Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Fast probabilistic spike deconvolution using optimized Markov chain Monte Carlo

Dylan Martins
University of California, Santa Barbara
Michael Goard
michaelgoard@ucsb.edu

University of California, Santa Barbara

Article

Keywords:

Posted Date: August 19th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-10425839/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Fast probabilistic spike deconvolution using optimized Markov chain Monte Carlo

Dylan M. Martins[1] **and Michael J. Goard**[2,3,*]

[1]Graduate Program in Dynamical Neuroscience, University of California, Santa Barbara, USA.
[2]Department of Molecular, Cellular, and Developmental Biology, University of California, Santa Barbara, USA.
[3]Department of Psychological and Brain Sciences, University of California, Santa Barbara, USA.
[*]michaelgoard@ucsb.edu

ABSTRACT

Two-photon calcium imaging allows monitoring of large populations of neurons, but individual action potentials are often obscured under typical recording conditions. Algorithms for spike deconvolution estimate the underlying spike train of a neuron given an observed calcium trace. One powerful approach, Markov chain Monte Carlo (MCMC) spike inference, models the underlying calcium dynamics, iteratively adding, removing, and shifting spikes in an inferred spike train in continuous time to sample from the posterior probability of the spike train given the observed calcium signal. However, MCMC is computationally taxing, requiring many hours of compute time for typical datasets. Here, we introduce a method for Optimized MCMC Spike Inference (OMSI) that preserves the performance of the full probabilistic framework while achieving a >90% reduction in compute time. Benchmarking using simulated data and simultaneous electrophysiology and calcium imaging confirms that our approach maintains accuracy comparable to traditional MCMC and matches or exceeds non-probabilistic alternatives. These performance gains establish continuous-time probabilistic inference as a practical, high-throughput tool for large-scale calcium imaging pipelines.

Introduction

Two-photon calcium imaging has revolutionized our ability to monitor large neural populations[1–5] and track the same cells over time[1,6,7]. However, under common imaging conditions calcium imaging cannot resolve individual action potentials from $\Delta F/F$[8], in part due to the slow kinetics of calcium indicators. Algorithms have been developed which attempt to closely estimate the true spike train underlying the observed calcium dynamics, allowing experimenters to quantify neural activity closer to the timescale of many rapid neural computations and behaviors. Since calcium movement across the cellular membrane[9] and the nonlinear dynamics of calcium sensors[10–13] are complex, inferring the underlying timing of action potentials from a calcium trace is not trivial.

Existing deconvolution methods each have limitations. While fast and sufficiently accurate for many use-cases, deterministic approaches such as OASIS[14] lack quantitative measures of per-spike uncertainty, predict spike rate in arbitrary units proportional to the scale of input fluorescence, and infer binned activity rather than continuous-time spikes. Deep learning tools such as CASCADE[15] require GPUs for maximum performance, require extensive training data which does not always generalize across experimental conditions, offer limited interpretability given the "black box" nature of deep learning, give no per-spike probability distribution, and infer binned activity. For many experimental aims, these limitations are not consequential and OASIS and CASCADE can serve as rapid means of inferring spike rates from observed calcium dynamics. Markov chain Monte Carlo (MCMC) is a general class of algorithms that systematically explore high-dimensional parameter spaces to approximate complex distributions. MCMC has been used to reconstruct underlying spike trains from ambiguous fluorescence traces[16–18]. While sampling the joint posterior over amplitude, decay, and noise parameters, it iteratively proposes additions, removals, and shifts of spike events in continuous time. It provides a per-spike uncertainty quantification, requires no training data, infers continuous-time spikes, and can be expected to generalize across conditions. While these advantages can be highly valuable depending on experimental aims, existing implementations of MCMC inference applied to spike deconvolution are time-consuming and computationally expensive for large-scale datasets.

Here, we present a method for Optimized MCMC Spike Inference (OMSI) that preserves the performance of the full probabilistic framework while significantly reducing compute time. Several targeted optimizations collectively reduce compute time by >90%. The method scales well with cell count and recording duration, completing inference on a 20-minute, 500-neuron session in ∼5 minutes on standard CPU hardware compared to ∼2.5 hours using existing implementations. We validate this approach on both simulated data and simultaneous electrophysiology-calcium imaging recordings and confirm that accuracy is preserved. These performance advantages establish probabilistic continuous-time spike inference as an efficient and scalable

approach for large-scale calcium imaging analysis.

Results

Optimizations

We make several targeted optimizations to increase the speed of Markov chain Monte Carlo spike inference without sacrificing accuracy.

i. We choose to reevaluate the change in log likelihood ratio for every proposal in a local support window rather than over the duration of the full recording. This has no accuracy cost because only a narrow window of the calcium traces is generally impactful on the local spike train (Figure S1a) and addition, removal, and shift proposals do not affect the calcium dynamics minutes before or after that local window. This optimization reduces the cost of proposing spike modifications (additions, removals, or shifts) and fixes the cost relative to the size of the support window T_{supp} rather than the duration of the full recording T. The support window length is determined adaptively per cell by truncating the kernel where its amplitude falls below 1% of its peak value. For $\tau_d = 1.2$ s at 30 Hz, this corresponds to approximately 6 seconds. Short window sizes show no decrease in overall spike train accuracy (F_β=0.595 at $T_{supp} = 13.0$ s vs. F_β=0.597 at the full 2400 s recording length). We characterized compute speed and inference accuracy as a function of the length of the support window used by generating a population of 50 cells with a 40 minute recording duration sampled at 30 Hz and with $\tau = 1.2$ and increased the size of T_{supp} step-wise. We found that accuracy does not change as a function of T_{supp} while T_{supp} remains above ~ 6 seconds while compute time increases dramatically as T_{supp} passes ~ 150 seconds and continues to increase as it approaches full recording duration (13.2 sec/cell at $T_{supp} = 13.0$ s vs. 369.0 sec/cell at the full 2400 s T_{supp}, a $\sim$28x increase).

ii. The number of spike addition and removal proposals per sweep is set to $\lceil T/500 \rceil$, scaling linearly with recording duration. This ensures that longer recordings receive proportionally more proposals per sweep, preventing undersampling of the spike train relative to its length. The denominator of 500 was chosen empirically to provide adequate coverage across typical recording durations.

