High-Density Electrophysiological Data Drift Correction by 1-rank Structure

Shijie Gu
Joint Graduate Group in Bioengineering
UC Berkeley - UCSF
shijiegu@berkeley.edu

Abstract

Recent technology development in neuroscience has allowed us to record high-density electrophysiological brain activities over a continuous stretch of time in behaving animals. This high-density data often has motion artifacts, commonly called drift, which are detrimental to downstream analysis. In this manuscript, we propose a method to remove such artifacts. Simplifying the recently published method on the same topic in the field, our formulation explicitly exploits the inherent 1-rank structure of the 2-dimensional corrected data matrix. Our method sees good performances in simulated datasets as well as in real datasets.

1 Introduction: electrophysiological recording and its “drift"

One common way neurons communicate with each other is through electric signals. When the voltage between the inside and the outside of a neuron surpasses a boundary, it gives a signal, called an action potential, or “spike". The spikes occur in an all-or-none fashion, which could be understood as the “1" signal in the binary signaling, with “0" being the resting signal. The shape of the rise and fall of the voltage signal of a spike is stereotyped, which means that a given neuron usually has approximately the same amplitude and time course for all spikes. If we plot each spike by its time of occurrence (t_n), waveform amplitude (A_n), and detected location (c_n) in the detected depth of the brain, then we have a plot on the left side of Figure 1. As we will see, this stereotypical spike amplitude for each neuron is the basis for the 1-rank structure of the recordings.

Neuroscientists obtain such spikes in the brain by first recording raw electric signals. This is typically done by inserting electrodes into the brain, and voltage is continuously recorded. Then, “spike sorting" algorithms are applied to the raw data to extract spikes for downstream scientific analysis. A typical “spike sorting" algorithm first detects spikes by some thresholding on the raw voltage data and then clusters spikes with similar spike waveforms and detected locations into one putative neuron.

One challenge during the clustering processes in spike sorting is that it is motion sensitive. Imagine two neurons that have very similar waveforms but are slightly shifted along the recording sites at slightly different depths in the brain. If the electrode drifts up, the bottom neuron will take the place of the top neuron. Since the two neurons have similar waveforms, it is likely that the spike sorting algorithm could mistake the spikes of the bottom neuron at this time as the spikes of the top neuron before the drift started. On the other hand, the bottom neuron’s signal would be thought to be abscent by the spike sorting algorithm. The right side of Figure 1 gives a real example of some neurons’ activities missing that appear due to motion.

This motion in recording elecctrodes has been a challenge in eletrophysiological recordings. Scientists had been restricted to analyze a short period of time during which motion is minimized.

Recent years’ development of high-density electrophysiological recording devices, such as the Neuropixel probes ([1]) has significantly increased the channel density. We now have access to dense

Preprint. Under review.

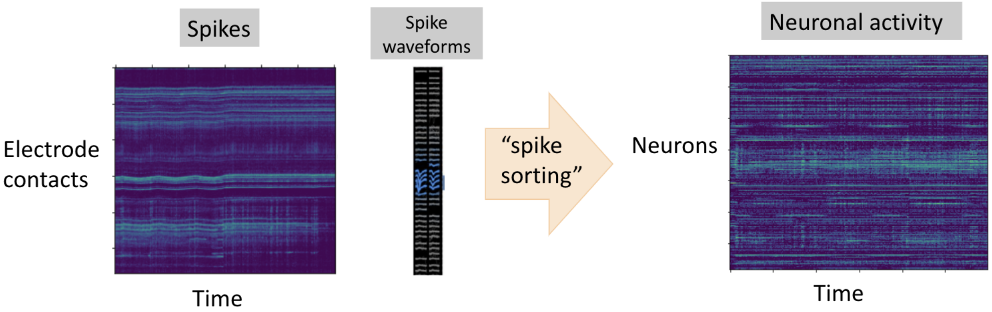

Figure 1: Background. **Left**: Spikes detected from raw data. Each data point in the figure is a “spike" (see text), described by three numbers: time, position and amplitude. Each spike is plotted in row as its position (as the location of its best detected channel), in column as its occurrence in time, and in color scale its amplitude. Drift manifests itself from the horizontal“wiggles" in the plot. **Right**: “Sorted" data are the kind neuroscientists often analyze. The effect of motion in the raw data is shown by some neurons’ activities periodically appearing and disappearing.

enough data to correct for motion artifacts. These devices, often called “probes" have densely packed recording sites along an axis. Intuitively, when the channel density is low, if the brain is shifted along the axis of the electrode, we will obtain a totally new set of data from another set of neurons at all channels, and therefore would have almost no information as to what motion occurred. The high-density recording devices allow us to infer the global motion from data across channels. We illustrate what this motion might look like and detail on how we infer the motion in subsequent sections.

2 Problem statement

In Figure 1, we plot all the spikes detected in a 10-minutes long recording. We see that at the beginning of the data, the data show strong horizontal bands, which are likely from well-isolated neurons without any motion. These bands, however, turned into wiggles later, which are manifests of the drift motion: the original channel-neuron correspondences are somewhat shifted. Importantly, the motion is cohesive enough that we can get a rough sense of what the motion might be by simply looking at the figure thanks to the high-density channels. The goal of the drift correction or drift tracking is to infer this motion that is either consistent across the whole probe depth – rigid estimation, or varying across the probe – nonrigid motion, and correct for this motion in the data so that in the downstream spike sorting stage, minimal mistakes (see above) are made. To express this problem in mathematical notations, we want to estimate the shift amount $\{d(k,t)\}$ for each recording channel k at time point t.

