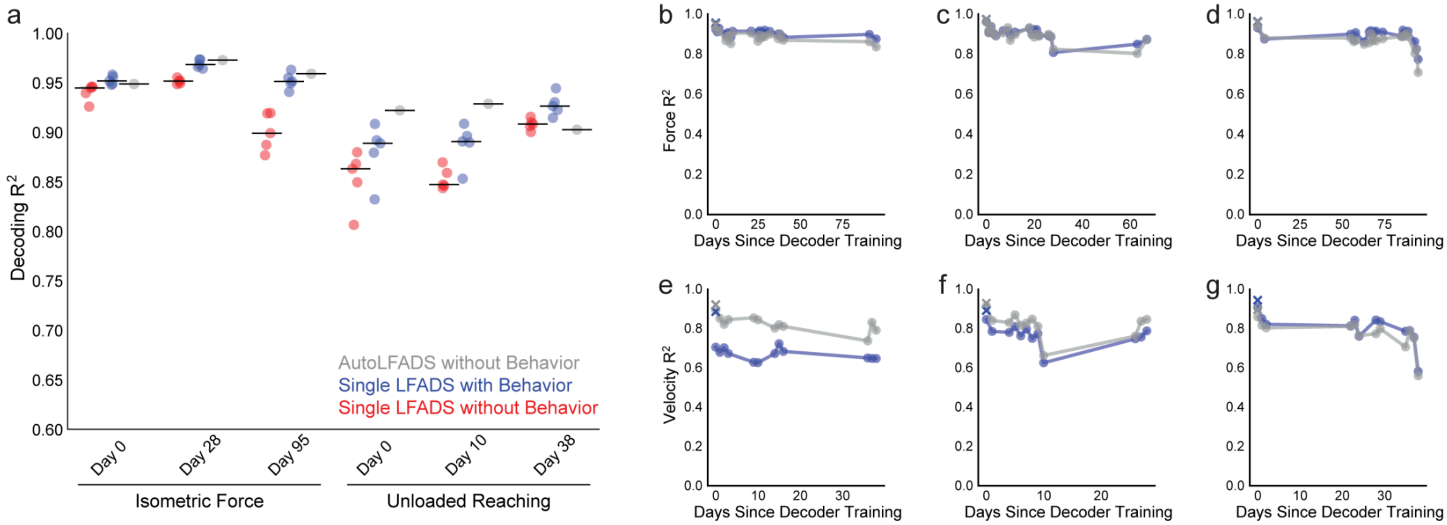
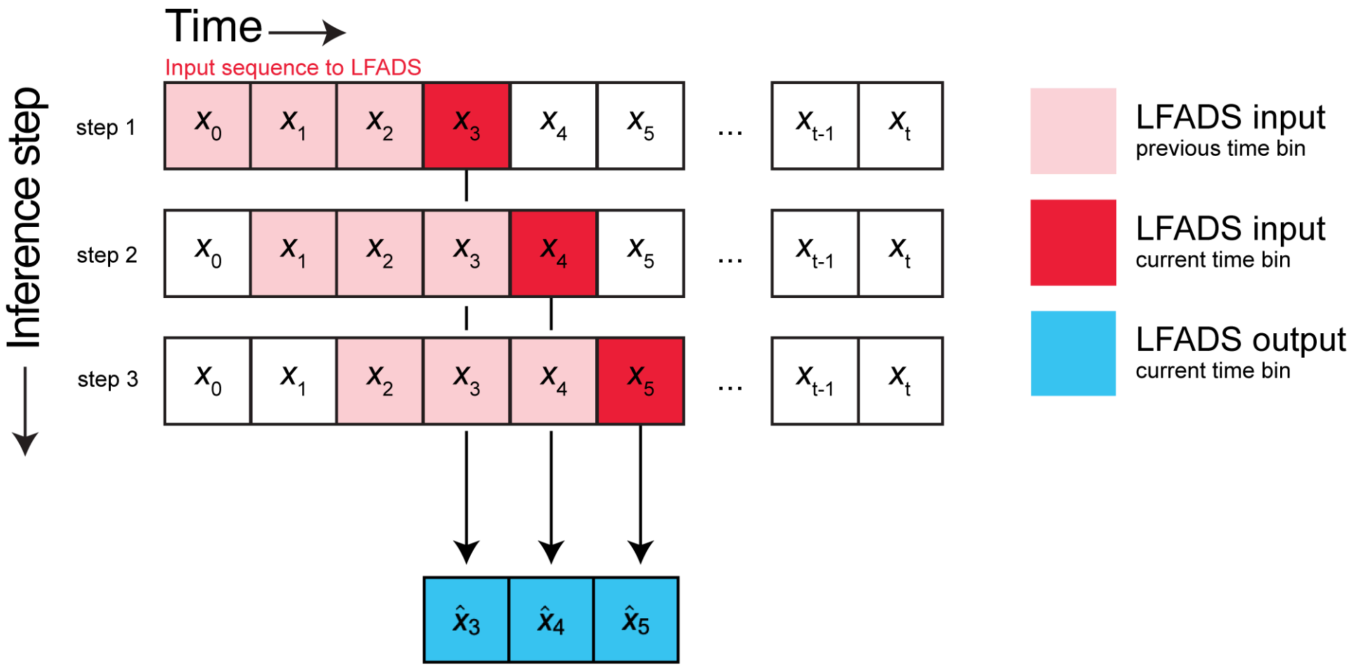
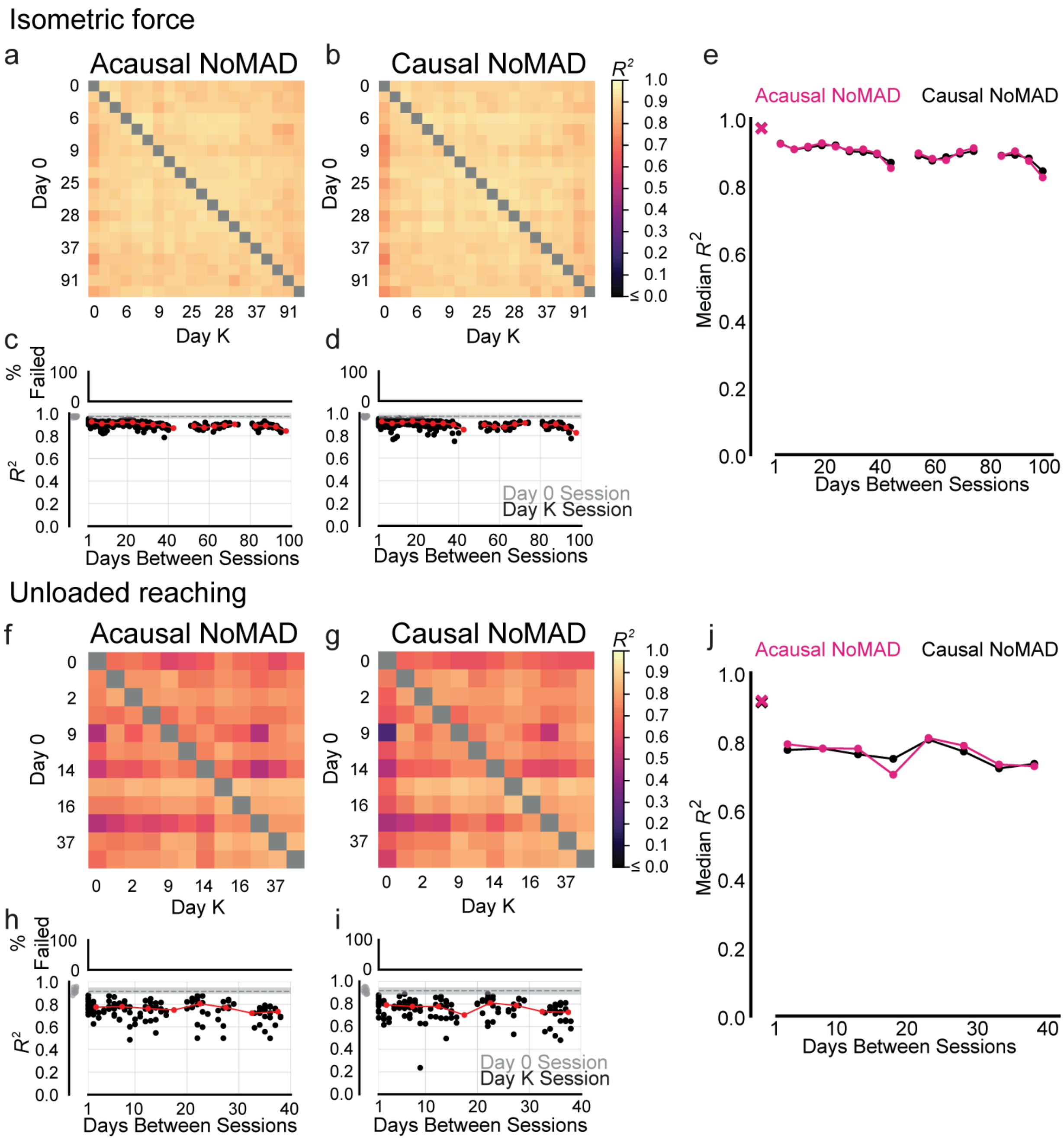
Supplementary Figures