iii. Rather than beginning with a FOOPSI[19]-based initial estimate upon which to begin sweeps, we use non-negative least squares (NNLS; Figure S2a) which is ~ 1.5 times faster than initializing with FOOPSI by median compute speed (Figure S2b; NNLS: 78.19 msec/cell, FOOPSI: 121.45 msec/cell; for comparison, FOOPSI is implemented here as an L_1-penalized least-squares deconvolution solved with L-BFGS-B). Despite this speed advantage, NNLS still provides a sufficient starting spike train which closely matches FOOPSI initialization on a synthetic dataset of 200 cells recorded at 30 Hz with $\tau = 1.2$ (Figure S2a). The key distinction between these methods is that NNLS does not impose an explicit L_1 sparsity penalty, unlike FOOPSI. Traces are highly correlated with one another (Figure S2c; corr=0.982).

iv. In typical MCMC spike inference, the early sweeps are used to approach an accurate estimate but do not directly contribute to the ultimate probability distribution. Rather, they are used to "burn in" an improved estimate of spike times. Thereafter, the remaining sweeps contribute to the per-spike probability distributions without being contaminated by early, less accurate predicted spike trains. In contrast to existing methods which run a fixed number of sweeps per cell, OMSI monitors two sequential convergence criteria checked every 50 sweeps. This serves to focus compute time on more challenging cells while avoiding needless compute expense on low-firing or high-noise cells. Burn-in is declared complete once the amplitude trace A stabilizes. The last 100-sweep window is split in half, and burn-in ends when the two half-means agree to within 0.01%. Post-burn-in, the sampler collects at least 300 additional sweeps before testing whether the spike-count distribution has stabilized. Convergence is declared when two halves of the post-burn-in spike count record differ by less than $10^{-1.5} \approx 3.16\%$ of the second-half mean and sampling terminates at that point, with a hard ceiling of 2,000 total sweeps. These thresholds were chosen empirically using a synthetic population of 200 cells (Figure S1b-c).

v. To support continuous-time spike positions with sub-frame precision, the calcium trace must reflect each spike's fractional offset within its bin rather than treating all spikes as falling at bin centers. Naively, this would require a separate infinite impulse response (IIR) filter sweep for each spike, making trace reconstruction an $O(T * K)$ operation where K is the number of spikes. Instead, spike contributions are first binned into two accumulator arrays, one weighted by e^{δ_k/τ_r} and one by e^{δ_k/τ_d}, where δ_k is the intra-bin offset, so that the IIR filter can be run in a single causal sweep from $t = 0$ to $t = T - 1$, reducing reconstruction to $O(T)$ regardless of spike count. Both accumulator construction and the subsequent IIR passes are pre-compiled with Numba[20], keeping per-proposal overhead small enough that high-frequency proposals remain practical.

vi. Evaluating the likelihood of a proposed spike addition or deletion requires the kernel's self-energy (the squared-norm term against which the residual-kernel dot product is compared) over the region of the trace it affects, and recomputing this energy integral from scratch on every proposal would add unnecessary overhead for long recordings. After each accepted update to the rise or decay time constant, the truncated impulse response and its cumulative energy arrays are rebuilt in a single pass of length L_{supp} (the support of the kernel) and cached. Subsequent addition, removal, and movement proposals then look up this energy term directly from the precomputed array in $O(1)$ rather than re-summing kernel values, whereas the residual-kernel dot product itself must still be summed over the L_{supp} frames of the support window for every proposal, so overall proposal evaluation remains $O(L_{\text{supp}})$ (see Methods). Because we check thousands of spike proposals but rarely update the time constants, the minor cost of keeping the cache updated pays off with significant time savings.

vii. We screen out cells with low signal-to-noise ratio (SNR < 2.0) to avoid wasting compute time on cells for which we do not expect successful inference is achievable. This threshold is set empirically rather than at the noise floor. A pure Gaussian trace scores SNR ≈ 2.64 by this metric, so a cutoff intended to exclude pure noise would need to exceed that value. Instead, we swept SNR across a population of synthetic cells and measured the resulting inference quality directly, finding that F_β and CosMIC scores approach a usable level ($\sim$0.5) for genuine calcium traces near SNR $= 2.0$, with quality degrading sharply below it (Figure S1d; at the nearest tested SNR values, F_β=0.482 and CosMIC=0.349 at SNR $= 2.084$, versus F_β=0.333 and CosMIC=0.225 at SNR $= 1.748$).

viii. For fast indicators at high frame rates where the rise and decay poles are closely spaced ($g_d - g_r < 0.15$) and the decay spans fewer than 40 frames, as occurs for GCaMP8 recorded at $\geq$100 Hz, the double-exponential AR(2) kernel is nearly coincident. At 122 Hz a nominal rise time of $\tau_r \approx 5$ ms spans less than one frame (≈ 0.6 frames), making the rise pole numerically indistinguishable from the decay pole. OMSI detects this automatically for every cell and addresses it by switching to a single-exponential AR(1) kernel (decay pole only, zero rise pole). Additionally, the empirical firing rate λ is re-estimated from the current spike count on every sweep rather than held fixed, allowing the sampler to self-correct from any over-estimation in the initial NNLS spike count; this is an empirical-Bayes-style approximation rather than an exact hierarchical treatment of λ (see Sensor-specific parameterization for fast indicators in Methods for discussion of this approximation).

ix. For indicators with a decay time constant shorter than 0.6 s, the Metropolis–Hastings update for the time constants τ_r and τ_d is disabled automatically. For fast indicators, the calcium transient is brief in absolute duration regardless of acquisition rate, leaving too little of the transient's characteristic shape, relative to the noise level, to separately constrain τ_r and τ_d; sampling more finely at a higher frame rate does not resolve this, since it is the short duration of the underlying event, not the number of frames used to sample it, that limits identifiability. Holding the initialization-derived time constants fixed avoids this identification problem with no meaningful accuracy cost.

x. We execute our Python code at C-speed using Numba[20] and parallelize across CPU cores using Ray[21]. Wherever possible, we use float32 values in place of float64 values and initialize arrays once at the outset of an iterative process and overwrite existing values instead of re-initializing a new array each time.

