Contents lists available at ScienceDirect

Journal of Neuroscience Methods

journal homepage: www.elsevier.com/locate/jneumeth

Removing nonlinear misalignment in neuronal spike trains using the Fisher-Rao registration framework

Zishen Xu [a], Xinyu Zhou [a], Yiqi Xu [a], Wei Wu [a,*]

[a] *Department of Statistics, Florida State University, 117 N Woodward Ave., Tallahassee, FL 32306–4330, USA*

ARTICLE INFO

Keywords:
Fisher-Rao registration
Spike trains
Time warping
Nonlinear alignment

ABSTRACT

Background: The temporal precision in neural spike train data is critically important for understanding functional mechanism in the nervous systems. However, the timing variability of spiking activity can be highly nonlinear in practical observations due to behavioral variability or unobserved/unobservable cognitive states.
New method: In this study, we propose to adopt a powerful nonlinear method, referred to as the Fisher-Rao Registration (FRR), to remove such nonlinear phase variability in discrete neuronal spike trains. We also develop a smoothing procedure on the discrete spike train data in order to use the FRR framework.
Comparison with existing methods: We systematically compare the FRR with the state-of-the-art linear and nonlinear methods in terms of model efficiency and effectiveness.
Results: We show that the FRR has superior performance and the advantages are well illustrated with simulation and real experimental data.
Conclusions: It is found the FRR framework provides more appropriate alignment performance to understand the temporal variability in neuronal spike trains.

1. Introduction

The temporal precision issue in neural spike train data has become one critical topic over the past few decades. Investigators started to realize the impact of temporal phase variability in spike data in both biological experiments (London et al., 2010; Bruno, 2011) and theory derivations (Brette, 2015; Denève and Machens, 2016). Various studies have shown the importance of the temporal precision in the analysis of the spike trains (Butts et al., 2007; Bair and Koch, 1996; Reich et al., 1997). However, very few methodologies have been developed to effectively solve this problem. The alignment on spike train data is challenging because the observations are discrete point processes, where the traditional methods for continuous signals such as EEG and fMRI cannot be directly used. In addition, multiple units are recorded simultaneously in various experiments and a *common* phase variability is often needed. Therefore, a desirable method should be able to identify the underlying alignment structure which represents temporal variation for all units.

The simplest and most commonly used method to handle the temporal precision problem is to record one or more time markers simultaneously during the experiment. A time marker is usually a special time point that represents a meaningful event, such as the time of stimulus or the time of subject’s reaction. With time markers, researchers can align the spike train data accordingly through a shifting model or a piecewise linear model. In other words, if there is one time marker, then the alignment is simply a time shift, i.e., we add a constant time to each trial so that the transformed time markers have the same value in all trials. If there are multiple time markers, then the spike time between two time markers will be linearly transformed to make the time markers in every trial be the same.

This way of alignment is easy to apply and efficient, but it has three drawbacks. Firstly, the selection of time markers can be tricky. One needs to know in advance which events will be highly related to the patterns in the signal in order to select an effective time marker, which is often unknown in an exploratory investigation. Randomly choosing one event to be the time marker may result in large errors. For example, for olfactory coding, alignment based on stimulus time will cause the result to be unreliable (Cury and Uchida, 2010; Shusterman et al., 2011, 2018). Secondly, the relation between the time marker and the patterns in the signal is unknown. A shift or linear transformation to align the

[*] Corresponding author.

https://doi.org/10.1016/j.jneumeth.2021.109436
Received 3 July 2021; Received in revised form 29 November 2021; Accepted 2 December 2021
Available online 7 December 2021

time marker may not represent the true relation and may not be able to uncover the hidden information. Thirdly, because the time marker is a part of the observations in the experiment, the recordings may contain random errors or systematic errors, which will affect the alignment performance. Therefore, compared to the alignment based on time markers, an unsupervised alignment method that depends on the spike train data only will be more desirable. Because time marker is not required in the method, it will not suffer these three issues.

Motivated by the registration on discrete events, the Dynamic Time Warping (DTW) method (Berndt and Clifford, 1994) has been adopted to align spike train data. Chi et al. (2007)) combined DTW and linear convolution to identify spike patterns and Cao et al. (2016)) proposed a spike classification method based on DTW. DTW is an unsupervised, algorithm-based method to align sequences. It has the advantage of handling nonlinear warping relations, but suffers from nonrobust and inconsistent framework (see detailed discussions in Zhao et al. (2020))). Recently, Williams et al. (2020)) proposed a reconstruction-error based method for temporal precision problems. The method builds an unsupervised model that contains a template matrix and a warping matrix. The model assumes that the warping function is linear or piecewise linear and the training of the model is a minimization of the reconstruction error, i.e., the difference between the original data and the warped template. We will call this method "piecewise linear model" in this paper. The piecewise linear model has proper performance according to Williams et al. (2020)) that it can successfully uncover hidden patterns in the experimental data. However, the warping function can change the time domain, which may make the aligned spike time out of the observed time interval. In addition, as the warping is restricted to be piecewise linear, its ability to handle drastic nonlinear relation is weak.

To address the weakness of the current unsupervised methods, we propose a new framework by using the Fisher-Rao Registration (FRR) method (Srivastava et al., 2011). FRR conducts the alignment by estimating a time warping function at each step, similar to that in the piecewise linear model. However, FRR has no restriction on the shape of the time warping function and it focuses on nonlinear time warping in a fixed time interval (WLOG, we use [0,1] in this paper). In addition, FRR is based on the Fisher-Rao distance, which is a proper metric for shapes (Srivastava and Klassen, 2016). This characteristic provides FRR many good mathematical properties such as robustness and efficiency, and can naturally lead to more in-depth analysis such as principal component analysis. In this paper, we will explore this method to align discrete spike trains and identify and remove temporal variability in the given data.

The rest of this paper is organized as follows. In section 2, we will firstly review the FRR method. Then the framework for spike train alignment based on FRR method will be described in detail. We will then provide a thorough comparison between the proposed method and commonly used ones. In section 3, we will apply the proposed method on one simulation and two real spike train datasets to evaluate its performance. Finally in section 4, we will summarize our study and discuss the future plan on the FRR alignment method.

2. Methods

In this section, we will at first review the FRR method, and then describe how it can be adapted to conduct alignment on the discrete spike train data.

2.1. Review of the FRR method

We assume that the input functions are in the space $\mathscr{F}_n = \{f : [0, 1] \rightarrow \mathbb{R}^n | f \text{ is absolutely continuous}\}$, $n \in \mathbb{N}$. The Fisher-Rao Registration (Srivastava et al., 2011) is a nonparametric, unsupervised method to align two multi-dimensional functions $f, g \in \mathscr{F}_n$. The alignment is done by searching for the optimal time warping in the set $\Gamma = \{\gamma : [0,1] \rightarrow [0,1] | \gamma(0) = 0, \gamma(1) = 1, \gamma'(t) > 0\}$. That is, the optimal warping from f to g minimizes the Fisher-Rao distance between g and the warped f:

$$\widehat{\gamma} = \underset{\gamma \in \Gamma}{\operatorname{argmin}}\, d_{FR}(f^{\circ}\gamma, g) \tag{1}$$

where d_{FR} represents the Fisher-Rao distance and ∘ represents function composition.

To solve the optimization problem in Eq. (1), we introduce the squared root velocity function (SRVF) to simplify the Fisher-Rao distance. The definition of SRVF is given as:

Definition 1. For any function $f \in \mathscr{F}_n$, its squared root velocity function is defined as:

$$q_f(t) = \begin{Bmatrix} f'(t) \Big/ \sqrt{\|f'(t)\|} & f'(t) \neq \mathbf{0}, \\ \mathbf{0} & f'(t) = \mathbf{0} \end{Bmatrix}$$

where $\| \cdot \|$ is the conventional l_2 Euclidean norm for vectors in $\mathbb{R}^n$. From this definition, we know that the SRVF of an element in $\mathscr{F}_n$ will also be a function from [0,1] to $\mathbb{R}^n$. There are some properties about the SRVF which are useful in our problem:

Property 1. The Fisher-Rao distance between any two functions $f, g \in \mathscr{F}_n$ can be transformed to the $\mathbb{L}_2$ distance between their SRVFs q_f and q_g as:

$$d_{FR}(f, g) = \sqrt{\sum_{i=1}^{n} \| q_{f_i} - q_{g_i} \|_2^2} = \|_2^2 \; \| q_f - q_g \|_2^2 \; \|$$

where $q_f = (q_{f_1}, \cdots, q_{f_n}), q_g = (q_{g_1}, \cdots, q_{g_n})$, $\| \cdot \|_2$ is a vector of classical L_2 norms for corresponding components in a function in $\mathscr{F}_n$. $\| \cdot \|$ is the l_2 Euclidean norm for vectors. **Property 2.** For any function $f \in \mathscr{F}_n$ and time warping function $\gamma \in \Gamma$, the SRVF of $f \circ \gamma$ is given as follows:

$$q_{f\circ\gamma}(\cdot) = (q_f{}^{\circ}\gamma)(\cdot)\sqrt{\gamma'(\cdot)} = \sqrt{\gamma'(\cdot)} \begin{bmatrix} q_{f_1}[\gamma(\cdot)] \\ q_{f_2}[\gamma(\cdot)] \\ \vdots \\ q_{f_n}[\gamma(\cdot)] \end{bmatrix}$$

where $q_f = (q_{f_1}, \cdots, q_{f_n})$ is the SRVF of f. According to Property 2, we can see that $\| q_{f\circ\gamma} \|_2^2 = \int q_f^2[\gamma(t)]\gamma'(t)dt = \int q_f^2(u)du = \| q_f \|_2^2$. Hence, time warping on functions will not change the L_2 norm of their SRVFs. By using the SRVF, the optimization problem for FRR in Equation 1 can be simplified to:

$$\hat{\gamma} = \underset{\gamma \in \Gamma}{\operatorname{argmin}} d_{FR}(f^{\circ}\gamma, g) = \underset{\gamma \in \Gamma}{\operatorname{argmin}} \| \| (q_f{}^{\circ}\gamma)\sqrt{\gamma'} - q_g \|_2 \; \| \tag{2}$$

Then the solution of Eq. (2) can be computed efficiently through a dynamic programming method (Srivastava et al., 2011). In this way, we can obtain the aligned functions $f^{\circ}\hat{\gamma}$ and g, together with the optimal alignment $\hat{\gamma}$.

One reason to adopt the Fisher-Rao distance is that the same warping on the input will not change the Fisher-Rao distance, i.e. $d_{FR}(f \circ \gamma, g \circ \gamma) = d_{FR}(f, g)$. SRVF functions also have the property that

$$\| q_{f\circ\gamma} - q_{g\circ\gamma} \|_2 = \| (q_f{}^{\circ}\gamma)\sqrt{\gamma'} - (q_g{}^{\circ}\gamma)\sqrt{\gamma'} \|_2 = \| q_f - q_g \|_2$$

Hence, all time warping functions are isometries to both the original function space and the SRVF space. If each component of the original function is limited to be within the space of probability density functions: $\mathscr{F}_n^{(p)} = \{f : [0,1] \rightarrow \mathbb{R}^n | f = (f_1, f_2, \cdots, f_n)' \text{ that } \forall i = 1, 2, \cdots, n, f_i \geq 0, \int f_i(t)dt = 1\}$, then the time warping operation will be slightly modified to keep the warping within the same space $\mathscr{F}_n^{(p)}$. This is clearly given in the following remark.