Algorithm 1 Drift Correction Overview

```
    Input: Detected spikes $\{(t_n, c_n, A_n)\}$ from raw data        ▷ See Figure 1
    Output: Shift amount $\{d(k,t)\}$ for each recording channel $k$ at time point $t$.
```

3 The current method

Although the signals are time series, the latest method to correct for drift is by first chunking data (e.g. 1 second) into intervals of time and then transforming each chunk into a 2-dimensional histogram of location and amplitude, followed by applying image registration techniques on them. ([2]).

Specifically, the method optimizes the following problem:

$$\min_{d,F} \sum_t \|F - f_t(d_t)\|_F^2 \tag{1}$$

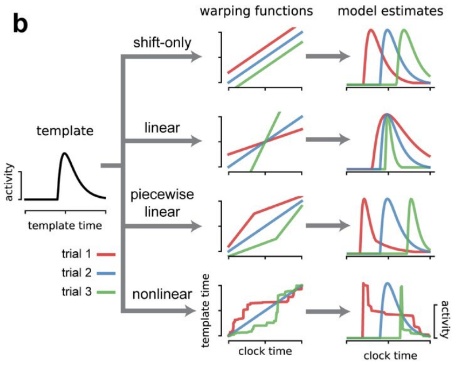

Figure 2: Illustration of different warping functions and their effects. The figure is taken directly from Figure 1b of [3].

where $\|\|_F$ denotes Frobenius norm. f_t is the 2-d histogram formed at time bin t, F is the inferred template, and d_t is the shift for the histogram at time t.

The algorithm iteratively optimizes d or F, holding the other variable fixed. In solving for the template F, the solution is simply the mean of all the current estimate of drift-corrected version of templates. In solving for rigid motion, the algorithm iteratively refine the best shift through greedy search. For nonrigid motion, the algorithm divides the whole length into equally spaced parts and finds the best rigid shift for each part, followed by interpolation to stitch together the whole length of the probe.

4 The proposed change of formulation

This recently published method is practical and gives good performance. Here we propose a couple of changes and a simplified 1-dimensional model that captures the essence of the above method, while adding some extra flexibility.

4.1 Estimating the template: rank-1 approximation

If data is reshaped into a 3-dimensional tensor with each chunk tiled against each other in the 3rd dimension, the above method can be formulated almost as solving for “1-rank structure" of the corrected data tensor and the necessary motion correction simultaneously. The above method actually enforces a stronger requirement than “1-rank structure", where the first principle component of the tensor is described by one vector along each dimension. In the above method, the scalar elements in the vector along the dimension of time are forced to be of the same number, therefore a stronger constraint. We could relax this constraint by only asking for a rank-1 approximation of the tensor.

4.2 Estimating motion: warping functions

As mentioned above, the current method only estimates translational motions, either globally or locally. However, we could allow *stretching* and *compressing* to be inferred as well. Shifting, stretching, and compressing can be expressed by linear functions taking original integer time indices and outputting the warped ones. For completeness, we have included a diagram here (Figure 2, taken from Figure 1b of [3]). Given its richness and that nonlinear models such as that given by DTW algorithms ([4]) are often over-fitting to noise ([3]), we are only considering piece-wise linear functions in this work. When referring to piece-wise linear functions with n different functions, we refer to them as “pwise-(n-1)" models.

Because the output time indices might not be integers, we will need to interpolate between integer data points. In this work, we use linear interpolation. This extra interpolation process is also necessary in the above method when stitching multiple parts of the histogram together.

5 A simplified 1-dimensional model with proposed changes

In this manuscript, I explored the possibility of attacking the problem directly from the alignment of 1-dimensional data at each time point (one column of data in Figure 1) but still simultaneously estimating the motion through 1-rank approximation. There are reasons why directly working on locally aggregated 1-d data might be better: 1) it is an average of the histogram in the amplitude direction, and could offer higher signal to noise ratio, 2) it is more memory efficient.

Specifically, given a spike plot data X similar to that plotted in Figure 1, I solve the following optimization problem

$$\min_{V,U,d_t} \sum_t ||X_{:t} - v_t U(d_t^{-1})||_2^2 \iff \min_{V,U,d_t} \sum_t ||v_t U - X_{:t}(d_t)||_2^2 \tag{2}$$

where U can be thought of as an inferred 1-dimensional template, v_t is the gain modulator at time t, and d_t is the invertible time warping function with its inverse being d_t.

This formulation reveals the similarity between our simplified formulation and the proposed 2-d histogram version in equation 1. Particularly, without the "gain modulator" v, then the model could be considered as a direct simplification of the problem to 1-dimension and is a 1-d time warping problem. See Figure 5 for a comparison between the models.

5.1 Related literature with similar formulations

If we denote an auxiliary variable $E = X_{:t} - v_t U(d_t^{-1})$, then the problem can be written as:

$$\min_{V,U,d_t,E} ||E||_F^2 \tag{3}$$

$$\text{s.t. } X_{:t} = v_t U(d_t^{-1}) + E_{:t} \tag{4}$$

This is reminiscent of the formulation of image alignment literature in the field computer vision. For example, in RASL [5], to find alignments of natural images like faces, the authors formed the following problem:

$$\min_{A,d_t,E} \text{rank}(A) + \gamma ||E||_1$$

$$\text{s.t. } X_{:t} = A_t(d_t^{-1}) + E_{:t}$$

where matrix A can be understood as a concatenation of the aligned images, each reshaped into a column vector.

If we enforce the matrix A in above to be rank-1, then we can immediately see that our formulation and the RASL formulation are the same, with different norms of the error being used in the objective function. With this, we also implemented the norm-1 objective for error minimization, but have so far only focused on characterization of the norm-2 version of the algorithm as most of the time we found that both give similar results.

5.2 Solving the problem through coordinate descent/iteratively solving for U, v, and d

Similar to [3] and [2], we solve the problem through coordinate descent/iteratively solving for the one-rank approximation of the original data and d while holding the other variable. In holding the warping function, we can solve for U and v simultaneously as detailed below. If the objective value landscape with respect to warping is smooth, then we could linearize the warping function (as in [5]) and this iteratively solving schema would be very similar to ADMM (the alternating direction method of multipliers) as the problem would then be convex. However, as we will see, we could not linearize the warping function.

