chatgpt.com/) was used only to revise grammar in writing.

A.3 SUPPLEMENTARY MATERIALS

Figure 11: Example schematic waveforms of y_{true} and y_{pred} with their corresponding mean R^2 and Dim-R^2 scores. Each example shows sine waves with varying phases phases across 5 channels, where 2 channels have been replaced with Gaussian noise of specified variance. Rows share the same y_{true} noise channel variance; columns share the same y_{pred} noise channel variance. Legends show variances.

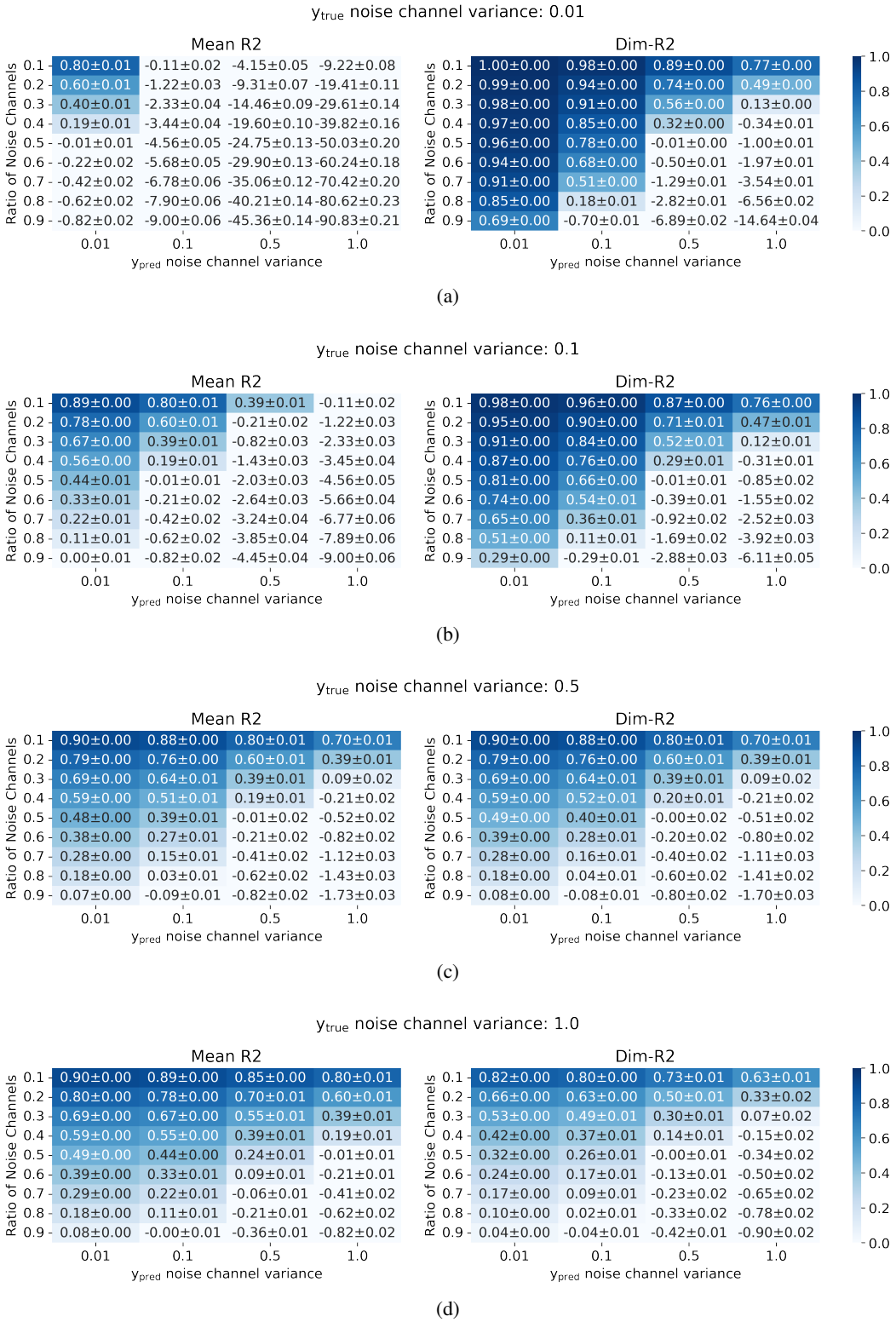


Figure 12: Dim-R2 highlights the presence of channels with high predictive accuracy in the presence of noisy channels. Scores were measured on simulated sinusoidal data (Fig. 8) across hyperparameter sweeps. Each entry shows the mean±standard deviation across 100 random repetitions.