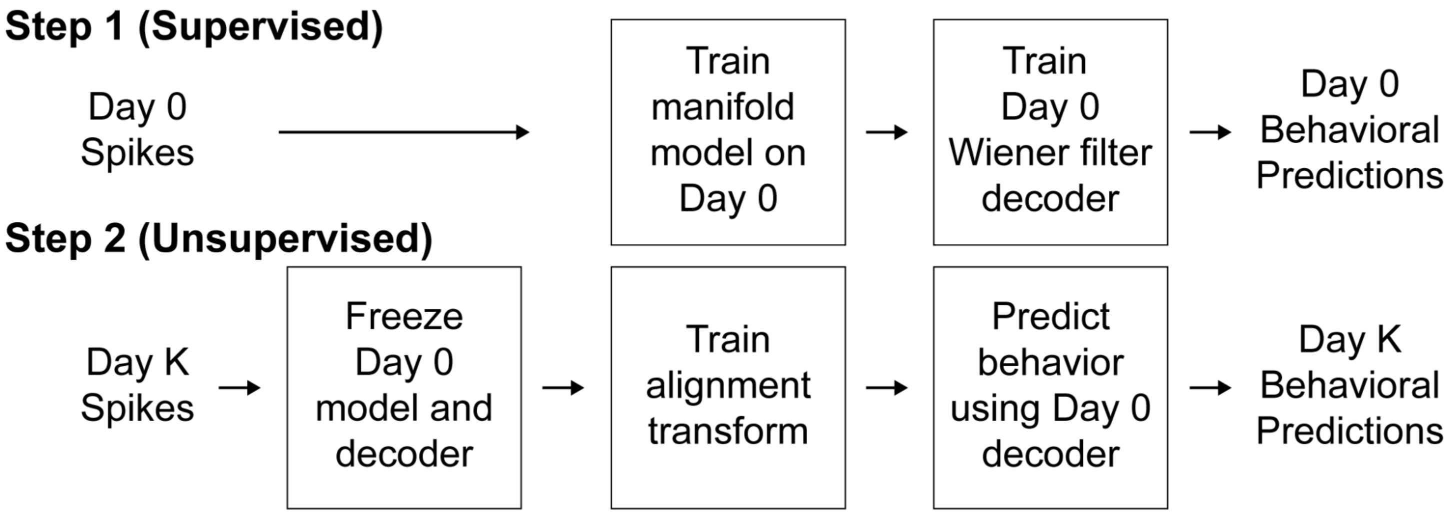
Supplementary Figure 1 | Effect of behavioral readout on decoding accuracy and alignment performance. (a) Comparison of behavioral decoding performance for three datasets from the isometric force task (left) and three datasets from the unloaded reaching task (right) using either a single LFADS model with the behavioral readout and trained with fixed hyperparameters (as described in Fig. 2; blue), a single LFADS model without the behavioral readout and trained with fixed hyperparameters (red), or an AutoLFADS model without the behavioral readout (gray). Single points denote the decoding R^2 for one model. For the single LFADS models, long black bars denote the median performance across five separately trained models with different random initializations. For the AutoLFADS model, the long black bar is included as a visual guide representing the decoding performance of the final model. (b) Alignment performance when aligning all other datasets to a single LFADS model with behavioral readout (blue) and to the AutoLFADS model without behavioral readout (gray) for isometric force Day 0. (c-d) Same as (b) but for isometric force Day 28 (c) and Day 95 (d). Isometric force datasets show significant increases in alignment performance for two of the three tested datasets using the behavioral readout models versus AutoLFADS models without the readout (p = 1.34e-5, 0.138, 1.91e-5 for Day 0, Day 28, Day 95 in one-sided Wilcoxon signed-rank test). (e-g) Same as (b) but for unloaded reaching Day 0 (e), Day 10 (f), and Day 38 (g). There are not significant increases in alignment performance using the behavioral readout models versus AutoLFADS models for the unloaded reaching task for two of the three datasets (p = 1.00, 1.00, 2.44e-3 for Day 0, Day 10, Day 38 in one-sided Wilcoxon signed-rank test).



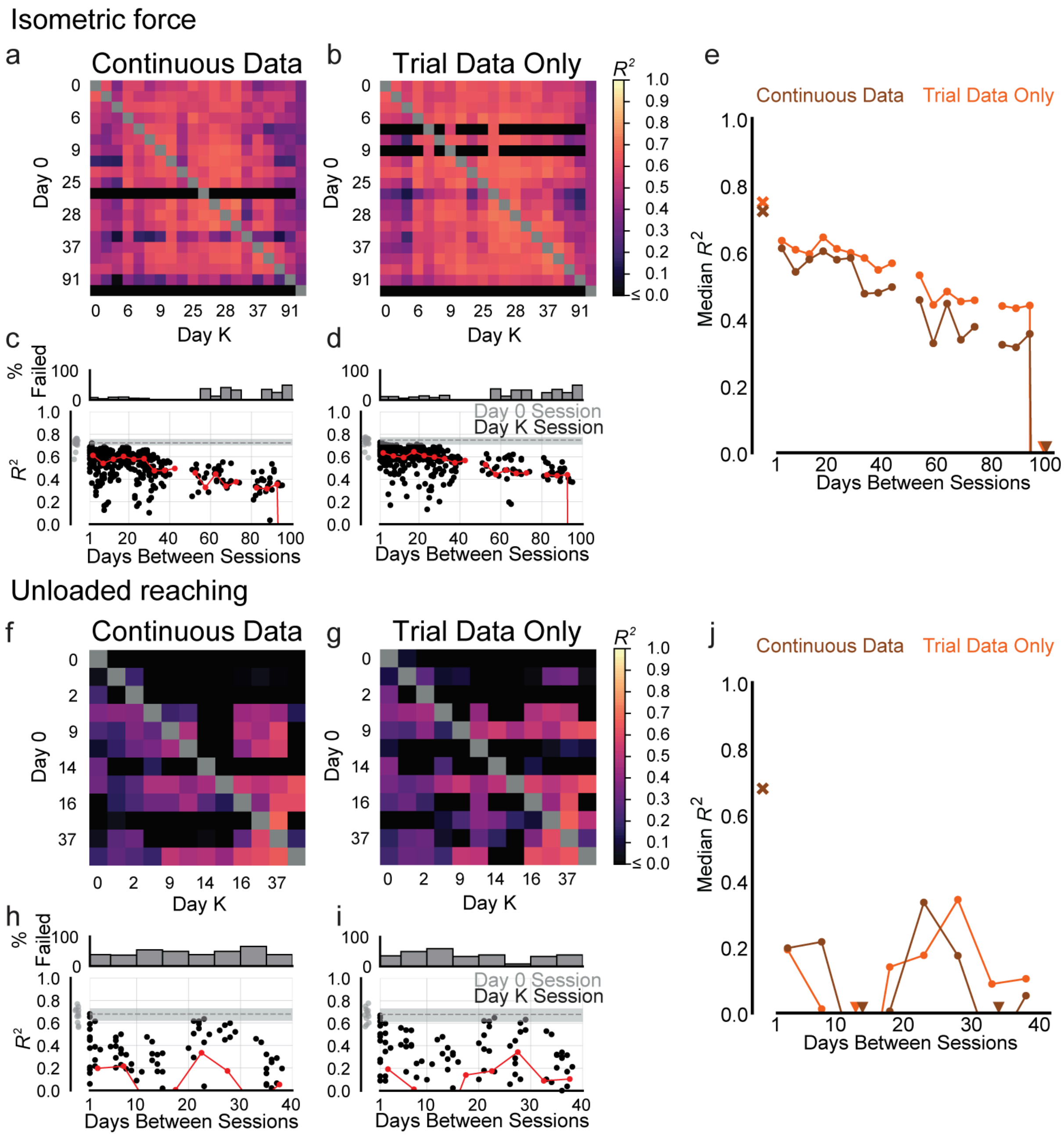
Supplementary Figure 2 | Schematic of sliding window approach for LFADS causal inference. For causal inference, we construct a window of input data that is the same size as the sequence length used for model training. The majority of this window consists of previously observed data. At each inference step, a single new bin of input data is added to the window. After inference on this input sequence, we obtain a sequence of model outputs. The time bin of the output sequence corresponding to the current time bin is taken as the new sample of the model's inferred output. The full output is therefore assembled one-at-a-time as new time bins are encompassed by the sliding window.