5.3 Solving for the template

Holding the warping function, the objective 5.1 is convex in both the gain v and the template U. In addition, solving for v and U amounts to solving for a rank-1 approximation for the warped data X

by (By the Best Rank-r Approximation Theorem). Without the gain term v, the template U can be simply solved as the mean of currently warped data across time.

5.4 Solving for the warping

To find a method for solving for warping, we first explore the landscape of the objective values if we change the warping. To be able to visually inspect the effect, we start with a simple 2-knot piecewise warping model, where we change the knot location (x,y locations) and calculate its loss with ground truth. The ground truth data is simulated through first generating a template through i.i.d. exponential distribution (with mean of 1), and then warped with a 1-knot piecewise linear model. This gives us a 2D colormap showing the objective values across knot locations.

We explored the landscape as we vary simulated data sparsity and smoothness (Figure 3). The simulated data is generated as follows: we first pick one single time's data (corresponding to one row as in the left side of Figure 1). Then for each data point we run a Bernoulli(p) with p being the probability of data point being dropped, following by various levels of Gaussian smoothing, with its standard deviation noted in the figure subplot titles.

We see that all the landscapes are complex. Note that we can generally see the slopes of the two optimal/ground truth linear functions when data is smooth, but this pattern is less obvious when data is not smooth. When data is "spiky" (nonsmooth and sparse), there is lack of gradient, and optimal values exist in sparse holes.

Real data (see section 7) in our context belongs to this "spiky" category with 1 second time averaging. (Figure 4) This prompted us that we could only use greedy algorithms to find the best warping. To this end, we have only used the greedy search proposed in [3]. Essentially, in each iteration, we perform a perturbation around the current estimate of knot locations. The perturbation magnitude is proportion to the temperature which decreases exponentially over iterations.

6 Characterizations of the simplified 1-dimensional drift correction model with simulated data

In this section, we characterize the simplified 1-dimensional drift correction model with simulated data. We consider the following models: 1) (shift-only) models that only model rigid motion or shifting, 2) (pwise-1) warping models can have up to 2 piecewise warping functions joined at 1 point and without the gain v–v is forced to be 1, 3) (pwise-2), 4) (pwise-2), 5) (r1-pwise-1) similar to r1-pwise-1 but with the gain v term, 6), (r1-pwise-2), 6) (r1-pwise-3). See the preamble in Section 5 for the formulations of the warping models, and Figure 5 for an illustration.

As in the section 5.4, to simulate data, we warp one single column of real spiking data (as in Figure 1) by different 2-knot piece-wise warping functions to produce different rows for a matrix of data. It is worth noting that here we are generating each rows independently. This is different from the real data as real data often look more continuous in time (row). However, this difference does not matter for testing the models as this continuity is not considered by the model.

6.1 Robustness to i.i.d. noise

We further process the generated data by adding iid Gaussian noise with standard deviation equal to 0.2, 0.4, 0.6 and 1 of that of data.

We plot the performance of various model's in Figure 6. Note that we are now plotting data's transpose as in Figure 1.

We see that all piecewise models overall work well under various noise conditions. Rank 1 models seem to be able to be able to denoise better. For example, all the rank-1 models capture the smaller valued peak at round time point 2000 better than their plain piecewise counterparts. Further, the templates (U) that they inferred do not have negative values (Figure 7), which are true for ground truth. The negative values in the inferred template for other models could be thought of as over-fitting noise.

