Rescuing neural spike train models from bad MLE

Diego M. Arribas **Yuan Zhao** **Il Memming Park**
Department of Neurobiology and Behavior
Center for Neural Circuit Dynamics
Stony Brook University, Stony Brook, NY 11790, USA
{diego.arribas,yuan.zhao,memming.park}@stonybrook.edu

Abstract

The standard approach to fitting an autoregressive spike train model is to maximize the likelihood for one-step prediction. This maximum likelihood estimation (MLE) often leads to models that perform poorly when generating samples recursively for more than one time step. Moreover, the generated spike trains can fail to capture important features of the data and even show diverging firing rates. To alleviate this, we propose to directly minimize the divergence between neural spike trains and model generated spike trains, using spike train kernels. We develop a method that stochastically optimizes the maximum mean discrepancy induced by the kernel. Experiments performed on both real and synthetic neural data validate the proposed approach, showing that it leads to well-behaving models. Using different combinations of spike train kernels, we show that we can control the trade-off between different features which is critical for dealing with model-mismatch.

1 Introduction

Determining the functional relationship between stimuli and neural responses is a central problem in neuroscience. A standard approach is to build a probabilistic generative model and estimate its parameters by maximizing the likelihood of the data under the model. This framework has been applied in diverse scenarios describing the activity of single neurons and coupled populations of neurons, extracting low dimensional latent dynamics underlying the data, and decoding stimuli that produces neural activity. The extracted model parameters are useful as they allow one to gain insights on the relationship between the observed neural activity, its covariates and the intrinsic dynamics.

However, maximum likelihood estimation (MLE) focuses on making the data likely under the assumed model without really assessing the behavior of the actual samples that the model generates. MLE often leads to models that are unstable, operate at unphysiological regimes or generate samples that fail to capture relevant features of the data. This harms model interpretation and it's a big drawback if the obtained model is intended to be used in simulations.

In machine learning, big improvements in generative modeling were achieved when alternative approaches leading to different loss functions were considered. In their original formulation, Generative Adversarial Networks [1] minimize an approximation to the Jensen-Shannon divergence by training a discriminator model that evaluates sample quality. More recent works have proposed to minimize other loss functions such as the Wasserstein distance [2] and the Maximum Mean Discrepancy (MMD) [3–5].

While these works have focused on the use of deep neural networks to generate synthetic images, neuroscience models are usually autoregressive models and they emphasize interpretability. Here we propose to complement likelihood based approaches with MMD minimization for the autoregressive models that are typically used in neuroscience. Using spike train kernels, MMD can evaluate

Preprint. Under review.

different similarity measures between the generative model's samples and the data. We show that the framework is flexible and can be adapted to find models that capture different features of the data.

2 Approach

2.1 Problem Statement – autoregressive models, MLE, and data generation

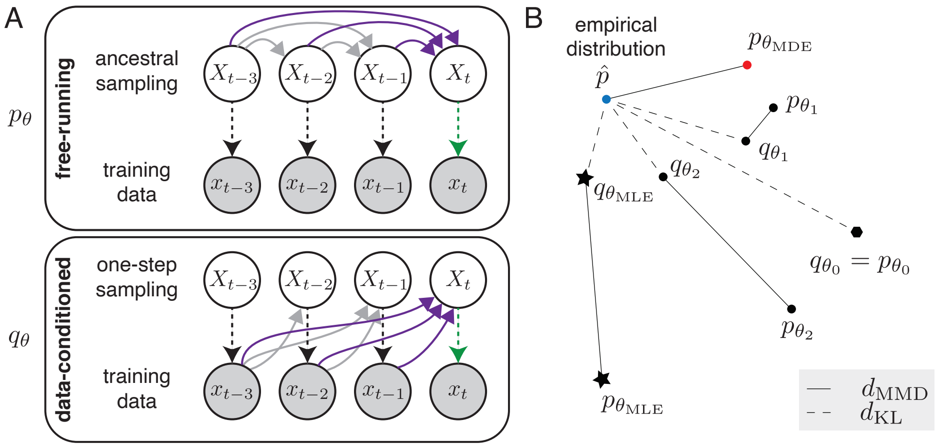

Figure 1: **(A) Two distinct likelihoods.** The free-running corresponds to the joint distribution of the probabilistic model. Solid line denotes conditional dependence and dashed line denotes evaluation of the likelihood. The data-conditioned likelihood always conditions on the actual observed past to predict the next time step. **(B) Cartoon depicting relative closeness of distributions.** The data-conditioned likelihood q_θ is optimized to obtain the maximum likelihood estimate θ_{MLE} which has minimum KL-divergence wrt the empirical distribution $\hat{p}$, but can lead to a model with unrealistic free-running behavior $p_{\theta_{\mathrm{MLE}}}$. Note that d_{KL} is not a metric. See Sec. 2.3 for details.