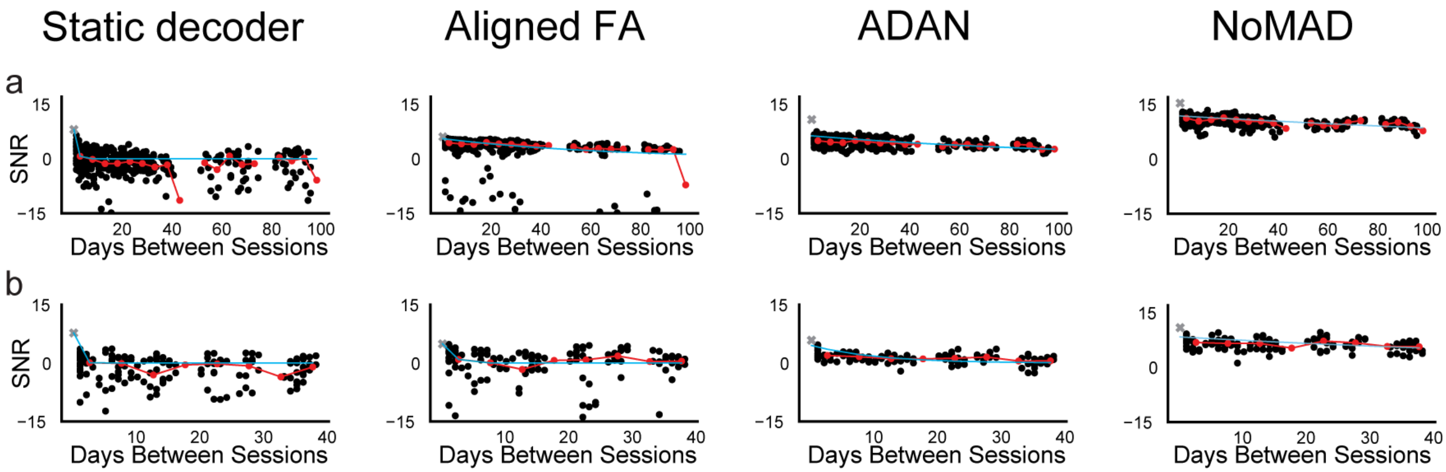
Supplementary Figure 3 | Comparison of Causal and Acausal NoMAD. (a-b) R^2 of aligned force decoding for Acausal NoMAD (a) and Causal NoMAD (b) applied to all pairs of sessions of the isometric force task (20 sessions spanning 95 days). (c-d) Top: Percentage of Day K decoding failures (R^2<0) in 5-day bins. Bottom: Decoding performance as a function of days between sessions. Black points indicate Day K decoding performance for single pairs of days. Red points show the median Day K performance within each bin of width 5 days. Gray points denote initial performance for Day 0 decoders (i.e., evaluated on held-out same-day data) for each of the 20 datasets. Gray dashed lines indicate median decoding performance of Day 0 decoders, and shaded regions around them represent the first and third quartiles. There is no significant decrease between the two NoMAD inference approaches (p = 1.0 in one-sided Wilcoxon signed-rank test). (e) Median R^2 of force decoding within each 5-day bin. Triangles indicate points that fall below the limit of the y-axis and x-markers indicate median within-day force decoding performance on Day 0. (f-g) Same as (a-b) but for the unloaded reaching task (12 sessions spanning 38 days). (h-i) Same as (c-d) but for the unloaded reaching task. There is no significant decrease between Causal and Acausal NoMAD (p = 0.99 in one-sided Wilcoxon signed-rank test). (j) Same as (e) but for the unloaded reaching task.



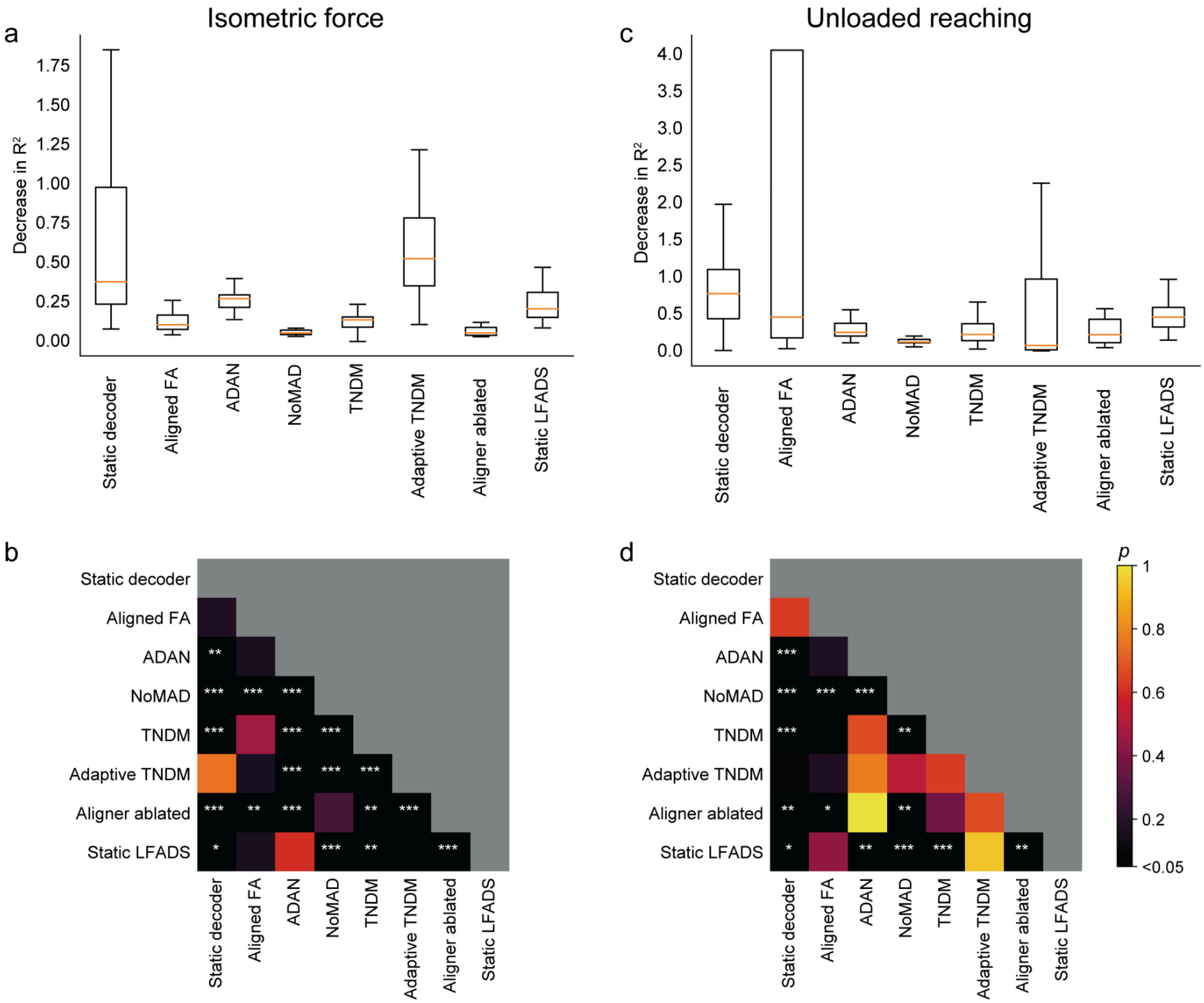
Supplementary Figure 4 | Schematic of model and decoder training procedure. The same model and decoder training procedure is used to produce results for all compared methods (NoMAD, ADAN, and Aligned FA). On some initial Day 0, we use a supervised training dataset containing both neural activity and behavior. We use this to train our manifold model (LFADS for NoMAD, an autoencoder for ADAN, and factor analysis for Aligned FA) and obtain manifold estimates on Day 0. The manifold estimates and behavior are then used to train a Wiener filter decoder on Day 0 (see *Methods - Neural Decoding*). On some later Day K, we assume we only have access to an unsupervised dataset (i.e., new neural data but no new behavioral data). We hold the manifold model and trained decoder from Day 0 fixed. We train the appropriate alignment transformation (the alignment network in NoMAD, the GAN in ADAN, and the Procrustes alignment in Aligned FA) to obtain updated manifold estimates on Day K. These Day K manifold estimates are then used with the Day 0 Wiener filter decoder to yield Day K behavioral predictions.



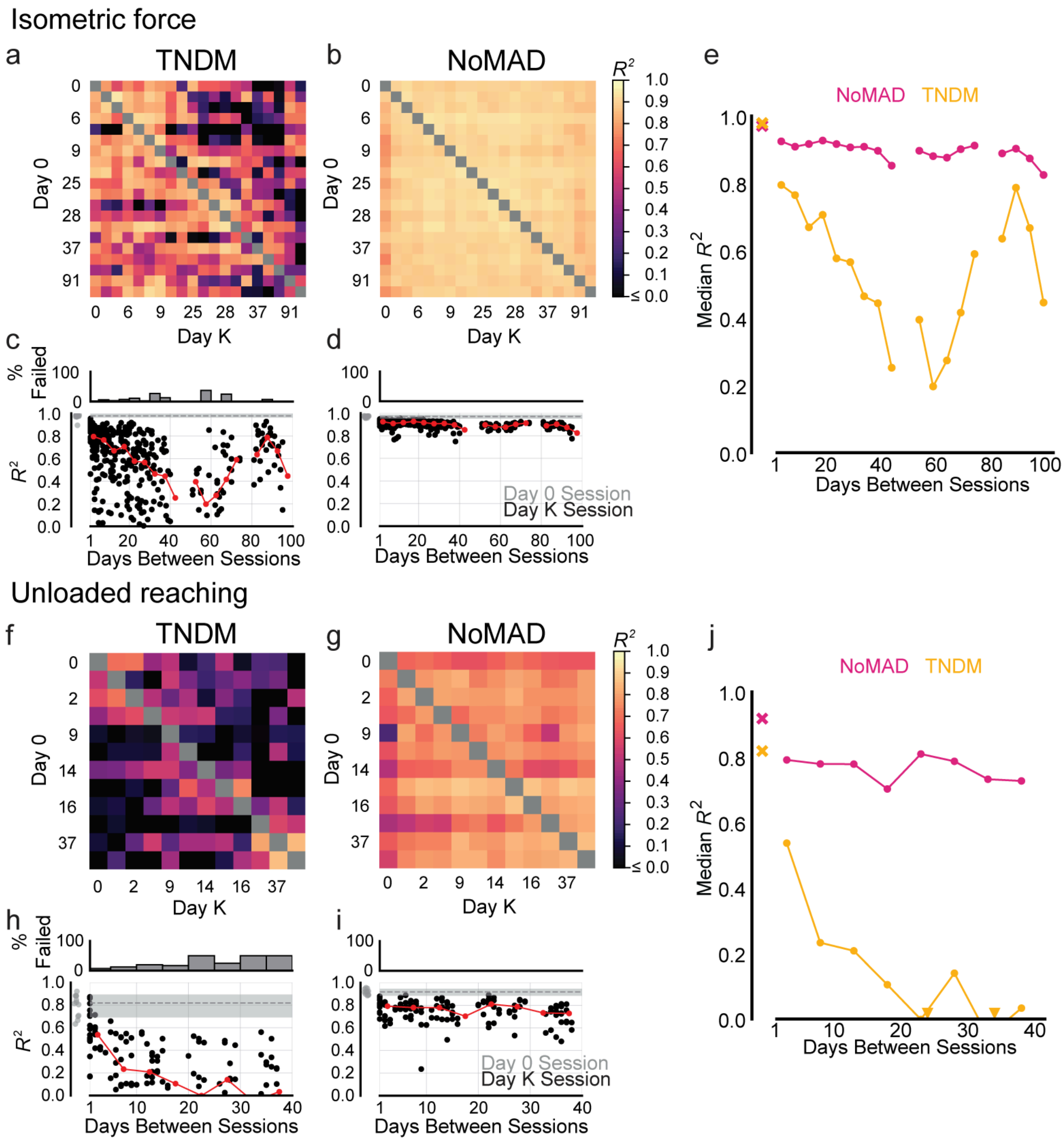
Supplementary Figure 5 | Comparison of Aligned FA using the continuous data vs. within-trial data only. (a-b) R^2 of aligned force decoding for Aligned FA with the continuous data (a) and Aligned FA using within-trial data only (b) applied to all pairs of sessions of the isometric force task (20 sessions spanning 95 days). (c-d) Top: Percentage of Day K decoding failures (R^2<0) in 5-day bins. Bottom: Decoding performance as a function of days between sessions. Black points indicate Day K decoding performance for single pairs of days. Red points show the median Day K performance within each bin of width 5 days. Gray points denote initial performance for Day 0 decoders (i.e., evaluated on held-out same-day data) for each of the 20 datasets. Gray dashed lines indicate median decoding performance of Day 0 decoders, and shaded regions around them represent the first and third quartiles. (e) Median R^2 of force decoding within each 5-day bin. Triangles indicate points that fall below the limit of the y-axis and x-markers indicate median within-day force decoding performance on Day 0. There is a significant advantage of the Trial Data Only models over the Continuous Data models (p = 1.34e-10 in one-sided Wilcoxon signed rank test). (f-g) Same as (a-b) but for the unloaded reaching task (12 sessions spanning 38 days). (h-i) Same as (c-d) but for the unloaded reaching task. (j) Same as (e) but for the unloaded reaching task. There is a significant advantage of the Trial Data Only models over the Continuous Data models for this task (p = 7.04e-3 in one-sided Wilcoxon signed rank test).