Evaluation on simulated data

To evaluate accuracy and required compute time, we generated a population of 500 synthetic neurons sampled at 30 Hz over a 20-minute recording, $\tau_d = 1.2$ sec, AR(2) kernel, Poisson spike trains drawn to span a broad range of trace excess kurtosis values (excess $K_4 \in [0, 25]$ over an exponential distribution). A subset of cells, 15%, were designated as "bursty," receiving 2-5 additional spikes (producing burst lengths of 3-6 spikes total) following each primary spike with probability 60%; additional spikes are spaced at multiples of 2 high-resolution time bins after the primary spike, where the high-resolution grid is upsampled 10-fold relative to the population's frame rate (e.g. 2 bins $\approx$6.7 ms apart at 30 Hz). Full details of synthetic data generation are given in Methods. When comparing to existing MCMC, we choose to compare to the dominant implementation of MCMC in use, CaImAn-Matlab[22]. Each method was run on the same synthetic fluorescence matrix with matched noise and ground truth spike times. Detection accuracy was quantified using F_β score ($\beta = 0.5$, weighting precision four times as heavily as recall) with a 100 milliseconds coincidence window (chosen to match[8]), and supplemented by CosMIC score[23].

F_β score measures a weighted harmonic mean of precision and recall and demands a one-to-one match between the prediction and ground truth. It tends to penalize over-prediction, such that if a burst of ten predicted spikes is inaccurately inferred for a single ground truth spike, the ground truth spike will be paired to only one of the predicted spikes and the rest will accumulate towards the false positive rate, reducing the F_β score. In contrast, CosMIC convolves both ground truth and estimated spike trains with a smoothing pulse and measures the overlapping area of the two smoothed signals. It is generally

more forgiving towards a burst of false positives because the tight cluster of false positives is smoothed into a continuous bump in activity. Concretely, F_β measures exact event counts, while CosMIC measures temporal activation overlap.

Example cells spanning the excess kurtosis range illustrate OMSI's high-quality matching with ground truth spike events (Figure 1a). Similarly, CaImAn MCMC and CASCADE spike time predictions closely match ground truth. OASIS tends to predict more spikes than exist in ground truth. While OASIS performs worse on F_β score than the other methods, which perform very well on the simulated dataset, all methods score with a median above 0.6 (Figure 1c; median F_β: OMSI=0.992, CaImAn MCMC=0.979, CASCADE=0.998, OASIS=0.729). Using CosMIC score, OMSI and CASCADE perform best, followed closely by CaImAn MCMC, then OASIS (Figure 1d; mean CosMIC: OMSI=0.621, CaImAn MCMC=0.598, CASCADE=0.626, OASIS=0.467).

Wall-clock compute times on both an absolute (Figure 1e) and time per cell (Figure 1f) basis show a near-tie between OMSI and CASCADE when CASCADE is run on the GPU (5.63 minutes and 5.50 minutes, respectively) while CaImAn MCMC is more than an order of magnitude slower (197.20 min.). When run on the CPU, CASCADE is modestly slower (6.85 min.) than on the GPU. Note that we performed this analysis with a mid-range GPU (Nvidia RTX 3060 with 12 GB VRAM), and with higher-powered GPU hardware, the speed advantage of GPU-accelerated CASCADE would likely increase. In contrast, OASIS runs orders of magnitude faster than other methods (0.06 min.). As a per-cell rate, this amounts to OMSI, CaImAn MCMC, CASCADE-CPU, CASCADE-GPU, and OASIS completing inference on this simulated dataset at 0.68, 23.66, 0.82, 0.66, and 0.01 sec/cell (Figure 1f). The relationship between spike count and the time spent inferring spike times for that cell was negligible (linear regression slope ≈ 0, $R^2 \approx 0$), indicating that OMSI's compute time does not scale appreciably with spike train density (Figure 1g).

Our proposed method, OMSI, completes inference on this simulated population using 97.1% less wall-clock time than CaImAn MCMC while computing the inferred spike train with the same underlying statistical approach. Indeed, it achieves MCMC deconvolution at speeds matching or exceeding modern deep learning approaches (OMSI is approximately 17.9% faster than CASCADE run on CPU, and 2.4% slower than CASCADE run on GPU). Together, these results establish that on synthetic data, OMSI preserves the accuracy of the full Bayesian framework of MCMC while reducing wall-clock time by 1.5 orders of magnitude (i.e., $\sim$35x faster) relative to the CaImAn MCMC implementation.

CASCADE and OASIS yield a binned spike rate rather than continuous-time spike event onsets. As such, they are at a systematic disadvantage in all metrics used in this manuscript. We apply fixed detection thresholds to transform these binned rates into continuous-time spikes. For OASIS, the chosen threshold is 1.0σ, where σ is the per-cell MAD-estimated noise standard deviation of the $\Delta F/F$. For CASCADE, a fixed threshold of 0.5 is used, retaining only frames where the model assigns at least 50% probability to a spike. In both cases, a minimum inter-peak distance of 5% of the frame rate is applied. That CASCADE and OASIS can at times match or exceed the performance of MCMC methods suggests that our thresholding approach is a fair method of generating comparable outputs. To further demonstrate that we are choosing ideal thresholds, we created a dataset of 100 simulated cells with $\tau_d = 1.2$ recorded at 30 Hz for 600 seconds and evaluated the precision, recall, F_β, and CosMIC scores at a range of detection thresholds (Figure S3). We found that at all detection thresholds tested $[0.0, 2.0]$ our thresholds of 0.5 for CASCADE and 1.0 for OASIS were ideal for each method (CASCADE at threshold=0.5: F_β=0.953, the maximum across all tested thresholds, and CosMIC=0.622, within 0.003 of the maximum CosMIC=0.625 at threshold=0.4; OASIS at threshold=1.0: CosMIC=0.465, the maximum across all tested thresholds, and F_β=0.753, within 0.014 of the maximum F_β=0.767 at threshold=1.5).

Speed and accuracy benchmarks while sweeping across parameters

To characterize the computational and statistical properties of OMSI more systematically, we performed five benchmarks, each varying a single axis while holding all others constant. All benchmarks utilize synthetic data.

We first examined how inference quality and wall-clock time co-vary with sweep count, running OMSI and CaImAn MCMC at eight fixed sweep budgets from 10 to 3,000 (with adaptive stopping disabled). CASCADE and OASIS do not perform iterative sweeps and were excluded from this benchmark. The benchmark had conditions fixed at 50 cells, a 5-minute recording, 30 Hz, and $\tau_d = 1.2$ seconds. Unsurprisingly, OMSI out-performs CaImAn MCMC in compute time by more than an order of magnitude and scales better with sweep count (Figure 2a; at 3,000 sweeps, OMSI=47.6 s vs. CaImAn MCMC=2352.9 s, a $\sim$49x difference). The second benchmark measured scalability, sweeping cell count from 50 to 3,000 at fixed duration (5 min) and recording duration from 5 to 120 min at fixed cell count (N=100). The OMSI and CASCADE (both on GPU and CPU) methods perform similarly, while OASIS required far less compute time, whereas CaImAn MCMC required far higher (Figure 2b; at N=3,000 cells, time-per-cell: OMSI=0.255 s, CASCADE-GPU=0.239 s, CASCADE-CPU=0.211 s, OASIS=0.0014 s, CaImAn MCMC=4.258 s). Finally, we tested scaling with recording duration and found similar results (Figure 2c; at the longest duration completed by both, 3,600 s/60 min, total time: OMSI=153.4 s vs. CaImAn MCMC=14106.2 s, a $\sim$92x difference).