We denote by x_t an observation at time t and by X_t the corresponding random variable in the stochastic process. Given a sequence of observations or time series, $\{x_t\}_{t\in\mathbb{N}}$, a general autoregressive model predicts X_t based on the past $X_{<t}$, which can be concisely encapsulated as the conditional probability $p(X_t|X_{<t}, u_{<t}, \theta)$, where θ denotes the model parameters and $u_{<t}$ are optional (observed) covariates. The standard maximum likelihood estimation (MLE) procedure yields the parameters that maximizes the likelihood for predicting this one-step prediction,

$$\theta_{\mathrm{MLE}} = \operatorname*{argmax}_{\theta} \prod_{t} p(X_t = x_t | X_t = x_{<t}, u_{<t}, \theta). \tag{1}$$

The disparity between the one-step prediction and longer-term forecasting is common in many autoregressive models, including recurrent neural networks (RNNs) [6–9]. For example, sequences freely generated from natural language models trained to predict the next token typicaly exhibit over-representation of unnaturally long sequences. A fundamental issue common to these problems is the difference between the free-running joint distribution and the data-conditioned joint predictive distributions (Fig. 1A):

$$p(X_t, X_{t+1}, \ldots X_{t+p}|x_{<t}, \theta) = \prod_{s=0}^{s=p} p(X_{t+s}|X_{t:t+s-1}, x_{<t}, \theta) \qquad \text{(free-running)} \tag{2}$$

$$q(X_t, X_{t+1}, \ldots X_{t+p}|x_{<t+p}, \theta) := \prod_{s=0}^{s=p} p(X_{t+s}|x_{<t+s}, \theta) \qquad \text{(data-conditioned)} \tag{3}$$

where $X_{t:t+s-1}$ denotes $\{X_t, X_{t+1}, \ldots, X_{t+s-1}\}$. In equation (2), the autoregressive model's joint distribution is conditioned on its own prediction after time t while in equation (3) the conditioning is always done on the observations as in typical MLE (eq. (1)). This conditioning of data is similar to the *teacher forcing* in the context of recurrent neural networks [10].

For example, for an autoregressive dependency that induces self-excitation, (2) and (3) can behave vastly differently—assuming the observed data are from a stable system, conditioning on the observed history (3) produces stable one-step predictions. However, for the same parameters, if one sampled trajectories using ancestral sampling from (2), runaway self-excitation might be generated. To illustrate this, consider one of the simplest autoregressive models, the *linear* autoregressive model of order p, AR(p):

$$X_t = \sum_{\tau=1}^{p} a_\tau X_{t-\tau} + \epsilon_t \tag{4}$$

where $\{\epsilon_t\}_t$ are white Gaussian noise, and $\theta = \{a_1, \ldots, a_p, \mathrm{var}(\epsilon_t)\}$. For AR($p$) models, the MLE could result in unstable parameter regime of self-amplification when the poles of the linear system are close to the unit circle [11, Ch. 10]. Fortunately, the condition of stability is exactly known for AR(p) models, and an estimator that constrains parameters to lie within the stable regime has been developed [11, Ch. 10]. For instance, for AR(1), $|a_1| < 1$ guarantees stability and stationarity, while $|a_1| > 1$ guarantees instability.

However, beyond those *linear* models, the intractability of the free-running distribution makes it difficult to directly optimize for it, For instance, the autoregressive point process models, often referred to as the generalized linear models (GLMs) in neuroscience [12, 13], suffer from the issue of instability as well:

$$X_t \sim \mathrm{Poisson}\left(\lambda(X_{<t}, u_{<t}, \theta)\right) \tag{5}$$

$$\lambda(X_{<t}, u_{<t}, \theta) = \exp\left(\sum_{\tau=1} h_\tau X_{t-\tau} + \sum_{\tau=1} a_\tau u_{t-\tau} + b\right) \tag{6}$$

where $\theta = \{\{h_\tau\}_\tau, \{a_\tau\}_\tau, b\}$ are the parameters, $\{h_\tau\}_\tau$ and $\{a_\tau\}_\tau$ are referred to as the *history filter* and *stimulus filter*, respectively, and $b \in \mathbb{R}$ is the bias.

Self-excitation in the inferred history filter has been observed on both short and long time scales, with different proposed causes: bistable or overdispersed data [14], periodic or bursting patterns [15], omitted covariates [16], ramping firing rates [17], and lack of data [18]. These history filters have the potential to generate runaway self-excitation. Recently, a number of approaches have attempted a resolution for the point process GLM and related models: Gerhard et al. [19] approximate the free-running distribution over the history using a quasi-renewal process approximation which is later extended by Chen et al. [18]. Rule and Sanguinetti [20] also provide an approximation of this distribution using moment matching. On the other hand, Hocker and Park [21] use Gibbs sampling to obtain the marginalized free-running likelihood for multi-step prediction which is computationally prohibitive. In this paper, we take a fresh stab at this problem.

2.2 Goodness-of-fit measures

To complicate the matter, the goodness-of-fit measures often assume the data-conditioned likelihood. As a result, the standard log-likelihood based measures such as deviance, information, or pseudo-r^2 [13] as well as the quantification of interval distribution using the time-rescaling theorem [22] fail to reliably predict whether the fit model would generate free-running samples similar to the data [19, 21]. We explore various forms of goodness-of-fit measures for GLM-like models which could be also useful for fitting.

