Scalable inference of functional neural connectivity at submillisecond timescales

Arina Medvedeva
Flatiron Institute
New York, NY
amedvedeva@flatironinstitute.org

Edoardo Balzani
Flatiron Institute
New York, NY
ebalzani@flatironinstitute.org

Alex H Williams
Flatiron Institute, New York University
New York, NY
awilliams@flatironinstitute.org

Stephen L Keeley
Fordham University
New York, NY
skeeley1@fordham.edu

Abstract

The Poisson Generalized Linear Model (GLM) is a foundational tool for analyzing neural spike train data. However, standard implementations rely on discretizing spike times into binned count data, limiting temporal resolution and scalability. Here, we develop Monte Carlo (MC) methods and polynomial approximations (PA) to the continuous-time analog of these models, and show them to be advantageous over their discrete-time counterparts. Further, we propose using a set of exponentially scaled Laguerre polynomials as an orthogonal temporal basis, which improves filter identification and yields closed-form integral solutions under the polynomial approximation. Applied to both synthetic and real spike-time data from rodent hippocampus, our methods demonstrate superior accuracy and scalability compared to traditional binned GLMs, enabling functional connectivity inference in large-scale neural recordings that are temporally precise on the order of synaptic dynamical timescales and in agreement with known anatomical properties of hippocampal subregions. We provide open-source implementations of both MC and PA estimators, optimized for GPU acceleration, to facilitate adoption in the neuroscience community[1].

1 Introduction

As recording technologies in neuroscience advance, there is a growing need to improve the scalability of statistical methods for analyzing neural spiking activity. A key challenge in understanding neural computation lies in accurately estimating functional connectivity—the statistical dependencies between neurons that reflect synaptic interactions. The Poisson Generalized Linear Model (GLM) is a powerful tool for this purpose, capable of inferring both stimulus encoding properties and coupling between spiking units. However, the standard implementation of the GLM requires binning the timeseries data into a large design matrix, $\mathbf{X}$, of discrete spike counts. The time resolution of this binning is often coarse ($\sim$ 1 to 10 ms) [1–5] compared to the timescale of synaptic dynamics, which rise and fall at submillisecond timescales [6–8]. This means conventional GLM implementations fail to capture synaptic coupling filters on a biophysically realistic scale [1, 3–5, 9]. Moreover, as the bin size decreases, $\mathbf{X}$ grows in size, posing significant computational and memory storage challenges.

[1]The Poisosn point process GLM code is available at `https://github.com/macari216/poisson-process-glm.git`

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

We find that for most modern neural datasets, storing $\mathbf{X}$ in memory is infeasible, requiring users to batch $\mathbf{X}$, which renders inference unstable even with state-of-the-art optimizers.[2]

Here, we propose methods that avoid these issues by considering the limit of infinitely small time bins, in which case the model becomes a Poisson point process (see e.g. Chapter 19 of [10]). Although point process models have been explored by the neuroscience community [11–17], most prior work either develops theoretical tools for continuous-time models without presenting fitting procedures (e.g., convexity of the log-likelihood [11] or error bounds [12]), or explores related model classes [15, 16], or uses numerical integration methods that do not scale to large datasets [17]; therefore, we limit our benchmark comparison to discrete-time GLM implementations [1, 4, 13, 14]. In our setting of interest, a point process model is able to capture fine-scale spike time correlations between co-recorded neurons, which can be indicative of monosynaptic connections [6, 7]. Furthermore, inputs to the model can be represented as a sequence of spike times instead of a large design matrix. However, to fit the point process model, we must numerically approximate an analytically intractable integral that appears in the likelihood function. We provide two approaches to deal with this integral: 1) a Monte Carlo sampling-based approach (MC) and 2) a second-order polynomial approximation, inspired by prior work [4, 18, 19]. Both methods demonstrate improvements in accuracy over conventional approaches while maintaining computational tractability. Additionally, the polynomial approximation yields a closed-form expression for the Poisson log-likelihood that is quadratic in the GLM parameters, enabling fast and efficient computation. We also propose generalized Laguerre polynomials scaled by an exponential as a new set of basis functions for GLM inference. While these polynomials retain the desirable temporal smoothing properties of the traditionally used raised cosine basis [20, 21], they offer orthogonality and closed-form integral solutions, enabling efficient filter identification.

We validate our models on both simulated and real spiking data. In simulations, we find that both MC and PA approaches scale favorably in compute time with recording length and population size, and show improved filter recovery compared to both the discrete polynomial approximate method and traditional GLMs. We then apply our method to real spiking data, where we analyze spike-time recordings from multiple rodent hippocampal regions [22] in a dataset whose size is computationally prohibitive for traditional batched GLMs. We show that recovered coupling filters align with empirical cross-correlograms (CCGs) with sub-millisecond temporal precision, suggesting the model is able to accurately identify monosynaptic coupling between neurons. In addition, we are able to use our model to isolate specific coupling filters that identify putative excitatory connections in the rodent hippocampus. We show that these isolated filters coincide with anatomical connectivity structure that is well-established in studies of hippocampal anatomy [23, 24], suggesting GLMs operating at this resolution provides new opportunities in the identification of neural circuitry from spike-train recordings.

2 Background

2.1 Discrete-time Poisson GLMs

Generalized linear models provide a useful tool for predicting spiking activity of a single neuron $\boldsymbol{y} = (y_1, \ldots, y_T)$ given recent population spiking activity or external stimuli $\boldsymbol{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T)$, and a set of model parameters $\boldsymbol{w}$. The spike counts y_t are conditionally Poisson distributed, $y_t \sim \text{Poisson}(y_t|\boldsymbol{w}, \boldsymbol{x}_t)$, and the model log-likelihood is written as:

$$\log p(\boldsymbol{y} \mid \boldsymbol{X}, \boldsymbol{w}) = \sum_{t=1}^{T} y_t \log(\Phi(\boldsymbol{x}_t^T \boldsymbol{w})) - \Phi(\boldsymbol{x}_t^T \boldsymbol{w}) \tag{1}$$

where $\Phi(\boldsymbol{x}_t^T \boldsymbol{w})$ is the predicted firing rate at time bin t and $\Phi : \mathbb{R} \to \mathbb{R}$ is a monotonically increasing, convex, and nonnegative function (e.g., exponential or softplus). The central goal of the Poisson GLM, in the identification of $\boldsymbol{w}$, is to find smooth time-varying statistical dependencies between either external stimuli or individual neuronal spike trains and post-synaptic firing rates in a neural population (Fig 1**A**). These filters are typically estimated using a linear combination of a small number

[2]While one can in principle represent $\mathbf{X}$ in a sparse matrix format to alleviate computational burden, there is currently limited support for sparse matrix routines in libraries that are compatible with modern GPUs.

of smooth basis functions and a nonlinearity to assure non-negative firing rates. The filters within neural populations reflect temporally delayed correlated firing, so called "functional connectivity," and are often thought of as a proxy to anatomical synaptic connections, reflecting how populations of neurons influence each other through either excitatory or inhibitory dynamics.