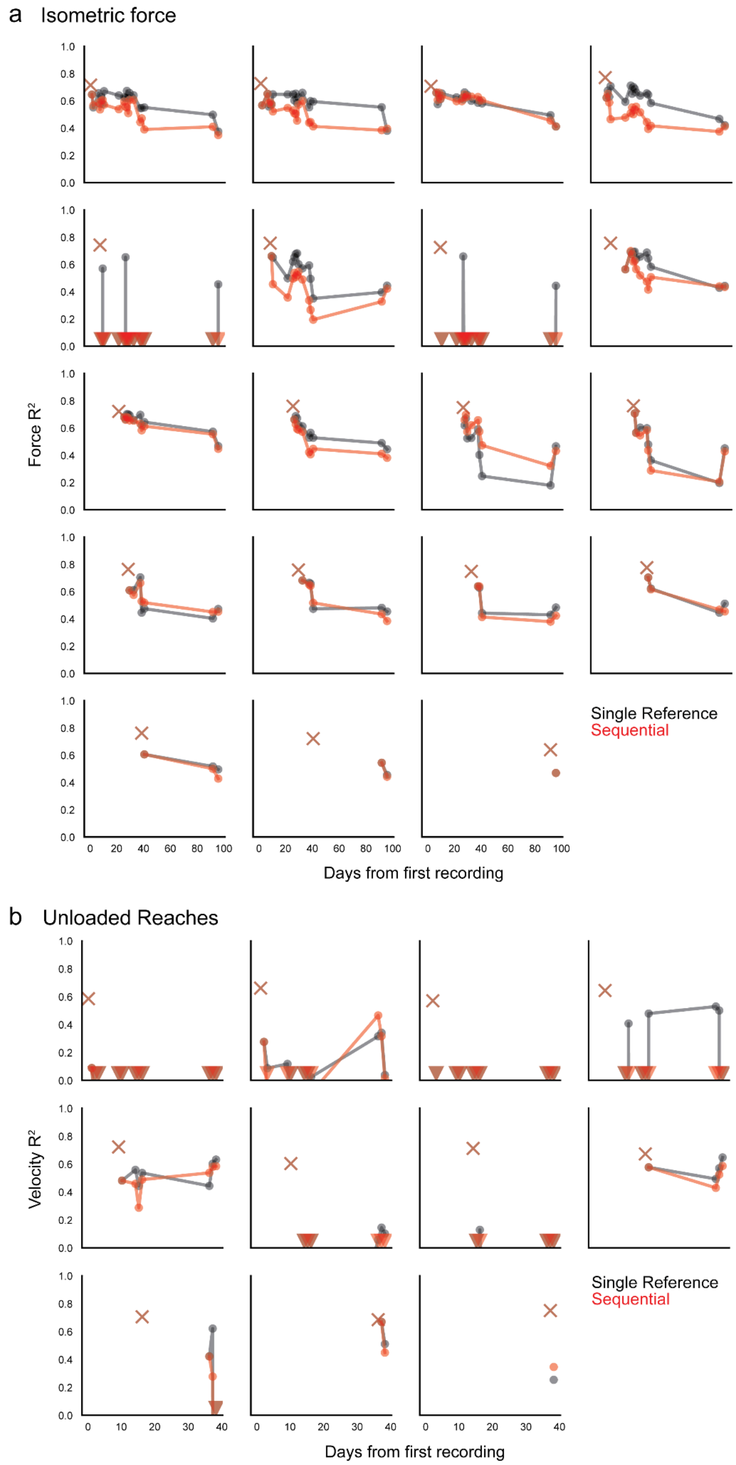
Supplementary Figure 6 | Exponential decay of decoding performance by method. Exponential curve fits (blue) to the median data (red) for (a) the isometric force datasets and (b) the reaching datasets. Decoding performance (R^2) for individual pairs of days is shown by the black dots. The median R^2 of the Day 0 decoder or classifier applied within the training day is shown by the gray x. Data is from experiments in Figures 3 and 4.



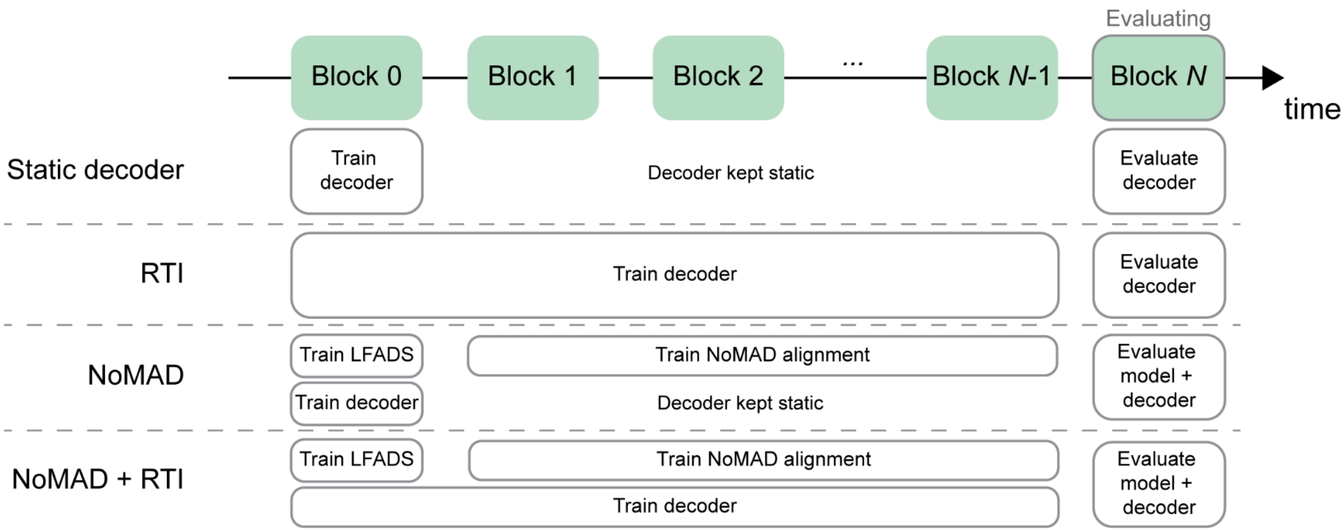
Supplementary Figure 7 | Quantification of the decrease in decoding performance for pairs separated by one day. We collected all pairs of sessions that are separated by one day and measured the drop in decoding R^2 between the initial decoder calibration and the method tested on the following day. (a) Summary of decrease in R^2 for pairs of sessions separated by a single day (*n*=22) for all methods for the isometric force task. Outliers have been removed for visual clarity. (b) Heat map of *p*-values computed using a two-sided Wilcoxon signed-rank test for isometric force results. Gray squares indicate untested pairs due to symmetry. Colored squares indicate values as delineated in the colorbar at right. Significant differences are indicated with an asterisk (* < 0.05, ** < 0.01, *** < 0.001). (c-d) same as a-b but for unloaded reaching task (*n*=16).



