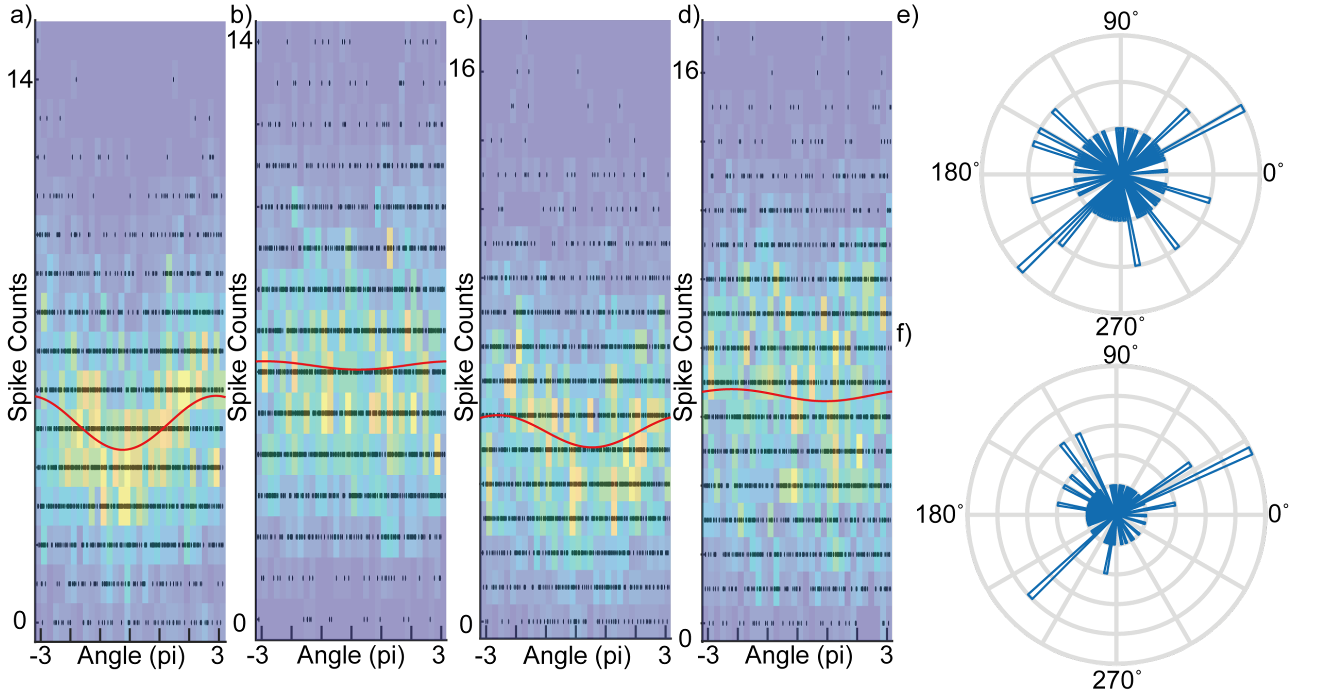