The remaining two benchmarks assess sensitivity to biophysical and experimental parameters. We first varied the indicator decay time constant over a realistic range ($\tau_d \in \{0.2, 0.5, 0.8, 1.2\}$ sec). Performance does not vary monotonically with τ for all

methods, but OASIS notably remains significantly lower than other methods in precision across the tested range (Figure 2d; at $\tau = 0.2$ s, the fastest tested, precision: OASIS=0.802, OMSI=0.987, CASCADE=0.981, CaImAn MCMC=0.998). We next varied the signal-to-noise ratio (SNR) using populations generated over a realistic range of SNR (SNR $\in \{1, 2, 5, 10, 20, 50, 100\}$). We found that below SNR $= 2$, the performance of most methods suffered significantly, performing poorly at both precision and recall (Figure S1d; at the lowest tested SNR $= 1$: OMSI precision=0.187/recall=0.108, CASCADE precision=0.256/recall=0.478, OASIS precision=0.233/recall=0.011, CaImAn MCMC precision=0.065/recall=0.881). Notably, CaImAn MCMC appears less sensitive to the high-noise conditions by recall score, but will nevertheless provide a poor inference given its near-zero precision (recall=0.881, precision=0.065 at SNR $= 1$). In such high noise environments, inferring individual spike times is not feasible. In higher SNR conditions beyond SNR $= 2$, all methods perform above precision and recall of 0.6 and increase to near-perfect performance at the highest tested SNR values, with the exception of OASIS which continues to score lower than alternative methods in precision (at SNR $= 5$, the lowest level beyond 2, all methods exceed 0.6 in both precision and recall; at the highest tested SNR $= 100$, precision/recall reach OMSI=0.999/0.993, CASCADE=1.000/0.986, CaImAn MCMC=0.959/0.945, while OASIS reaches recall=0.999 but only precision=0.670).

We next evaluated precision and recall across imaging frame rates ($f_s \in \{7.5, 10, 20, 30, 50, 100\}$ Hz). All methods perform worse at low sample rates. While CASCADE performs well at 20-30 Hz, at low sample rates ($\leq$ 10 Hz), CASCADE performance falls in both precision and recall (Figure 3a) more than is observed in MCMC methods (at 7.5 Hz: CASCADE precision=0.634/recall=0.552, vs. OMSI precision=0.937/recall=0.914 and CaImAn MCMC precision=0.795/recall=0.713). At higher sample rates ($\geq$ 20 Hz), CASCADE recall plateaus at a rate similar to that of OMSI and CaImAn MCMC (at 30 Hz: CASCADE recall=0.978, OMSI recall=0.993, CaImAn MCMC recall=0.962). When combined into an F_β score, the pattern is clear: OASIS performs worse than alternatives at all sample rates, both MCMC methods score better than CASCADE at low sample rates, but both MCMC methods and CASCADE perform similarly at high sample rates (Figure 3b). OASIS generally performs worse than all alternative methods in precision, and as a result, F_β (median across all tested sample rates: OASIS precision=0.667/F_β=0.702, vs. OMSI precision=0.975/F_β=0.977, CASCADE precision=0.994/F_β=0.990, CaImAn MCMC precision=0.945/F_β=0.944).

When cells are recorded at 30 Hz, CASCADE performs very well on both F_β and CosMIC scores, but performs poorly on cells sampled at slower rates (as shown in Figure 3a,b). To further investigate this, we created two populations of 50 neurons with a $\tau = 1.2$ simulated for 300 seconds and evaluated these cells sampled at either 7.5 Hz or 30 Hz, using CASCADE models pre-trained on data at the corresponding frame rate. The slower 7.5 Hz sampled cells were predicted worse than the faster cells by F_β and CosMIC (Figure 3c; median F_β=0.563 at 7.5 Hz vs. 0.985 at 30 Hz; median CosMIC=0.305 at 7.5 Hz vs. 0.713 at 30 Hz). Poor performance at 7.5 Hz likely reflects two contributing factors: the reduced temporal resolution inherent to lower frame rates, which limits the information available for spike detection, and the fact that the 7.5 Hz CASCADE training data was not originally acquired at 7.5 Hz but was instead downsampled from 30 Hz recordings. This further demonstrates the limitation of spike inference methods that require training data, which may not always generalize to other recording conditions.

Together, these results confirm that OMSI maintains the advantages of MCMC spike inference at dramatically shorter compute times.

Evaluation on ground-truth datasets

We analyzed a publicly available dataset[8] featuring simultaneous two-photon calcium imaging (Cre-driver lines expressing either GCaMP6f, $\tau_d = 0.5$ s, or GCaMP6s, $\tau_d = 1.2$ s) and electrophysiological recordings from mouse visual cortical cells, yielding matched fluorescence traces and ground-truth spike trains (N=105 cells total after inclusion filtering; see Methods). Imaging frame rate was not uniform across this dataset: low-zoom recordings were acquired at 30 or 60 Hz, while high-zoom recordings were acquired at their native rate, which ranged from approximately 118-180 Hz depending on the session; all recordings were analyzed at their native frame rate rather than resampled to a common rate.