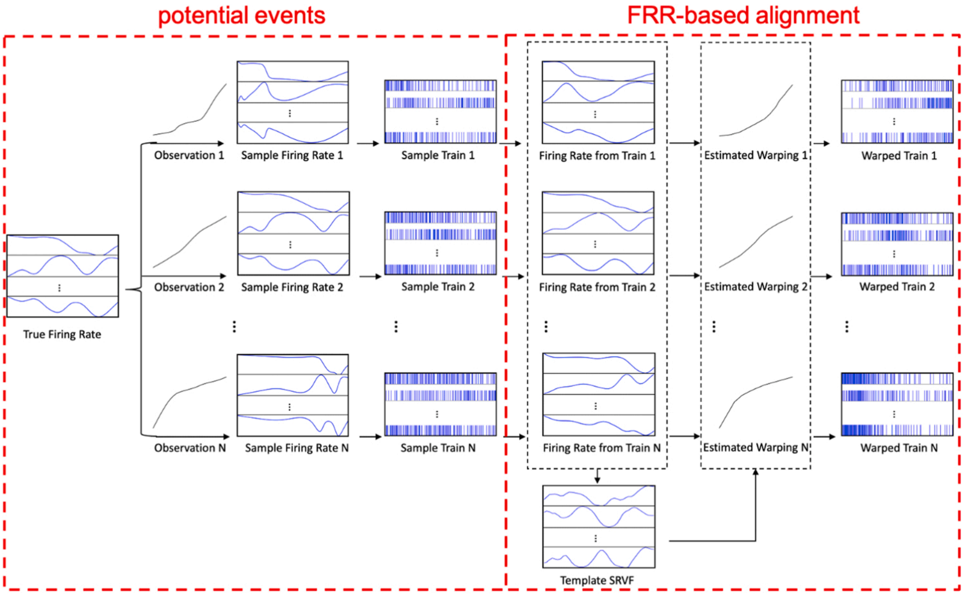

Fig. 1. An illustration of the overall process for spike train alignment.

Remark 1. If the function for alignment is within the density function space $f \in \mathscr{F}_n^{(p)}$, we will consider the warping on f to be $(f;\gamma) = (f^\circ\gamma)\gamma'$ and the SRVF of f to be $q_f = \sqrt{f}$. The properties about SRVF will still hold in this case.

We can see that under Remark 1, the warped function $(f;\gamma)$ is still within $\mathscr{F}_n^{(p)}$ and $q_{(f;\gamma)}$ is still $(q_f{}^\circ\gamma)\sqrt{\gamma'}$, so the SRVF space with L_2 distance is still isometry to the original function space with Fisher-Rao distance.

We point out that based on the pairwise alignment between two functions, the FRR method can be used to align a group of functions by matching every function to a target function. This target function can be set by the average of all functions by taking into account both the amplitude and phase. The detailed procedure is given in (Srivastava et al., 2011).

2.2. FRR-based spike train data alignment framework

2.2.1. Overall framework

The overall process of the alignment on spike train data is illustrated in Fig. 1. The panel in the left side represents the potential events before we observe the spike data, and the right side indicates the FRR-based alignment process.

For the potential events, we assume that there is a potentially true firing rate pattern for every neuron, and the experiment conditions such as stimulus are repeated in different trials. Each trial in the experiment corresponds to a noised, time-warped version of the true firing rate curve. Since all the neurons are observed at the same time in a trial, the time warping function for each neuron is assumed to be the same. Based on the warped firing rate functions, spike train data are generated accordingly and observed during the data collection process.

For the spike trains alignment, the first step is to transform the observed spike train data to functions, either to estimated firing rate functions or to estimated probability density functions of the spike time. Then after computing the SRVFs of the estimated functions, we can apply FRR to obtain the estimated time warping functions together with the template SRVF. In the end, adjusting the spike time data through the estimated time warping functions will give us the aligned spike train data.

According to Fig. 1, three assumptions are required for the FRR-based spike train alignment:

1. All the spike train observations are repeated trials under the same experiment settings.
2. All the neurons are observed simultaneously in every trial, i.e. all the signals are recorded at the same time.
3. All the trials are independent realizations with no warping noise on average.

The first and third assumptions jointly indicate that it is reasonable to assume every observation comes from a time-warped true firing rate. The second assumption suggests that all neurons in a trial share the same time warping function. According to the third assumption, we are able to assume the mean of all the time warping functions is the identity function $\gamma_{id}(t) = t$ for $t \in [0,1]$.

2.2.2. Spike-to-function transformation

- **Type of functional data**

  To transform spike train data to functional data, we propose two choices for the type of functions in this paper. The first choice is the estimated firing rate for the trial based on the spike train data. It can be naturally assumed that the firing rate function is absolutely continuous and nonnegative, so the functions for alignment can be seen as from the space $\mathscr{F}_n$ introduced in section 2.1. In this way, the time warping is applied through $f\circ\gamma$ and SRVF is computed by $q_f = f'/\sqrt{\| f' \|}$. The second choice is the estimated probability density function for the trial based on the spike time collections. In this case, the functions for alignment can be seen as from the space $\mathscr{F}_n^{(p)}$ introduced in section 2.1. according to Remark 1, we should apply time warping through $(f^\circ\gamma)\gamma'$ and compute SRVF by $q_f = \sqrt{f}$.

  Although these two approaches are clearly different, their estimation processes are similar. Denote $F_{j,k}^{(smooth)}(t)$ as the estimated firing rate value and $P_{j,k}^{(smooth)}(t)$ as the estimated density function value for neuron j, trial k at time t. Then $F_{j,k}^{(smooth)}(t)$ represents the count of spikes within a small neighborhood of t and $P_{j,k}^{(smooth)}(t)$ represents the probability of having a spike within a small neighborhood of t. Empirically, probability is estimated by frequency, so $P_{j,k}^{(smooth)}(t)$ can be seen as the frequency of spikes within the neighborhood of t; that is, $P_{j,k}^{(smooth)}(t) = F_{j,k}^{(smooth)}(t)/\mathrm{m}_{j,k}$, where $m_{j,k}$ is the total number of spikes in trial k for neuron j. From this we can see that the difference between the two estimation results is simply a constant multiplication. Therefore, in the following section, we will only introduce the methods for estimating the firing rate $F_{j,k}^{(smooth)}$. If the estimated density function is needed, one can firstly estimate the firing rate, and then divide it by the total number of spikes in the trial to obtain $P_{j,k}^{(smooth)}$.

Both of these two choices have their advantages: Estimating the firing rate follows the nature of the data generating process in our overall framework. As shown in Fig. 1 left half, we assume the firing rate is what is warped in a trial, not the density function. In addition, if we estimate the firing rate, we are able to get the estimated true firing rate (the template), which is not feasible if we estimate the density function. On the other hand, the SRVF will not be based on derivatives in the estimation of density functions. In the case that the estimated firing rate is extremely non-smooth, the density based alignment may perform better as the error in the SRVF will be much smaller.

- **Estimation approaches**

In this paper, we propose two estimation methods to transform the spike train data to functional data. The first method is binning. We firstly set a sequence of consecutive time bins. Then raw firing rate can be calculated by counting the number of spikes in each bin. Specifically, suppose the time bin boundaries are: $t_1, t_2, \cdots, t_N$, i.e. the time bins are: $[t_1, t_2)$, $[t_2, t_3)$, $\cdots$, $[t_{N-2}, t_{N-1})$, $[t_{N-1}, t_N]$, and the spike time sequence is $s_1, s_2, \cdots, s_{m_{j,k}}$ for neuron j and trial k, then the raw firing rate for bin i will be:

$$F_{j,k}^{(raw)}(t_i) = F_{i,j,k}^{(raw)} = \left\{ \begin{array}{ll} \dfrac{1}{t_{i+1} - t_i} \displaystyle\sum_{r=1}^{m_{j,k}} I_{[t_i, t_{i+1})}(s_r) & i = 1, 2, \cdots, N-2, \\ \dfrac{1}{t_N - t_{N-1}} \displaystyle\sum_{r=1}^{m_{j,k}} I_{[t_{N-1}, t_N]}(s_r) & i = N-1, N \end{array} \right\}$$

For easy computation, the raw firing rate $F_{i,j,k}^{(raw)}$ at time bin $[t_i, t_{i+1}]$ is assigned as $F_{j,k}^{(raw)}(t_i)$, the value of function $F_{j,k}^{(raw)}$ at the left time bin boundary t_i, for $i = 1, 2, \cdots, N-1$, and we set $F_{j,k}^{(raw)}(t_N)$ to be the same as $F_{j,k}^{(raw)}(t_{N-1})$. The raw firing rate from binning will then be smoothed to compute the SRVF for FRR alignment. In this paper, we tried two smoothing methods: Gaussian kernel smoothing (Wand and Jones, 1994) and cubic smoothing spline (De Boor and De Boor, 1978). The Gaussian kernel smoothing method is similar to a moving average approach that for a time point t_i and a pre-determined window size d, the smoothed function value $\widetilde{F}_{j,k}^{(smooth)}(t_i)$ will be a weighted average of the raw function values $F_{j,k}^{(raw)}(t_{\max\{1,i-d\}})$, $F_{j,k}^{(raw)}(t_{\max\{1,i-d\}+1})$, $\cdots$, $F_{j,k}^{(raw)}(t_{\min\{N,i+d\}})$, where the weights are determined by a Gaussian density, i.e.:

$$\widetilde{F}_{j,k}^{(smooth)}(t_i) = \frac{1}{\sum_{r=-d}^{d} w_r} \sum_{r=\max\{-i+1,-d\}}^{\min\{N-i,d\}} w_r \, F_{j,k}^{(raw)}(t_{i+r})$$

where $w_r = \exp\{-\frac{1}{2\sigma_b^2}(t_{i+r} - t_i)^2\}$ is the Gaussian density weight with σ_b to be the standard deviation hyper-parameter. The cubic smoothing spline is a nonparametric regression method with a penalty on the roughness. It will estimate a function $\widetilde{F}_{j,k}^{(smooth)}$ based on cubic splines to optimize the objective function:

$$(1-\lambda)\sum_{i=1}^{N}[F_{j,k}^{(raw)}(t_i) - \widetilde{F}_{j,k}^{(smooth)}(t_i)]^2 + \lambda \int [\widetilde{F}_{j,k}^{(smooth)\prime\prime}(t)]^2 dt$$

where $\lambda \in [0,1]$ is a hyper-parameter that controls the smoothness. When $\lambda = 0$, $\widetilde{F}_{j,k}^{(smooth)}$ will pass all the points $(t_i, F_{j,k}^{(raw)}(t_i))$ for $i = 1, 2, \cdots, N$. Increasing λ will make $\widetilde{F}_{j,k}^{(smooth)}$ smoother and when $\lambda = 1$, $\widetilde{F}_{j,k}^{(smooth)}$ will be the least square regression line. To be consistent with the Gaussian kernel smoothing output, function values at $t_1, t_2, \cdots, t_N$ will be recorded to save $\widetilde{F}_{j,k}^{(smooth)}$. As cubic spline estimation is second order differentiable, $\widetilde{F}_{j,k}^{(smooth)}$ has continuous derivative, which can guarantee a smooth SRVF, so we will prefer to use this method for most cases.