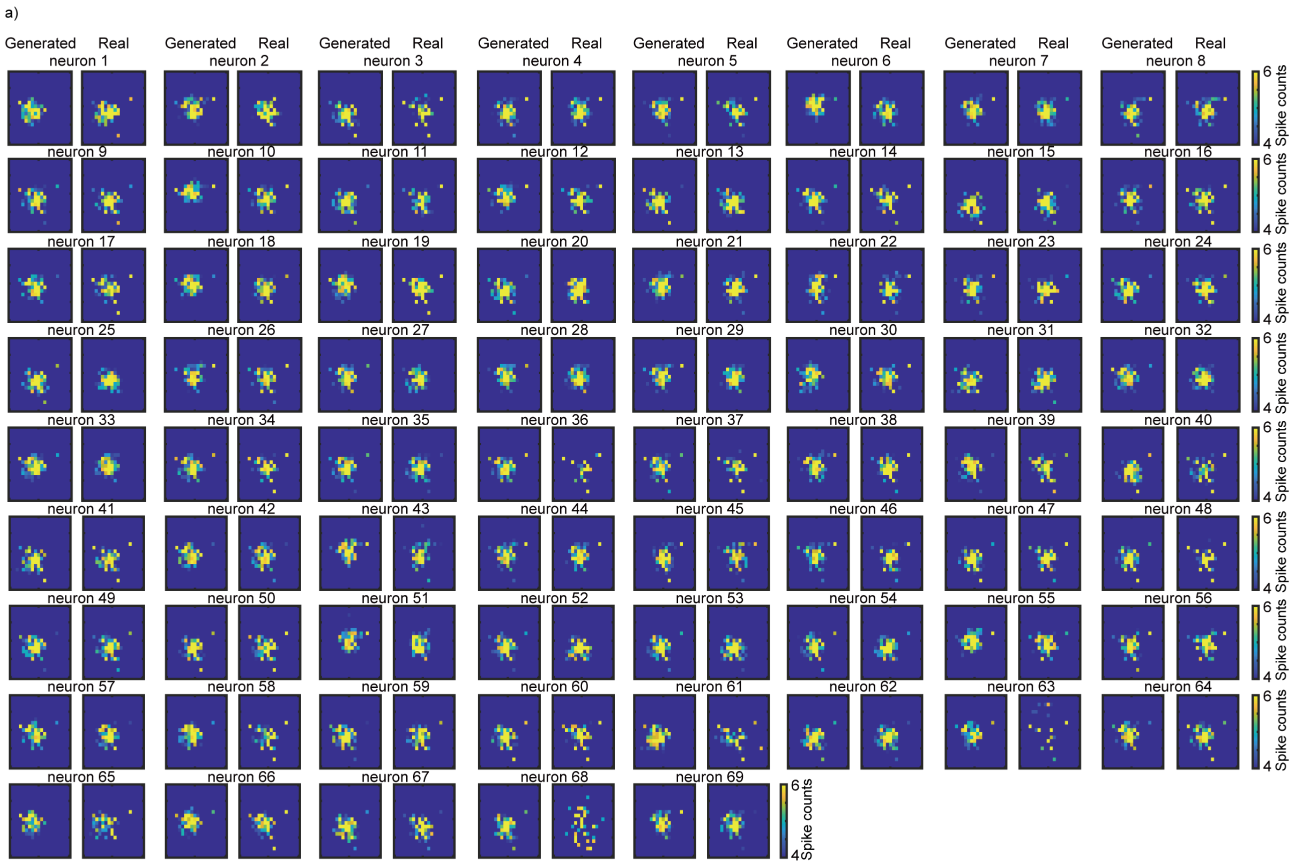
Contents