Example cells demonstrate generally accurate inference across all methods except OASIS, which overestimates spike counts on this dataset (Figure 4a). To more accurately reflect standard recording conditions, this dataset includes both full-resolution ("high-zoom") and reduced-resolution ("low-zoom") recordings of separate imaging sessions (N=87 high-zoom and N=18 low-zoom cells); these are treated throughout as independent recordings and are never pooled or cross-registered against one another, but are instead reported separately by zoom condition in every analysis (Figure 4b,c,f). Predictably, precision and recall are generally higher for high-zoom compared to low-zoom recordings, and precision is higher for low τ_d compared to high τ_d, though recall shows the opposite dependence on τ_d (Figure 4b,c; pooled across methods, median precision/recall: high-zoom=0.867/0.606 vs. low-zoom=0.802/0.430; $\tau_d = 0.5$ s precision=0.892 vs. $\tau_d = 1.2$ s precision=0.698, but $\tau_d = 0.5$ s recall=0.508 vs. $\tau_d = 1.2$ s recall=0.689). Precision is comparable among OMSI, CaImAn MCMC, and CASCADE at both zoom levels, whereas OASIS exhibits low precision, particularly in the high-zoom condition (median precision, high-zoom/low-zoom: OMSI=0.942/0.854, CaImAn MCMC=0.750/0.840, CASCADE=1.000/0.951, OASIS=0.229/0.601). While OASIS achieves the highest recall, this is a natural consequence of its tendency to over-predict, given that it rarely misses spikes but

frequently generates false positives. The other three methods maintain similar, more balanced recall values (median recall: OASIS=0.955, OMSI=0.511, CaImAn MCMC=0.562, CASCADE=0.415).

Furthermore, F_β scores are similarly robust for OMSI, CaImAn MCMC, and CASCADE (Figure 4d), while OASIS generally underperforms (median F_β: OMSI=0.745, CaImAn MCMC=0.599, CASCADE=0.754, OASIS=0.376). Surprisingly, CosMIC scores remain relatively consistent across all methods (Figure 4e), likely because F_β measures exact event counts while CosMIC measures temporal activation overlap (median CosMIC: OMSI=0.295, CaImAn MCMC=0.277, CASCADE=0.247, OASIS=0.311). When evaluating the data grouped by τ and zoom condition, F_β scores indicate that CaImAn MCMC, OMSI, and CASCADE perform comparably, despite some expected variability across individual recording datasets (e.g., at $\tau_d = 0.5$ s, high-zoom: OMSI=0.700, CaImAn MCMC=0.711, CASCADE=0.782). OASIS matches the performance of the other methods only under low-zoom conditions, falling behind elsewhere (OASIS median F_β: $\tau_d = 0.5$ s low-zoom=0.670 and $\tau_d = 1.2$ s low-zoom=0.602, both comparable to or exceeding the other methods in those same low-zoom groups, vs. $\tau_d = 0.5$ s high-zoom=0.195 and $\tau_d = 1.2$ s high-zoom=0.415, far below the other methods' $\sim$0.7-0.8 in those high-zoom groups). Together, these results demonstrate that OMSI consistently matches or exceeds the performance of CaImAn MCMC for continuous-time spike inference on ground-truth datasets.

The four methods were also evaluated using an expanded dataset of many sensors (accessed through[15]), reported here for GCaMP6f, GCaMP6s, GCaMP8f, GCaMP8m, and jRCaMP. Three individual datasets were excluded from this comparison: GCaMP7f, GCaMP8s, and XCaMP. These recordings showed temporal misalignment between calcium imaging and electrophysiological data which was not consistent between cells. Because all methods performed uniformly poorly on these excluded recordings, they did not provide a reliable baseline for methodological comparison. GCaMP5k, OGB1, and jGECO were also excluded from this comparison.

Example traces from several sensors demonstrate high inference accuracy for OMSI, even in cells exhibiting lower excess kurtosis (Figure S4a). Furthermore, the F_β (Figure S4b) and CosMIC (Figure S4c) scores indicate that OMSI and CaImAn MCMC achieve comparable performance across most sensors, with a few notable exceptions where one method outperforms the other. For example, OMSI exceeds CaImAn MCMC's accuracy on GCaMP6s (median F_β: OMSI=0.697 vs. CaImAn MCMC=0.642; median CosMIC: OMSI=0.318 vs. CaImAn MCMC=0.237). Importantly, the relative performance of two methods can vary drastically between accuracy metrics, for example when tested on the jRCaMP sensor, OMSI outperforms CaImAn MCMC by CosMIC score, but CaImAn MCMC outperforms OMSI by F_β (median CosMIC: OMSI=0.356 vs. CaImAn MCMC=0.211; median F_β: OMSI=0.474 vs. CaImAn MCMC=0.659). Thus the combination of scores, and viewing raw spike trains, is necessary to understand their relative advantages fully.

Notably, for GCaMP8m, OMSI yields lower F_β and CosMIC scores than CaImAn MCMC in a subset of cells, and for GCaMP8f, OMSI yields a modestly lower F_β score despite a higher CosMIC score (Figure S4b,c; median F_β: GCaMP8f OMSI=0.786 vs. CaImAn MCMC=0.825, GCaMP8m OMSI=0.493 vs. CaImAn MCMC=0.867; median CosMIC: GCaMP8f OMSI=0.406 vs. CaImAn MCMC=0.348, GCaMP8m OMSI=0.150 vs. CaImAn MCMC=0.394). Closer inspection of individual cells revealed that while OMSI predicts spike trains accurately for many cells, it drastically over-predicts in others (Figure S5). This discrepancy accounts for the somewhat bimodal distribution observed in OMSI's inference performance. Because OMSI performs well on synthetic data with matching decay times to GCaMP8 sensors, we suggest that the decline in performance for GCaMP8 sensors could in part be attributable to the high-noise calcium traces that do not always show clear alignment between ground truth spike times and observed calcium transients. The mean SNR values for the GCaMP8 datasets are substantially lower than those of other sensors (GCaMP6f: 54.99 ± 71.24, GCaMP6s: 73.64 ± 60.18, GCaMP8f: 13.27 ± 12.57, GCaMP8m: 14.71 ± 10.17). Finally, while CosMIC evaluates OASIS surprisingly favorably across this collection of datasets, OMSI and CaImAn MCMC generally maintain similar performance profiles across the majority of the sensors tested.

Discussion

As dataset sizes continue to grow, the computational demands of spike inference become an increasingly important practical constraint. Here we presented OMSI, a set of targeted optimizations to Markov chain Monte Carlo spike inference that collectively reduce compute time by >90% while preserving the accuracy of the full probabilistic framework.