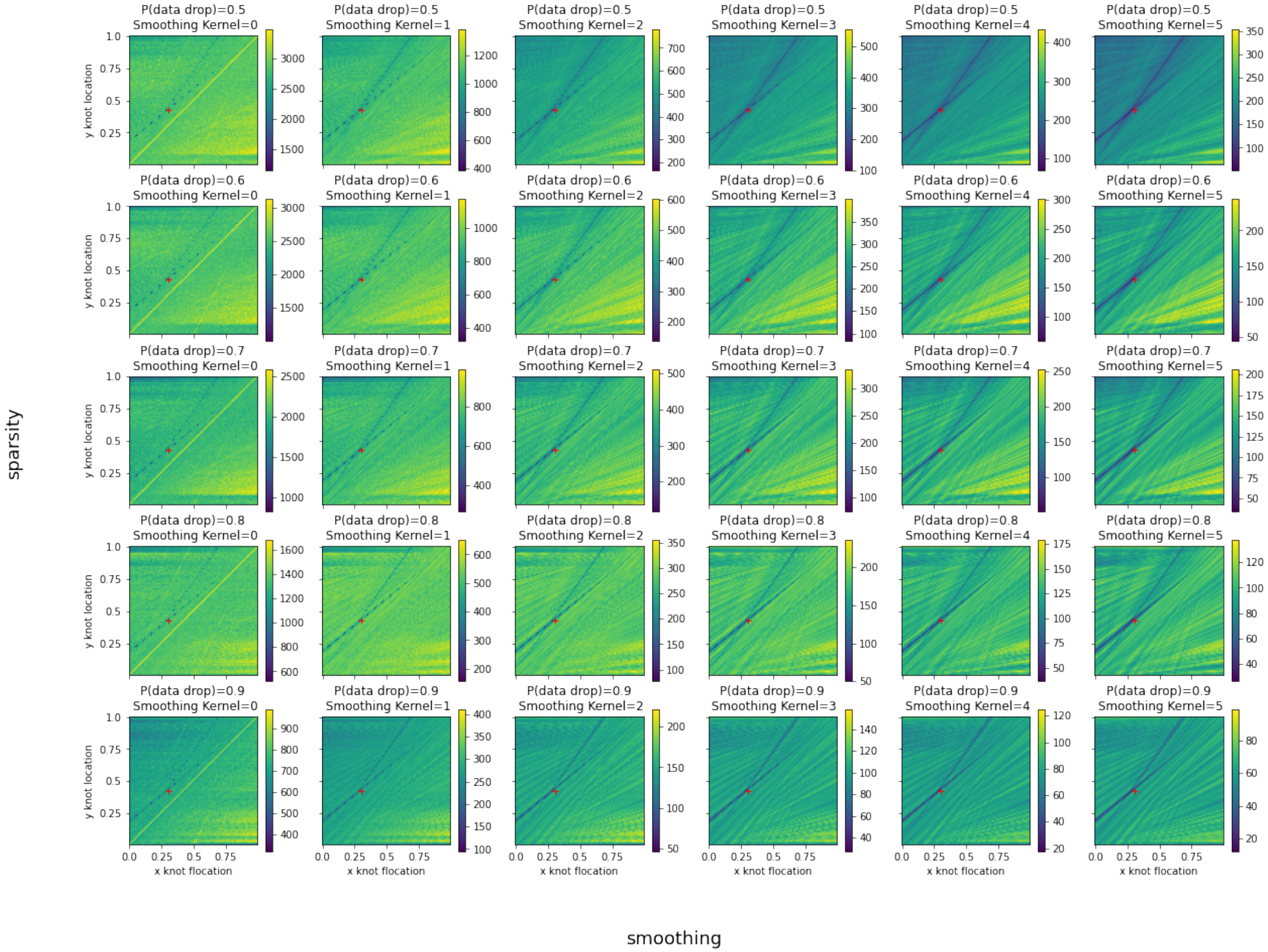

Figure 3: Objective value landscape with respect to one single “knot" in the warping function under various data sparsity and smoothness conditions. The data to warp starts from a generated template, then warped with a pwise-1 warping model (kept as ground truth), and then processed (data dropping and smoothing, see text for details). The ground truth best warping knot is denoted by the red cross.

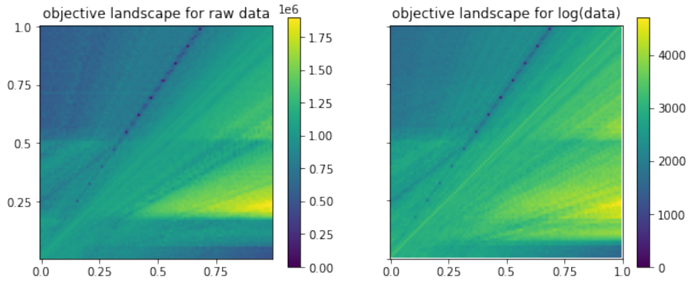

Figure 4: Objective value landscape with respect to one single “knot" in the warping function with real data. The data to warp starts from a single column of real data as in Figure 1.

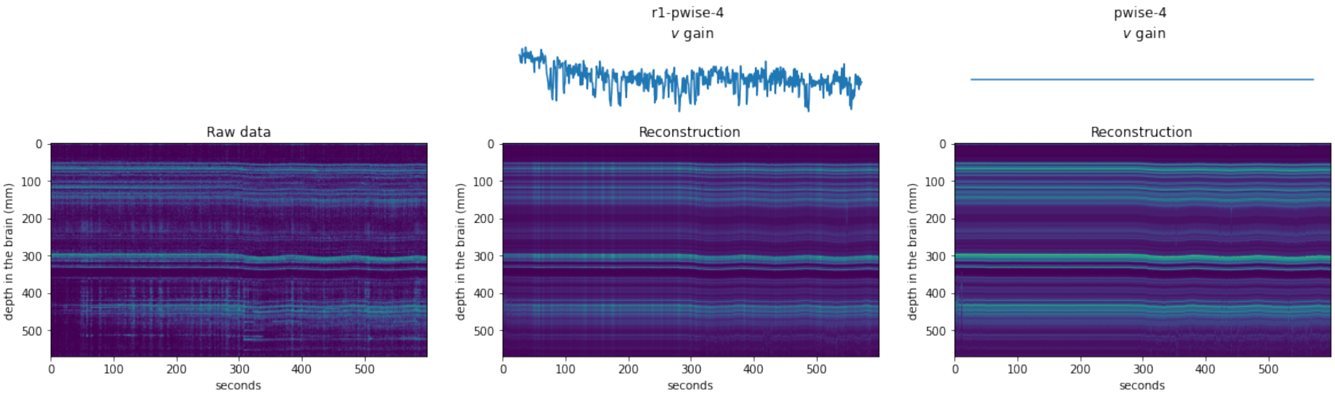

Figure 5: **The difference between model “pwise-r" and “r1-pwise-r"**. “r1-pwise-r" has a *gain* term v that modulates the template. The gain term helps model the sudden change in neural signal strength, capturing abrupt global change in signal. Note that the peaks of the gain term coincides with abrupt changes in the signal. This gain term is equivalent to the projection of rows of the data onto the first principle component. The inclusion of this term in the fitting model is intended to help reduce the contribution of these moments to the loss function.