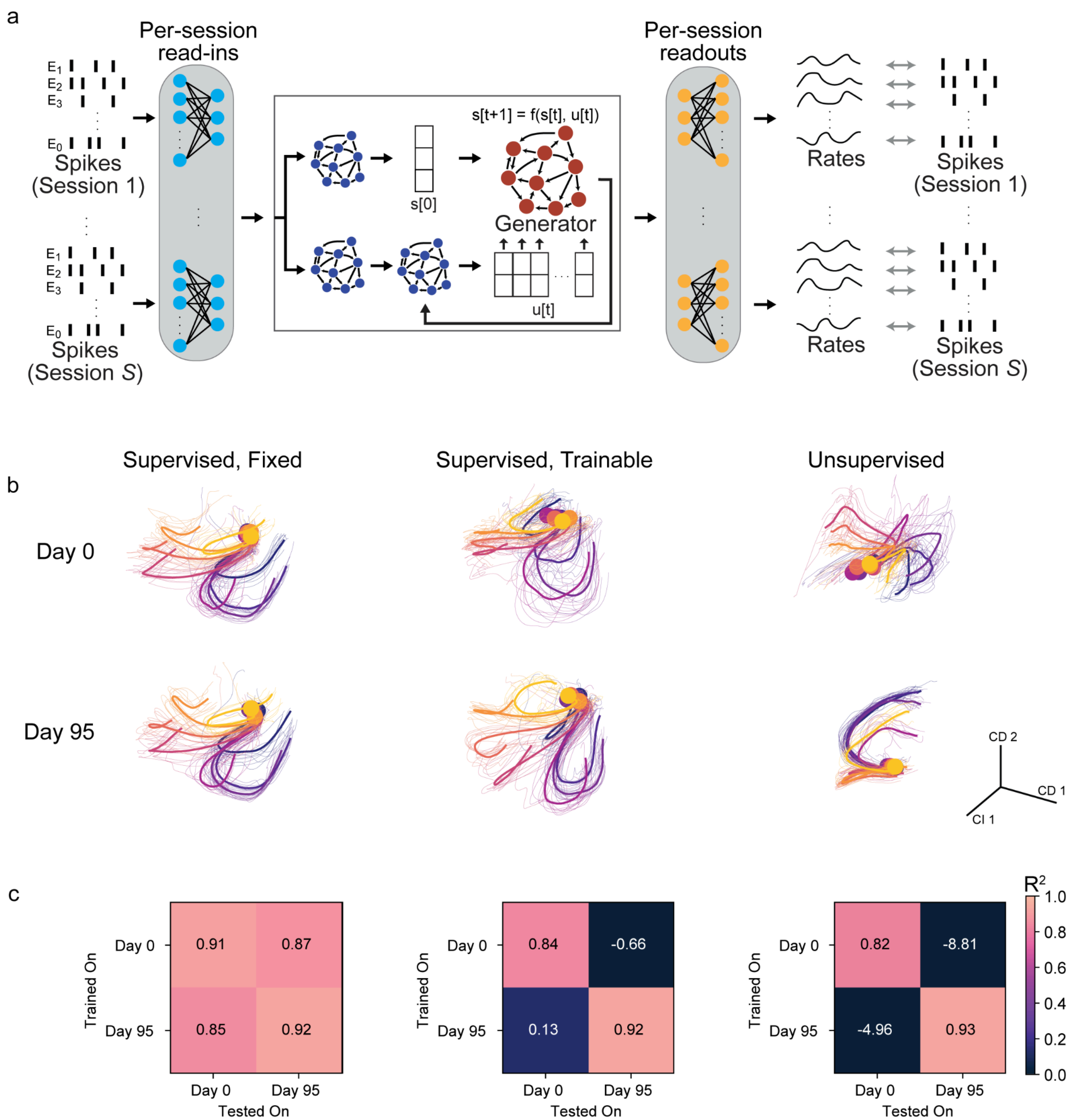
Supplementary Figure 8 | Comparison of NoMAD to fixed TNDM. The LFADS models used in this work are similar to TNDM as both models aim to reconstruct both neural data and behavior. LFADS with the behavioral readout imposes no constraints on how the model achieves this, so long as neural and behavioral reconstruction losses are being optimized. TNDM requires latent variables to be separated into independent behaviorally-relevant and irrelevant latent variables. (a-b) R^2 of aligned force decoding for TNDM (a) and NoMAD (b) applied to all pairs of sessions of the isometric force task (20 sessions spanning 95 days). (c-d) Top: Percentage of Day K decoding failures ($R^2<0$) in 5-day bins. Bottom: Decoding performance as a function of days between sessions. Black points indicate Day K decoding performance for single pairs of days. Red points show the median Day K performance within each bin of width 5 days. Gray points denote initial performance for Day 0 decoders (i.e., evaluated on held-out same-day data) for each of the 20 datasets. Gray dashed lines indicate median decoding performance of Day 0 decoders, and shaded regions around them represent the first and third quartiles. NoMAD significantly outperforms fixed TNDM (p = 1.30e-63 in one-sided Wilcoxon signed-rank test). (e) Median R^2 of force decoding within each 5-day bin. Triangles indicate points that fall below the limit of the y-axis and x-markers indicate median within-day force decoding performance on Day 0. (f-g) Same as (a-b) but for the unloaded reaching task (12 sessions spanning 38 days). (h-i) Same as (c-d) but for the unloaded reaching task. NoMAD significantly outperforms fixed TNDM (p = 1.28e-23 in one-sided Wilcoxon signed-rank test). (j) Same as (e) but for the unloaded reaching task.



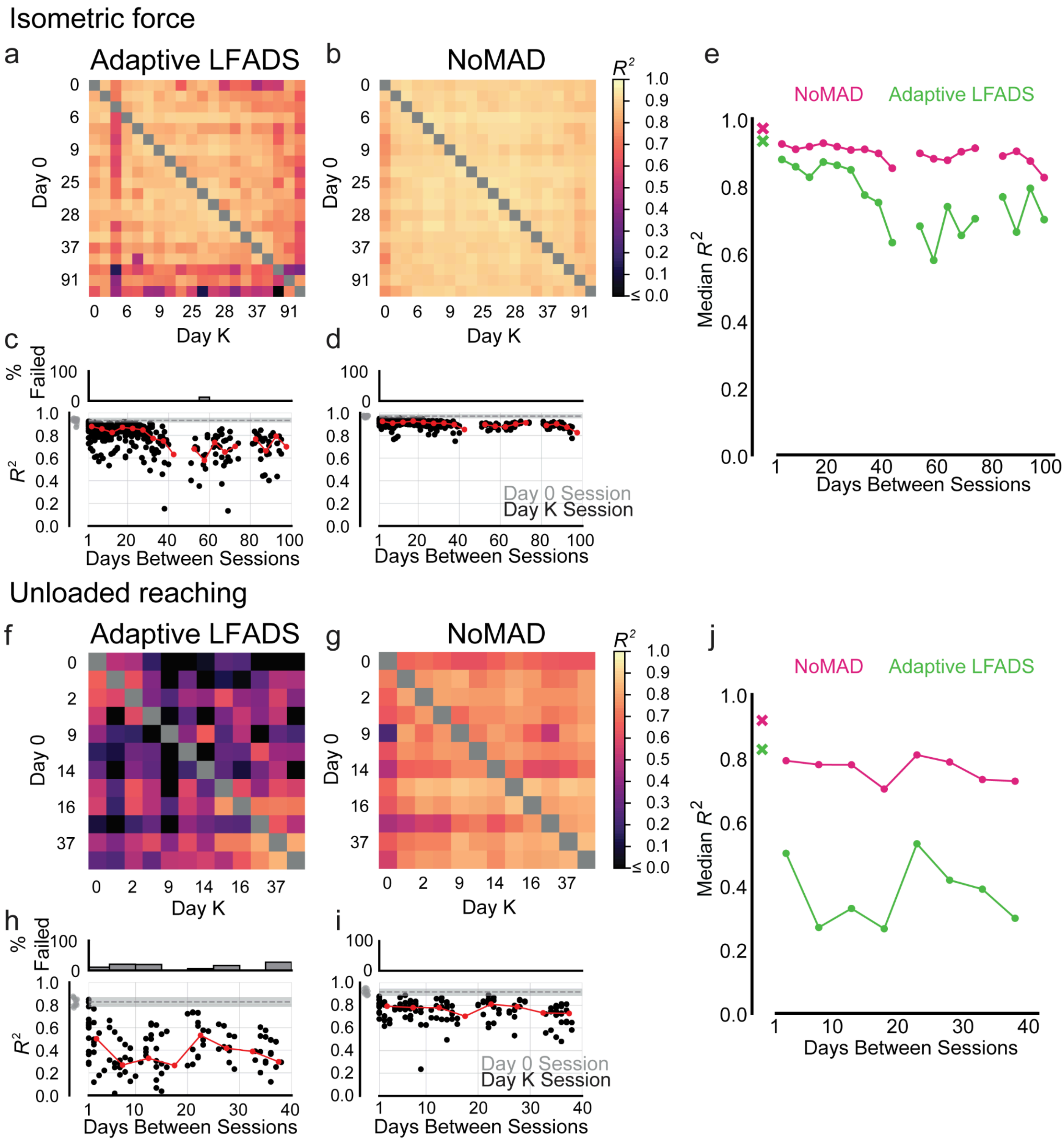
Supplementary Figure 9 | Comparison of Aligned FA applied sequentially versus using a single reference day. In the original Aligned FA demonstration, alignment is applied in a chaining fashion so that each subsequent dataset is aligned to the most recent point ("Sequential" alignment), . This is in contrast to the approach used in this work, where each alignment is performed with respect to a single reference Day 0 dataset regardless of its distance in time from the Day K dataset being aligned ("Single Reference"); each subsequent dataset is always aligned back to the same reference point. (a) Comparison of Aligned FA approaches for isometric force dataset (*n*=20 sessions). Each subplot shows force R^2 over time for either the Single Reference approach (black) or the Sequential approach (red). The initial calibration R^2 is marked using an "x" and the subsequent aligned points are marked with each dot. Points with R^2 below 0 are marked with a triangle on the x-axis. The Single Reference approach significantly outperforms the Sequential approach (p = 1.41e-10 in one-sided Wilcoxon signed-rank test). (b) Comparison of Aligned FA approaches for unloaded reaching dataset (*n*=12 sessions). Conventions the same as in (a). The Single Reference approach significantly outperforms the Sequential approach (p = 3.79e-5 in one-sided Wilcoxon signed-rank test).