Supplementary Figures

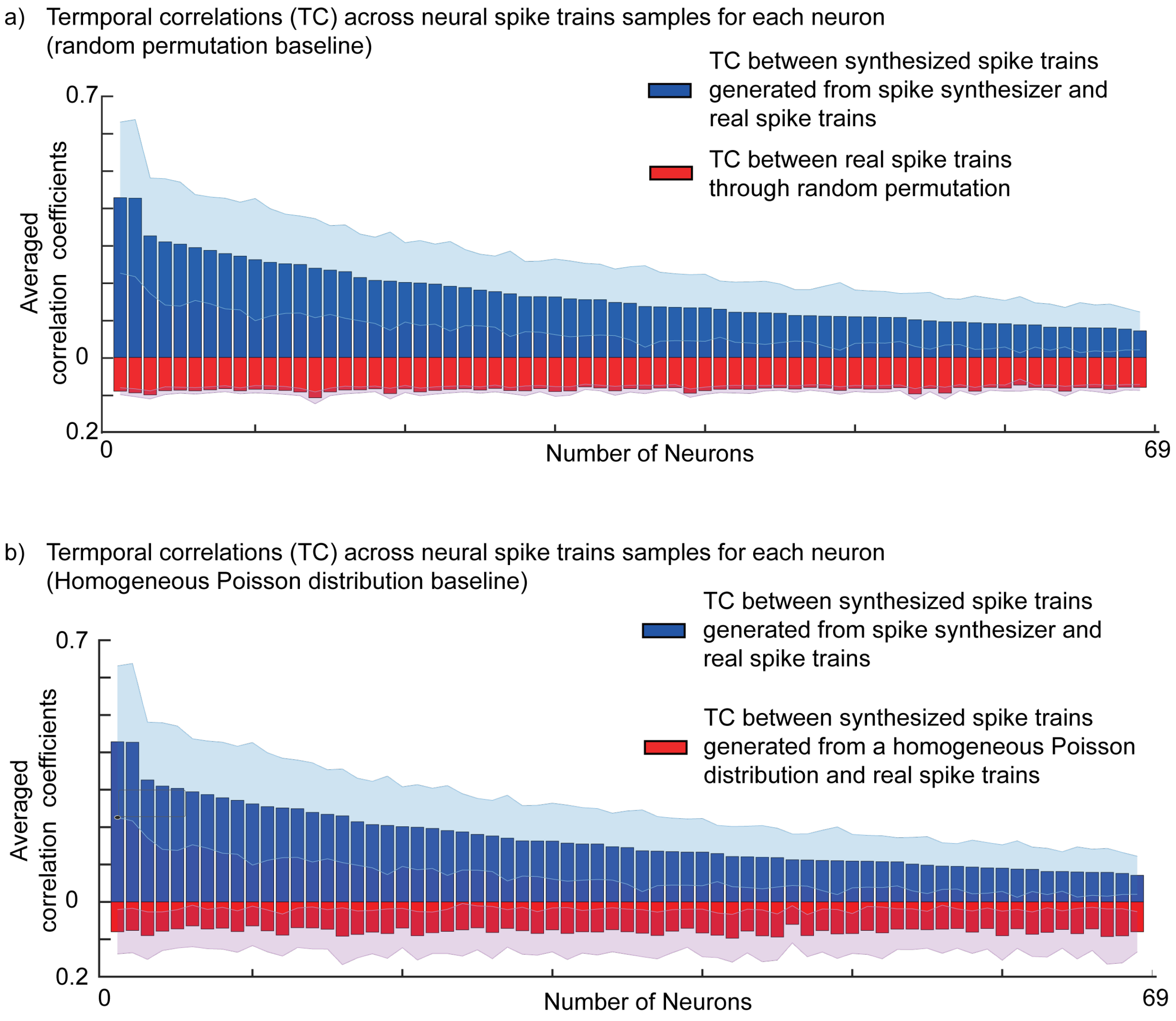
Supplementary Discussion



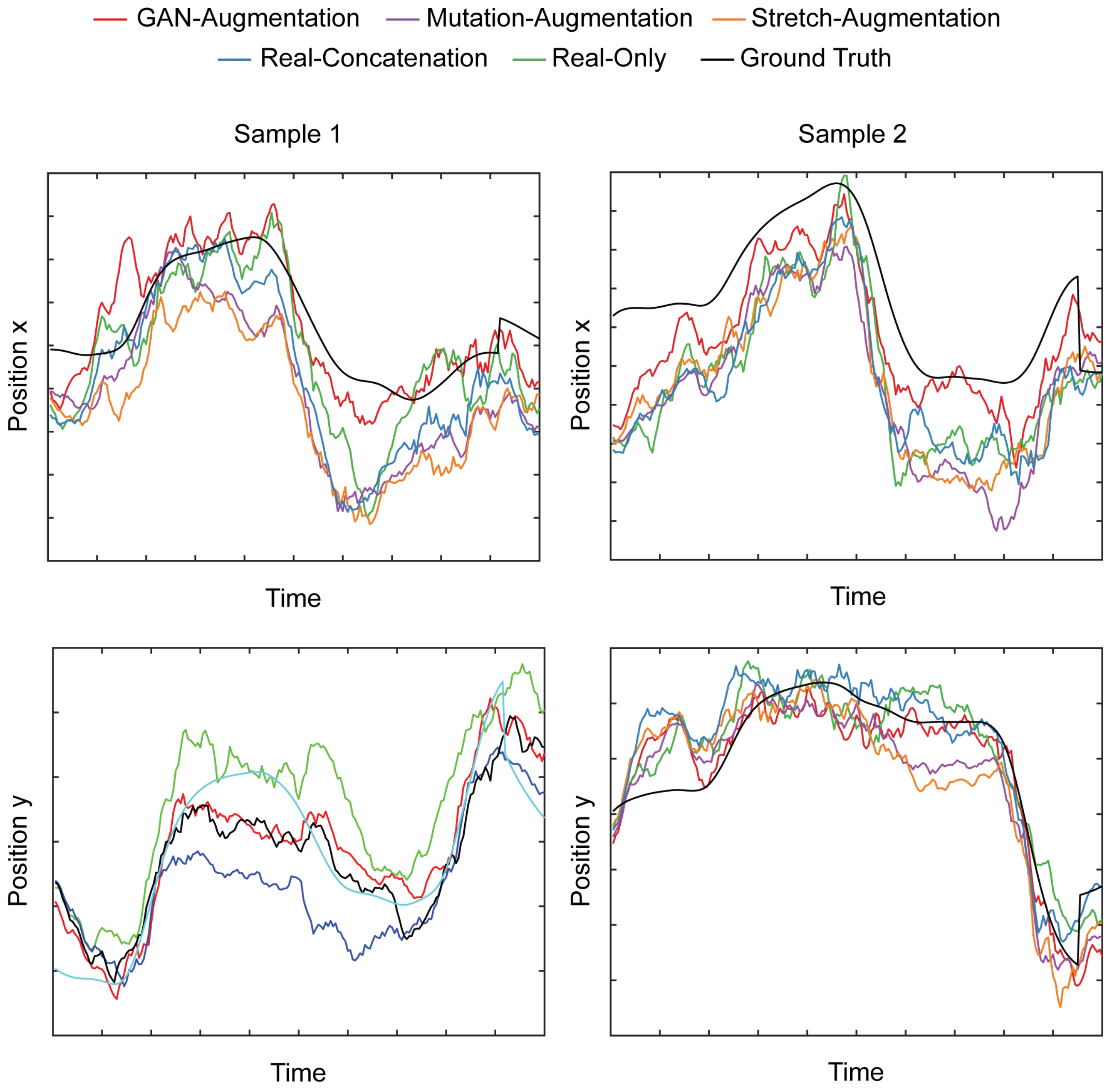
Supplementary Fig. 1 | Velocity neural tuning curves. For (a-d), we calculated the hand velocity direction for each 300ms and calculated the spike counts during that 300ms. We plotted the spikes counts vs hand velocity direction for each real and virtual neuron. The red line is the velocity neural tuning curve fitted by a cosine function. The black dot is the spike counts for each bin at each angle. The heatmap counts how many black dots are in an area. a) real velocity neural tuning curve for neuron 32. b) generated velocity neural tuning curve for neuron 32. c) real velocity neural tuning curve in velocity space for neuron 57. d) generated velocity neural tuning curve for neuron 57. e) histogram of preferred direction for real neurons. f) histogram of preferred direction for virtual neurons.