The last step is a linear transformation on the function values. For neuron j trial k, we firstly make the function nonnegative:

$$\overset{\approx}{F}_{j,k}^{(smooth)} = \left\{ \begin{array}{ll} \widetilde{F}_{j,k}^{(smooth)} & \text{if } \forall u \in [t_1, t_N], \widetilde{F}_{j,k}^{(smooth)}(u) \geq 0, \\ \widetilde{F}_{j,k}^{(smooth)} - \min\limits_{u \in [t_1, t_N]} \widetilde{F}_{j,k}^{(smooth)}(u) & \text{if } \exists u \in [t_1, t_N], \widetilde{F}_{j,k}^{(smooth)}(u) < 0 \end{array} \right\}$$

Then we make the integral to be consistent with the number of spikes in the trial, and in this way we can obtain the estimated firing rate:

$$F_{j,k}^{(smooth)} = \frac{m_{j,k} \overset{\approx}{F}_{j,k}^{(smooth)}}{\int_{t_1}^{t_N} \overset{\approx}{F}_{j,k}^{(smooth)}(u) du}$$

where $m_{j,k}$ is the number of spikes for neuron j in trial k and $[t_1, t_N]$ is the observed time interval, which is also the boundary of the first and last time bin. The second method for spike-to-function transformation is kernel density estimation Silverman (1986)). This approach is designed for density curve estimation. As discussed in the previous section, the estimated firing rate is just a constant multiplication of the estimated density function, so the kernel density approach works for both choices. In this paper, we adopt the Gaussian kernel estimator and the estimated firing rate can be computed by

$$F_{j,k}^{(smooth)}(t) = \sum_{i=1}^{m_{j,k}} \frac{\exp[-\frac{1}{2\sigma^2}(t - s_i)^2]}{\int_{t_1}^{t_N} \exp[-\frac{1}{2\sigma^2}(u - s_i)^2] du}$$

where $s_1, s_2, \cdots, s_{m_{j,k}}$ is the spike time sequence for neuron j and trial k, and $[t_1, t_N]$ is the observed time interval, which is also the boundary of the first and last time bin. $\sigma > 0$ is a hyper-parameter which controls the interaction between spike time. A close-to-zero σ will result in a spike-like $F_{j,k}^{(smooth)}$ and a large σ will make $F_{j,k}^{(smooth)}$ smooth. Similar to the binning method, we can record the function value at $t_1, t_2, \cdots, t_N$ to represent $F_{j,k}^{(smooth)}$.

Both the binning method and the kernel density estimation method will require a large enough true firing rate, i.e. large enough total number of spikes in a trial. If the true firing rate function values are small over the observed time interval, few spikes will be observed and we can hardly estimate the desired function.

- **Performance measurement**

To measure the performance of the alignment, the most common way is to make raster plots. If the original data do not have clear pattern in the plots but the aligned data do, it means some hidden information is uncovered and the alignment is likely to work properly. In the case that there are time markers recorded with the data, we can consider them in the aligned data. Because time markers are not used at all in the alignment framework, if they are clearly concentrated in the aligned data, then it reveals that the alignment effect is appropriate.

To quantify the alignment performance, one can use the difference between the true and estimated time warping function, or the difference between the true and estimated firing rate function. For example, the root mean square errors (RMSE) with respect to the L_2 distance for functions on warping functions and firing rates respectively are given as

$$RMSE^{(\gamma)} = \sqrt{\frac{1}{K}\sum_{k=1}^{K}\int[\gamma_k(t) - \hat{\gamma}_k(t)]^2 dt} \tag{3}$$

$$RMSE^{(f)} = \sqrt{\frac{1}{M}\sum_{j=1}^{M}\int[f_j(t) - \hat{f}_j(t)]^2 dt} \tag{4}$$

where K is the total number of trials and M is the total number of observed neurons. γ_k and f_j represent the time warping function for trial k and the true firing rate for neuron j respectively, while $\hat{\gamma}_k$ and $\hat{f}_j$ represent the estimated time warping function for trial k and the estimated true firing rate for neuron j respectively. A smaller value of $RMSE^{(\gamma)}$ or $RMSE^{(f)}$ indicates a better alignment performance.

Another way is to use the reconstruction R-square, proposed by Williams et al. (2020)). Basically, one can use the inverse of the estimated time warping function on the estimated true firing rate, so that a fitted warped firing rate can be computed:

$$\hat{F}_{j,k} = \hat{f}_j \circ \hat{\gamma}_k^{-1}$$

where $\hat{f}_j$ is the estimated true firing rate for neuron j and $\hat{\gamma}_k$ is the estimated warping function for trial k. Then we can compare $\hat{F}_{j,k}$ to the raw firing rate $F_{j,k}^{(raw)}$, or to the estimated firing rate $F_{j,k}^{(smooth)}$ and compute the R-square value:

$$R_B^2 = 1 - \frac{\sum_{i=1}^{N}\sum_{j=1}^{M}\sum_{k=1}^{K}[F_{j,k}^{(raw)}(t_i) - \hat{F}_{j,k}(t_i)]^2}{\sum_{i=1}^{n}\sum_{j=1}^{M}\sum_{k=1}^{K}[F_{j,k}^{(raw)}(t_i) - \overline{F}_j^{(raw)}]^2} \tag{5}$$

$$R_S^2 = 1 - \frac{\sum_{i=1}^{N}\sum_{j=1}^{M}\sum_{k=1}^{K}[F_{j,k}^{(smooth)}(t_i) - \hat{F}_{j,k}(t_i)]^2}{\sum_{i=1}^{n}\sum_{j=1}^{M}\sum_{k=1}^{K}[F_{j,k}^{(smooth)}(t_i) - \overline{F}_j^{(smooth)}]^2} \tag{6}$$

where

$$\overline{F}_j^{(raw)} = \frac{1}{NK}\sum_{i=1}^{N}\sum_{k=1}^{K}F_{j,k}^{(raw)}(t_i)$$

and

$$\overline{F}_j^{(smooth)} = \frac{1}{NK}\sum_{i=1}^{N}\sum_{k=1}^{K}F_{j,k}^{(smooth)}(t_i).$$

For the estimated density function, we can compute the R-square based on the fitted density:

$$\hat{P}_{j,k} = (\hat{P}_j \circ [\hat{\gamma}_k^{-1}])[\hat{\gamma}_k^{-1}]'$$

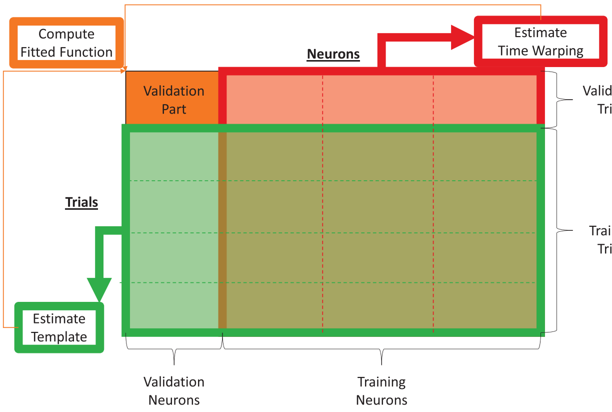

Fig. 2. Graphical illustration of the cross-validation for optimal hyper-parameter search.

where $\hat{P}_j$ is the estimated template density function for neuron j and $\hat{\gamma}_k$ is the estimated warping function for trial k. The associated R-square on density is defined as:

$$R_P^2 = 1 - \frac{\sum_{i=1}^{N}\sum_{j=1}^{M}\sum_{k=1}^{K}[P_{j,k}^{(smooth)}(t_i) - \hat{P}_{j,k}(t_i)]^2}{\sum_{i=1}^{n}\sum_{j=1}^{M}\sum_{k=1}^{K}[P_{j,k}^{(smooth)}(t_i) - \overline{P}_j^{(smooth)}]^2} \tag{7}$$

where $\overline{P}_j^{(smooth)} = \frac{1}{NK}\sum_{i=1}^{N}\sum_{k=1}^{K}P_{j,k}^{(smooth)}(t_i)$. A large R-square value suggests that the alignment result can well represent the original data, but the ability to recover the original data does not mean that the alignment is meaningful or reasonable. Therefore, the reconstruction R-square value is not as good as the RMSE. However, RMSE will not be available when we do not have the true time warping function and true firing rate. In this case we can only use R-square to quantify the alignment performance.

Based on the RMSE or R-square, we propose a cross-validation method to search for optimal hyper-parameters. The detailed procedure is given in Algorithm 1 as follows. In summary, the method partitions observed dataset into two parts: a training part and a validation part, with respect to trials and neurons. In other words, the entire trials are separated into training trials and validation trials, and the entire neurons are separated into training neurons and validation neurons. The partition for the trials and the neurons are independent, so one can set different number of folds for the trials and the neurons. As illustrated in Fig. 2, the FRR alignment will be applied on the training trials, entire neurons to estimate the template. Then it will be applied again on the entire trials, training neurons to estimate the time warping functions. In this way, the estimated templates cover all the neurons, and the estimated time warping functions cover all the trials. Using the estimated templates for the validation neurons and the estimated time warping functions for the validation trials, we can get the fitted firing rate functions (validation trials and neurons only). By calculation on all folds, we can obtain the estimated template, estimated time warping function and fitted firing rate function for every trial and neuron, and the RMSE or R-square can be computed accordingly. Therefore, for each selected hyper-parameter, we are able to compute the corresponding cross-validation performance value (either RMSE or R-square), and finally a grid search can be done to select the optimal one.