Throughout this work, we will focus primarily on estimating functional connectivity filters using the GLM, and we will use the exponential nonlinearity, $\Phi(\cdot) = \exp(\cdot)$, as this is a common choice in neuroscience and simplifies the log-likilhood objective. However, all of the methods here trivially work with an augmented $\mathbf{X}$ to include stimuli, and with alternative nonlinearities, such as softplus, which is another common choice in the field (see Supplement S.4 for more details).

The traditional approach described above requires discretization of the time series, with a bin size commonly chosen within the range from hundreds of milliseconds to one millisecond, depending on the system and stimulus (features) [1, 4, 25]. However, if the goal is to identify functional monosynaptic connections between neurons, which is a common motivation in modern GLMs, even 1 ms resolution is not sufficient. Electrophysiological recordings in experimental neuroscience have shown that synaptic dynamics are often highly transient, with the rise and fall in firing occurring within 1–5 ms following a presynaptic spike [26, 7]. This means that even bin sizes as small as 1 ms fail to accurately identify peak amplitude and timing (Fig 1**B** and **C**), which may be important for cell-specific synapse properties or distinguishing correlation firing patterns from synaptic activity.

For discrete-time GLMs, sampling at finer than 1 ms resolution demands prohibitively large memory allocations. The dimensionality of the feature space $\mathbb{R}^{NJ}$ depends on the number of neurons N in the recording and the number of basis functions J used to describe each neuron's activity history. For a given dataset, this results in a design matrix $\boldsymbol{X} \in \mathbb{R}^{T \times NJ}$. For long recordings from a large number of neurons, computing and storing this design matrix with a sufficiently small bin size becomes non-trivial. As shown in Fig 1**E**, simulating a dataset of 200 neurons at 1 ms or .1 ms resolutions for 10-100 minutes would require an $\mathbf{X}$ matrix of 10^{10}–10^{12} bits, necessitating batched gradient calculations. In contrast, storing only spike times drastically reduces memory usage, making GLM computations far more tractable for modern high-resolution (submillisecond) datasets.

While batching the design matrix $\mathbf{X}$ for discrete-time Poisson GLM optimization is a sensible approach, it poses significant problems when practically fitting the model. In particular, due to the sparse firing patterns of neural activity, the variance in gradients across batches can be very large. Even when implementing a state-of-the-art stochastic variance-reduced gradient (SVRG) optimization which guarantees an unbiased gradient estimates and minimal memory overhead [27], we find that in practice the variance of our updates is too large to achieve good fits as compared to discrete GLMs using small enough datasets to not require batching (Fig 3,4). Consequently, batched approaches are not only quite slow—requiring, for example, 5 hours on a dataset of 250 neurons with recording length 1000 seconds binned at 0.1 ms resolution—but they can lead to inaccurate model fits.

2.2 The Polynomial-Approximate GLM

Previous work has shown that approximating the nonlinearity in the Poisson likelihood with a polynomial can be effective tool for scaling GLMs [18, 19, 4]. These approaches use an orthonormal set of Chebyshev polynomials which provide a good approximation to GLM non-linearities over a wide range of values, and are effective even for just second order polynomial approximations [18]. Considering the exponential nonlinearity, the approximation can be written as $\exp(x)\Delta = a_2x^2 + a_1x + a_0$, where Δ is the time bin size and a_2, a_1, a_0 are the optimal Chebyshev coefficients that minimize the mean squared error between the nonlinearity and quadratic approximation across the specified range $[x_0, x_1]$.Using this approximation, the GLM log-likelihood can be written as:

$$\log p(\boldsymbol{y} \mid \boldsymbol{X}, \boldsymbol{w}) \approx \sum_{t=1}^{T} \boldsymbol{w}^\top \boldsymbol{x}_t^\top (y_t - a_1 \mathbf{1}) - a_2 \boldsymbol{w}^\top \boldsymbol{x}_t^\top \boldsymbol{x}_t \boldsymbol{w} \tag{2}$$

where terms that do not depend on $\mathbf{w}$ are dropped, and $\mathbf{1}$ is a vector of ones. Because the log-likelihood is quadratic in the parameters, one can directly compute a maximum a posteriori (MAP) estimate using the sufficient statistics ($\sum_{t=1}^{T} \boldsymbol{x}_t$, $\sum_{t=1}^{T} y_t \boldsymbol{x}_t$, and $\sum_{t=1}^{T} \boldsymbol{x}_t \boldsymbol{x}_t^\top$). For more information on this approach, see [4].

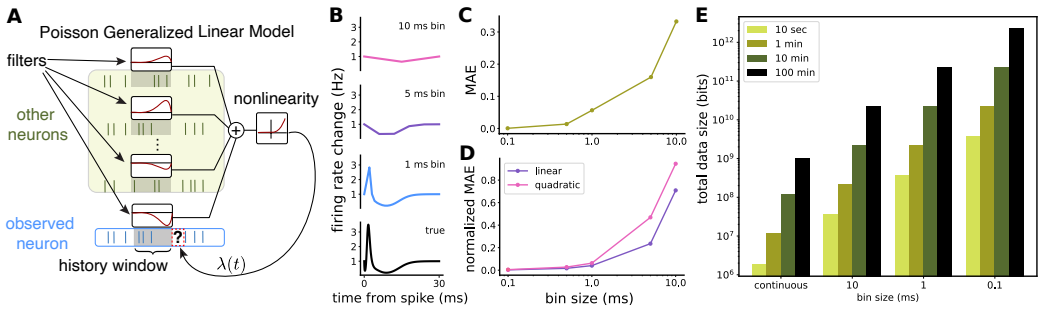

Figure 1: **A** Schematic of GLM for neuronal filter identification; **B** Simulation of realistic timescale post-synaptic conductance change as estimated by a GLM binned at 10, 5 and 1 ms bins; **C** Mean absolute error (MAE) on filter accuracy from **B** at various bin sizes; **D** Normalized error of discrete-time sufficient statistics from continuously generated Poisson rates estimated using various bin sizes; **E** Memory storage of spike times and $\mathbf{X}$ for 200 neurons at various recording lengths and bin sizes.