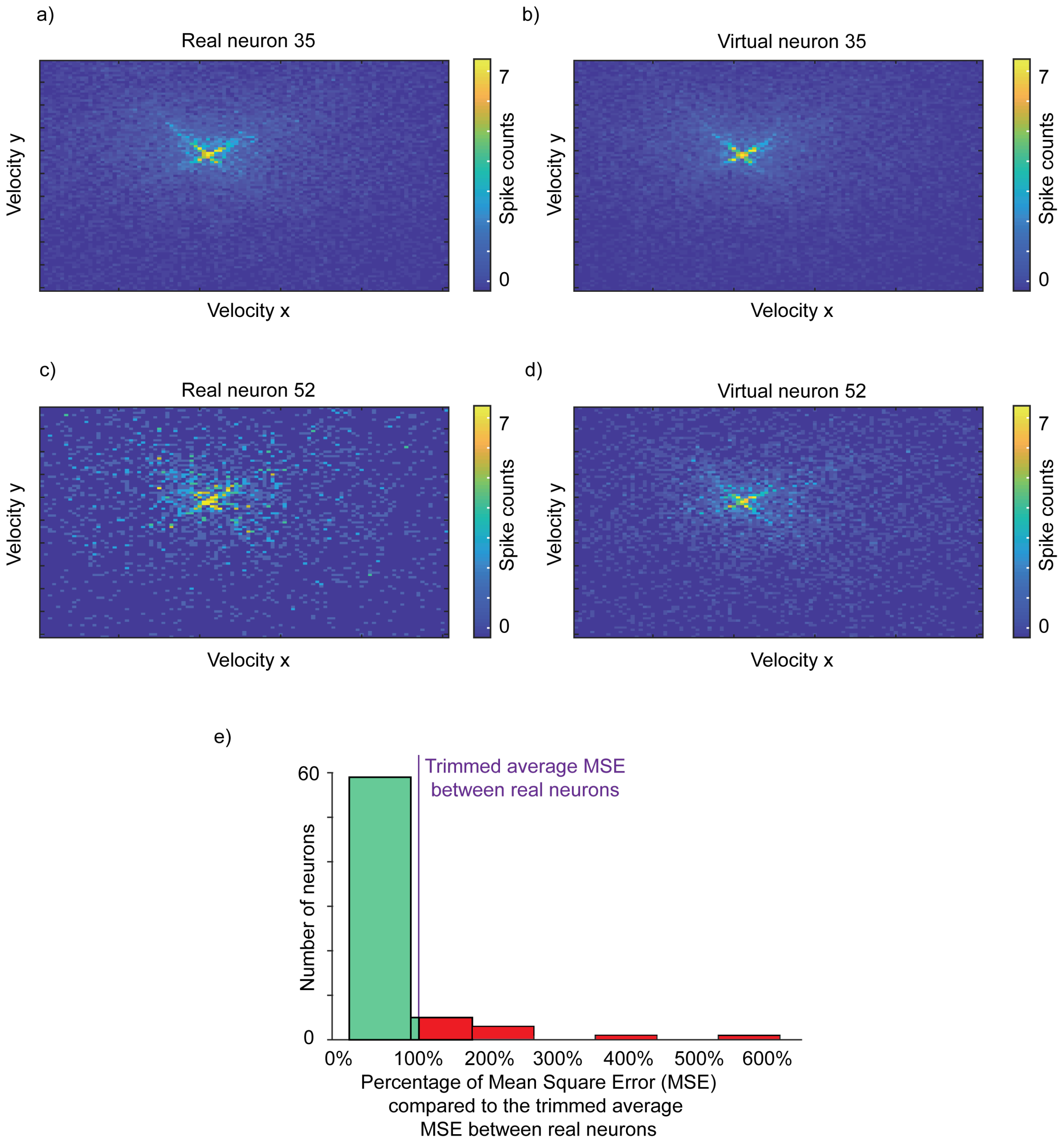
Supplementary Fig. 2 | Position activity maps. a, Position activity maps for all virtual and real neurons with clipped colour bar.