Algorithm 1. Cross-Validation Grid Search (Hyper-parameter Tuning).

```
Input:
- The spike data $S = \{S_{ij}\}_{\substack{i=1...K \\ j=1...M}}$ with $K$ trials and $M$ neurons.
- The number of folds on trials $C_t$ and on neurons $C_n$.
- The estimation method from spike to firing rate / density function $M(s;\theta)$ with $s$ to be the input spike data and $\theta$ to be the hyper-parameter.
- The measure of performance $P(s,\hat{H},\hat{\gamma},\hat{f})$ with $s$ to be the input spike data and $\hat{H},\hat{\gamma},\hat{f}$ to be the estimated template, estimated time warping and fitted firing rate / density for $s$.
- The set of hyper-parameter values to search $\{h_1, h_2, \cdots, h_r\}$.
Iteration:
(1) Random shuffle set $\{1...K\}$ and partition it into $C_t$ parts: $T_1 ... T_{C_t}$, i.e. $\forall\, i,j = 1...C_t$ and $i \ddagger j$, $T_i \cap T_j = \emptyset$; $\cup_{i=1}^{C_t} T_i = \{1...K\}$.
(2) Random shuffle set $\{1...M\}$ and partition it into $C_n$ parts: $N_1 ... N_{C_n}$, i.e. $\forall\, i,j = 1...C_n$ and $i \ddagger j$, $N_i \cap N_j = \emptyset$; $\cup_{i=1}^{C_n} N_i = \{1...M\}$.
for each $\alpha = 1, 2...r$ do
for each $\beta = 1, 2...C_t$ do
for each $\gamma = 1, 2...C_n$ do
(3.1) Estimate the firing rate / density functions from the spikes: $\{F_{ij}^{(\alpha)}\}_{\substack{i=1...K \\ j=1...M}} = M(S;h_\alpha)$.
(3.2) Get the trial indices for train: $T_\beta^c = \cup_{\substack{i=1 \\ i\neq\beta}}^{C_t} T_i$ and valid: $T_\beta$.
(3.3) Get the neuron indices for train: $N_\gamma^c = \cup_{\substack{i=1 \\ i\neq\gamma}}^{C_n} N_i$ and valid: $N_\gamma$.
(3.4) Apply FRR on training trials and entire neurons, i.e. $\{F_{ij}^{(\alpha)}\}_{\substack{i\in T_\beta^c \\ j=1...M}}$, to obtain the estimated template $\{\hat{H}_j^{(\alpha)}\}_{j=1...M}$.
(3.5) Apply FRR on entire trials and training neurons, i.e. $\{F_{ij}^{(\alpha)}\}_{\substack{i=1...K \\ j\in N_\gamma^c}}$, to obtain the estimated time warping $\{\hat{\gamma}_i^{(\alpha)}\}_{i=1...K}$.
(3.6) Compute the fitted firing rate / density $\{\hat{f}_{ij}^{(\alpha)}\}_{\substack{i\in T_\beta \\ j\in N_\gamma}}$ by $\hat{f}_{ij}^{(\alpha)} = \hat{H}_j^{(\alpha)} \circ (\hat{\gamma}_i^{(\alpha)})^{-1}$, for $i \in T_\beta, j \in N_\gamma$.
(3.7) Record the estimated template, time warping and fitted function for validation: $\{\hat{H}_j^{(\alpha)}\}_{j\in N_\gamma}$, $\{\hat{\gamma}_i^{(\alpha)}\}_{i\in T_\beta}$ and $\{\hat{f}_{ij}^{(\alpha)}\}_{\substack{i\in T_\beta \\ j\in N_\gamma}}$
end for
end for
(4) After the iteration, we have the estimated template, time warping and fitted function for all trials, all neurons: $\{\hat{H}_j^{(\alpha)}\}_{j=1...M}$, $\{\hat{\gamma}_i^{(\alpha)}\}_{i=1...K}$ and $\{\hat{f}_{ij}^{(\alpha)}\}_{\substack{i=1...K \\ j=1...M}}$, so the measure of performance for $h_\alpha$ can be computed as: $\hat{p}_\alpha = P(S, \{\hat{H}_j^{(\alpha)}\}_{j=1...M}, \{\hat{\gamma}_i^{(\alpha)}\}_{i=1...K}, \{\hat{f}_{ij}^{(\alpha)}\}_{\substack{i=1...K \\ j=1...M}})$.
end for
Output:
$\hat{p}_1, \cdots, \hat{p}_r$ are the measurement of performance with respect to the hyper-parameters $h_1, \cdots, h_r$. One can choose the $h_\alpha$ with the best $\hat{p}_\alpha$.
```

2.3. Advantages of the FRR-based spike train alignment

Various approaches have been proposed to address the alignment problem on spike train data. These approaches include a model-free method (Ventura, 2004), Dynamic Time Warping (DTW) and its variants (Keogh and Pazzani, 2001; Cuturi and Blondel, 2017; Lawlor et al., 2018), scalable latent identification (Duncker and Sahani, 2018), and piecewise linear models (Williams et al., 2020). Within these methods, two of the most commonly used ones are the DTW and piecewise linear models. We point out that although the DTW method can properly identify nonlinear time warping in the given neural signals, it suffers undesired mathematical properties such as inconsistent sampling points, unstable normalization, and pinching effect. A thorough comparison between FRR and DTW is already discussed in detail in (Zhao et al., 2020). In this manuscript, we focus on the comparison between FRR and the piecewise linear models (Williams et al., 2020). The FRR alignment has the following advantages: .

1. **Allow all types of time warping functions:** The piecewise linear models assume the warping is either a simple time shift or a piecewise linear function. For mild phase variability, such methods can provide efficient and accurate representation. However, these models may suffer (of robustness and complexity) when more drastic time warpings are observed. In contrast, the FRR alignment can be given in any nonlinear form and has no restriction on the type of warping function. No matter what the true time warping is, the FRR alignment can work efficiently and effectively.
2. **Invariant on the time domain:** In the piecewise linear models, the time warping function cannot be fixed in any time domain. As a result, the aligned spikes can be out of the given observed time interval, which makes the result difficult to interpret and may lead to information loss. In the FRR method, the time warping functions are monotone increasing from [0,1] to [0,1].
3. **Minimal hyper-parameter tuning:** In the piecewise linear models, one needs to carefully select the hyper-parameters in the model, including the number of knots, the regularization strength for both the template and the warping, and the number of iterations for optimizing the warping function. This may be a highly demanding task in practical use. In contrast, in the FRR method, the only hyper-parameters are the ones in the smoothing procedure, so there are at most two hyper-parameters (when using binning with Gaussian kernel smoothing), and the tuning process is straightforward.

3. Experimental results

In this section, we will apply the FRR alignment on simulated and real experimental datasets to evaluate its performance. Firstly in Section 3.1, a simulation dataset is used for alignment. The true firing rate functions and true time warping functions are known, so we simply compare the estimated curves to the true curves by computing $RMSE^{(\gamma)}$ and $RMSE^{(f)}$. Then in Sections 3.2 and 3.3, the method is applied on two real datasets. Because there is no ground truth in these real-world datasets, *RMSE* will not be a valid measure and we will focus on the behavior of the aligned time markers.

3.1. Simulation example

To evaluate the performance of the FRR alignment on spike train data, we conduct a simulation study following the structure in Fig. 1, where the potential true firing rate and true time warping function for every trial are known, so that a real application of the FRR alignment can be imitated and at the same time the actual performance can be accessed. To better understand the method, we separate the simulated data to two parts: a training part for template estimation and a test data for time warping estimation based on the output from the training data. In total, we simulate 30 trials for training and 10 trials for test. Both include 50 observed neurons. The observed time interval for the spike and the firing rate is [0,20]. The generated potential true firing rate functions and true time warping functions are shown in Fig. 3, including 50 true firing rate functions, 30 true time warping functions for training and 10 true time warping functions for test. The true firing rate function is generated by smoothed random walk within 0–10 in function value. For the true time warping function, we firstly random generate some points within 0 and 1 to be the break points. Then we sort these points and connect them by concatenated cubic functions:

$$f(x) = \begin{cases} y_1 + a(x - x_1)^3 & \text{if } x \leq (x_1 + x_2)/2 \\ y_2 + a(x - x_2)^3 & \text{if } x > (x_1 + x_2)/2 \end{cases}$$

where (x_1, y_1) and (x_2, y_2) are the two points to connect and $a = 4(y_2 -$

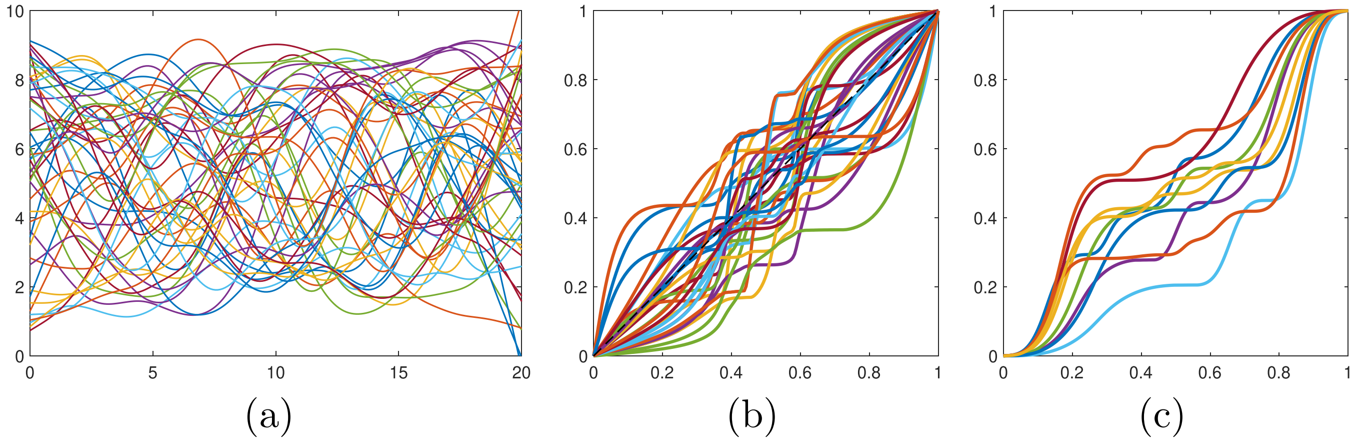

Fig. 3. The potential true functions in the simulation. (a) The potential true firing rate functions. (b) The true time warping functions for the training part. (c) The true time warping functions for the test part.

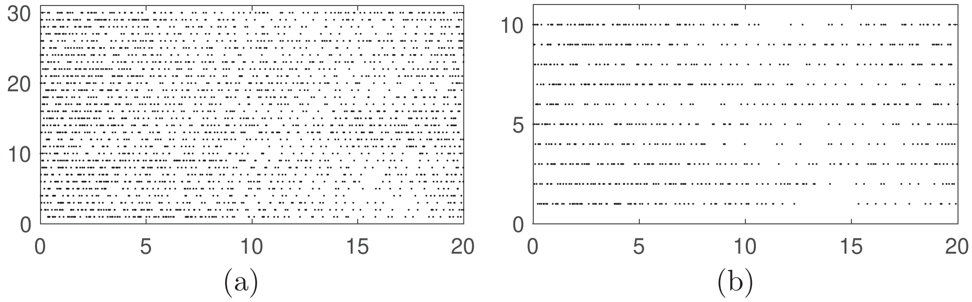

Fig. 4. The simulated spike data for the 10th neuron as an example. (a) The 30 training trials. (b) The 10 test trials.

$y_1)/(x_2 - x_1)^3$. In this way, the resulting curve will be in Γ as defined in section 2.1. After composing the firing rate and the time warping, we randomly generate spikes on the warped firing rate functions. An example of the spike data for the 10th neuron is shown in Fig. 4, containing 30 trials for training and 10 trials for test.

For the FRR alignment, binning with cubic smoothing spline is applied to estimate the firing rate from the spike data. The time bins are $[0, 0.5), [0.5, 1), \cdots, [19.5, 20]$ and the hyper-parameter λ is searched through a cross-validation grid search with 4 folds on trials and 5 folds on neurons. To imitate the application on real data, we use the reconstruction R-square in Eqs. (5) and (6) as the measure of performance. The resulting learning curve is shown in Fig. 5 (a). As can be seen, the R_B^2 curve is nearly monotonically decreasing, and therefore the value that can maximize the R_S^2 is chosen to be the final hyper-parameter ($\hat{\lambda} = 0.2436$). Based on the FRR alignment, the estimated template and estimated time warping functions for the training and the test are shown in Fig. 6 (a), (b) and (c) respectively. Compared to Fig. 3, we can see the estimated curves in general have similar shape as the true curves. Some curve segments become straight in the time warping estimation due to information loss during random generation of spikes from firing rate. Because the potential true firing rate and true time warping are known, we adopt the RMSE in equation 3 and 4 as the measure of performance. The resulting values are shown in Table 1 row 1.