We find that the second-order polynomial approximation is helpful in significantly reducing the computational time of the GLM, but batched sufficient statistics can still carry a large computational load and can be time-consuming on datasets with fine temporal resolution. Moreover, the binning of the design matrix introduces an error in the estimation of the linear and quadratic sufficient statistics that accumulates with increasing number of spikes in the recording (Fig 1**D**) (see Supplement S.6.2 for more details).

3 The Poisson process GLM model

To improve the scalability and accuracy of these traditional GLM approaches, we instead consider a continuous-time Poisson Process GLM log-likelihood given by:

$$\log p(\boldsymbol{y} \mid \boldsymbol{X}, \boldsymbol{w}) = \sum_{k=1}^{K} \log \lambda(y_k) - \int_0^T \lambda(t)\, dt \tag{3}$$

Here, a time-varying Poisson rate $\lambda(t)$ is evaluated at time points designated by observed spike times y_k of the post-synaptic neuron $\boldsymbol{y} = (y_1, \ldots, y_K)$, and the second term integrates the rate over the duration of the entire recording $[0, T]$. The firing rate at time t is then given by:

$$\lambda(t; \boldsymbol{X}, \boldsymbol{w}) = \Phi\left[\sum_{\boldsymbol{x}_s \in \mathcal{X}(t,H)} \boldsymbol{w}_{n_s}^\top \boldsymbol{\phi}(t - t_s)\right] \tag{4}$$

where $\mathbf{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_S)$ represents the full set of S spikes and each spike $\boldsymbol{x}_s = (n_s, t_s)$ indicates that neuron $n_s \in 1, \ldots, N$ fired at time t_s; $\mathcal{X}(t, H)$ denotes the set of spikes occurring in the history window $[t - H,\, t]$; $\boldsymbol{w}_{n_s} \in \mathbb{R}^J$ is a subset of weights associated with neuron n_s; and $\boldsymbol{\phi} : [0, H] \to \mathbb{R}^J$ denotes a nonlinear mapping onto J temporal basis functions. In this work, we select history window length H of 4-6 ms to encompass expected neuronal dynamical effects. While $\mathbf{X}$ can be easily augmented to include external stimuli, here we restrict our analysis to spike history, primarily focusing on the role of neural interactions and intrinsic dynamics at synaptically relevant timescales.

Given that the intensity function $\lambda(t)$ is defined analytically, the first term in the Poisson process log-likelihood can be computed exactly. However, the nonlinearity Φ makes the cumulative intensity function (CIF) $\int_0^t \lambda(\tau)\, d\tau$ intractable, and thus the second term of the log-likelihood requires approximation. Here, we propose two methods to approximate this integral: 1) a Monte Carlo sampling-based approach (MC) with an unbiased estimator for the CIF; and 2) a polynomial approximation (PA) that yields an expression quadratic in the GLM parameters, independent of bin size or recording length.

3.1 Monte-Carlo sampling for the CIF

To compute the second term in the objective function, $\int_0^T \lambda(t)dt$, we approximate the integral with a Monte Carlo estimate. Instead of simple uniform sampling, we employ stratified sampling: the time support $[0, T]$ is divided into M equal subintervals, and sample points $\boldsymbol{\tau} = (\tau_1, \ldots, \tau_M)$ are drawn uniformly from each subinterval. Then,

$$\frac{T}{M} \sum_{m=1}^{M} \lambda(\tau_m) \approx \int_0^T \lambda(t)dt \tag{5}$$

provides an unbiased estimator of the integral that exhibits lower variance compared to uniform Monte Carlo sampling (see Chapter 8 in [28]). Thus, our loss function for a fixed sample of $\boldsymbol{\tau}$ is:

$$f(\boldsymbol{w}, \boldsymbol{\tau}) = \frac{T}{M} \sum_{m=1}^{M} \lambda(\tau_m) - \sum_{k=1}^{K} \log \lambda(y_k) \tag{6}$$

Where the second term can be computed exactly. We can employ standard gradient-based optimization procedures on this objective selecting a different $\boldsymbol{\tau}$ at every iteration.

3.2 The Polynomial-Approximate continuous GLM

Alternatively, we can use a polynomial approximation method inspired by Zoltowski and Pillow [4] and Huggins et al. [18] to derive a tractable, scalable form for the log-likelihood's CIF. By fitting a second-order polynomial with coefficients a_2, a_1, a_0 to minimize the mean squared error (MSE) against the true nonlinearity over a specified range, we reformulate the objective into a sum of integrals over linear terms (individual basis functions) and quadratic terms (basis function pairs). Depending on the choice of basis functions, these integrals may admit analytic solutions, enabling efficient evaluation of the log-likelihood. The polynomial-approximate CIF is written as:

$$\begin{aligned}
\int_0^T \lambda(t)dt &= \int_0^T \Phi\left(\sum_n \sum_{t_s \in \mathcal{X}_n} \boldsymbol{w}_n^\top \boldsymbol{\phi}(t - t_s) \right) dt \\
&\approx a_2 \int_0^T \left(\sum_n \sum_{t_s \in \mathcal{X}_n} \boldsymbol{w}_n^\top \boldsymbol{\phi}(t - t_s) \right)^2 dt + a_1 \int_0^T \sum_n \sum_{t_s \in \mathcal{X}_n} \boldsymbol{w}_n^\top \boldsymbol{\phi}(t - t_s)\, dt + T a_0 \\
&= a_2 \boldsymbol{w}^\top \mathbf{M} \boldsymbol{w} + a_1 \mathbf{m}^\top \boldsymbol{w} + T a_0
\end{aligned} \tag{7}$$

Here, $\mathcal{X}_n$ denotes the set of spikes from neuron n and the linear term includes a defined vector $\mathbf{m} \in \mathbb{R}^{NJ}$ that contains N concatenated $\boldsymbol{\varphi}$ vectors scaled by respective total number of spikes per neuron, S_n: ($\mathbf{m} = S_1\boldsymbol{\varphi}, S_2\boldsymbol{\varphi} \ldots S_N\boldsymbol{\varphi}$), where $\boldsymbol{\varphi}$ is a vector of precomputed integrals for each of the J basis function over $\tau = t - t_s$. That is, $\varphi_j = \int_0^H \phi_j(\tau)d\tau$.