A statistical divergence is a non-negative function that quantifies how dissimilar two distributions are [23], thus it can be used as a goodness-of-fit measure: smaller the divergence, the better the fit. Consider the Kullback-Leibler (KL) divergence $d_{KL}(\hat{q} \parallel q(\theta)) := E_{\hat{q}}\left[\log \frac{\mathrm{d}\hat{q}}{\mathrm{d}q(\theta)}\right]$ between the empirical data distribution $\hat{p}$ and the data-conditioned likelihood $q(\theta)$ from (3), where $\hat{p}(\{X_t\}_t) = \prod_t \delta(X_t - x_t)$. In this context, the standard MLE is equivalent to minimizing the KL divergence, that is,

$$\theta_{\mathrm{MLE}} = \operatorname*{argmin}_\theta d_{KL}(\hat{q} \parallel q(\theta)) = \operatorname*{argmax}_\theta q(\{x_t\}_t \mid \{x_t\}_t, \theta) = \operatorname*{argmax}_\theta \prod_t p(x_t \mid x_{<t}, \theta) \tag{7}$$

As our goal is to find a model that can generate time series that resemble the data, it naturally leads to the following minimum divergence estimation (MDE):

$$\theta_{\mathrm{MDE}} = \operatorname*{argmin}_\theta d(\hat{p} \parallel p(\theta)) \tag{8}$$

where d is a divergence and p is the free-running likelihood of (2). From this first principle, our challenge is finding a divergence such that the optimization (8) is computationally feasible. Unlike in the standard MLE, using KL divergence generally results in an intractable objective function.

Here we investigate to use of a widely used kernel-induced statistic called maximum mean discrepancy (MMD) [24]. Given a positive definite kernel k, we can embed a probability measure p in the corresponding reproducing kernel Hilbert space $\mathscr{H}$, i.e., $p \mapsto p_k := \int k(\cdot, x)\, \mathrm{d}p(x) \in \mathscr{H}$. MMD measures the distance between the kernel embeddings,

$$d_{MMD}(p, q) = \|p_k - q_k\|_{\mathscr{H}} = \sup_{f \in \mathcal{F}} \left(\mathop{\mathrm{E}}_{X \sim p}[f(X)] - \mathop{\mathrm{E}}_{X \sim q}[f(X)] \right) \tag{9}$$

where $\mathcal{F}$ is a unit-ball in $\mathscr{H}$ [24]. Depending on the choice of kernel, MMD differentially weighs the features in the input space. Hence, MMD can be tuned to produce goodness-of-fit statistics sensitive to that of the inducing kernel. Importantly, if the kernel k is *characteristic*, the induced MMD is a divergence (and a metric), i.e., it vanishes if and only if the two distributions are identical [25].

2.3 MMD Professor Forcing

Given parameter θ and a time series $\{x_t\}_t$, if the free-running (2) and data-conditioned (3) distributions agree, we might expect $p(\theta)$ to be a good description of $\hat{p}$. If the data is stable, the free-running behavior of the model won't diverge from the data-conditioned behavior. This is the idea introduced by Professor forcing [9] where they included a (KL-)divergence minimization between the hidden states generated by the free-running and teacher forcing modes. Then, MMD based goodness-of-fit that measure the closeness between $q(X \mid x, \theta)$ and $p(X \mid \theta)$ can be used. Note that for autoregressive models, the agreement between the free-running and data-conditioned distributions is trivially achieved if the model does not depend on the history (e.g., Poisson process). Thus this MMDs cannot be optimized on its own. Figure 1B illustrates the ideas here with various possible model parameters and their corresponding conditional distributions. While both q_{θ_1} and q_{θ_2} are both only slightly worse than $q_{\theta_{\mathrm{MLE}}}$ in explaining data, their corresponding free-running distributions can be very dissimilar (e.g., p_{θ_2} and $p_{\theta_{\mathrm{MLE}}}$) or very close (e.g., p_{θ_1}). Informally, if both $d_{\mathrm{KL}}(\hat{p}, q_\theta)$ and $d_{\mathrm{MMD}}(q_\theta, p_\theta)$ are small, we might expect $d(\hat{p}, p_\theta)$ to be small to. Hence, a loss function that optimizes both $d_{\mathrm{KL}}(\hat{p}, q_\theta)$ and $d_{\mathrm{MMD}}(q_\theta, p_\theta)$ can lead to both faithful and stable generative models.

3 Minimizing empirical MMD

Given a kernel k and its associated feature map $\phi : \mathscr{X} \mapsto \mathscr{H}$, we have the kernel trick $\langle \phi(x), \phi(x') \rangle_{\mathscr{H}} = k(x, x')$ and we can write the (squared) MMD between the empirical data distribution $\hat{p}$ and the free running distribution p on $\mathscr{X}$ as,

$$d_{\mathrm{MMD}}(\hat{p}, p)^2 = \| \mathop{\mathrm{E}}_{x \sim \hat{p}}[\phi(x)] - \mathop{\mathrm{E}}_{x' \sim p}[\phi(x')] \|^2_{\mathscr{H}} \tag{10}$$

$$= \mathop{\mathrm{E}}_{x, x' \sim \hat{p}}[k(x, x')] + \mathop{\mathrm{E}}_{x, x' \sim p}[k(x, x')] - 2 \mathop{\mathrm{E}}_{x \sim \hat{p}, x' \sim p}[k(x, x')]. \tag{11}$$