Restricting likelihood evaluation to a local support window eliminates the dominant $O(T)$ cost of full-trace residual recomputation without measurable loss of accuracy, as windows as short as 6 seconds produce inference quality indistinguishable from full-recording evaluation. Complementing this, the number of spike addition and removal proposals per sweep is scaled linearly with recording duration as $\lceil T/500 \rceil$, ensuring that longer recordings receive proportionally more proposals and preventing undersampling of the spike train relative to its length. Replacing FOOPSI initialization with block-wise NNLS provides a comparable starting point for the sampler at roughly half the initialization cost. Adaptive convergence detection focuses compute time where it is most needed, terminating early on low-firing or high-noise cells and allocating additional sweeps to cells with complex spike trains. Together with $O(T)$-to-$O(1)$ kernel energy caching, float32 arithmetic, Numba

compilation, and Ray-based parallelization, these changes reduce wall-clock time for a 20-minute, 500-neuron session from approximately 2.5 hours to approximately 5 minutes on standard CPU hardware.

Benchmarking on both simulated data and ground-truth simultaneous electrophysiology-calcium imaging recordings confirms that this speed gain does not come at the cost of accuracy. On simulated data, OMSI matches CaImAn MCMC on F_β and CosMIC scores while completing inference 97% faster. On ground-truth recordings, OMSI performs comparably to CaImAn MCMC across a broad range of sensors, zoom conditions, and decay time constants. Performance on GCaMP8m is lower than OMSI's performance on other sensors tested with ground truth data, but given the results of synthetic tests for the same sensor decay time, may be attributable to the available dataset itself.

The appropriate choice of spike inference method depends on experimental aims. For many or most applications, OASIS provides accurate spike trains at near-instantaneous compute speed on standard hardware. CaImAn MCMC has comparably high accuracy, but longer compute times make it impractical for large datasets. For applications in which neither continuous-time action potentials nor a per-spike probability distribution are important, OASIS and CASCADE are likely sufficient algorithms. If a user's recording conditions fall approximately within the bounds of available ground truth data and the user does not need per-spike probability distribution, CASCADE is well suited to many such datasets. Its ability to deconvolve spikes with rapid GPU acceleration offers a distinct advantage, especially given recent advances in GPU hardware. Optimized Markov chain Monte Carlo Spike Inference (OMSI) is the appropriate choice when a user requires accurate continuous-time action potential onsets, a per-spike probability distribution, or a method that generalizes across experimental conditions without requiring matched training data. Where MCMC inference has historically been a computational bottleneck for large-scale datasets, OMSI makes it a practical option, establishing probabilistic continuous-time spike inference as an efficient and scalable tool for modern calcium imaging analysis pipelines.

Methods

We follow the Reversible Jump Markov chain Monte Carlo approach described previously for spike inference[16–18].

Observation model

We model the fluorescence trace y at frame t as a linear combination of a calcium transient ($G_s(t)$) driven by unobserved spikes (s) scaled by per-spike amplitude (A), a baseline (b), an exponentially decaying (g_d^t) initial calcium offset (C_{in}) and additive Gaussian noise (ε_t):

$$y(t) = A * G_s(t) + b + C_{in} * g_d^t + \varepsilon_t$$

where $\varepsilon \sim N(0, \sigma^2)$ and $s = \{s_1, ..., s_K\}$ for a total of K spikes, and with the decay pole $g_d = e^{-\Delta t/\tau_d}$.

Calcium transient model

A two-component AR(2) process with rise time τ_r and decay time $\tau_d > \tau_r$ has a single spike kernel

$$h(u) = \frac{e^{-u/\tau_d} - e^{-u/\tau_r}}{g_d - g_r}$$

for lag $u \geq 0$, where $g_r = e^{-\Delta t/\tau_r}$. The total calcium transient is

$$G_s(t) = \sum_k h(t - s_k)$$

for spike index k.

Sub-frame spike timing

When evaluating the kernel, spikes are mapped to enclosing bin $t_k = \lceil s_k/\Delta t \rceil - 1$ and a sub-frame offset

$$\delta_k = s_k - \Delta t * \lceil s_k/\Delta t \rceil \leq 0$$

The rise component weight $w_k^r = A * e^{\delta_k/\tau_r}$ and decay component weight $w_k^d = A * e^{\delta_k/\tau_d}$ vary continuously with the spike's position within the bin. This approach allows two spikes in the same bin that only slightly differ to produce different calcium transients.

Forward model

The calcium trace is created using two one-pole causal infinite impulse response (IIR) filters so that spikes at different positions within a bin can contribute different amplitudes. Spike contributions are binned into two accumulators, which allows the IIR filter to be run in a single sweep from $t = 0$ to $t = T - 1$ and without the accumulators, we would need to run a separate IIR filter pass for each individual spike. This approach turns an $O(T * K)$ operation into an $O(T)$ operation. The accumulators take the form

$$\tilde{s}^r(t) = \sum_{k:t_k=t} e^{\delta_k/\tau_r} \quad \text{and} \quad \tilde{s}^d(t) = \sum_{k:t_k=t} e^{\delta_k/\tau_d}$$

to get the filtered outputs

$$G^r(t) = \tilde{s}^r(t) + g_r G^r(t-1) \quad \text{and} \quad G^d(t) = \tilde{s}^d(t) + g_d G^d(t-1)$$

The combined transient is then

$$G_{\tilde{s}}(t) = \frac{-G^r(t) + G^d(t)}{g_d - g_r}$$

computed in a single pass. All IIR operations are pre-compiled with Numba[20] to minimize overhead.

Inference

Closely following existing methods[16–18], we perform inference with spike addition, removal, and movement operations performed in sequential sweeps during which a set of Gibbs and Hamiltonian Monte Carlo updates are used to select cell-specific parameters.

Throughout this section we write $\mathrm{SSR}(s) = \sum_t (y_t - \hat{y}_t(s))^2$ for the sum of squared residuals of the calcium trace reconstructed from spike train s, and $\Delta\mathrm{SSR} = \mathrm{SSR}(s) - \mathrm{SSR}(s')$ for the reduction in squared error achieved by a proposed spike train s' relative to the current one s. Under the Gaussian observation model above, the corresponding change in log-likelihood is $\Delta\ell = \ell(s') - \ell(s) = \Delta\mathrm{SSR}/(2\sigma^2)$; the implementation tracks the unnormalized quantity $-\mathrm{SSR}(s)$ (the normalizing constant is identical across proposals and cancels in every acceptance ratio below), so all proposal evaluations are expressed here directly in terms of $\Delta\mathrm{SSR}$ rather than $\Delta\ell$.