The quadratic term is a symmetric block matrix $\mathbf{M} \in \mathbb{R}^{NJ \times NJ}$ with $N \times N$ blocks of size $J \times J$. Each block $\mathbf{M}_{n,n'}$ corresponds to a neuron pair (n, n') and accumulates the contributions from all spike pairs $(t_s, t_{s'})$ with $t_s \in \mathcal{X}_n$ and $t_{s'} \in \mathcal{X}_{n'}$. The entry at position (j, j') of the block is given by:

$$[\mathbf{M}_{n,n'}]_{j,j'} = \sum_{\substack{t_s \in \mathcal{X}_n \\ t_{s'} \in \mathcal{X}_{n'}}} \int_{\delta_{t_s,t_{s'}}}^{H} \phi_j(\tau)\phi_{j'}(\tau - \delta_{t_s,t_{s'}})\, d\tau, \tag{8}$$

where $\delta_{t_s,t_{s'}} = |t_s - t_{s'}|$ is the spike time difference. This integral is nonzero only when $\delta_{t_s,t_{s'}} \leq H$, i.e., when the spike pair is within the interaction window. Therefore, if these basis function products can be expressed analytically and integrated in closed form, we only need to compute all pairwise spike time differences within the window $[t_s - H,\, t_s]$ and sum the $J \times J$ integral evaluations.

Given the quadratic expression of the CIF, the first term of the log-likelihood can be computed exactly when using the exponential inverse link function. The contributions from presynaptic spikes are pre-computed as neuron-specific vectors $\boldsymbol{\psi}_n = \sum_{k=1}^{K} \sum_{t_s \in \mathcal{X}_n(y_k, H)} \boldsymbol{\phi}(y_k - t_s)$, yielding the compact

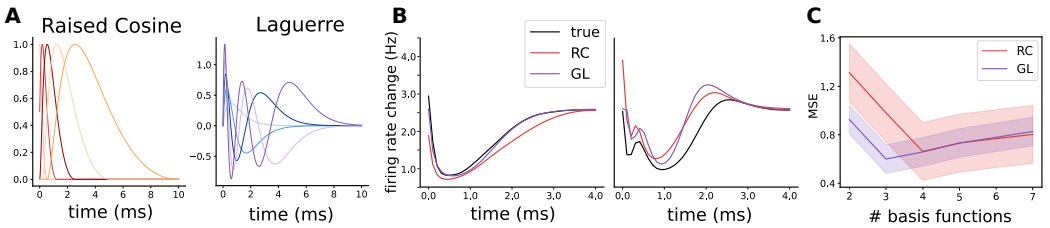

Figure 2: **A** Visualization of the first 5 RC and GL basis functions; **B** Best-performing fits of both bases onto filters generated from 100 RC bases; **C** Error on filter reconstruction for varying number of bases for both models.

form $\sum_{n=1}^{N} \boldsymbol{w}_n^\top \boldsymbol{\psi}_n = \boldsymbol{w}^\top \mathbf{k}$ where $\mathbf{k} \in \mathbb{R}^{NJ}$ concatenates all $\boldsymbol{\psi}_n$. Now, the full log-likelihood can be approximated as:

$$\begin{aligned} \log p(\mathbf{y} \mid \mathbf{X}, \boldsymbol{w}) &= \sum_{k=1}^{K} \log \lambda(y_k) - \int_0^T \lambda(t) dt \\ &\approx \boldsymbol{w}^\top (\mathbf{k} - a_1 \mathbf{m}) - a_2 \boldsymbol{w}^\top \mathbf{M} \boldsymbol{w} \end{aligned} \tag{9}$$

which admits a closed-form solution for model parameters $\boldsymbol{w}$. For additional details, the full derivation of the quadratic polynomial approximation to the Poisson process log-likelihood and its extension to non-canonical link functions (e.g., softplus), please refer to Supplement S.4 and S.6.

We define the approximation range for the nonlinearity Φ based on estimates of the postsynaptic neuron's firing rate. In simulations, where ground-truth binned firing rates are available, the approximation range is set between the 2.5th and 97.5th percentiles of these rates, mapped back through the inverse link function (i.e., $\log(\cdot)$ when $\Phi = \exp$). For real data, where firing rate distributions are not directly accessible, we center the range at the inverse of the mean firing rate and determine its bounds by maximizing cross-validated log-likelihood, following the approach of [4]. In our analyses of neural recordings, we use an approximation interval spanning 3–7 Hz around the mean rate. Notably, wider intervals accommodate more variability in the estimated filter amplitudes but increase approximation error. As a result, polynomial approximation methods produce higher error when estimating the true underlying filters (simulated data) or CCGs (real data, see Figs. 4**B**, **D**, and 5**B,C**).

3.3 Generalized Laguerre polynomials as basis functions

We propose using scaled generalized Laguerre (GL) polynomials as basis functions for GLM temporal filters. Unlike raised-cosine (RC) bases, these functions are orthogonal under the weight $t^\alpha e^{-t}$ and thus can provide more efficient representation of filter variability with fewer basis functions [29]. These polynomials have the added feature of following an approximate gamma-function envelope, in line with fine time-scale rises and slow decays that correspond to biophysical synaptic and neuronal dynamics (Fig 2**A**). The parameter $\alpha > -1$ controls the long time-scale delay of the filter, $\alpha = 0$ yielding standard Laguerre polynomials. We additionally add a coefficient c to the input variable t that scales the rise-time of the bases. We set $c = 1.5$ and $\alpha = 2$ throughout the manuscript based on initial model exploration, but find that varying these values does not dramatically change model performance (Fig. S2**D**).