To make a comparison, we have also tried the piecewise linear model (Williams et al., 2020) on the same simulation data (the spike data). The time bins are chosen to be the same $[0, 0.5), [0.5, 1), \cdots, [19.5, 20]$ and other hyper-parameters are searched by the method provided in the given paper. The learning curve is shown in Fig. 5 (b). According to the output, we select the best 3 models: piecewise-3, piecewise-4 and piecewise-5. The resulting time warping functions are shown in Fig. 7 and the estimated template functions are shown in Fig. 8. The results are not as good as the result from the FRR alignment, especially the time warping functions, compared to Fig. 3. The RMSE values are given in

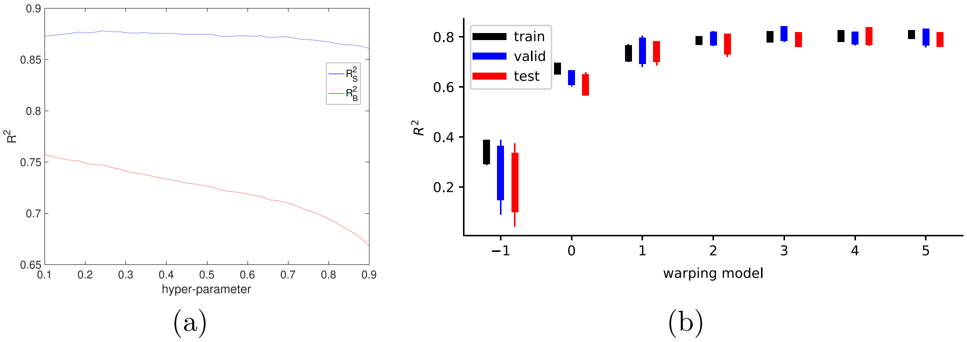

Fig. 5. The learning curve for hyper-parameter search. (a) The cross-validation grid search in FRR alignment. (b) The hyper-parameter search for piecewise linear model.

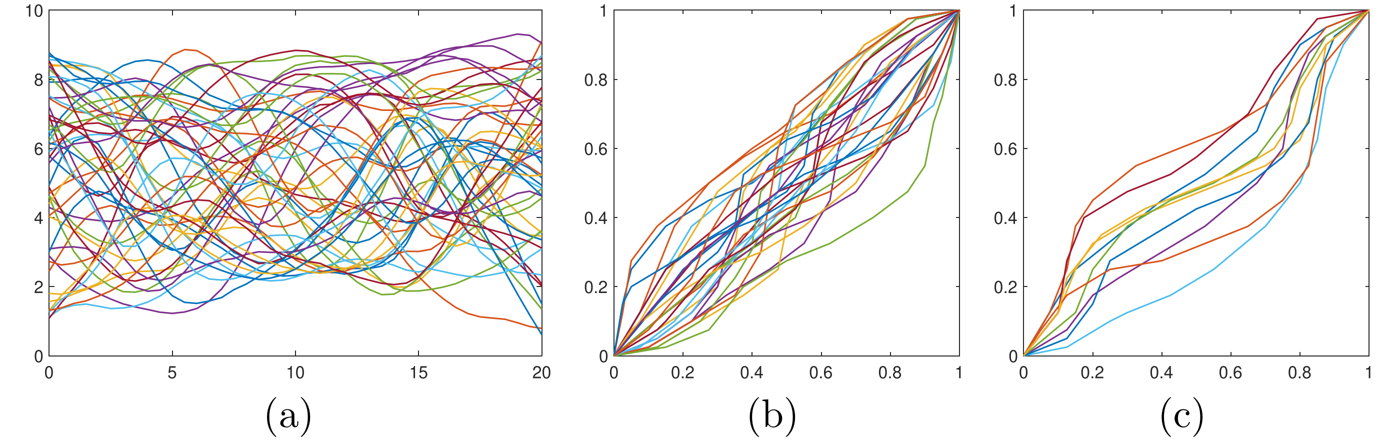

Fig. 6. The estimated functions through FRR alignment. (a) The estimated template firing rate functions. (b) The estimated time warping functions for the train part. (c) The estimated time warping functions for the test part.

Table 1
The performance of the alignment methods with respect to the true functions.

Type of Model	$\text{RMSE}^{(\gamma)}_{(\text{train})}$	$\text{RMSE}^{(\gamma)}_{(\text{test})}$	RMSE $^{(f)}$
FRR	0.0365	0.0398	2.4566
Piecewise-3	0.0427	0.0553	2.8807
Piecewise-4	0.0391	0.0430	2.5526
Piecewise-5	0.0416	0.0463	2.6647

Table 1 row 2 to row 4. It also agrees to the conclusion that the FRR alignment provides better results than piecewise linear model.

In conclusion, based on the simulation outcomes, the proposed FRR alignment method can properly estimate the potential true firing rate functions and true time warping functions. It can provide reasonable alignment results on the spike train data.

3.2. Rat motor cortex data alignment

The second dataset is a real experimental recording of motor cortical activity and was used in (Williams et al., 2020). The dataset contains 1266 trials on 30 neurons. The experiment was to train rats to press a lever twice and if the time between the pressing is within a target interval, then the rats receive food reward. The time for the two pressing was also recorded as time markers. The spike data was preprocessed so that the time for the first pressing is consistent across trials at 500 ms and the trials are sorted by the second pressing time. The time interval for the spike data is [0,2000] ms. For easy computation, we chose the neurons whose average number of spikes per trial was larger than 5 to be the “main-focus” neurons. Then we included only the trials that had more than 1 spike for the main-focus neurons in our study. In this way, 644 out of 1266 trials were considered in the computation and 15 out of 30 neurons became the main-focus neurons.

The plots for the raw data are displayed in Fig. 9 (a). The first 6 rows are 6 example neurons, where the blue points represent the first pressing and the red points represent the second pressing. The last row is the histogram for the two pressing time. Similarly, the blue bars are for the first pressing and the red bars are for the second pressing. As the original data is aligned by the first pressing, the histogram has only one blue bar. The standard deviation (STD) and interquartile range (IQR) for the two taps and the distance between the two taps are shown in Table 2 row 1. From the plots of the raw data, we can hardly see any pattern.

A naive way of alignment is to align both the two pressing through a manual linear time warping. Because the first pressing time is already aligned, the time warping function will be a linear scaling of the time after the first pressing, i.e. the time warping function γ_j for trial j will be:

$$\gamma_j(t) = \begin{cases} t & t \in [0, C_1), \\ \dfrac{c_j - C_1}{C_2 - C_1}(t - C_1) + C_1 & t \in [C_1, C_2), \\ \dfrac{1 - c_j}{1 - C_2}(t - C_2) + c_j & t \in [C_2, 1] \end{cases}$$

where $C_1 = (500 - 0)/2000 = 0.25$ and $C_2 = 0.61775$ are the average time of the first and second pressing scaled to be within [0,1]. c_j is the time of the second pressing scaled to be within [0,1] for trial j. Through this alignment, we obtained Fig. 9 (b), and some example time warping functions are given in Fig. 10 (a). It can be seen that the two pressing time is perfectly aligned, but there is still no clear pattern on aligned spike trains.

We then tried the FRR method on the spike data. We used only the "main-focus" neurons in the FRR alignment, and when the estimated time warping functions for all the trials were obtained, we applied them on all the neurons to get the aligned spikes. After comparing the output from different settings, we chose the density approach with binning and cubic smoothing spline as the final FRR model. The result is shown in Fig. 9 (d), where some time warping function examples are given in Fig. 10 (c). From the raster plots of the example neurons and the histogram, we can see the alignment concentrates the time for tap 2 a lot, while making the time for tap 1 a little disperse. In addition, it is clear that patterns exist in the spike data. We can observe oscillations in the raster plots, especially the second, third and fifth row. These findings are consistent with the result of the shifting model in the original paper Williams et al. (2020)), as shown in Fig. 9 (c), with some time warping functions for illustration shown in Fig. 10 (b). Compared to Fig. 9 (a) and (b), the oscillation patterns only exist in the aligned output. This indicates that the alignment based on time markers or simple manual linear time warping may not be able to uncover the hidden information in the data. The STD and IQR for the two taps and the distance between taps are shown in Table 2, where the shifting model output is in row 2 and the FRR method output is in row 3. In comparison with the raw data, the FRR alignment has the distance between taps and the time for tap 2 more concentrated since its IQR value is apparently much lower (the STD value is nearly the same). On the other hand, it can be seen that the shifting model and the FRR method have similar STD and IQR for the second tap, which implies they have similar effect on concentrating the times for tap 2. However, for tap 1 time, the FRR method has much lower STD and IQR compared to the shifting model. In terms of the distance between taps, as the shifting model will not change the space between spikes, it has the same IQR and STD as the raw data; that is, the tap1-tap2 distance is not concentrated at all in the shifting model. Therefore, we can see that the FRR method is able to find potential