Minimizing MMD is then equivalent to minimizing the difference in the statistics represented by ϕ between the two distributions. Given M samples x drawn from the data $\hat{p}$ and N samples x' drawn from the model distribution p, an unbiased empirical estimator of MMD is

$$\hat{d}_{MMD}(\hat{p}, p)^2 = \sum_{\substack{i=1 \\ i \neq j}}^{N} \sum_{j=1}^{M} \frac{k(x_i, x_j)}{N(N-1)} + \sum_{\substack{i=1 \\ i \neq j}}^{N} \sum_{j=1}^{M} \frac{k(x'_i, x'_j)}{M(M-1)} - 2 \sum_{i=1}^{N} \sum_{j=1}^{M} \frac{k(x_i, x'_j)}{NM} \tag{12}$$

Unlike for MLE, computing $\hat{d}_{MMD}$ involves generating samples from the model and measuring their similarity to the data. We propose to minimize MMD by gradient descent on the model parameters. We provide two different variants that rely on different assumptions on the kernel used, and result in different bounds on the variance of the MMD gradient estimator.

3.1 Minimizing empirical MMD using the score function estimator

In general, the only dependence of MMD on the model parameters is through the expectations over the model's samples in equation (10). For models with tractable likelihood, given a sample x' we can evaluate $p(x'; \theta)$. Using the log-derivative trick [26] (derivation in the Appendix), we can rewrite MMD's gradient as

$$\nabla_\theta\, d_{MMD}(\hat{p}, p)^2 = 2E_{x, x' \sim p}[\nabla_\theta \log p(x'; \theta) k(x, x')] - 2E_{x \sim \hat{p}, x' \sim p}[\nabla_\theta \log p(x'; \theta) k(x, x')]. \tag{13}$$

where $\nabla_\theta \log p(x'; \theta)$ is known as the score function. Then, given M samples x from $\hat{p}$ and N samples x' from p, we can compute a stochastic empirical estimate of MMD's gradient

$$\nabla_\theta \, d_{MMD}(\hat{p}, p)^2 \approx 2 \sum_{\substack{i=1 \\ i \neq j}}^{N} \sum_{j=1}^{M} \frac{\nabla_\theta [\log p(x'_j, \theta)] k(x'_i, x'_j)}{M(M-1)} - 2 \sum_{i=1}^{N} \sum_{j=1}^{M} \frac{\nabla_\theta [\log p(x'_j, \theta)] k(x_i, x'_j)}{NM} \tag{14}$$

This is the score function estimator of MMD's gradient. In principle, this procedure can be used to minimize MMD for arbitrary kernels in the space of spike trains [27].

3.2 Minimizing empirical MMD for model based kernels

The gradient estimator (14) for (squared) MMD relies fully on weighting the gradient of the score function and this may result in large variance and slow convergence [26]. In this section we will propose a less general but more efficient model based MMD formulation. Following the Professor Forcing framework [9], we will encourage a model's free running dynamics to match its data conditioned dynamics. Specifically, we propose to use the MMD induced by model based kernels. Given a model, a data sample x, a model sample x', and a kernel based on the model, we will measure the similarity between the samples by using a feature obtained from the model conditioned on each one of them. Explicitly, we can write the feature map $\phi : x(t) \mapsto \nu_\theta(t) \in \ell_2$ such that the observable $\nu_\theta(t)$ is causal, that is, it only depends on $x(s), s < t$.

We illustrate this idea with an example using the autoregressive GLM of equation (5). Given a spike train x and GLM parameters θ, the model assigns a conditional intensity (CI) $\lambda_t(x, \theta)$. If the CI conditioned on the model's free-running spike trains is similar to the CI conditioned on the data, we might expect the model to match its data conditioned and free running behaviors and improve free running spike trains overall. A natural kernel choice could then be $k(x, x'; \theta) = \langle \lambda(x, \theta), \lambda(x', \theta) \rangle$ resulting in the biased

$$\hat{d}_{MMD}(\hat{p}, p)^2 = \sum_t (\lambda_t^p(\theta) - \lambda_t^{\hat{p}}(\theta))^2 \tag{15}$$

where $\lambda_t^p(\theta)$ and $\lambda_t^{\hat{p}}(\theta)$ are the mean CIs conditioned on the data samples and free running samples respectively at time t, i.e., $\lambda_t^p(\theta) = \mathrm{E}_{x \sim p} \, \lambda_t(x, \theta)$. This particular model based MMD measures the sum of squared difference between mean CIs but in general any differentiable feature obtained by conditioning the model on samples can be used resulting in optimized parameters. Model based MMDs are not characteristic in general; The statistics they can match will always be limited by the used model's capacity and zero MMD does not imply equal distributions.

As model based kernels introduce an explicit dependence of MMD on the optimized model parameters, the optimization procedure is more robust and convergence is improved. We use the partial derivative with respect to the model parameters while keeping the model generated samples fixed.

$$\nabla_\theta \, \hat{d}_{MMD}(\hat{p}, p)^2 \approx \sum_{\substack{i=1 \\ i \neq j}}^{N} \sum_{j=1}^{M} \frac{\nabla_\theta \, k(x_i, x_j; \theta)}{N(N-1)} + \sum_{\substack{i=1 \\ i \neq j}}^{N} \sum_{j=1}^{M} \frac{\nabla_\theta \, k(x'_i, x'_j; \theta)}{M(M-1)} - 2 \sum_{i=1}^{N} \sum_{j=1}^{M} \frac{\nabla_\theta \, k(x_i, x'_j; \theta)}{NM} \tag{16}$$

This provides a more computationally efficient and numerically stable optimization than the estimator in section 3.1. As we explained in section 2.3 the parameters θ have to be optimized using this MMD in combination with a likelihood term.