These orthogonal polynomials better capture filters in fewer basis functions than the standard RC basis. We demonstrate this on a simulated all-to-one coupled GLM whose filters are generated from 100 raised cosine bases.We simulate an 8-neuron population over a 1000-second recording, with the postsynaptic neuron's baseline firing rate set to 3 Hz. On these data, we fit the continuous MC GLM using either the standard RC or GL sets of 2-7 bases. We find coupling filters are better matched using GL in fewer bases functions, with the best performing model being 3 GL bases. Figure 2**B** shows filter matches using 3 GL and 4 RC bases, and 2**C** shows the mean error $\pm$ standard deviation across all simulated filters. For more details on the properties of the generalized Laguerre basis and comparison to RC, refer to Supplement S.5.

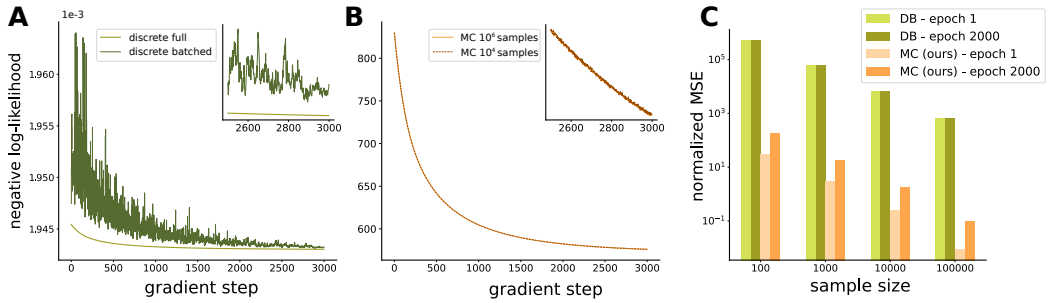

Figure 3: **A** First 3000 evaluations of negative log-likelihood objective on simulated data for batched and full discrete GLM with batch size = 10^4 time bins; **B** Same as **A** for the continuous GLM with MC optimizer for sample sizes comparable to the evaluations in **A**; **C** Normalized MSE of the stochastic gradients relative to the full gradient at the beginning and end of the optimization procedure for discrete and MC models, across different batch and sample sizes.

These bases also have the advantage of admitting straightforward closed-form solutions for both single and pairwise product basis function integrals. Given a generalized Laguerre polynomial of degree n and parameter α, noted by $L_n^{(\alpha)}(ct)$, integrals I_n of these bases have the form:

$$I_n = \int_0^H L_n^{(\alpha)}(ct) ct^{\alpha/2} e^{-ct/2}\, dt = \sum_{k=0}^{n} C_n \int_0^H t^{k+\alpha/2} e^{-ct/2}\, dt \tag{10}$$

where C_n is a polynomial constant that depends on n and α. This admits exact integration via the lower incomplete gamma function $\gamma(a, x) = \int_0^x t^{a-1} e^{-t}\, dt$, with similar closed-form solutions available for pairwise basis function evaluations (see Supplement S.5 for derivations). While our polynomial approximation framework does not strictly require analytical solutions—as numerical integration remains computationally efficient—we found that using these closed-form expressions yielded optimal performance in both accuracy and speed for our implementation. For the remainder of this work, we run all simulations with 100 RC bases and fit all models with 3 to 5 GL bases.

4 Experiments

4.1 Stochastic gradient variance in discrete and continuous GLM

We first show that a naive approach to implementing traditional GLMs on modern datasets—batching the design matrix $\mathbf{X}$—fails to converge to the optimum due to high variance of gradient estimates across batches. The discrete batched (DB) approach performs parameter updates on small subsets of data, resulting in highly inaccurate gradients. When comparing DB to the full approach (on datasets small enough for the full design matrix $\mathbf{X}$ to fit in memory), we find that the GLM log-likelihood converges poorly under gradient descent in the batched case, failing to reach the global optimum achieved by the unbatched version (Fig. 3**A**). This gradient variability is a function of batch size, but even for batch sizes that push memory limits, gradient error remains prohibitively high on large datasets (Fig 3**C**). We therefore look to other approaches for scaling GLMs to large datasets.

Our Monte Carlo (MC) approach also introduces stochasticity in gradient estimates as different samples approximate the CIF integral. However, this variability is substantially lower than that of the discrete batched approach, resulting in much more stable inference with better log-likelihood values (Fig. 3**B**). This improved stability arises from two key differences: first, the spike term (first term in the log-likelihood) is always computed exactly over all observed spikes rather than a subset; second, although MC sample size affects the accuracy of the CIF integral estimate (the second term), stratified sampling ensures uniform coverage of the entire recording duration. In Fig. 3**C**, we quantify the resulting improvement in gradient accuracy by computing the expected squared error between the true and stochastic gradients, normalized by the squared norm of the initial gradient:

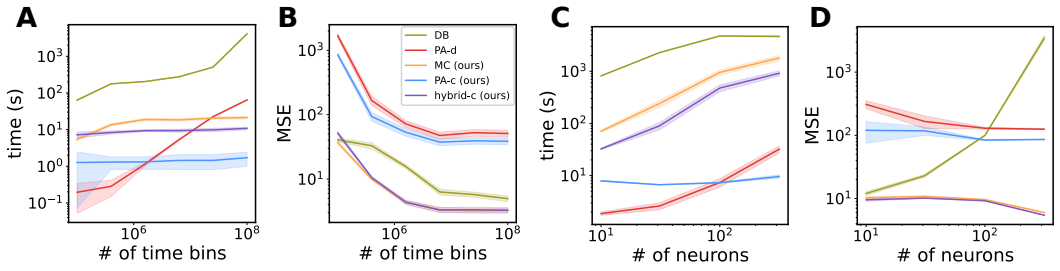

Figure 4: **A** Time to completion for discrete batched GLM (DB), discrete PA (PA-d), and our continuous models as recording length increases in size; **B**, filter accuracy for the models in **A**; **C**, same as **A** as the number of neurons in the population increases; **D**, same as **B** for the model fits in **C**.

$$\mathbb{E}\frac{||\nabla_p - \tilde{\nabla}_p||_2^2}{||\nabla_1||_2^2} \tag{11}$$

where ∇_p is the true gradient at step p and $\tilde{\nabla}_p$ is the corresponding stochastic gradient. Throughout inference, this error remains orders of magnitude higher in the DB model compared to the continuous sampling-based MC approach. Note that the error increases toward the end of training for both methods, as accurately estimating increasingly small gradient steps becomes more difficult as models approach convergence.