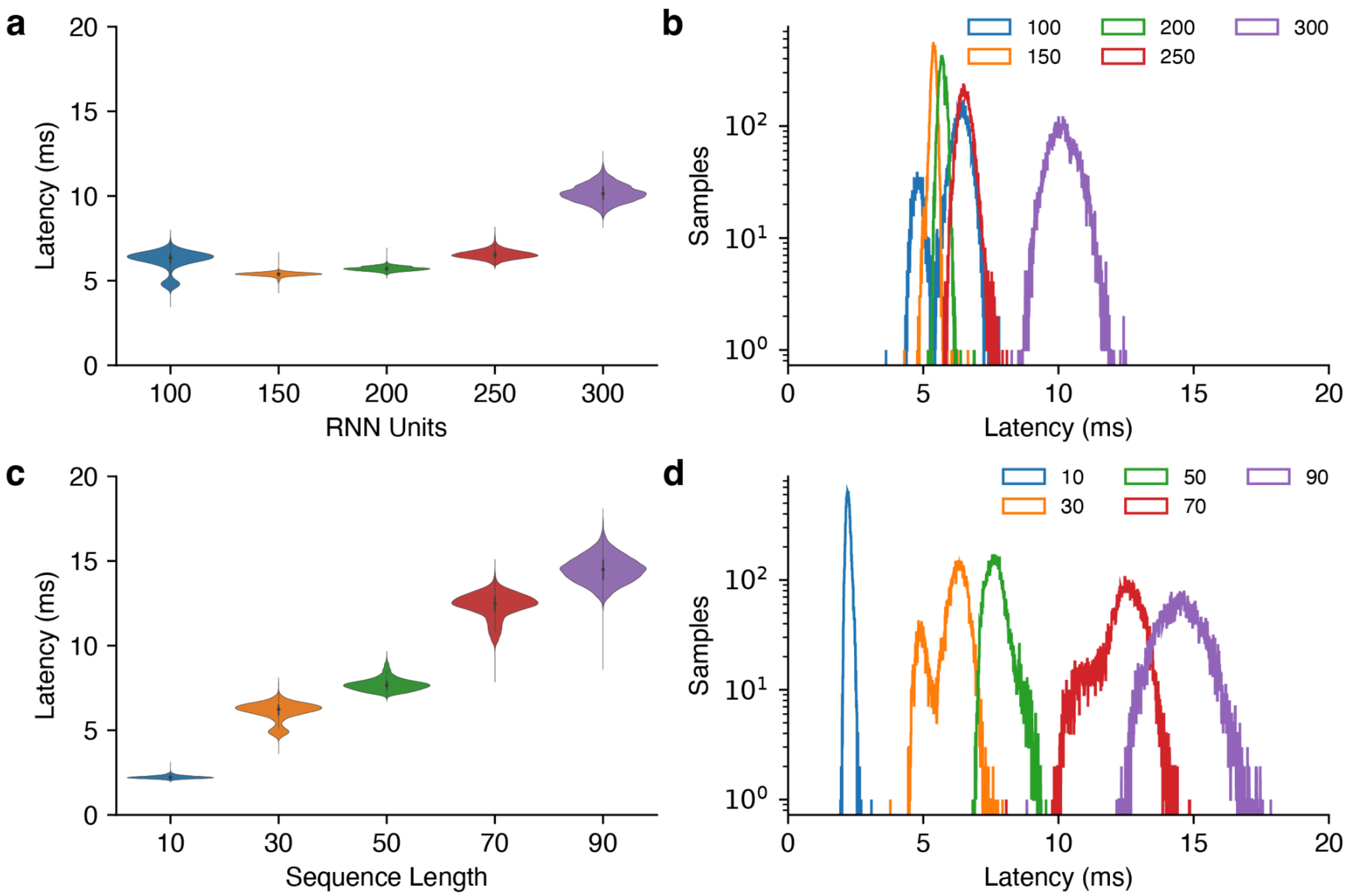
Supplementary Figure 10 | Methodology for applying retrospective target inference (RTI) and NoMAD on closed-loop iBCI data. Schematic of experimental design for various decoding approaches tested. For the Static decoder, we train the decoder on the first block and apply it to the block of interest. For RTI, we train the decoder using all *N*-1 historical blocks and apply it to the block of interest. For NoMAD, we first train the initial LFADS model and decoder on the first block. We then use the next blocks up to but not including the block of interest to train NoMAD alignment. We then apply the model and decoder to the block of interest. Finally, we combine this with RTI to update both the model and decoder before applying to the block of interest.



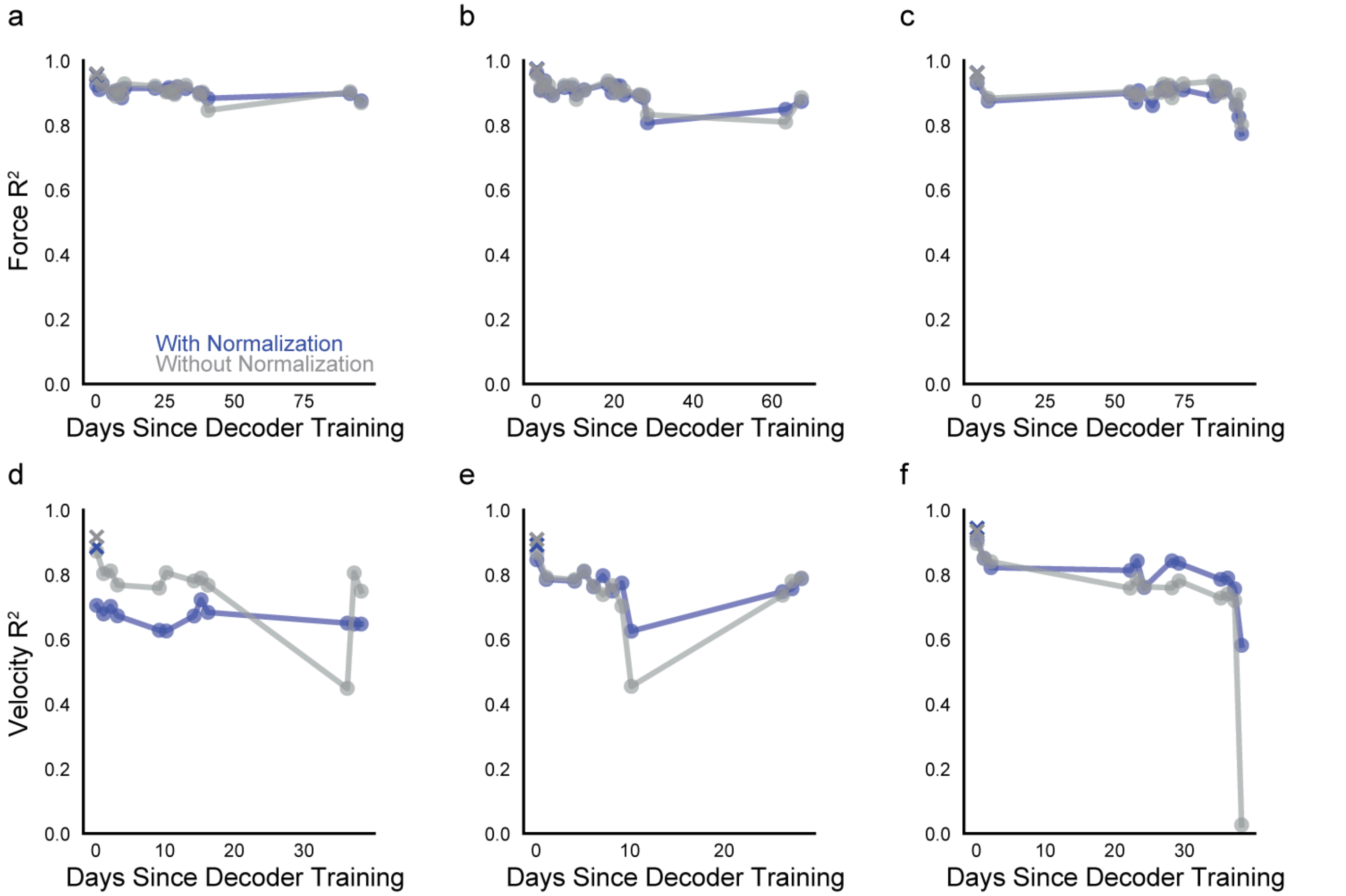
Supplementary Figure 11 | Delineation of LFADS "stitching" procedure and necessity of supervised read-in initialization. (a) LFADS stitching is a method described in Pandarinath et al., 2018[1] that allows a single manifold and set of dynamics to be used to describe many sessions of neural data simultaneously. The method works by initializing a read-in matrix for each session using principal components regression (PCR) to obtain weights that map the condition-averaged data from each session to the shared principal components of all sessions. These read-in matrices are held fixed during training, encouraging the data being input to the LFADS model to approximate a shared space. A single LFADS model is trained to learn the manifold and dynamics across all sessions. This manifold and dynamics are then mapped back out to the neural data from each session using a per-session read-out matrix that is initialized using the PCR weights but allowed to train with the rest of the model. As the PCR step requires us to know information about the behavior in order to obtain condition averages, this model is supervised with respect to behavior and therefore not a suitable solution to the iBCI decoder stability problem. We note that inference for the models in this figure is performed as in the original stitching demonstration[1]. (b) We trained three LFADS stitching models to find the shared manifold and dynamics of Day 0 and Day 95 from the isometric force task: one with PCR initialization as described in (a) ("Supervised, Fixed"), one where the Per-session read-ins are initialized randomly and allowed to train with the rest of the model ("Unsupervised"), and one where the Per-session read-ins are initialized with PCR but allowed to train with the rest of the model ("Supervised, Trainable"). We then fit dPCA to visualize the LFADS factors corresponding to Day 0 and applied these dPCA parameters to the LFADS factors corresponding to Day 95. Colors indicate trajectories for the different target locations. Thick lines indicate the trial average, and thin lines denote single trials (10 representative trials per condition are shown here). In the Supervised, Fixed model, the factors on both days share clear correspondence in structure, indicating that the model successfully identified a manifold and set of dynamics that describes both sessions well. In the Supervised, Trainable and the Unsupervised model, there is a clear difference in the dPC structure between the two sessions, indicating that shared structure was not successfully identified. (c) We train linear decoders from the shared factors to predict both within (trained and tested on the same day) and across (trained and tested on different days) sessions for all model types. Good generalization of the decoder indicates that the shared factors space describes the behavior of both sessions well; poor generalization may mean a lack of shared structure in the learned representation across sessions. Within-session decoding performance is high for both sessions and all models. Across-session decoding performance for the Supervised, Fixed model indicates good generalization. However, across-session performance for the Supervised, Trainable model and the Unsupervised model yields a very low or negative R^2 indicating a strong lack of decoder generalization.