4 Experiments

We demonstrate in the following experiments that it is possible to minimize MMD as described before in both simulated and real data.

4.1 Toy example: Learning an autoregressive GLM without MLE

To illustrate our formulation we will start by showing we can recover a GLM's parameters by directly minimizing MMD without resorting to MLE. We drew 50 spike train samples from an autoregressive

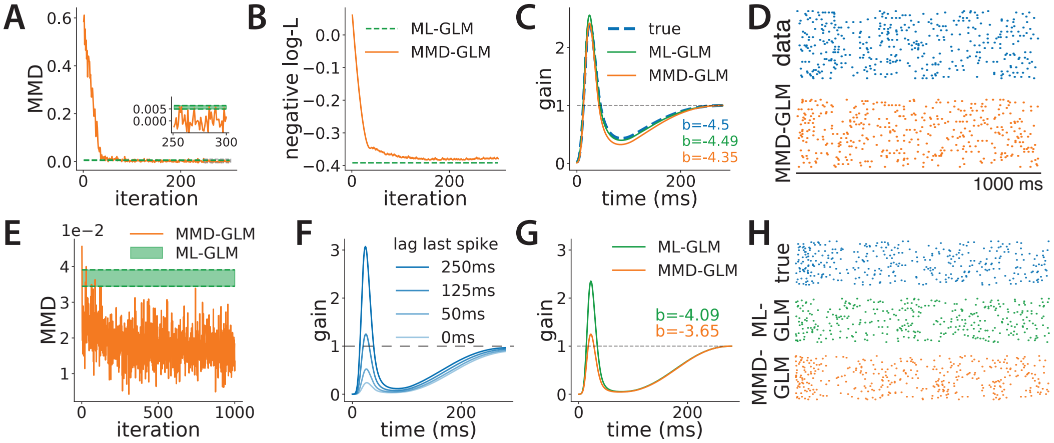

Figure 2: Learning a GLM spike history filter by minimizing MMD. **(A-D) without model mismatch, MLE and MMD minimization agree.** A) MMD estimates during minimization. B) Negative log-likelihood relative to a homogeneous Poisson process with the same rate during training. C) History filter and bias used to generate the data and estimated by MLE and MMD minimization. D) Samples used for ML and MMD training (top) and samples drawn MMD optimized GLM (bottom) **(E-H) with model mismatch, MLE and MMD minimization can disagree.** Data generated from a GQM [28] with time dependent gain shown in (F). Note the mismatch in the inferred history filter in (G).

GLM consisting of a bias and history filter (Figure 2C, D). In this example, we use the kernel

$$k(x, x') = \exp\left(-\frac{1}{\sigma} \int_0^T (I_x(t) - I_{x'}(t))^2 dt \right) \tag{17}$$

where $I_x(t) = \sum_j \Theta(t - t_j^x)$ with $\Theta(t)$ the Heaviside function. This is an example of a characteristic kernel [27] so with sufficient data we expect to find the true GLM parameters if and only if MMD is 0. At each optimization step, we draw 200 samples from the model and update the parameters by computing MMD and its gradient following equations (12) and (14). MMD converges to values around 0 (Figure 2A) and we recover accurate estimates of the true bias b and history filter. We emphasize that the negative log-likelihood decreases during the first part of the optimization although it is not being used. While MLE also retrieves an accurate estimate of the true parameters, the solution found by the two procedures is slightly different due to finite number of samples. MMD is slightly above 0 for the ML-GLM while the likelihood is slightly smaller for the MMD-GLM. In a real application, where there is model mismatch or a non-characteristic kernel is used, the parameters found by MLE and MMD minimization will not be the same. We illustrate this by simulating a model mismatch (Figure 2 E to H). Here we sample from a model in which the history filter depends on the time of the previous spike. The history filter evoked for different lags to the last spike is illustrated in Figure 2. As GLMs can't capture this dependency, MMD induced by the characteristic kernel of equation (17) is different from 0 for MLE (Figure 2). By initializing a GLM at the ML estimated parameters and minimizing MMD further we obtain different parameters. This toy example illustrates that in general MLE and MMD minimization are different and will yield different models.