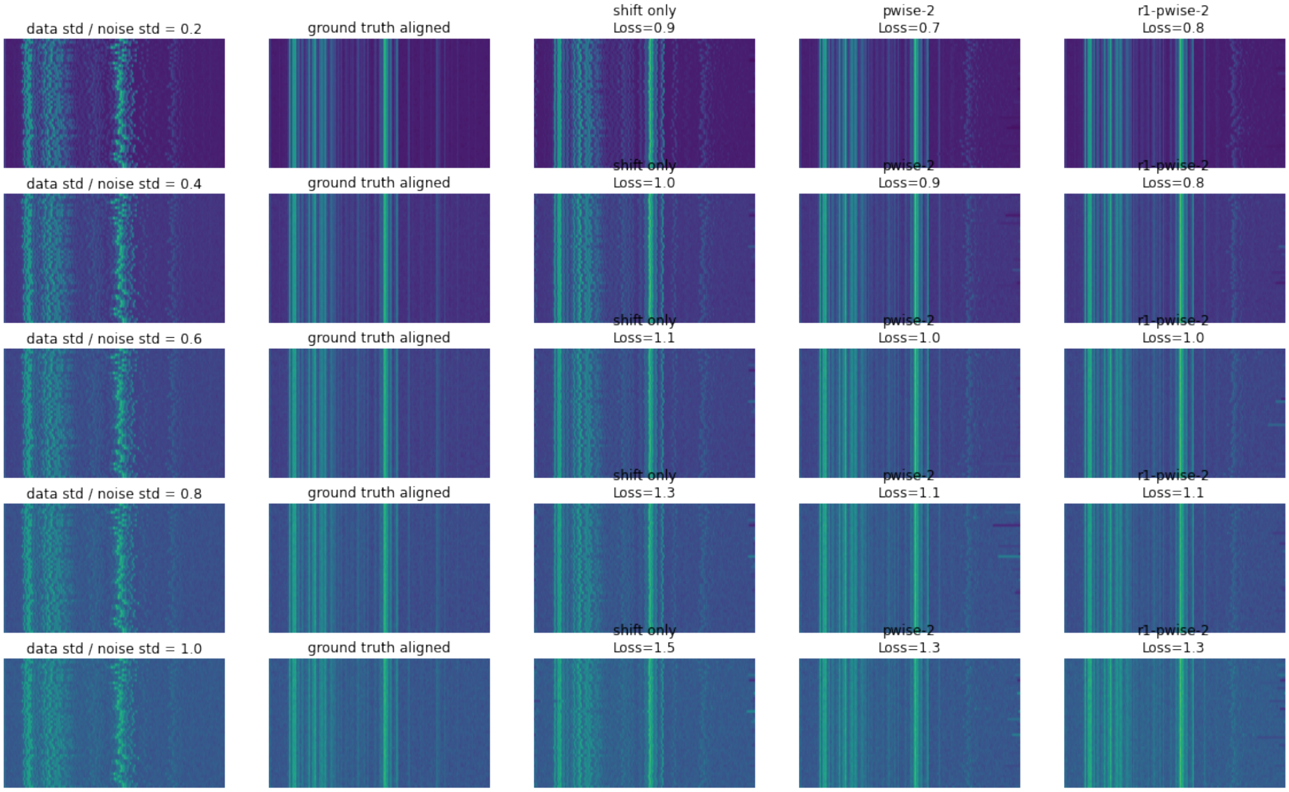

Figure 6: **Simulated data under various iid Gaussian Noise condition.** Note that we are now plotting data’s transpose as in Figure 1 to accentuate warped vs straight strips.

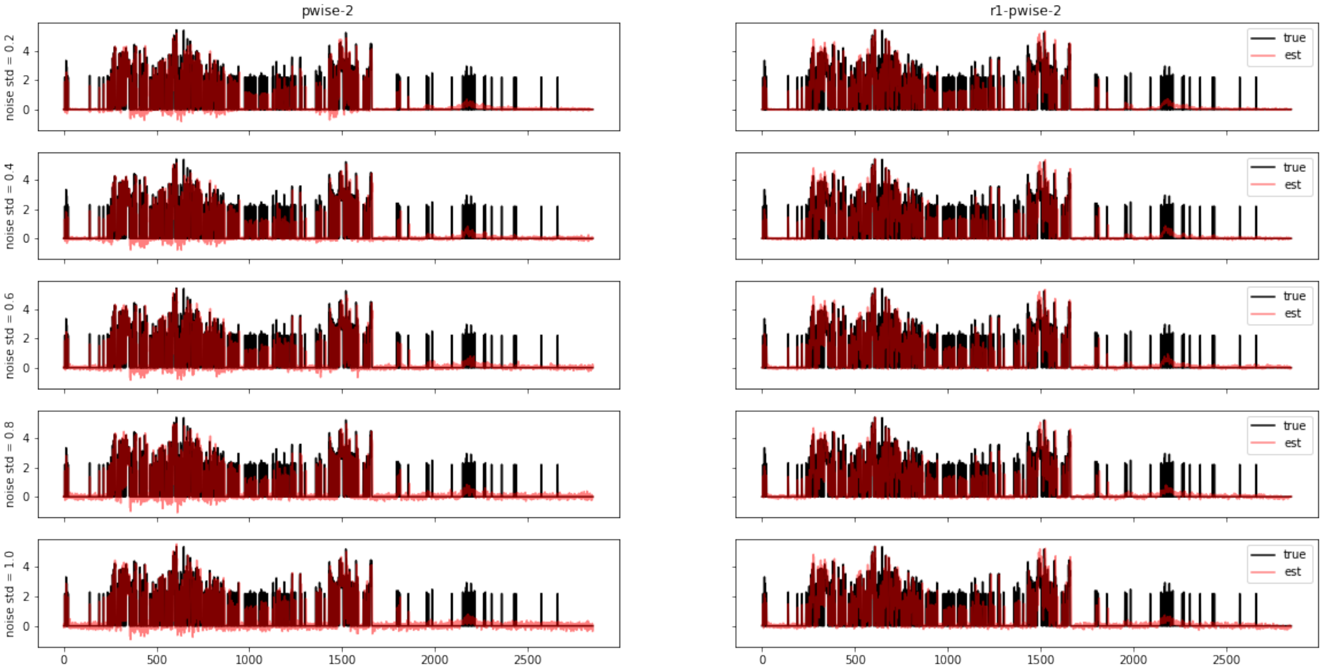

Figure 7: Models' inferred template U given simulated data contaminated with various levels of noise