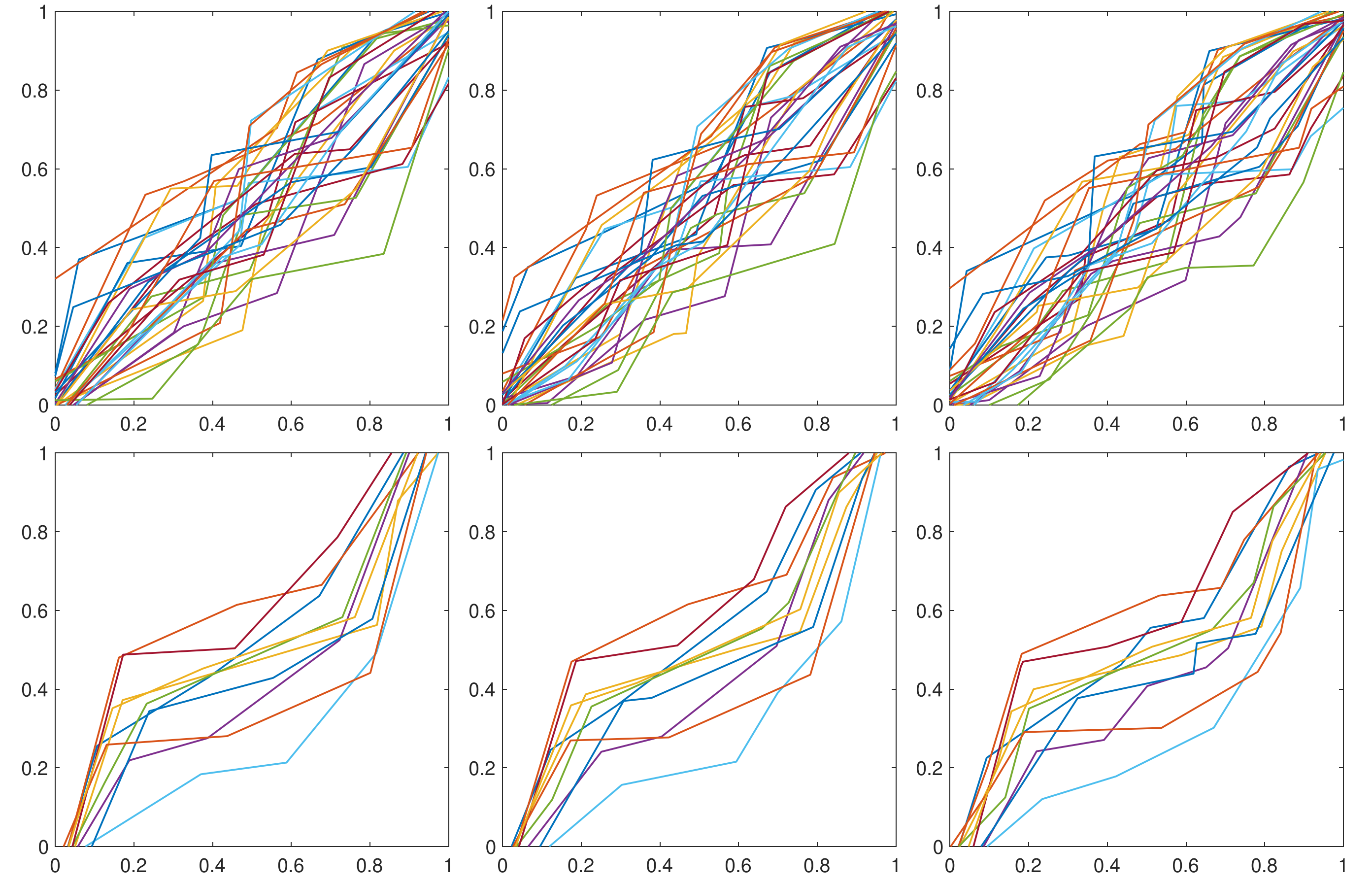

Fig. 7. The estimated time warping functions through piecewise linear model. Top row: training trials. Bottom row: test trials. Left column: piecewise-3. Middle column: piecewise-4. Right column: piecewise-5.

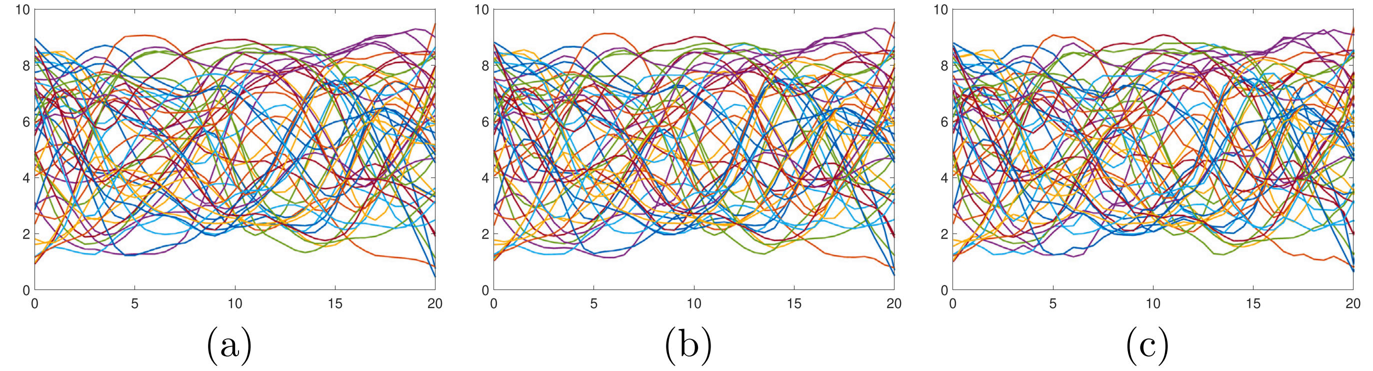

Fig. 8. The estimated template functions through piecewise linear model. (a) piecewise-3. (b) piecewise-4. (c) piecewise-5.

patterns in the data and take into consideration the subject's major actions simultaneously, while the piecewise linear model cannot.

3.3. Monkey decoding data alignment

The third experimental dataset is macaque monkey neuron decoding dataset, first introduced by Wu and Srivastava (2011)). The experiment trained monkeys to use right arm moving a cursor on a screen to a target location, and during this process, the extracellular spiking activities were recorded through implanted electrodes. There are four target locations on the screen, forming the four vertices of a rectangle. We will refer to the target at the top left corner as target 1, the target at the bottom left corner as target 2, the target at the bottom right corner as target 3 and the target at the top right corner as target 4. Each trial begins when the cursor comes to the first target. Then when one target is reached, the next one will appear on the next counterclockwise vertex. The trial ends when the cursor reaches the last target, which is the same as the first target, making the trajectory a closed path (illustrated in Fig. 11). In the experiment, 110 neurons were observed over 60 trials. The spike time is preprocessed so that all the spikes are within 0–5 s. The time that the cursor reaches the target is also recorded and preprocessed, so there are five time markers. The first one is at 0 s and last one is at 5 s. The remaining three time markers are within 0–5 s

The raster plots of the raw data are shown in Fig. 12 (a). The first 5 rows are the raster plots for 5 example neurons. The cyan, red, green and blue points represent the times that targets 1, 2, 3 and 4 are reached, respectively. The last row is the histogram for the times that targets 2, 3 and 4 are reached, respectively. Because the recording starts at the time

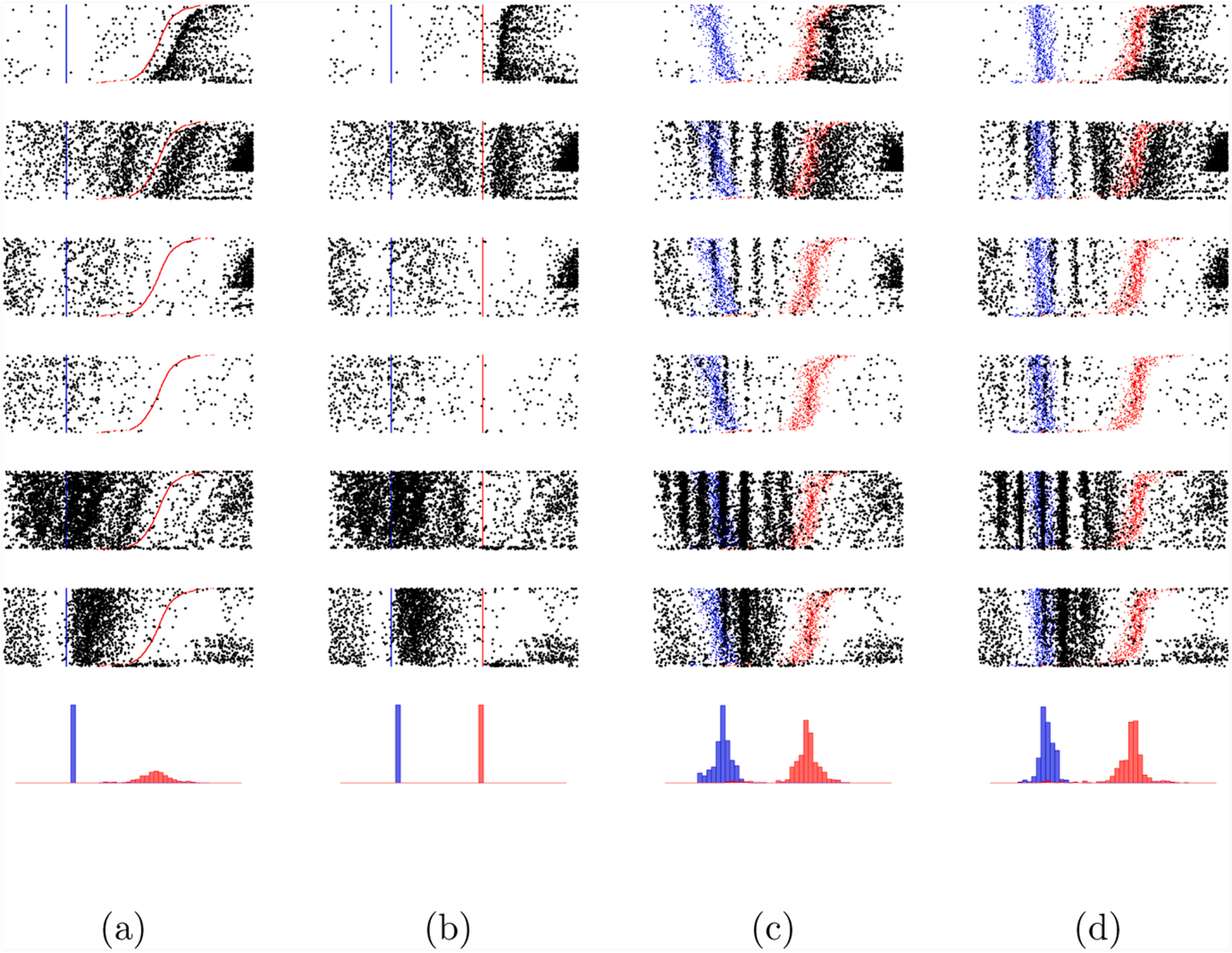

Fig. 9. The motor cortex spike data. Row 1 to row 6: 6 example neurons from the total 30 neurons, where the blue points represent first pressing and the red points represent the second pressing for each trial. Row 7: the histogram of the two pressing time where blue bars are for the first pressing and red bars are for the second pressing. Column (a): the raw data, aligned by the first pressing. Column (b): the data aligned by the two pressing time through a manual linear time warping. Column (c): the data aligned by the shifting model in the original paper (Williams et al., 2020). Column (d): the data aligned by the FRR method.

Table 2
The standard deviation (STD) and interquartile range (IQR) of the distance between taps, tap1 time marker and tap2 time marker with respect to different cases.

Method	Tap1-Tap2 Dis		Tap1		Tap2	
	STD	IQR	STD	IQR	STD	IQR
Raw Data	138.05	153.98	0.00	0.00	138.05	153.98
Shifting Model	138.05	153.98	81.16	90.00	141.40	99.50
FRR Method	141.29	138.12	62.66	81.78	143.85	99.03

target 1 is reached and ends at the time target 1 is reached again, it can be seen that the cyan time markers are already aligned on the two sides in the plots. However, the red, green and blue time markers are distributed randomly. The points are mixed, making it difficult to obtain information from the raw data. To measure the concentration of the time markers, we compute the standard deviation (STD) and interquartile range (IQR) of the times that targets 2, 3 and 4 are reached, respectively, which are shown in Table 3 row 1.

To apply the FRR method, we pre-selected the neurons such that only the neurons with at least 1 spike in every trial, or the neurons that have at least 5 spikes per trial on average, will be used in FRR alignment. In total, 76 out of 110 neurons were selected. Then after the time warping functions were estimated, we applied them on all the neurons to obtain aligned spikes. Different FRR method settings were tried and we finally chose the density approach with binning and cubic smoothing spline to run the FRR alignment. The output raster plots are shown in Fig. 12 (c) and some example time warping functions are shown in Fig. 13 (b). It can be seen that the cyan time markers are unchanged, while the red, green and blue time markers are much more concentrated in comparison with the raw data. The STD and IQR for the middle three time markers are shown in Table 3 row 3. We can see a huge drop in the STD and IQR values with respect to the raw data, which also supports that the time markers become concentrated after the FRR alignment. In addition, from the aligned results, we can see that there are some behavioral patterns in the data. For Fig. 12 (c) row 2, the spikes are mainly within the red and green time markers. Similarly for Fig. 12 (c) row 3, the spikes are mainly within the green and blue time markers. In Fig. 12 (c) row 4, there is a strong signal corresponding to each of the red, green and blue time markers. As the time markers are not aligned in the raw data, these patterns cannot be seen clearly. With this information, we will be able to know which neuron can tell us more information about the subject's action and thus can work better in decoding.