4.2 Stabilizing GLMs in real data

We show here how our procedure can be used to encourage stable GLM parameters that don't suffer from runaway self-excitation. We use two small datasets used in Gerhard et al. [19] and Chen et al. [18] that are prone to give unstable ML parameters. As MLE is a convex problem for point process GLMs and it is not computationally expensive, we initialized our optimization at MLE and minimize NLL + αMMD i.e. the negative log-likelihood plus an MMD penalty term that acts as a regularizer. We use a model based kernel to reduce noise in the optimization procedure and accelerate convergence. We used the kernel

$$k(x, x'; \theta) = \sum_{\tau} C_{H_x}(\tau) C_{H_{x'}}(\tau) \tag{18}$$

where $\theta = h_{\tau\tau}$ $C_x(\tau) = \sum_{t=1}^{\tau} H_x(t;\theta) H_x(t+\tau;\theta)$ and $H_x(t;h) = \sum_\tau h_\tau x_{t-\tau}$. This kernel measures the similarity between the autocorrelations of two history filter convolved spike trains and therefore will

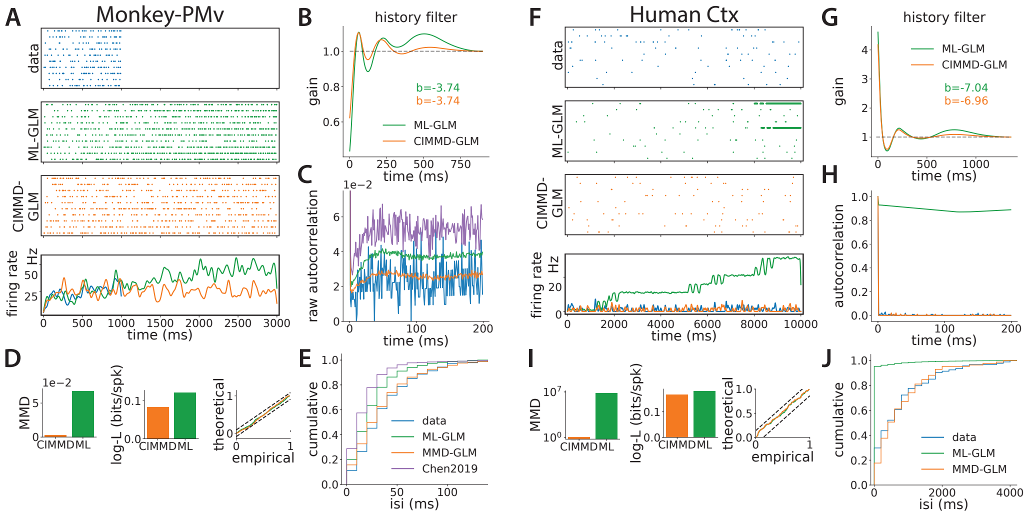

Figure 3: **Multi-objective optimization that combines the likelihood and CI-MMD infers superior generative models.** Data used in [19]. A) 1 second repeated trials frmo Monkey PMv area, model generated spike trains, and corresponding PSTH. Note that ML-GLM's firing rate continuously ramps up. B) Inferred history filters C) Autocorrelation of CIMMD-GLM reproduces that of real data. Purple trace from [18] D) Goodness-of-fits: MMD, normalized log-likelihood, interval statistics. E) CIMMD-GLM best reproduces the full ISI distibution. (F-J) same for Human cortex data.

encourage the autocorrelations of the data conditioned and free running H_t to match. The obtained history filters decrease the amount of late self excitation while preserving the overall shape found by MLE and the samples obtained from the recovered GLMs show stability for both datasets. MMD is greatly reduced at the expense of the likelihood decreasing. For the Human dataset, our method shows notable improvements with a small likelihood reduction. Samples drawn from the resulting models also capture the ISI distribution and the autocorrelation of the data better than MLE.

4.3 Capturing different features in the data

In our last experiment we want to explore how the use of different kernels leads to models with distinct properties. We used a dataset previously used in Hocker and Park [21] and (original reference?) We used 200 trials for training the models and 200 for validating. We initialized all models at MLE and minimized NLL + αMMD. For each optimization we always drew at least 100 trials from the model at each iteration step to compute MMD and its gradient. Figure 4 shows some selected models we obtained for different kernels together with different statistics typically measured in spiking data.

MLE achieves good likelihood values both in the training and validation sets but fails to capture any other features in the data due to self excitation that causes diverging firing rates in a significant number of trials. Although no unstable trials were observed for models 1, 3, 4 and 5 out of the 400 trials generated for each model, we do not guarantee their stability and they could show a small proportion of trials with diverging firing rates if the sampling is repeated. Model 2 showed runaway self excitation for 3 out of 400 trials meaning we know the model is not stable and will produce diverging firing rates with low probability. Although straightforward to compare the difference between conditioned and free running CIs between different models (Figure 4D), a divergence in this quantity was always a signature of trials with diverging firing rates.

Models 1, 2 and 3 used kernels that act directly on the spike train space and were therefore optimized using the score function estimator of section 3.1. Models 1 and 2 were obtained with different penalty strengths using the kernel

$$k(x, x') = \exp\left(- \frac{(\nu_x - \nu_{x'})^2}{\sigma} \right) \tag{19}$$

where ν_x is the mean firing rate of sample x. In theory, this gaussian kernel should enforce the optimized model to match the whole firing rate data distribution. Model 3 was optimized by matching