4.2 Continuous GLM model performance

We compare model performance and runtime across five approaches: a DB GLM with an SVRG optimizer [27] (DB); the polynomial approximation method of Zoltowski and Pillow [4] (PA-d); our continuous-time polynomial approximation (PA-c); our sampling-based Monte Carlo method (MC); and a hybrid approach that initializes MC inference with PA-c estimates (a "warm start"), reducing optimization steps and accelerating convergence. First, we evaluate performance on simulated data from an all-to-one coupled GLM ($N = 8$), varying recording duration from 10 to 10^4 seconds, which spans the range of modern neuroscience recordings, with the bin size set to 0.1 ms for discrete models. (Fig. 4**A**,**B**). Next, we assess scalability by simulating a random, sparsely (10%) connected GLM with increasing population size ($N = 10$ to $N = 350$) with a fixed recording length $T = 100$ sec (Fig. 4**C**). We evaluate model performance by computing the mean squared error (MSE) between the estimated and true filters.

While SVRG guarantees convergence given enough passes through the full data, in practice we find that even when its runtime exceeds that of all other models by orders of magnitude, DB still underperforms, which is particularly evident at larger population sizes (Fig. 4**D**). The PA-d method is computationally efficient for smaller dataset sizes but eventually scales poorly in time and neuron number due to the cost of batch-computing sufficient statistics. In contrast, continuous-time methods utilize GPU-parallelized scans over the data, making them largely insensitive to recording length while increase only moderately with population size (Fig. 4**A,C**). In terms of estimation accuracy, the polynomial approximation methods (PA-d and PA-c) are less accurate, as expected, due to their approximations in the log-likelihood. The MC and hybrid models achieve the best filter recovery, with the hybrid approach offering the best tradeoff between speed and accuracy (Fig. 4**B,C**). We note here also that PA-c slightly outperforms PA-d due to inaccuracies present in binned data, though both use identical nonlinearity and approximation ranges. Further discussion of the discretization error and example filters from all models are provided in Supplement S.2.

4.3 Evaluation on hippocampal data

The hippocampus is a highly interconnected brain region essential for memory formation and retrieval. Its canonical trisynaptic circuit comprises the dentate gyrus (DG), CA3, and CA1 subregions, with distinct connectivity: the DG projects sparsely to CA3 via mossy fibers, with reciprocal connections back from CA3, and CA3 drives CA1 via the Schaffer collaterals (Fig.5**A**). Additionally, CA3 exhibits

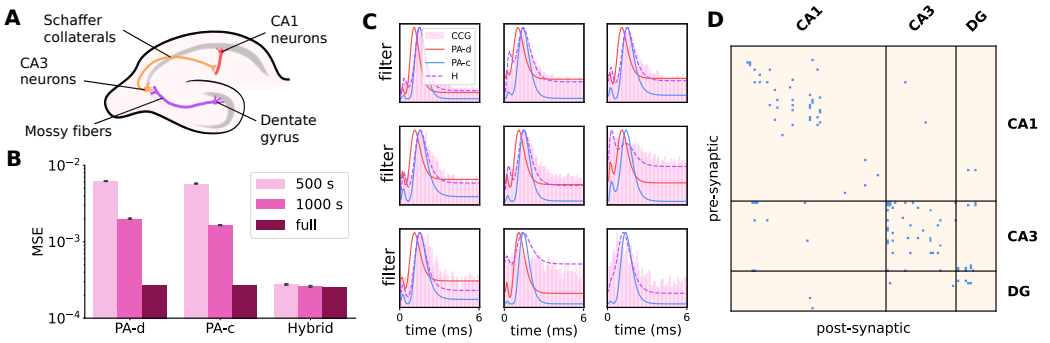

Figure 5: **A** Schematic of hippocampal anatomy; **B** Alignment of filter estimates on subsets of data with CCGs calculated from full dataset; **C** Example estimated filters with overlayed CCGs selected from high firing rate neuron pairs; **D** Putative excitatory connections across hippocampal subregions.

dense recurrent excitatory (EE) connectivity—a hallmark feature supporting autoassociative memory dynamics [23]. While this anatomical framework is well-established [24, 30], inferring monosynaptic connectivity and population-level spiking dynamics from multi-region electrophysiological recordings remains a significant statistical challenge. Cross-correlograms (CCG) based methods are computationally demanding at large scale and require additional processing to extract interpretable synaptic coupling patterns [31, 32, 25]. In contrast, GLMs offer a compact, efficient alternative that reduces parameter count while capturing temporal structure. This setting thus presents an opportunity to evaluate our continuous-time GLM models, which operate at submillisecond temporal resolution.

We use publicly available data from the Allen Institute consisting of 106 neurons ($N_{CA1} = 62$, $N_{CA3} = 28$, $N_{DG} = 16$) recorded with a single probe over approximately 2.7 hours [22]. All models are run with ridge regularization ($\beta = 1000$), a common choice for GLMs [3, 4, 9], to encourage sparsity in synaptic connections (see Supplement S.1 for more hyperparameter details). To assess filter accuracy, we compute the MSE between CCGs calculated on the full dataset and filter estimates from hybrid PA-MC (H), PA-c, and PA-d models on various subsets of the data. We exclude the discrete batched model (DB) from this analysis, as running it to convergence on the full dataset would be computationally infeasible. We find that our filters empirically match the pairwise CCGs, with the hybrid model showing the closest alignment even with only 500 seconds (8.3 minutes) of data, a small fraction of the full 2.7-hour recording (Fig.5**B, C**). While CCGs serve as a proxy for putative connections and cannot fully isolate synaptic effects from common input or indirect pathways, they provide a useful benchmark for evaluating filter estimates. Furthermore, after pre-selecting filters with peaks between 0.3–2.5 ms—indicative of excitatory connections—we find a connectivity structure that closely reflects known hippocampal anatomy (Fig. 5**D**, Table 1). The CA3 network exhibits the highest density of recurrent excitatory connections ($\sim 4\%$), consistent with anatomical estimates [24], while also showing bidirectional communication with the dentate gyrus [30] and Schaffer collateral projections to CA1 (Table 1. Notably, cross-region couplings tend to exhibit longer temporal delays (measured as time from filter onset to peak) than intra-regional latencies, consistent with axonal conduction times between structures and suggesting physiological validation of our identified connections. Fit results on the full dataset ($N = 623$ neurons across all probes) and a comparison showing improved performance with Generalized Laguerre versus raised cosine basis functions are provided in Supplement S.1.3.