For comparison, we also tried the piecewise linear model. The cross-validation search of the hyper-parameters, as shown in Fig. 14, informed that the number of knots to be 4, the template smoothness penalty to be

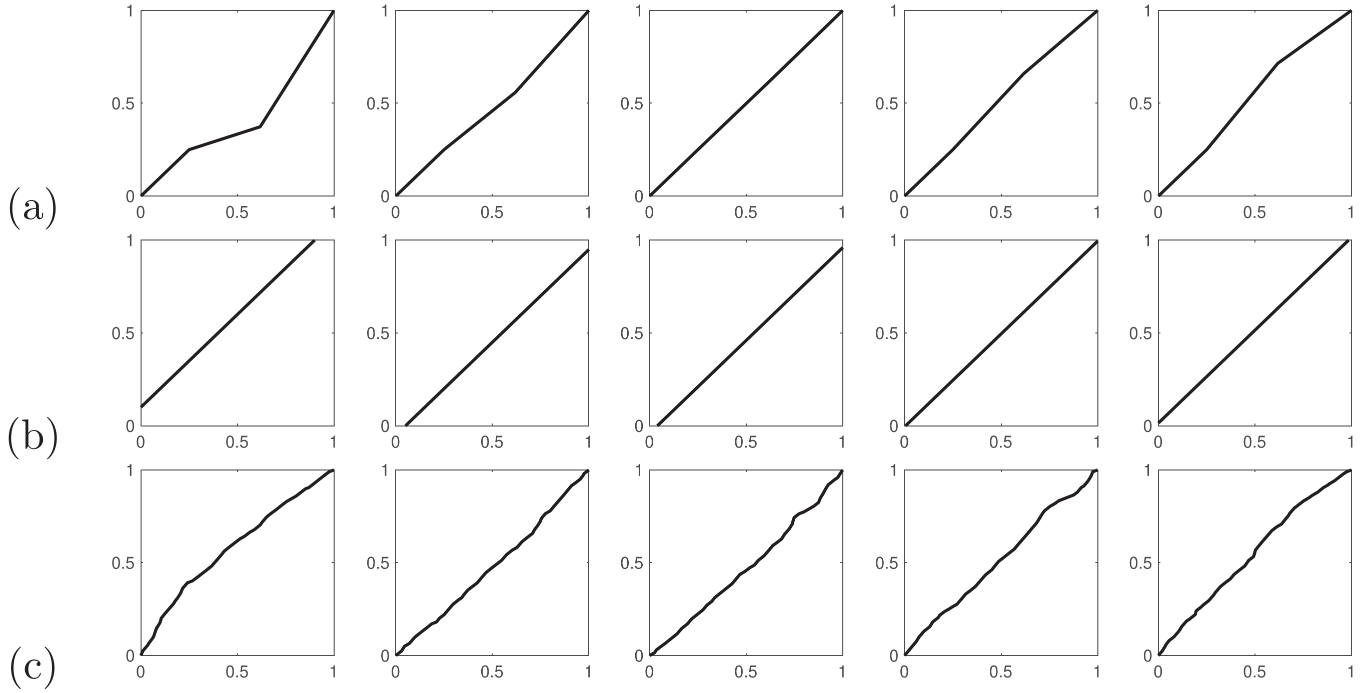

Fig. 10. Illustration of the time warping functions in the alignments. The columns correspond to the time warping functions for trials 1, 100, 300, 500 and 600, respectively. Row (a): by the manual linear time warping. Row (b): by the shifting model in (Williams et al., 2020). Row (c): by the FRR method.

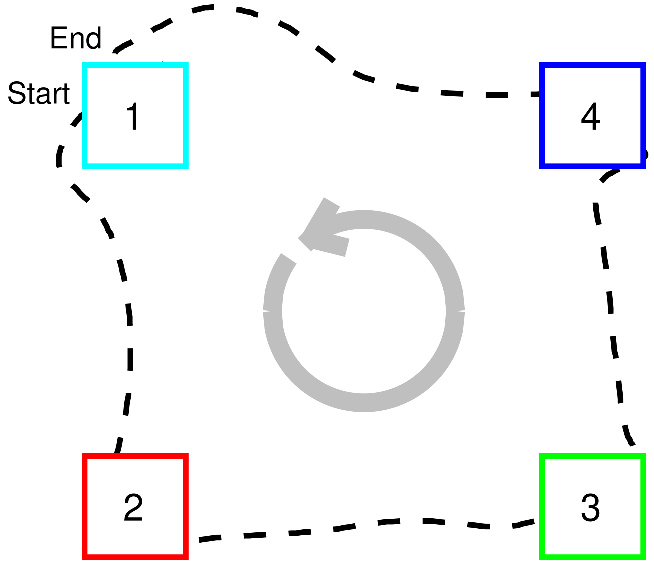

Fig. 11. Illustration of the experiment task. The data used in this analysis have the starting point at the target located on the top left corner (target 1, the cyan square 1). The monkey will move the cursor from target 1 to the target on the bottom left corner (target 2, red box 2), the target on the bottom right corner (target 3, green square 3), the target on the top right corner (target 4, blue box 4) and in the end back to target 1, as shown by the gray circular arrow in the center. The dashed black line is one of the observed trajectory in the experiment, shown as an example.

0.7057, the warping smoothness penalty to be 0.0016, the number of total iterations to be 50 and the number of iterations for warping optimization to be 50 were the optimized setting. The corresponding output plots are shown in Fig. 12 (b), and the STD and IQR values for the red, green and blue time markers are given in Table 3 row 2. Some time warping functions for illustration are given in Fig. 13 (a). As compared to the result of the FRR alignment, both methods have the ability to concentrate the middle three time markers, where the piecewise linear model is better in target 2 (STD 0.18 vs 0.19 and IQR 0.20 vs 0.28) and the FRR method is better in target 4 (STD 0.10 vs 0.20 and IQR 0.14 vs 0.21). However, in the result of the piecewise linear model, the originally aligned cyan time markers become dispersed. Considering how each trial begins and ends, it is difficult to interpret this aligned output. This is because many trials will have starting point or ending point outside the target time interval after alignment. Therefore, although both the FRR method and the piecewise linear model can give comparable alignment results in terms of the times that target 2, 3 and 4 are reached, the FRR method can do much better when the alignment on target 1 time is considered.

4. Summary and future work

The temporal precision in neural spike train data is an important topic, whereas it is still under-explored in the field. In this paper, we proposed a method for spike train data alignment, based on an existing shape data registration framework - the FRR method. In section 2, we firstly reviewed how the FRR method works on functions in $\mathscr{F}_n$ and $\mathscr{F}_n^{(p)}$. Then the framework of alignment through FRR method on spike train data was illustrated and the method assumptions were given. Fig. 1 provides a summary for the overall process of the FRR-based alignment. After a discussion of the alignment framework, we found that the key to the method was to transform the discrete spike time to sampling points of function curves, either the estimated firing rate functions or the estimated probability density functions. We proposed two approaches: binning and kernel density estimation. Binning is to separate the time interval into time bins to compute the function values through counting and then smooth the obtained values to get the estimated curve. Kernel density estimation is to assign a function to each spike and then sum up all the functions. A cross-validation method was also provided for hyper-parameter tuning. At the end of Section 2, we compared the FRR method with commonly used piecewise linear models and showed its three advantages: (1) allow all types of time warpings, (2) invariant on the time domain, and (3) minimal hyper-parameter tuning.

In Section 3, application results of the FRR method were discussed. We firstly applied the method on a simulated dataset, where the true

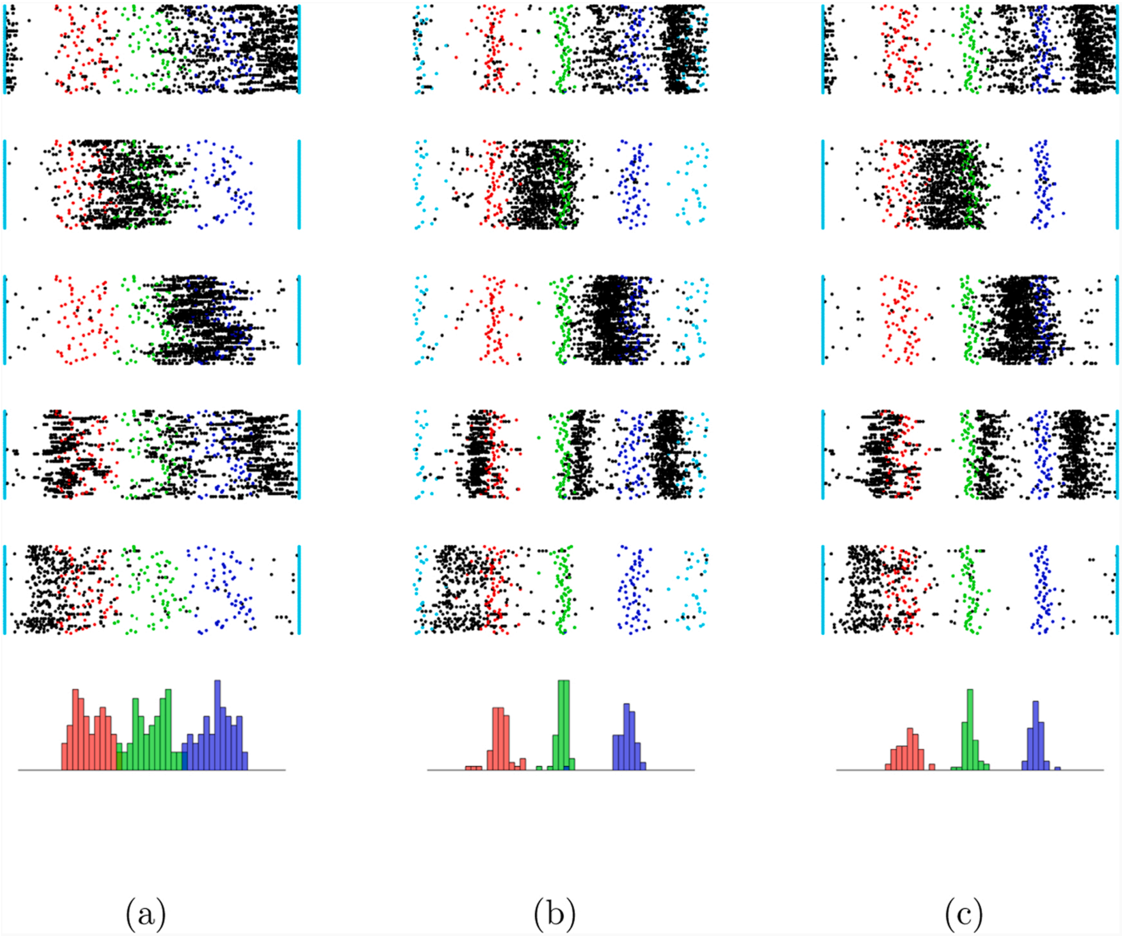

Fig. 12. The monkey decoding data. Row 1 to row 6: 6 example neurons from the total 110 neurons. The cyan, red, green, blue points represent the times targets 1, 2, 3, 4 are reached, respectively. Row 7: the histogram of the time when target 2 (red), 3 (green), 4 (blue) are reached, respectively. Column (a): the raw data, aligned by the time target 1 is reached. Column (b): the data aligned by a 4-knot piecewise linear model. Column (c): the data aligned by the FRR method.