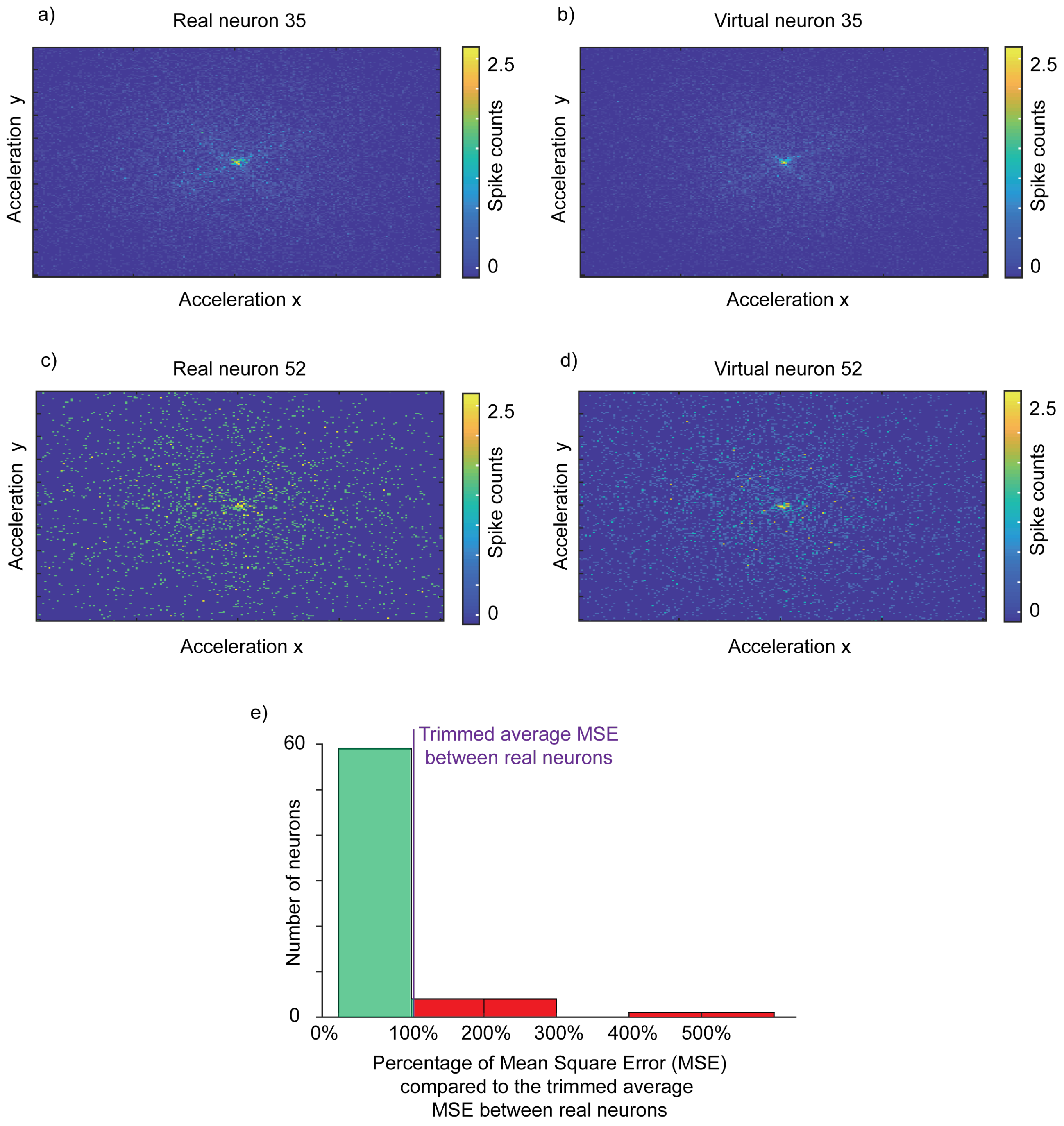
Supplementary Fig. 3 l Correlations across neural spike trains samples for each neuron sorted by the averaged correlation coefficient for each neuron. a, Random permutation baseline. Blue curve is the correlation between synthesized (from the spike synthesizer) and real neural data with shaded blue error bar (mean + / - S.D., sample number = 63). Red curve is the correlation between real spike trains through random permutation with shaded red error bar (mean + / - S.D., sample number = 63). **b,** Homogeneous Poisson distribution baseline. Correlations across neural spike trains samples for each neuron sorted by the averaged correlation coefficient for each neuron. Blue curve is the correlation between synthesized (from the spike synthesizer) and real neural data with shaded blue error bar (mean + / - S.D., sample number = 63). Red curve is the correlation between synthesized neural (from a homogeneous Poisson distribution) and real neural data with shaded red error bar (mean + / - S.D., sample number = 63).



Supplementary Fig. 4 l Visualization examples of actual movement trajectory for GAN-Augmentation, Mutation-Augmentation, Stretch-Augmentation, Real-Concatenation and Real-Only methods compared to ground truth.



Supplementary Fig. 5 l Normalized velocity activity map, constructed as the histogram of neural activity as a function of velocity. a, Velocity activity map for real neuron 35 normalized across the workspace. **b,** corresponding velocity activity map for virtual neuron 35. **c, d,** Velocity activity maps for real and virtual neuron 51. **e,** Histogram of mean squared error between the real and generated activity maps for all neurons. The purple line is the trimmed averaged mean square error (based on 99% samples, 0.11) between real neurons. It provides a reasonable bound for quantifying the difference between real and virtual neurons.



Supplementary Fig 6 I Normalized acceleration activity map, constructed as the histogram of neural activity as a function of acceleration. a, acceleration activity map for real neuron 35 normalized across the workspace. **b,** corresponding acceleration activity map for virtual neuron 35. **c, d,** acceleration activity maps for real and virtual neuron 51. **e,** Histogram of mean squared error between the real and generated activity maps for all neurons. The purple line is the trimmed averaged mean square error (based on 99% samples, 0.10) between real neurons. It provides a reasonable bound for quantifying the difference between real and virtual neurons.