5 Conclusion

We developed a continuous-time GLM implementation capable of identifying fine-timescale coupling filters in modern large-scale neural recordings, rendering modern datasets (hundreds of neurons recorded for thousands of seconds) trainable in minutes with sub-millisecond precision. Our focus has been on detecting potential synaptic connections through coupling filters, complementing existing approaches [7, 31], with a key advantage of being able to rapidly screen candidate connections in large datasets.

Table 1: Putative excitatory connections and synaptic latencies across hippocampal regions.

Block	Pairs Total	Putative E	Fraction (%)	Mean Delay (ms)
CA3→CA3	784	30	3.83	1.75 ± 0.52
DG→DG	256	9	3.52	0.85 ± 0.31
CA3→DG	448	12	2.68	1.57 ± 0.83
CA1→CA1	3844	51	1.33	1.69 ± 0.75
DG→CA3	448	5	1.12	2.09 ± 0.33
CA3→CA1	1736	18	1.04	2.15 ± 0.34
CA1→DG	992	4	0.4	1.86 ± 0.29
CA1→CA3	1736	3	0.17	2.17 ± 0.06

Our work complements existing continuous-time modeling efforts which have different modeling goals or operate in smaller data regimes. In particular, Hawkes processes [16] represent a computationally efficient approach to identifying excitatory neuronal connections, but they cannot model inhibitory connections and thus occupy a different model class than the general Poisson process GLM. Other models, such as continuous Point-process latent variable models [33], share a similar likelihood construction but focus on identifying latent structure rather than fine-scale functional connectivity. To our knowledge, the only prior work that actually fits a continuous-time GLM [17] uses Gauss-Lobatto quadrature to approximate the integral in the log-likelihood. However, this approach requires inserting quadrature nodes between every spike time, making it computationally infeasible for the dataset sizes explored here (see Supplement S.3 for details). These fundamental limitations—model structure mismatch and computational infeasibility—precluded direct comparison to these methods in our benchmarks.

Our approach inherits several limitations from the broader class of Poisson GLMs, including the challenge of dissociating monosynaptic connections from correlated firing [2] and the difficulty of identifying true connectivity without overly penalizing weak dependencies or connections involving low-firing neurons [25, 32]. The Poisson distribution itself may be suboptimal for describing neural spiking due to its variance assumptions; flexible alternatives such as the negative binomial distribution [19, 34] could better capture spiking characteristics. Additionally, there is a fundamental trade-off between our two approximation methods: the PA approach enables faster inference through closed-form solutions but is inherently less accurate due to its global approximation of firing rates, while the MC approach is more accurate but requires multiple iterations to converge. Our hybrid model, which uses PA-based initialization followed by MC finetuning, is our attempt to balance this trade-off.

Key future directions include: more thorough evaluation of sparsity priors for population recordings, use of additional non-linearities, per-neuron approximation range optimization for our polynomial-approximate approach, and exploring variance reduction techniques [17, 35] for Monte Carlo sampling of the CIF. Additionally, extending the framework to incorporate latent population dynamics—for instance, by modeling shared low-dimensional trajectories at slower timescales similar to GPFA [33]—could help disentangle fast coupling dynamics from slower coordinated population activity, potentially improving both interpretability and generalization to held-out neurons.

6 Acknowledgments

This work was supported by the Simons Foundation. AHW was supported by the NIH BRAIN initiative (1RF1MH133778).

References

[1] Jonathan W Pillow, Jonathon Shlens, Liam Paninski, Alexander Sher, Alan M Litke, EJ Chichilnisky, and Eero P Simoncelli. Spatio-temporal correlations and visual signalling in a complete neuronal population. *Nature*, 454(7207):995–999, 2008.

[2] Abhranil Das and Ila R Fiete. Systematic errors in connectivity inferred from activity in strongly recurrent networks. *Nature Neuroscience*, 23(10):1286–1296, 2020.

[3] Il Memming Park, Miriam LR Meister, Alexander C Huk, and Jonathan W Pillow. Encoding and decoding in parietal cortex during sensorimotor decision-making. *Nature neuroscience*, 17(10):1395–1403, 2014.

[4] David Zoltowski and Jonathan W Pillow. Scaling the poisson glm to massive neural datasets through polynomial approximations. *Advances in neural information processing systems*, 31, 2018.

[5] Jacob L Yates, Il Memming Park, Leor N Katz, Jonathan W Pillow, and Alexander C Huk. Functional dissection of signal and noise in mt and lip during decision-making. *Nature neuroscience*, 20(9):1285–1292, 2017.

[6] Daniel Fine English, Sam McKenzie, Talfan Evans, Kanghwan Kim, Euisik Yoon, and György Buzsáki. Pyramidal cell-interneuron circuit architecture and dynamics in hippocampal networks. *Neuron*, 96(2): 505–520, 2017.

[7] Ian H. Stevenson. Circumstantial evidence and explanatory models for synapses in large-scale spike recordings. *Neurons, Behavior, Data analysis, and Theory*, 2023.

[8] BL Sabatini and WG Regehr. Timing of synaptic transmission. *Annual review of physiology*, 61(1): 521–542, 1999.

[9] Eric Hart and Alexander C Huk. Recurrent circuit dynamics underlie persistent activity in the macaque frontoparietal network. *elife*, 9:e52460, 2020.

[10] Robert E Kass, Uri T Eden, Emery N Brown, et al. *Analysis of neural data*, volume 491. Springer, 2014.

[11] Liam Paninski. Maximum likelihood estimation of cascade point-process neural encoding models. *Network: Computation in Neural Systems*, 15(4):243, 2004.

[12] Don H Johnson. Point process models of single-neuron discharges. *Journal of computational neuroscience*, 3:275–299, 1996.

[13] Wilson Truccolo, Uri T Eden, Matthew R Fellows, John P Donoghue, and Emery N Brown. 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, 2005.