A movement at time $s'_k = s_k + \nu$ where $\nu \sim N(0, \sigma^2_{move})$ is proposed, with Metropolis–Hastings acceptance ratio

$$\alpha_{\text{move}} = \min\left(1, \exp\left(\frac{\Delta\mathrm{SSR}_{\text{move}}}{\sigma^2}\right)\right)$$

Note that this exponent is twice $\Delta\ell_{\text{move}} = \Delta\mathrm{SSR}_{\text{move}}/(2\sigma^2)$, the change in log-likelihood implied by the Gaussian observation model. The movement acceptance ratio implemented here uses $\exp(\Delta\mathrm{SSR}_{\text{move}}/\sigma^2) = \exp(2\Delta\ell_{\text{move}})$, whereas the addition and removal ratios below use the factor of $2\sigma^2$ that follows directly from the observation model. A candidate time $t^* = U(0, T\Delta t)$ is proposed for addition based on an acceptance ratio using the empirical firing rate λ, estimated once from the initial NNLS spike count and held constant throughout sampling by default:

$$\alpha_{\text{add}} = \min\left(1,\ \exp\left(\frac{\Delta\mathrm{SSR}_{\text{add}}}{2\sigma^2}\right)\frac{T\Delta t}{K+1}\lambda\right)$$

and for removal,

$$\alpha_{\text{remove}} = \min\left(1,\ \exp\left(\frac{\Delta\mathrm{SSR}_{\text{remove}}}{2\sigma^2}\right) * \frac{K}{T\Delta t\lambda}\right)$$

For close-pole sensors (see Sensor-specific parameterization subsection), λ is instead re-estimated from the current spike count at the end of each sweep, allowing the chain to self-correct if the initial estimate is inflated; because α_{add} and α_{remove} above are derived assuming λ is a fixed prior parameter, this substitution is an approximate, empirical-Bayes-style modification rather than an exact step of the reversible-jump sampler (discussed further in Sensor-specific parameterization for fast indicators). The number of addition/removal proposals per sweep is set to $\lceil T/500 \rceil$, scaling with recording duration. The dominant cost to MCMC spike inference is re-computing the full squared residual sum $\mathrm{SSR}(s) = \sum_t (y_t - \hat{y}_t)^2$ over T frames after each proposal, a costly $O(T)$ operation. Here, we compute $\Delta\mathrm{SSR}$ using only the frames within a compact support window. L_{supp} is determined adaptively per cell by truncating the exponential kernel where its amplitude drops below 1% of its peak value, so the effective

window scales automatically with the indicator's decay time constant. For the truncated and normalized rise and decay kernels e_h and e_d of lengths L_h and L_d and where the current residual is $r(t) = y(t) - C_{current}(t)$, when adding a spike at time s^* the resulting change in squared error is

$$\Delta\text{SSR}_{\text{add}} = 2\sum_{k=0}^{L_h-1} r(t_0+k)w_k^h e_k^h - (w^h)^2 * \tilde{n}_{L_h}^h + 2\sum_{k=0}^{L_d-1} r'(t_0+k)w_k^d e_k^d - (w^d)^2 * \tilde{n}_{L_d}^d$$

where $\tilde{n}(j) = \sum_{i=0}^{j-1}(e_i)^2$ is the pre-computed cumulative energy of the kernel, a one-time $O(L)$ computation, and r' denotes the residual after the rise component's contribution has already been subtracted ($r'(t_0+k) = r(t_0+k) - w_k^h e_k^h$ for $k < L_h$, else $r' = r$); the rise term is evaluated first against the raw residual r, and the decay term is then evaluated against r' so that the two components' contributions to the same frames are not double-counted. For removal,

$$\Delta\text{SSR}_{\text{remove}} = -2\sum_{k=0}^{L_h-1} r(t_0+k)w_k^h e_k^h + (w^h)^2 * \tilde{n}_{L_h}^h - 2\sum_{k=0}^{L_d-1} r'(t_0+k)w_k^d e_k^d + (w^d)^2 * \tilde{n}_{L_d}^d$$

with r' defined identically except that the rise contribution is added back ($r'(t_0+k) = r(t_0+k) + w_k^h e_k^h$ for $k < L_h$), since removing the spike restores that component to the residual. The dot-product sums against r and r' must still be evaluated over the L_{supp} frames of the support window for every proposal. Only the self-energy terms $\tilde{n}_{L_h}^h$ and $\tilde{n}_{L_d}^d$ are precomputed and looked up in $O(1)$ (see Optimizations). Because the kernel support L_{supp} is truncated at a precision threshold of 1% peak amplitude, each proposal evaluation costs $O(L_{supp})$. When moving a spike as $s_k \to s_k'$, the combined rise-and-decay window spanning both the old and new spike positions is updated in a single pass, equivalent to

$$\Delta\text{SSR}_{\text{move}} = \Delta\text{SSR}_{\text{remove}}(s_k) + \Delta\text{SSR}_{\text{add}}(s_k'; r')$$

For only a shift proposal, notation is used to indicate $\Delta\text{SSR}_{\text{remove}}$ depends only on the existing spike (s_k) to be removed and $\Delta\text{SSR}_{\text{add}}$ depends on the current spike time (s_k') given the residual no longer containing the old spike's transient (r'). To avoid double-counting, a shift step evaluates the addition term against a corrected residual r' that omits the old spike, unlike a standalone addition which simply uses the raw residual r.

Parameter sampling

The amplitude A, baseline b, and initial calcium C_{in} have the Gaussian joint conditional posterior

$$p(A, b, C_{in} | s, y, \sigma) \propto N(\mu_{\text{post}}, \Sigma_{\text{post}})$$

where s is the spike train, y is the observed calcium trace, and σ is the noise variance. The inverse posterior covariance matrix is

$$\Sigma_{\text{post}}^{-1} = \frac{1}{\sigma^2} M^\top M + \lambda_{\text{prior}} I$$

where the design matrix M has rows $[G_s(t), 1, g_d^t]$ for each observed frame t, where λ_{prior} is the prior precision scale, and I is the identity matrix. Then, the posterior mean vector is

$$\mu_{\text{post}} = \Sigma_{\text{post}} \left(\frac{1}{\sigma^2} M^\top y + \lambda_{\text{prior}} \mu_{\text{prior}} \right)$$