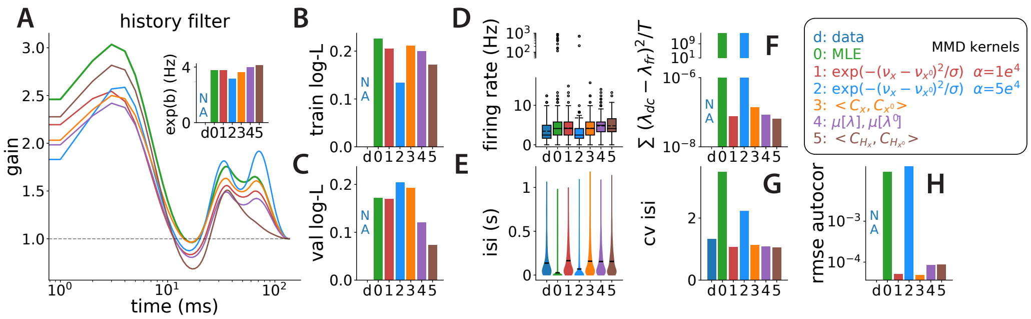

Figure 4: **The use of different kernels leads to different optimized parameters**. A) History filter and bias of the MLE and MMD optimized models. B-C) Training and validation log-likelihoods relative to a homogeneous Poisson process with the rate. D) Mean firing rate distributions. E) ISI distributions. F) Mean squared difference between mean data conditioned and free running CIs G) ISI Coefficient of Variation H) Root mean squared error of the predicted raw spike autocorrelations

the raw autocorrelations $C_x(\tau)$ between the data and model spike trains using the kernel

$$k(x, x') = \sum_{\tau} C_x(\tau) C_{x'}(\tau). \tag{20}$$

Models 4 and 5 used model based kernels and were optimized following section 3.2. Model 4 simply matched the CI baselines using $k(x, x') = (\sum_t \lambda_t(x; \theta))/T(\sum_t \lambda_t(x'; \theta))/T$ and model 5 uses the same kernel introduced in equation (18).

The optimization for models 1 and 3 converged to similar solutions that preserve the overall shape of the ML history filter but reducing the amount of self excitation. These two models were the best at capturing the raw spike autocorrelations in the data. Model 2 showed the biggest late self excitation but the smallest bias and its firing rate diverged for 3 samples. As most of the statistics shown here are very sensitive to the presence of these outliers, model 2 shows bad performance in general. Interestingly, model 2 kernel seems to have encouraged a matching of the whole firing rate distribution having most of its probability mass close to the real data (Figure 4 C). Model 2 also illustrates that the MMD penalty term can act as a regularizer on the likelihood obtaining better likelihood values for the validation data. The models 4 and 5 used model based kernels and optimization converged much faster than for the previous ones. Model 5 is the one that most reduces the late self excitation and in exchange best captures the 1st peak. Overall they showed worse validation likelihoods and spike train autocorrelations. Finally, we would like to note that due to the stochastic nature of the optimization procedure and the presence of local minima, different parameters can be found when repeating the optimization with the same hyperparameters.

5 Discussion

Taking ideas from generative modeling in machine learning, we propose to minimize alternative objective functions to the likelihood as a way to improve sample quality of neural generative models. Here we focused on formulating the framework and exploring the use of different kernels while limiting ourselves to a single model. However, the ideas exposed here can be easily applied to any autoregressive generative model and potentially the benefits could be bigger for more complex models than the point process GLM. In neuroscience the framework can be easily applied to coupled GLMs, rate and spiking neural networks, dimensionality reduction models and decoding.

The main limitation of our proposal is the need to sample and compute kernel similarity during the optimization procedure. Computation can be reduced in many ways. As we did here, MLE can be used to for good parameter initialization and optimization objectives that use MMD and the likelihood may accelerate convergence. MMD minimization updates could also be alternated with likelihood updates so MMD doesn't have to be computed at every step. For the score function estimator of MMD's gradient, there are available methods to control its variance and potentially accelerate convergence.

Broader Impact

Bridging the gap between statistical neuroscientific models such as autoregressive point processes and dynamical systems is substantial challenge not only from the perspective of generative modelling but also in terms of allowing a dynamical interpretation, that carries with it all the niceties that are afforded by stochastic dynamical systems. As such, while the motivation we drew up on comes from neuroscience, modelling, simulating and analyzing point process dynamics has a broad applicability to biological sciences, along with it translational application. Similarly, our method has potential use in modelling within social sciences, geophysics (e.g. earthquakes), astrophysics and finance. In many of those areas stable inference and simulation of future events would directly enable the ability to discern and shape social and economic trends, or effect policy safeguarding against baleful events such as volcano eruptions and stock market instabilities.

Acknowledgments and Disclosure of Funding

This work is supported by NSF CAREER IIS-1845836 and IIS-1734910.

References

[1] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. & Bengio, Y. Generative adversarial nets. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N. D. & Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 27*, pages 2672–2680. Curran Associates, Inc., 2014.

[2] Arjovsky, M., Chintala, S. & Bottou, L. Wasserstein GAN. 2017, 1701.07875.

[3] Dziugaite, G. K., Roy, D. M. & Ghahramani, Z. Training generative neural networks via maximum mean discrepancy optimization. *Uncertainty in Artificial Intelligence - Proceedings of the 31st Conference, UAI 2015*, pages 258–267, 2015, arXiv:1505.03906.

[4] Li, Y., Swersky, K. & Zemel, R. Generative moment matching networks. *32nd International Conference on Machine Learning, ICML 2015*, 3:1718–1727, 2015, arXiv:1502.02761.