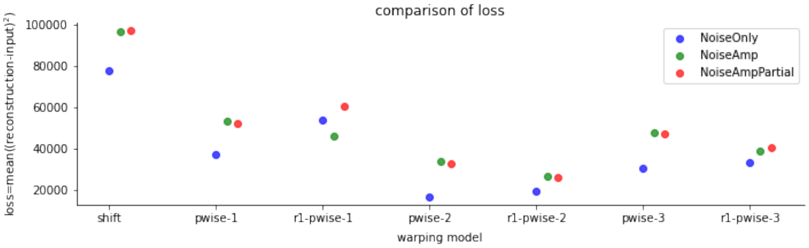

Figure 8: **Various models' reconstruction loss.** Note that a better loss to plot here is the loss between the models' reconstruction and the original warped data without contamination.

6.2 Robustness to (partial) amplitude modification

For another dataset ("NoiseAmp"), we further add in amplitude modulation for each row. For another dataset ("NoiseAmpPartial"), we add in amplitude modulation for only part (a continuous strip of about 30% of all entries) of randomly picked rows (about 10% of all the rows), and each randomly picked rows have different sections to be modulated.

In figure 8, we plot the reconstruction loss with various models. We see that in general knot number 2 gives the best result, and that higher knot numbers could potentially lead to a more complex optimization landscape and therefore hurt the performance. On the other hand, the loss does not capture exactly how well the models actually corrects warping as the higher loss for rank-1 linear models vs the gain-free linear model could be due to that fact that it is overfitting noise. As an example, we plotted (Figure 9) the correction results for piecewise-linear warping and rank-1-piecewise-linear warping for data with the dataset with partial amplitude modulation. Note that for the smaller valued peak at round time point 2000 is aligned slightly better for the rank-1-piecewise-linear warping model.

Then, we visualize each model's learned template (Figure 10) and their gain (Figure 6.2). A similar trend shows up for rank-1 models vs their plain counterparts, despite overall similar performances:

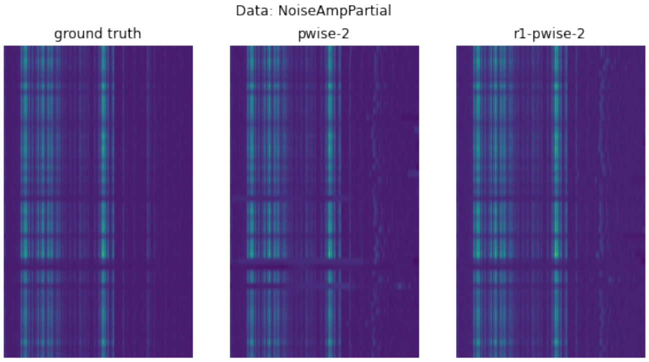

Figure 9: One example dataset comparing the performance of a piecewise linear warping model (pwise-2) vs its counterpart with “gain" (r1-pwise-2). Note that we are now plotting data’s transpose as in Figure 1 to accentuate warped vs straight strips.

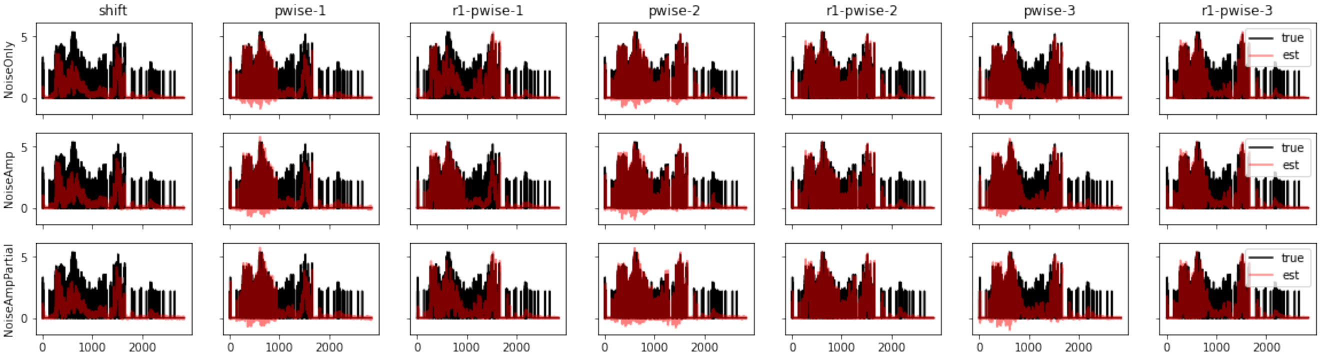

Figure 10: Models’ inferred template U given simulated data contaminated with various levels of noise or (partial) gain modulation.

rank-1 models denoise better; the templates (U) that they inferred do not have negative values. It is worth noting that, despite random partial modification of amplitude, the rank-1 models are able to recover the gain almost exactly. This is reminiscent of Chapter 4-5 (Matrix Recovery).

7 Application of the 1-dimensional drift correction model to real data

The real dataset comes from a recent publication [2] where data were recording from a head-fixed mouse while a motor controlled arm was moving the recording probe up and down the brain (Dataset *Imposed motion recordings at Steinmetz Lab*). Therefore, it can be considered to have a ground truth global rigid motion available.

We applied our method with 1d warping models and compared it to the 2D histogram method with 1 second time window averaging. We can visually see that the correction done by the three methods are overall very similar (Figure 12). We then used the corrected data for spike sorting and we found that motion artifacts in the neuronal activity shown in the unwarped data is now largely absent visually (Figure 13). Note that in each plot in Figure 13, there are no specific sequence in plotting rows and so there are no correspondences among subplots in their rows (neuronal activities).