[14] Zhe Chen, David F. Putrino, Demba E. Ba, Soumya Ghosh, Riccardo Barbieri, and Emery N. Brown. A regularized point process generalized linear model for assessing the functional connectivity in the cat motor cortex. In *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 5006–5009, 2009.

[15] Alex Williams, Anthony Degleris, Yixin Wang, and Scott Linderman. Point process models for sequence detection in high-dimensional neural spike trains. *Advances in neural information processing systems*, 33: 14350–14361, 2020.

[16] Scott W. Linderman and Ryan P. Adams. Scalable bayesian inference for excitatory point process networks, 2015. URL `https://arxiv.org/abs/1507.03228`.

[17] Gonzalo Mena and Liam Paninski. On quadrature methods for refractory point process likelihoods. *Neural computation*, 26(12):2790–2797, 2014.

[18] Jonathan Huggins, Ryan P Adams, and Tamara Broderick. Pass-glm: polynomial approximate sufficient statistics for scalable bayesian glm inference. *Advances in Neural Information Processing Systems*, 30, 2017.

[19] Stephen Keeley, David Zoltowski, Yiyi Yu, Spencer Smith, and Jonathan Pillow. Efficient non-conjugate gaussian process factor models for spike count data using polynomial approximations. In *International conference on machine learning*, pages 5177–5186. PMLR, 2020.

[20] Jonathan W Pillow, Liam Paninski, Valerie J Uzzell, Eero P Simoncelli, and EJ Chichilnisky. Prediction and decoding of retinal ganglion cell responses with a probabilistic spiking model. *Journal of Neuroscience*, 25(47):11003–11013, 2005.

[21] John G Proakis and Masoud Salehi. *Digital communications*. McGraw-hill, 2008.

[22] Allen Institute for Brain Science. Visual coding - neuropixels, 2023. URL `https://portal.brain-map.org/explore/circuits/visual-coding-neuropixels`. Dataset includes spike times, LFP, and behavior from mouse visual cortex during stimuli presentation.

[23] Alessandro Treves and Edmund T Rolls. Computational analysis of the role of the hippocampus in memory. *Hippocampus*, 4(3):374–391, 1994.

[24] David G Amaral and Menno P Witter. The three-dimensional organization of the hippocampal formation: A review of anatomical data. *Neuroscience*, 31(3):571–591, 1989. doi: 10.1016/0306-4522(89)90424-7.

[25] Naixin Ren, Shinya Ito, Hadi Hafizi, John M Beggs, and Ian H Stevenson. Model-based detection of putative synaptic connections from spike recordings with latency and type constraints. *Journal of neurophysiology*, 124(6):1588–1604, 2020.

[26] Stephanie C Seeman, Luke Campagnola, Pasha A Davoudian, Alex Hoggarth, Travis A Hage, Alice Bosma-Moody, Christopher A Baker, Jung Hoon Lee, Stefan Mihalas, Corinne Teeter, Andrew L Ko, Jeffrey G Ojemann, Ryder P Gwinn, Daniel L Silbergeld, Charles Cobbs, John Phillips, Ed Lein, Gabe Murphy, Christof Koch, Hongkui Zeng, and Tim Jarsky. Sparse recurrent excitatory connectivity in the microcircuit of the adult mouse and human cortex. *eLife*, 7, 2018.

[27] Robert M Gower, Mark Schmidt, Francis Bach, and Peter Richtárik. Variance-reduced methods for machine learning. *Proceedings of the IEEE*, 108(11):1968–1983, 2020.

[28] Art B. Owen. *Monte Carlo theory, methods and examples*. `https://artowen.su.domains/mc/`, 2013.

[29] Roelof Koekoek and Hendrik Gerrit Meijer. A generalization of laguerre polynomials. *SIAM journal on mathematical analysis*, 24(3):768–782, 1993.

[30] Helen E Scharfman. The ca3 "backprojection" to the dentate gyrus. *Progress in brain research*, 163: 627–637, 2007.

[31] Zach Saccomano, Sam Mckenzie, Horacio Rotstein, and Asohan Amarasingham. A causal inference approach of monosynapses from spike trains. *arXiv preprint arXiv:2405.02786*, 2024.

[32] Ryota Kobayashi, Shuhei Kurita, Anno Kurth, Katsunori Kitano, K. Mizuseki, Markus Diesmann, B. J. Richmond, and S. Shinomoto. Reconstructing neuronal circuitry from parallel spike trains. *Nature Communications*, 2019.

[33] Lea Duncker and Maneesh Sahani. Temporal alignment and latent gaussian process factor inference in population spike trains. *Advances in neural information processing systems*, 31, 2018.

[34] Jonathan Pillow and James Scott. Fully bayesian inference for neural models with negative-binomial spiking. *Advances in neural information processing systems*, 25, 2012.

[35] Aaron Defazio, Xingyu Alice Yang, Harsh Mehta, Konstantin Mishchenko, Ahmed Khaled, and Ashok Cutkosky. The road less scheduled. *arXiv preprint arXiv:2405.15682*, 2024.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We elaborate on all claims made in the abstract and introduction in the main paper, ensuring they accurately reflect our contributions.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss limitations of our work in the main text and in the supplement.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

Justification: Our paper does not include formal theoretical results such as theorems or lemmas, and therefore we do not require a formal set of assumptions or proofs. While we provide derivations and model formulations, these are not framed as theoretical claims requiring proof.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We have provided the most important information about all parameters, hyperparameters and other modeling details in the main paper with additional information available in the supplement.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We release code for simulation and model fitting, and we use publicly available data for our results, cited appropriately.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: Yes, we provide all required details in the main paper text and in the supplement.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We have included error bars for all results where they are appropriate and meaningful.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We specify the type of computer resorces in the main paper and provide detailes resorce profile in the appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: This paper adheres to the NeurIPs Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: We anticipate any direct societal impact of this work.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.

- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper does not involve the release of data or models with a high risk of misuse.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We use publicly available software libraries and properly cite the benchmark model we compare against. All assets used are under standard open-source licenses, and we respect their terms of use.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.

- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We provide open-source implementations of the models developed in this work. The code will be shared anonymously at submission and publicly upon acceptance.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve research with human subjects nor crowdsourcing.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: LLMs do not play a central or novel role in the method developed in this research.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.