Table 3
The standard deviation (STD) and interquartile range (IQR) of the time reaching target 2, 3 and 4 with respect to different cases.

Method	**Target 2**		**Target 3**		**Target 4**	
	STD	IQR	STD	IQR	STD	IQR
Raw Data	0.29	0.50	0.32	0.54	0.30	0.42
Piecewise Linear Model	0.18	0.20	0.10	0.12	0.20	0.21
FRR Method	0.19	0.28	0.11	0.12	0.10	0.14

firing rate functions and true time warping functions were known. The output showed that the FRR method had the firing rate functions and time warping functions precisely estimated. Although some patterns in the true curves were not caught, in general the shape of the estimated curves was close to the true curves. As a comparison, we also tried the piecewise linear model proposed by Williams et al. (2020)). Both the plots of the estimated curves and the *RMSE* values agreed that the FRR method had better performance in this study. Then we applied the

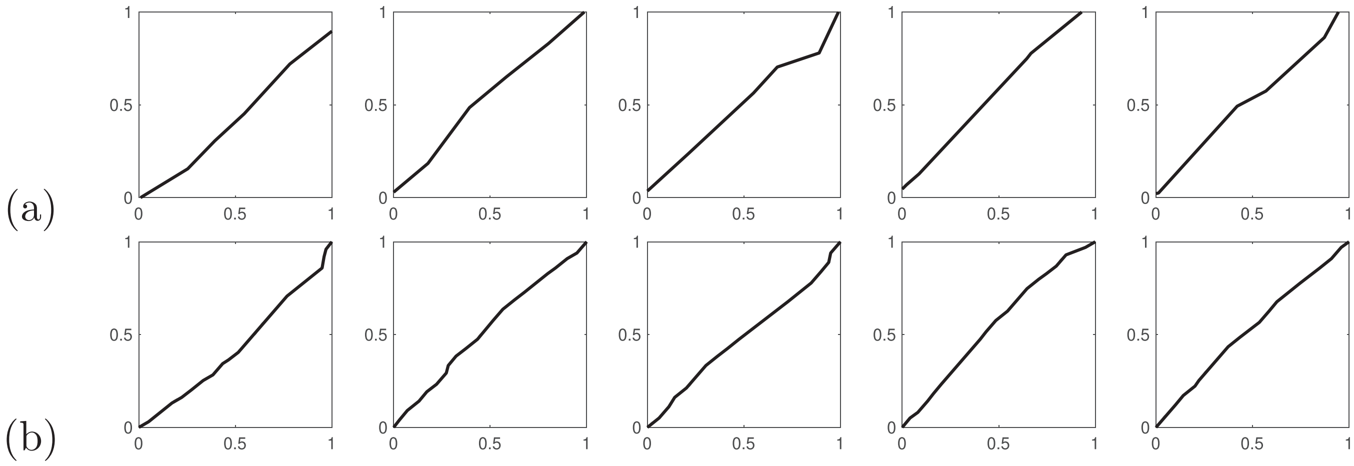

Fig. 13. Illustration of the time warping functions in the alignments. The columns correspond to the time warping function for trials 1, 15, 30, 45 and 55, respectively. Row (a): by the 4-knot piecewise linear model. Row (b): by the FRR method.

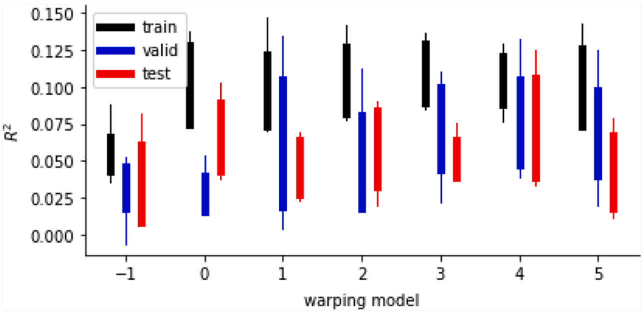

Fig. 14. The hyper-parameter search for piecewise linear model hyper-parameters. The 4-knot model is selected due to its high test R square values.

method on two real-world datasets. For the motor cortex dataset, two time markers were recorded together with the spike data. The raw data and a naive piecewise linear alignment based on the time markers on the raw data both contained no clear pattern. In contrast, the FRR alignment result showed oscillation patterns in many neurons, which agrees to the result of the shifting model. However, the shifting model cannot make the two time markers concentrated, while the FRR method can. This application indicated that the FRR method had the comparable ability to uncover hidden patterns as the piecewise linear model, but the FRR method can also take into consideration the object's major action information, which the piecewise linear model cannot. For the monkey decoding dataset, there were 5 time markers with two to be the starting point and ending point, and three in the middle, coming from 4 targets. It was difficult to obtain information from the raw data, and manually aligning all the time markers was a big challenge. Through the FRR alignment, we had all the time markers concentrated. As a result, patterns in the data can be easily obtained. In comparison, the piecewise linear models were able to align the middle three time markers, while the first and last time markers became dispersed.

In conclusion, the proposed FRR method is an effective way to handle temporal precision problems in spike train data, based on the theoretical properties and applications discussed in this paper. However, the method still has large room to be further improved. At first, the transformation from spike time to function values requests a large enough number of spikes. If the true firing rate is low, it will be unlikely to observe enough signals and in this way, the FRR method will have large errors due to poor estimation of the firing rate or density function. We will explore new methodologies to address this issue. Moreover, the bin size in the binning method and kernel function in the smoothing procedure in this paper are still ad hoc. We will investigate methods to integrate them under one framework. Finally, this paper has focused on the comparison of FRR and piecewise linear models. We will explore comparisons of FRR and other alignment methods on spike trains such as soft-DTW (Cuturi and Blondel, 2017) in the future.

CRediT authorship contribution statement

Zishen Xu: Designed the proposed framework, Conducted computational analysis, Wrote the paper. **Xinyu Zhou**: Assisted with data analysis, Wrote the paper. **Yiqi Xu**: Assisted with data analysis. **Wei Wu**: Designed the proposed framework, Supervised data analysis, Wrote the paper.

Declarations of interest

None.

References

Bair, W., Koch, C., 1996. Temporal precision of spike trains in extrastriate cortex of the behaving macaque monkey. Neural Comput. 8, 1185–1202.

Berndt, D.J., Clifford, J., 1994. Using dynamic time warping to find patterns in time series. KDD Workshop 359–370 (Seattle, WA, USA).

Brette, R., 2015. Philosophy of the spike: rate-based vs. spike-based theories of the brain. Front. Syst. Neurosci. 9, 151.

Bruno, R.M., 2011. Synchrony in sensation. Curr. Opin. Neurobiol. 21, 701–708.

Butts, D.A., Weng, C., Jin, J., Yeh, C.I., Lesica, N.A., Alonso, J.M., Stanley, G.B., 2007. Temporal precision in the neural code and the timescales of natural vision. Nature 449, 92–95.

Cao, Y., Rakhilin, N., Gordon, P.H., Shen, X., Kan, E.C., 2016. A real-time spike classification method based on dynamic time warping for extracellular enteric neural recording with large waveform variability. J. Neurosci. Methods 261, 97–109.

Chi, Z., Wu, W., Haga, Z., Hatsopoulos, N.G., Margoliash, D., 2007. Template-based spike pattern identification with linear convolution and dynamic time warping. J. Neurophysiol. 97, 1221–1235.

Cury, K.M., Uchida, N., 2010. Robust odor coding via inhalation-coupled transient activity in the mammalian olfactory bulb. Neuron 68, 570–585.

Cuturi, M., Blondel, M., 2017. Soft-dtw: a differentiable loss function for time-series. Int. Conf. Mach. Learn., PMLR 894–903.

De Boor, C., De Boor, C., 1978. A Practical Guide to Splines, volume 27. Springer-Verlag, New York.

Denève, S., Machens, C.K., 2016. Efficient codes and balanced networks. Nat. Neurosci. 19, 375–382.

Duncker, L., Sahani, M., 2018. Temporal alignment and latent gaussian process factor inference in population spike trains. bioRxiv, 331751.

Keogh, E.J., Pazzani, M.J., 2001. Derivative dynamic time warping. In: Proceedings of the 2001 SIAM international conference on data mining. SIAM, pp. 1–11.

Lawlor, P.N., Perich, M.G., Miller, L.E., Kording, K.P., 2018. Linear-nonlinear-time-warp-poisson models of neural activity. J. Comput. Neurosci. 45, 173–191.

London, M., Roth, A., Beeren, L., Häusser, M., Latham, P.E., 2010. Sensitivity to perturbations in vivo implies high noise and suggests rate coding in cortex. Nature 466, 123–127.

Reich, D.S., Victor, J.D., Knight, B.W., Ozaki, T., Kaplan, E., 1997. Response variability and timing precision of neuronal spike trains in vivo. J. Neurophysiol. 77, 2836–2841.

Shusterman, R., Sirotin, Y.B., Smear, M.C., Ahmadian, Y., Rinberg, D., 2018. Sniff invariant odor coding. Eneuro 5.

Shusterman, R., Smear, M.C., Koulakov, A.A., Rinberg, D., 2011. Precise olfactory responses tile the sniff cycle. Nat. Neurosci. 14, 1039–1044.

Silverman, B.W., 1986. Density Estimation for Statistics and Data Analysis, volume 26. CRC Press.

Srivastava, A., Klassen, E.P., 2016. Functional and Shape Data Analysis, volume 1. Springer.

Srivastava, A., Wu, W., Kurtek, S., Klassen, E., Marron, J.S., 2011, Registration of functional data using Fisher-Rao metric. arXiv:1103.3817.

Ventura, V., 2004. Testing for and estimating latency effects for poisson and non-poisson spike trains. Neural Comput. 16, 2323–2349.

Wand, M.P., Jones, M.C., 1994. Kernel Smoothing. CRC Press.

Williams, A.H., Poole, B., Maheswaranathan, N., Dhawale, A.K., Fisher, T., Wilson, C.D., Brann, D.H., Trautmann, E.M., Ryu, S., Shusterman, R., et al., 2020. Discovering precise temporal patterns in large-scale neural recordings through robust and interpretable time warping. Neuron 105, 246–259.

Wu, W., Srivastava, A., 2011. An information-geometric framework for statistical inferences in the neural spike train space. J. Comput. Neurosci. 31, 725–748.

Zhao, W., Xu, Z., Li, W., Wu, W., 2020. Modeling and analyzing neural signals with phase variability using Fisher-Rao registration. J. Neurosci. Methods 346, 108954.