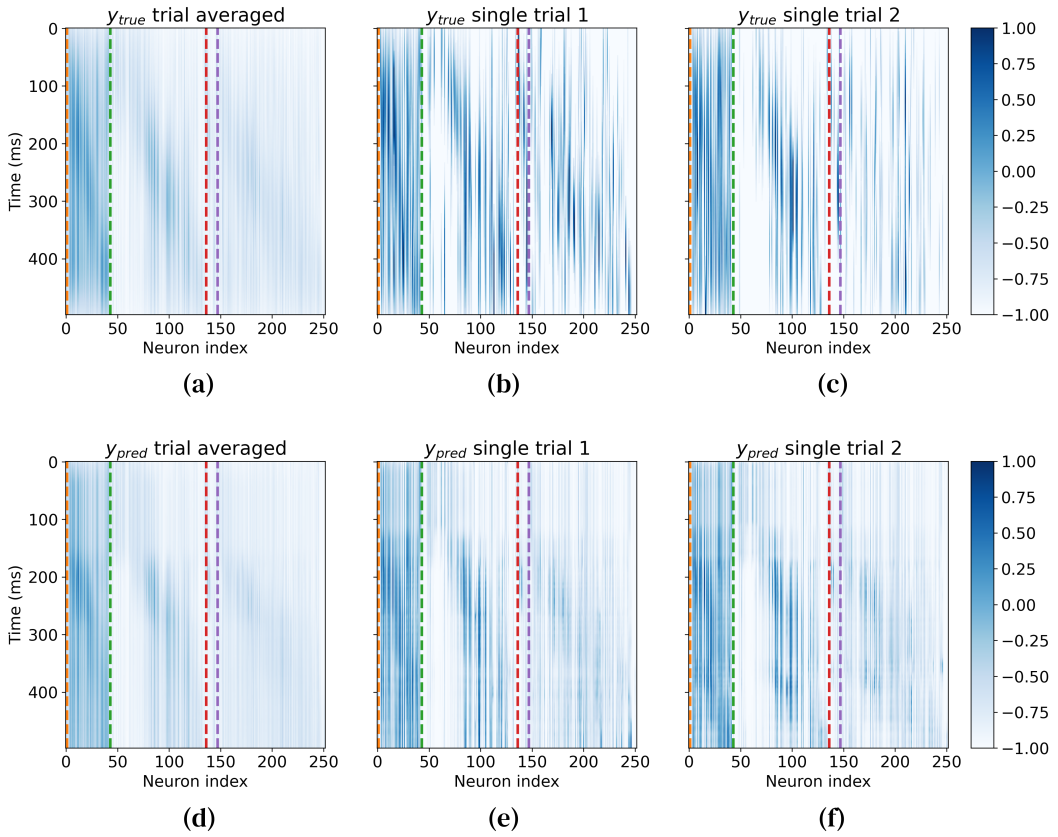


Figure 13: Examples of y_{true} and y_{pred} from DC-RNN trained to reproduce neural activity. This single session contains 78 trials, with 42, 93, 11, and 106 neurons from DCN, M1, Striatum, and Thalamus, respectively. This session corresponds to Session index 21 with a 50ms Gaussian filter in Fig. 10. Dashed lines separate neurons by brain region (from left): DCN (orange), M1 (green), Striatum (red), Thalamus (purple). (a) & (d) Trial averaged activity, (b) & (e) Single trial example 1, (c) & (f) Single trial example 2. (a)-(c) y_{true}, (d)-(f) y_{pred}.

	Trial averaged (a) & (d)	Single trial 1 (b) & (e)	Single trial 2 (c) & (f)
Dim-R2	0.92	0.44	0.57
Mean R2	0.57	-0.28	-0.41

Table 1: Dim-R2 and Mean R2 measured on y_{true} and y_{pred} of Fig. 13. Dim-R2 yields higher scores than mean R2 when the y_{true} and y_{pred} are similar, even in the presence of noise.