Supplementary Discussion

Stabilization

David et, al.[1] demonstrated that training an RNN decoder from many months of previously recorded data can be more robust to future neural variability. Here, our work is training a spike synthesizer (GAN) to learn good neural attributes from a single session of neural data. Inspired by that paper, our spike synthesizer could be trained with more general neural attributes across multiple sessions and monkeys. With more general neural attributes, the spike synthesizer could synthesize neural data that would enhance the cross-sessions and cross-subjects decoding. Our work differs in at least three aspects:

1) Our spike synthesizer could learn more generalizable neural attributes
2) The decoding approach generalizes to multiple sessions and subjects
3) We can achieve saturating performance with much less historical training data

Alan et, al.[2] demonstrated that a manifold-based stabilizer can help a BCI decoder recover proficient control under different instability conditions such as tuning change, drop-outs or baseline shifts. The authors hypothesized that “even though the specific neurons being recorded may change over time, the recorded population activity reflects a stable underlying representation of movement intent that lies within the neural manifold”. This is a sound hypothesis for the neural data collected from the same subject. However, this approach, which requires a subset of stable electrodes cannot be used across subjects or when the specific neurons change dramatically. Even within a subject, the method requires a substantial number of stable neurons in order to accomplish the realignment. It will fundamentally not work between subjects. In comparison, our spike synthesizer learned general neural attributes and could quickly adapt itself to new sessions or subjects using limited additional neural data.

Possible caveat

We could use large amounts neural data from multiple monkeys and multiple sessions to build a better embedding of GAN. This embedding would be expected to generalize better. Yet, for cross-subject decoding, more abundant neural data might yield lower performance for the already “good” covariates such as pos x and pos y. Since, in our work, the embedding of GAN is learned from only one monkey, just fine-tuning the read-out module is not good enough to synthesize neural data that match the distribution of the other subjects perfectly. Thus, the combination of large amounts of additional neural data with synthesized neural data could yield a lower performance on already good covariates. This caveat could be addresses by training our CC-LSTM-GAN with neural data from multiple monkeys and sessions.

Determine the weights for equation (11) and (12)

To better train a CC-LSTM-GAN, you must assign a big portion of the weights to the GAN loss discriminator and generator losses (equation (11) and (12)). Weight ratios of other properties (e.g., decoder loss or inner product loss) should be small compared to these. The exact weights were optimized using trial and error, as it takes 3-4 days to train the CC-LSTM-GAN, which limited our ability to explore the space of all possible combinations more thoroughly. The exact weights were optimized using trial and error as it takes 3-4 days to train the CC-LSTM-GAN, which limited our ability to explore the space of all possible combinations more thoroughly.

Collecting ground truth of the covariate of interest

It is true that “ground truth” kinematic signals are typically available only from animal (mostly monkey) studies. If it were not possible to construct such decoders without reference to actual movement, the entire field would have no relevance to paralyzed patients. Fortunately, this problem has been solved through the use of “observation” based decoders, trained not on actual kinematics, but on the instructed kinematic trajectory that the patient is asked to mimic[3].

Hyperparameters

Modules	Sample size	Timestep	Hidden dimensions	Training Epochs	Learning rates	optimizer	activation
Generator	128	200	200	4000	0.0006*	Adam[52]	tanh
Discriminator	128	200	200	4000	0.0003	SGD	sigmoid
GANta LSTM Deocder	128	200	200	200	0.003*	Adam	N/A
LSTM BCI decoder**	128	200	200	200	0.003*	Adam	N/A

*with exponential learning rates decay, ** same hyperparameters for all data augmentation methods

References

1. Sussillo, D., Stavisky, S. D., Kao, J. C., Ryu, S. I. & Shenoy, K. V. Making brain-machine interfaces robust to future neural variability. *Nat. Commun.* (2016) doi:10.1038/ncomms13749.
2. Degenhart, Alan D and Bishop, William E and Oby, Emily R and Tyler-Kabara, Elizabeth C and Chase, Steven M and Batista, Aaron P and Byron, M. Y. Stabilization of a brain--computer interface via the alignment of low-dimensional spaces of neural activity. *Nat. Biomed. Eng.* 1--14 (2020).
3. Hochberg, L. R. *et al.* Neuronal ensemble control of prosthetic devices by a human with tetraplegia. *Nature* (2006) doi:10.1038/nature04970.