[5] Ren, Y., Li, J., Luo, Y. & Zhu, J. Conditional generative moment-matching networks. *Advances in Neural Information Processing Systems*, (2):2936–2944, 2016, arXiv:1606.04218.

[6] Von Zuben, F. J. & Andrade Netto, M. L.de . Second-order training for recurrent neural networks without teacher-forcing. In *Proceedings of ICNN'95 - International Conference on Neural Networks*, volume 2, pages 801–806 vol.2, November 1995.

[7] Bengio, S., Vinyals, O., Jaitly, N. & Shazeer, N. Scheduled sampling for sequence prediction with recurrent neural networks. In Cortes, C., Lawrence, N. D., Lee, D. D., Sugiyama, M. & Garnett, R., editors, *Advances in Neural Information Processing Systems 28*, pages 1171–1179. Curran Associates, Inc., 2015.

[8] Huszár, F. How (not) to train your generative model: Scheduled sampling, likelihood, adversary? pages 1–9, 2015, arXiv:1511.05101.

[9] Goyal, A., Lamb, A., Zhang, Y., Zhang, S., Courville, A. & Bengio, Y. Professor forcing: A new algorithm for training recurrent networks. *Advances in Neural Information Processing Systems*, (Nips):4608–4616, 2016, arXiv:1610.09038.

[10] Williams, R. J. & Zipser, D. A learning algorithm for continually running fully recurrent neural networks. *Neural computation*, 1(2):270–280, June 1989.

[11] Ozaki, T. *Time Series Modeling of Neuroscience Data*. CRC Press, January 2012. ISBN 9781420094602.

[12] Truccolo, W., Eden, U. T., Fellows, M. R., Donoghue, J. P. & Brown, E. N. A point process framework for relating neural spiking activity to spiking history, neural ensemble, and extrinsic covariate effects. *Journal of neurophysiology*, 93(2):1074–1089, February 2005.

[13] Pillow, J. W., Shlens, J., Paninski, L., Sher, A., Litke, A. M., Chichilnisky, E. J. & Simoncelli, E. P. Spatio-temporal correlations and visual signalling in a complete neuronal population. *Nature*, 454(7207):995–999, August 2008.

[14] Weber, A. I. & Pillow, J. W. Capturing the dynamical repertoire of single neurons with generalized linear models. *Neural computation*, 29(12):3260–3289, December 2017.

[15] Østergaard, J., Kramer, M. A. & Eden, U. T. Capturing spike variability in noisy izhikevich neurons using point process generalized linear models. *Neural computation*, 30(1):125–148, January 2018.

[16] Stevenson, I. H. Omitted variable bias in GLMs of neural spiking activity. *Neural computation*, pages 1–32, October 2018.

[17] Park, I. M., Meister, M. L. R., Huk, A. C. & Pillow, J. W. Encoding and decoding in parietal cortex during sensorimotor decision-making. *Nature neuroscience*, 17(10):1395–1403, October 2014.

[18] Chen, Y., Xin, Q., Ventura, V. & Kass, R. E. Stability of point process spiking neuron models. *Journal of computational neuroscience*, 46(1):19–32, February 2019.

[19] Gerhard, F., Deger, M. & Truccolo, W. On the stability and dynamics of stochastic spiking neuron models: Nonlinear hawkes process and point process GLMs. *PLoS computational biology*, 13(2):e1005390, February 2017.

[20] Rule, M. & Sanguinetti, G. Autoregressive point processes as latent State-Space models: A Moment-Closure approach to fluctuations and autocorrelations. *Neural computation*, 30(10): 2757–2780, October 2018.

[21] Hocker, D. & Park, I. M. Multistep inference for generalized linear spiking models curbs runaway excitation. *International IEEE/EMBS Conference on Neural Engineering, NER*, pages 613–616, 2017.

[22] Brown, E. N., Barbieri, R., Ventura, V., Kass, R. E. & Frank, L. M. The time-rescaling theorem and its application to neural spike train data analysis. *Neural computation*, 14(2):325–346, February 2002.

[23] Pekalska, E. & Duin, R. P. W. *The Dissimilarity Representation for Pattern Recognition*. World Scientific Publishing Co., Inc., River Edge, NJ, USA, 2005. ISBN 9789812565303.

[24] Gretton, A., Borgwardt, K. M., Rasch, M., Schölkopf, B. & Smola, A. J. A kernel method for the two-sample-problem. *Advances in Neural Information Processing Systems*, pages 513–520, 2007.

[25] Sriperumbudur, B. K., Gretton, A., Fukumizu, K., Schölkopf, B. & Lanckriet, G. R. G. Hilbert space embeddings and metrics on probability measures. *Journal of machine learning research: JMLR*, 11(50):1517–1561, 2010.

[26] Ranganath, R., Gerrish, S. & Blei, D. M. Black box variational inference. In *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics*, 2014.

[27] Park, I. M., Seth, S., Rao, M. & Príncipe, J. C. Strictly positive definite spike train kernels for point process divergences. *Neural computation*, 24(8):2223–2250, August 2012, arXiv:1302.5964 [q-bio.NC].

[28] Park, I. M., Archer, E., Priebe, N. & Pillow, J. W. Spectral methods for neural characterization using generalized quadratic models. In *Advances in Neural Information Processing Systems (NIPS)*, 2013.