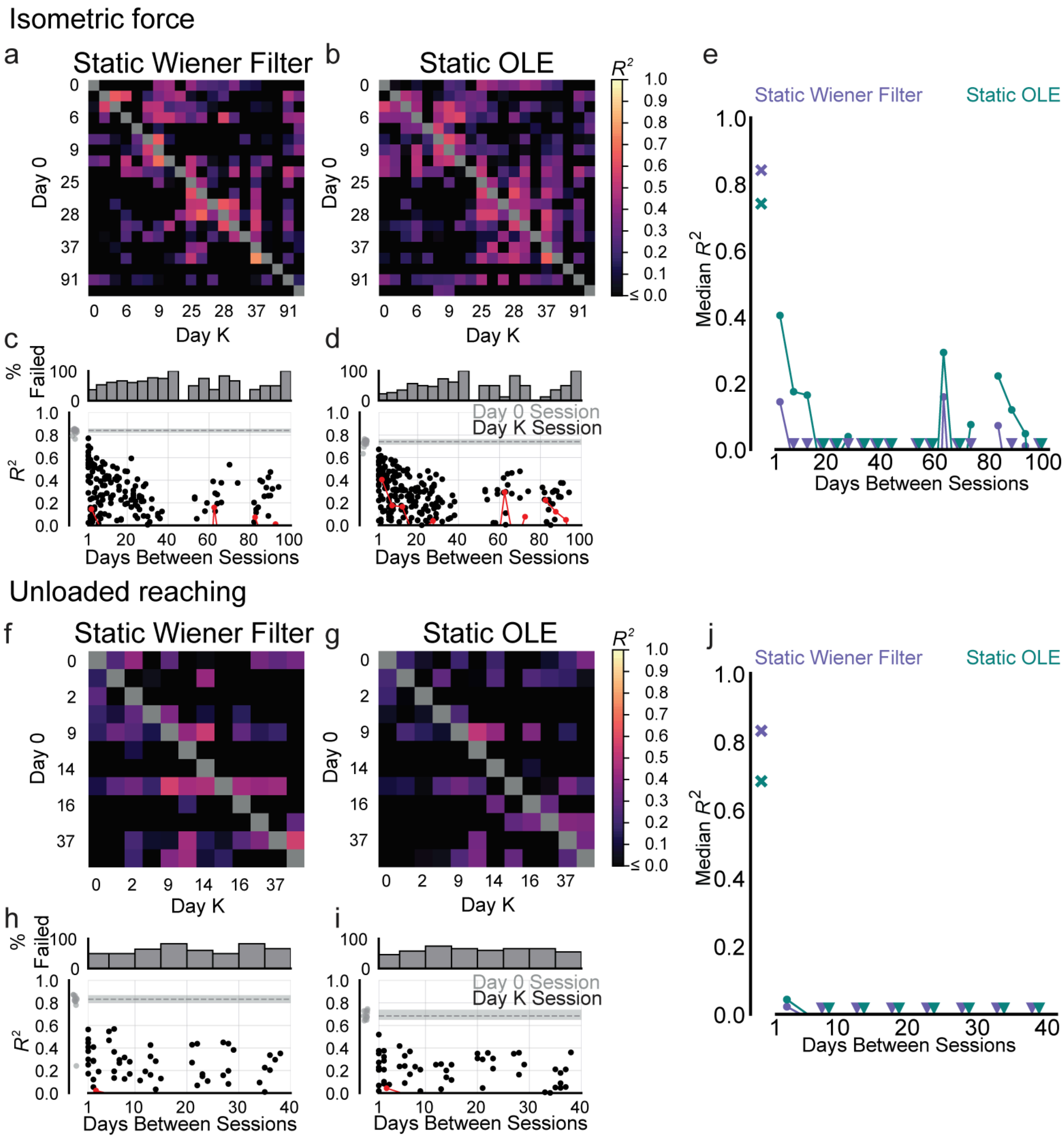
Supplementary Figure 12 | Comparison of NoMAD to Adaptive LFADS. LFADS models were trained without a behavioral read-out on Day 0. Alignment on Day K was performed without the alignment network or KL cost, relying only on reconstruction cost to update the read-in and read-out weights (see *Methods - Adaptive LFADS*). (a-b) R^2 of aligned force decoding for NoMAD with the alignment network and KL Divergence cost ablated (a) and standard NoMAD (b) applied to all pairs of sessions of the isometric force task (20 sessions spanning 95 days). (c-d) Top: Percentage of Day K decoding failures (R^2<0) in 5-day bins. Bottom: Decoding performance as a function of days between sessions. Black points indicate Day K decoding performance for single pairs of days. Red points show the median Day K performance within each bin of width 5 days. Gray points denote initial performance for Day 0 decoders (i.e., evaluated on held-out same-day data) for each of the 20 datasets. Gray dashed lines indicate median decoding performance of Day 0 decoders, and shaded regions around them represent the first and third quartiles. NoMAD models significantly outperform the ablated models (p = 1.99e-62 in one-sided Wilcoxon signed-rank test). (e) Median R^2 of force decoding within each 5-day bin. Triangles indicate points that fall below the limit of the y-axis and x-markers indicate median within-day force decoding performance on Day 0. (f-g) Same as (a-b) but for the unloaded reaching task (12 sessions spanning 38 days). (h-i) Same as (c-d) but for the unloaded reaching task. NoMAD models significantly outperform the ablated models (p = 1.05e-23 in one-sided Wilcoxon signed-rank test). (j) Same as (e) but for the unloaded reaching task.



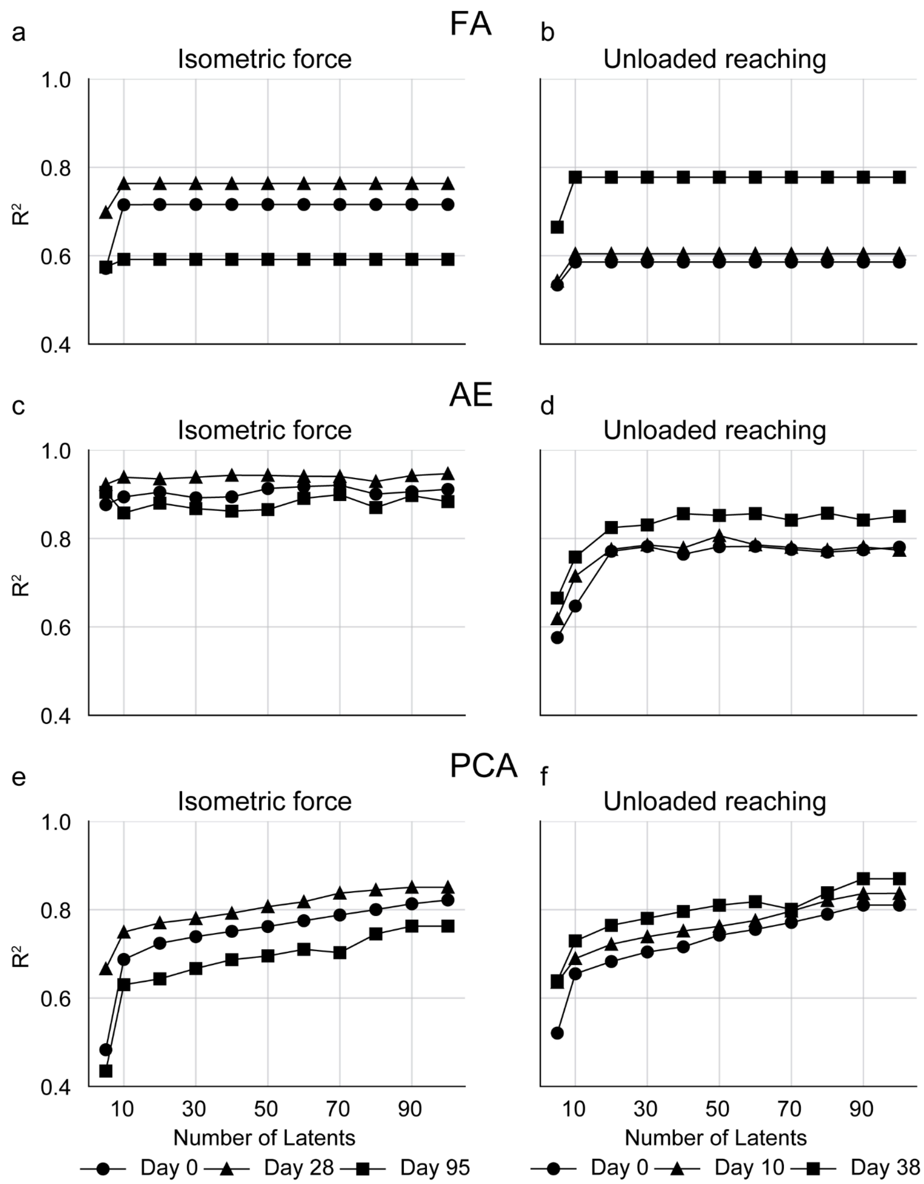
Supplementary Figure 13 | LFADS latency for various RNN sizes and input sequence lengths. Latency was benchmarked by sending 256 channels of sine wave data at a rate of 50 Hz (corresponding to a 20 ms bin size; *n*=12500 samples per architecture setting) into the LFADS Day 0 model (Fig. 2a). For this test, the hyperparameters from the LFADS Day 0 monkey model (Table 2) were used unless otherwise noted. a) When varying the size of the initial condition encoder, controller input encoder, controller, and generator RNNs, latency generally increased as the number of units in each RNN increased. b) Histograms of the latency measurements for each RNN size. c) Latency also increased as the input sequence length increased. d) Histograms of the latency measurements for each sequence length.