To further quantitatively see how much drift artifacts various correction methods have removed, we correlate sorted neuronal signals and motion vectors. We plot their absolute coefficient of correlation in Figure 14. We see that all methods have significantly reduced the overall correlation of neuronal activity with movement of the probe. It is worth noting that compared to the control (randomly shuffled neuronal data), there are still some correlation between neuronal signal and movement. This

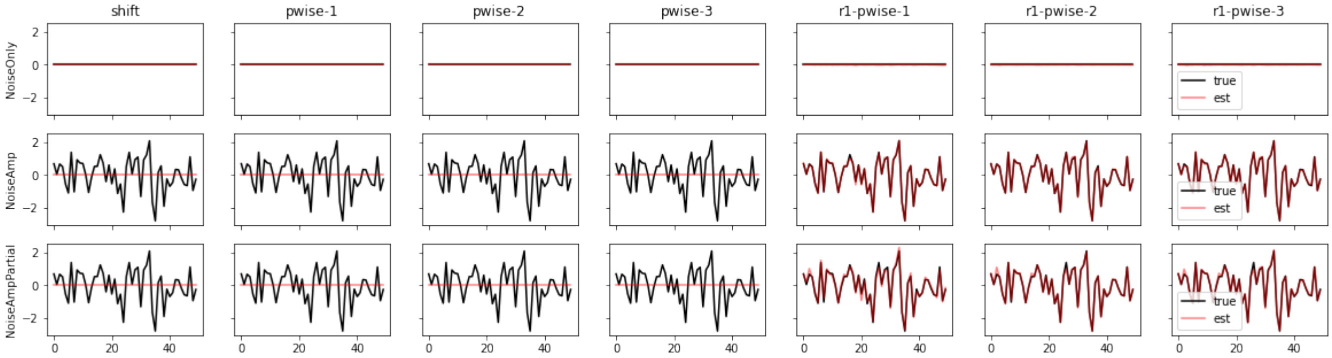

Figure 11: Models' inferred gain v given simulated data contaminated with various levels of noise or (partial) gain modulation.

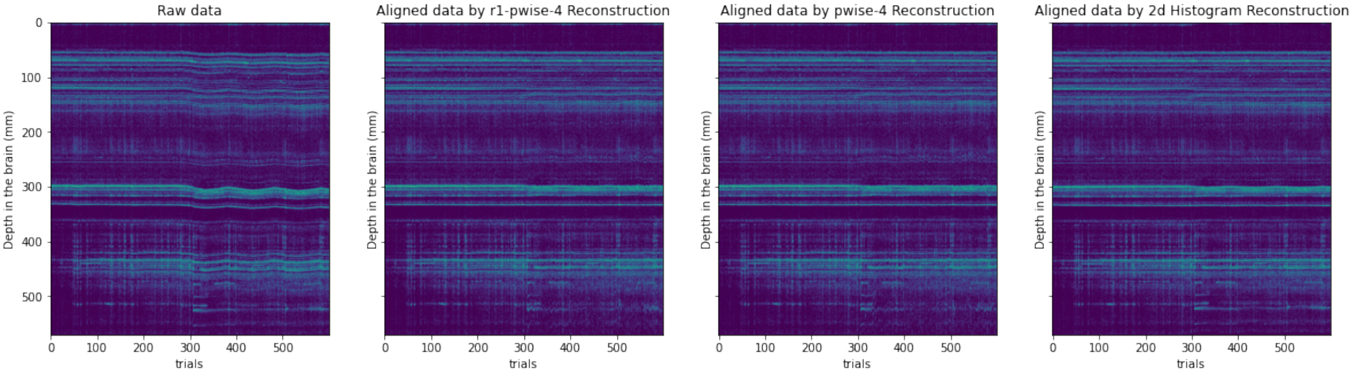

Figure 12: **Real data and corrected data under different methods.**

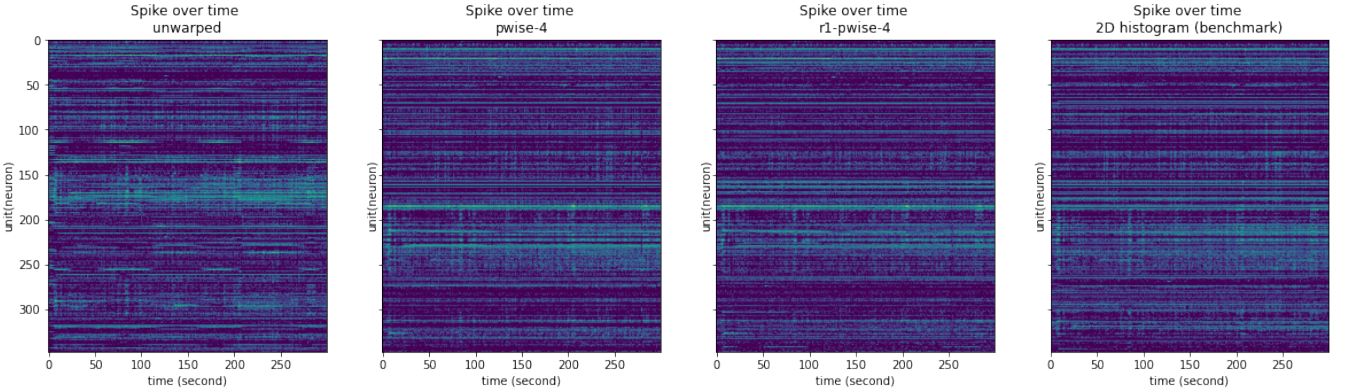

Figure 13: We used the corrected data for spike sorting and we found that motion artifacts in the neuronal activity shown in the unwarped data is now largely absent visually. Note that in each plot in the figure, there are no specific sequence in plotting rows and so there are no correspondences among subplots in their rows (neuronal activities).