Supplementary Figure 14 | Comparison of NoMAD alignment with and without normalization layer. To ensure that the normalization applied to data in the NoMAD model does not unfairly bias the model to outperform other methods, we compare models trained with and without normalization. Models trained with normalization are extracted for three example Day 0 models from Figs. 3-4 ("With Normalization", blue). New Day 0 models were then trained without normalization and NoMAD was used to perform alignment to them, also without normalization ("Without Normalization", gray). We denote the Day 0 model performance with an "x" and each aligned session's performance with a round marker for 3 sessions for each task. For the isometric force task, there is no significant improvement in the performance of models trained with normalization versus those without normalization for alignment to (a) Day 0, (b) Day 28, or (c) Day 95 (p = 0.66, 0.83, 0.96 in one-sided Wilcoxon signed-rank test). For the unloaded reaching task, normalization provided no significant improvement over models without normalization for Days (d) 0 or (e) 10, but there was some improvement for (f) Day 38 (p = 0.98, 0.42, 6.8e-3 in one-sided Wilcoxon signed-rank test).



Supplementary Figure 15 | Comparison of Static Wiener Filter decoder to Static Optimal Linear Estimator (OLE). We compare the performance of the Static Wiener Filter (reproduced from Static decoder in the main text) and a Static OLE, which uses no history and therefore has fewer parameters. (a-b) R^2 of aligned force decoding for Static Wiener Filter (a) and Static OLE (b) applied to all pairs of sessions of the isometric force task (20 sessions spanning 95 days). (c-d) Top: Percentage of Day K decoding failures (R^2<0) in 5-day bins. Bottom: Decoding performance as a function of days between sessions. Black points indicate Day K decoding performance for single pairs of days. Red points show the median Day K performance within each bin of width 5 days. Gray points denote initial performance for Day 0 decoders (i.e., evaluated on held-out same-day data) for each of the 20 datasets. Gray dashed lines indicate median decoding performance of Day 0 decoders, and shaded regions around them represent the first and third quartiles. (e) Median R^2 of force decoding within each 5-day bin. Triangles indicate points that fall below the limit of the y-axis and x-markers indicate median within-day force decoding performance on Day 0. (f-g) Same as (a-b) but for the unloaded reaching task (12 sessions spanning 38 days). (h-i) Same as (c-d) but for the unloaded reaching task. (j) Same as (e) but for the unloaded reaching task. Both decoders exhibit marked instability on both tasks.



Supplementary Figure 16 | Sweep of latent space dimensionality for within-day manifold models. We show the accuracy of the decoded output (R^2) for within-day manifold models fit to the data at various latent dimensionalities. For Aligned FA, we fit factor analysis (FA) models to the data pertaining to three different datasets for (a) the isometric force task (Days 0, 28, 95) and (b) the unloaded reaching task (Days 0, 10, 38) and found that performance saturates at a dimensionality of 10. For ADAN, we fit the initial autoencoder (AE) to three different dates for each of the (c) isometric force task and (d) unloaded reaching task. We find mixed trends across tasks; for the results in this manuscript we use a dimensionality of 10. Finally as a reference, we fit PCA to the datasets for the (e) isometric force task and (f) unloaded reaching task and find that as expected, decoding accuracy continues to increase until the full dimensionality of the data has been reached.

Supplementary Tables

Supplementary Table 1. Behavioral cutoffs for outlier removal.

Monkey	Behavioral Variable	Lower Bound, X	Upper Bound, X	Lower Bound, Y	Upper Bound, Y
Isometric	Force (mV)	-1500	800	-1100	600
Isometric	dF/dt (mV/t)	-100	100	-100	100
Reaching	Cursor position (cm)	0.5	1.5	-1.1	-0.25
Reaching	Cursor velocity (cm/t)	-0.025	0.025	-0.025	0.025

Supplementary Table 2. LFADS Day 0 architecture parameters.

Parameter	Monkey Value	Human Value
Low-D Read-In Output Dimensionality	50	100
Encoder RNN Units (initial conditions and controller input)	100	100
Generator RNN Units	100	100
Controller RNN Units	100	100
Controller Output Dimensionality	4	4
Factors Dimensionality	30	30
External Input Dimensionality	0	0

Supplementary Table 3. LFADS Day 0 model hyperparameters.

Hyperparameter	Value	Hyperparameter	Value
Dropout rate	0.05	Coordinated dropout rate	0.3
Batch size	1000	Sample validation rate	0
Initial learning rate	1e-3	Maximum gradient norm	200
Learning rate decay	0.95	Adam optimizer epsilon	1e-8
Learning rate decay patience	6	Autoregressive constant on controller output prior	10.0
L2 weights on controller and generator	1e-6	Noise variance on controller output prior	0.1
L2 weight on encoders	0	Noise variance on initial conditions prior	0.1
L2 ramping epochs	100	Minimum variance allowed for initial condition posteriors	0.0001
KL weight on initial conditions and controller inputs	1e-4	Max magnitude of GRU hidden state	5.0
KL ramping epochs	100	Controller input lag bins	1
Behavioral readout learning rate	1e-3	Total loss multiplier	1e4
Behavioral readout cost scale	0.1	L2 weight on low-D read-in	5e-3
Behavioral readout cost ramping epochs	100	Sequence length	30 bins

Supplementary Table 4. AutoLFADS Hyperparameter ranges for human datasets.

Hyperparameter	Low Searched Value	High Searched Value
Initial learning rate	1e-5	1e-3
Dropout rate	0	0.6
Coordinated dropout rate	0.01	0.99
L2 weight on generator	1e-5	1e-0
L2 weight on controller	1e-5	1e-0
KL weight on controller	1e-5	1e-3
KL weight on initial conditions encoder	1e-5	1e-3

Supplementary Table 5. NoMAD hyperparameters.

Hyperparameter	Value	Hyperparameter	Value
Initial learning rate	2e-3(M), 1e-4(H)	Maximum gradient norm	2000
Learning rate decay	0.95	L2 penalty weight on low-D read-in	5e-3
Learning rate decay patience	6	Batch size	300
NLL cost weight	10	Early stopping validation loss patience	40(M), 30(H)
NLL ramping epochs	100	Adam optimizer epsilon	1e-8
KL ramping epochs	10(M), 100(H)		
KL weight on initial conditions	1e-4		
KL weight on controller outputs	1e-4		

Supplementary Table 6. Aligned FA hyperparameters.

Parameter	Value
Latent dimensionality	10
Number of FA models to fit	5
Maximum number of EM iterations to fit FA	100000
EM stopping criteria	0.00001
Minimum private variance threshold	0.1
Number of rows of loading matrix to use for alignment	90
Alignment L2 norm threshold	0.01

Supplementary Table 7. ADAN hyperparameters.

Parameter	Value
Decoder latent dimensionality	10
Decoder batch size	64
Decoder epochs	400
Decoder learning rate	1e-3
Decoder steps	4
Decoder layers	1
ADAN epochs	200
ADAN batch size	4
ADAN discriminator learning rate	5e-5
ADAN generator learning rate	1e-4

Supplementary Table 8. TNDM Parameters & Hyperparameters.

Hyperparameter	Value	Hyperparameter	Value
L2 Regularization Weight	1	Update rate	0.0005
Initial Weight of Neural NLL	1.0	Dropout	0.15
Initial Weight of Behavioral Loss	0.2	Variance minimum	0.0001
Lambda q	100	Prior Variance	1
Initial Learning Rate	1e-2	Adam Optimizer Beta1	0.9
Adam Optimizer Beta2	0.999	Adam Optimizer Epsilon	1e-8
Learning Rate Decay Factor	0.95	Learning Rate Patience	10
Minimum Learning Rate	1e-5	Relevant Factors Dim	2
Irrelevant Factors Dim	2	Encoder Dim	64
Max Gradient Norm	200	Batch Size	16
L2 Regularization	1	Relevant initial conditions	64
Irrelevant initial conditions	64	Relevant decoder dim	64
Irrelevant decoder dim	64		

References

1. Pandarinath, C. *et al.* Inferring single-trial neural population dynamics using sequential auto-encoders. *Nat. Methods* **15**, 805–815 (2018).