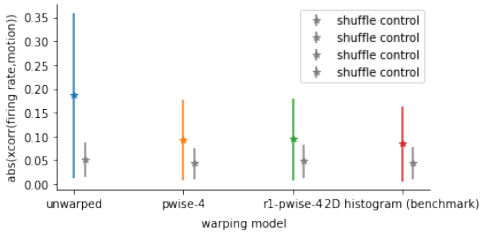

Figure 14: Quantifications on motion removal on **data with 1 second time window averaging** by various models. Plotted are the mean and standard deviation of the absolute correlations coefficient of neuronal activities from corrected data and ground truth global motion.

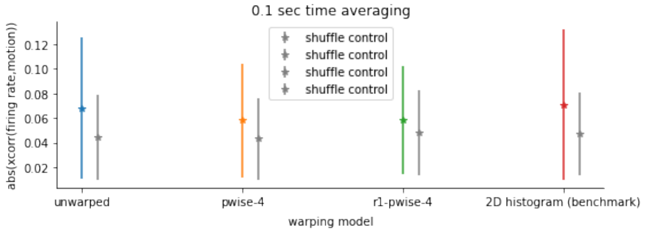

Figure 15: Similar to Figure 14, but here we apply methods to **data with 0.1 second time** averaging.

is likely due to the fact that some neurons do respond with motion stimuli, in addition to the residual artifacts that have not been removed.

Despite that we predict the 1-d methods would have a better signal to noise ratio, it actually does not outperform the 2-d histogram method on this dataset with 1 second time averaging window. This could be due to 1) There is minimally drift left in the dataset already, so any further improvement could not be easily done. 2) The majority of motion in this dataset is shifting, as manifested by the histogram of the slope of inferred motion (data not shown). Therefore, it is within the modeling capacity of 2D histogram method. 3) The 2D histogram method splits the whole data vector into intervals at different depths in the brain and find the best warping independently, it could be that this independent estimation helps in aligning some smaller peaks in the data (For example, 1d warping models in figure 6 have a significantly harder time aligning the small peaks.), and therefore compensates the low signal-to-noise ratio. 4) 2D histogram might be a good transformation of data to find warping. Future work could look into the landscape of the objective with respect to warping 2D histograms.

We also note that we have tried even smaller time intervals such as 0.1 second and we found that our method gives better results than the 2D histogram one (overall smaller correlation between neuronal signal and motion). We note that more detailed analysis is needed to show that our 1d method is indeed better. These further analysis includes manually curating sorted neurons such that noise units are eliminated and potential signal split into different neurons due to motion are inspected and compared across different methods.

8 Future work

As can be seen from the performance on simulated data, smaller peaks in the data are generally aligned less well. To this end, we have also tried norm-1 loss but have similar results. This is likely due to the overwhelmingly large contribution to the loss function for large peaks, and smaller peaks of the data might change only very little for the overall loss. It is possible for future work to consider a vector loss function (sectional loss) such that all sections of data have about equal losses contributions.

Another issue is in deciding on the knot number. Related to the aforementioned sectional loss, one could imagine an algorithm that adds knots if there is no way to achieve balanced loss.

Finally, online development or faster implementation might be worthwhile for future brain-machine interfaces.

9 Conclusion

We believe that authors of [2] and us have convergence arrival on the approach of using 1-rank structure to solving drift tracking problem. We have explicated the 1-rank structure formulation and explored the natural formulation and extensions under this setting, which includes the formal 1-rank approximation of corrected data with a “gain" for the template at each time and a richer repertoire of warping functions considered. We find that compared to the model without the “gain", the full 1-rank models can de-noise better. In application to real data, the explored options do not substantially improve upon the existing method ([2]) in one example data with 1 second time averaging window, potentially due to the minimally drift to be further corrected. Our method does show better performance with 0.1 second time averaging. However, more quantification will need to be done to fully declare its success.

Overall, this study could be offered as a companion supplement to the method in ([2]) and gives a fuller account of drift correction using 1-rank structure in the data.

Acknowledgments and Disclosure of Funding

The author would like to thank Dr. Yi Ma (UC Berkeley) for the guidance during this whole course (UC Berkeley EE290-001), my advisor Dr. Loren Frank (UCSF) for being supportive, Dr. Alex H. Williams (Stanford) for several correspondences from the conception of the project, Dr. Daniel Egert (UCSF) and Dr. Nick A. Steinmetz (University of Washington, Seattle) for their timely response to my inquiry for datasets used in the project.

Shijie Gu is supported by the UCSF-UC Berkeley BioEngineering Graduate Group Grant.

References

[1] Steinmetz N. Siegle J. et al. Jun, J. Fully integrated silicon probes for high-density recording of neural activity. *Nature*, 551:232–236, 2017.

[2] 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, Dimitar Kostadinov, Carolina Mora-Lopez, Junchol Park, Jan Putzeys, Britton Sauerbrei, Rik J. J. van Daal, Abraham Z. Vollan, Marleen Welkenhuysen, Zhiwen Ye, Joshua 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. *bioRxiv*, 2020.

[3] Niru Maheswaranathan Ashesh K. Dhawale Tucker Fisher Christopher D. Wilson David H. Brann Eric M. Trautmann Stephen Ryu Roman Shusterman Dmitry Rinberg Bence P. Ölveczky Krishna V. Shenoy Surya Ganguli Alex H. Williams, Ben Poole. Discovering precise temporal patterns in large-scale neural recordings through robust and interpretable time warping. *Neuron*, 105-2(3):246–259, 2020.

[4] H. Sakoe and S. Chiba. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Trans. on Acoust., Speech, and Signal Process.*, ASSP 26:43–49, 1978.

[5] John Wright Wenli Xu Yigang Peng, Arvind Ganesh and Yi Ma. Rasl: Robust alignment by sparse and low-rank decomposition for linearly correlated images. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 2010.