Rather than treating σ^2 as a fixed value, we use a gamma prior as

$$\frac{1}{\sigma^2} = \Gamma\left(\frac{n_0}{2}, \frac{n_0 \hat{\sigma}_{MAD}^2}{2} \right)$$

where $\hat{\sigma}_{MAD} = \text{median}(|\Delta y|/0.6745)$ is a robust noise estimate, 0.6745 is the 75th percentile of the standard normal distribution, and $n_0 = 50$ is a fixed concentration hyperparameter chosen empirically. The conditional posterior is

$$\sigma^2 | s, A, b, C_{in}, y \sim \text{inverse-gamma}\left(\frac{n_0}{2} + \frac{N_{\text{valid}}}{2}, \frac{n_0 \hat{\sigma}_{MAD}^2}{2} + \frac{SSR}{2} \right)$$

where N_{valid} is the number of non-NaN observations and

$$SSR = \Sigma_t (y_t - A * G_s(t) - b - C_{in} * g_d^t)^2$$

A new σ^2 is drawn from this distribution on every sweep. The fully Bayesian treatment propagates uncertainty in σ^2 into the posterior over spike times. The rise and decay time constants τ_r and τ_d are updated via Metropolis–Hastings with independent Gaussian proposals $\tau' \sim N(\tau, \sigma_\tau^2)$, constrained to the valid ranges $\tau_r < \tau_d$ and $\tau_d < \tau_{\max}$. The acceptance ratio is

$$\alpha_\tau = \min\left(1, \exp\left(\frac{\log C' - \log C}{2\sigma^2}\right)\right)$$

where $\log C = -\mathrm{SSR}$ is the unnormalized log-likelihood. When a proposed time constant is accepted, only the affected IIR component (G^r or G^d) is re-filtered over T frames and the other component is reused, halving the cost. After any accepted τ update, the truncated kernels e_h, e_d and their cumulative energy arrays e_{nh}, e_{nd} are rebuilt in a single pass of length L_{supp}, keeping subsequent proposal evaluations efficient.

For indicators with a decay time constant shorter than 0.6 s, time-constant updating is disabled automatically. For fast indicators, the per-spike calcium transient is brief in absolute duration regardless of frame rate, leaving too little of its shape (relative to the noise level) to separately constrain τ_r and τ_d; the initialization-derived time constants are held fixed for the remainder of sampling in these cases.

Initialization with block-wise NNLS

The chain is initialized using block-wise non-negative least squares (NNLS) deconvolution, replacing the constrained FOOPSI optimization of other implementations (which typically requires a dedicated convex solver). For benchmarking against FOOPSI (Figure S2), we implement it as an unconstrained L_1-penalized least-squares deconvolution solved with L-BFGS-B rather than a convex solver. The trace is divided into non-overlapping blocks of 400 frames. For each block, a lower-triangular Toeplitz system is solved with `scipy.optimize.nnls`. Calcium that spills from prior-block spikes is subtracted from the next block before each solve to preserve continuity across borders. This approach is significantly faster than FOOPSI in practice while producing an adequate starting point for MCMC burn-in. For close-pole sensors, block-wise NNLS is run at the full frame rate using the AR(1) kernel substituted automatically for the standard AR(2) kernel (see Sensor-specific parameterization subsection). For justification, see Figure S2.

Sensor-specific parameterization for fast indicators

When the difference between the discrete-time rise and decay poles is small ($g_d - g_r < 0.15$) and the decay time constant spans fewer than 40 frames, as occurs for fast indicators such as GCaMP8 recorded at high frame rates ($\geq$100 Hz), the double-exponential AR(2) kernel is nearly degenerate. At 122 Hz, a nominal rise time of $\tau_r \approx 5$ ms corresponds to only ≈ 0.6 frames; the rise dynamics are sub-frame and not resolvable in discrete time, making the two poles numerically indistinguishable. OMSI checks this criterion automatically for every cell, regardless of how it is invoked, and addresses it by substituting a single-exponential AR(1) model (decay pole only, zero rise pole) evaluated at the full, un-decimated frame rate. This kernel is always well-conditioned irrespective of how close the nominal AR(2) poles would have been, and no temporal resolution is discarded in the process. The empirical firing rate λ is re-estimated from the current spike count at the end of each sweep rather than being held fixed, so that the chain can self-correct if the initial NNLS estimate over-counts spikes.

This re-estimation is a heuristic, empirical-Bayes-style substitution rather than a formally derived step of the reversible-jump sampler: the birth-death acceptance ratios in Inference (Methods) are derived assuming a fixed Poisson prior rate λ, and replacing λ with a state-dependent estimate computed from the chain's own current spike count, on every sweep, means the sampler is no longer guaranteed to target the posterior implied by that fixed-λ prior; it has not been derived here as part of a hierarchical model in which λ is itself sampled or analytically integrated out. We adopt it because it empirically prevents an inflated initial NNLS spike count from persisting for close-pole sensors, but we note this approximation explicitly because it is applied precisely to the GCaMP8 conditions in which OMSI shows its weakest performance against ground truth (see Evaluation on ground-truth datasets in Results), and a departure from exact posterior targeting is a plausible contributing factor alongside the low SNR of those recordings.

Convergence detection and cell pre-screening

The use of a fixed sweep budget may waste compute time on simple traces or spend insufficient time on complex ones. We address this with an adaptive stopping rule for both an initial burn-in phase and for a subsequent sampling phase in which convergence is assessed. Burn-in is declared finished when the means of the first and second halves of the most recent 100-sweep window differ by less than 0.01% of the second-half mean.

After burn-in, the chain is terminated once the mean spike count in the first versus second half of the accumulated post-burn-in samples agree to within $10^{-1.5} \approx 3.16\%$, although there is an enforced 300-sweep minimum and 2000-sweep maximum. Both criteria are evaluated every 50 sweeps to limit overhead.

These thresholds were chosen empirically by sweeping threshold values across a synthetic population of 200 cells and measuring both inference accuracy and compute time. The burn-in amplitude threshold of 0.01% represents a balance between

rapid termination and sufficient chain initialization: looser thresholds reduce burn-in duration but risk initializing with a poorly-conditioned sampler, whereas stricter thresholds ensure robust initialization at modest computational cost (Figure S1c, right panel). The post-burn-in spike-count convergence threshold of $10^{-1.5} \approx 3.16\%$ similarly optimizes computation time while maintaining convergence reliability (Figure S1b, right panel). The adaptive evaluation every 50 sweeps introduces minimal overhead while allowing the sampler to terminate early on low-complexity cells or continue sampling on cells with complex spike trains, subject to the hard minimum of 300 sweeps and maximum of 2,000 sweeps.

When auto-termination is used, the effective sample size adapts to each cell's mixing behavior, rather than imposing a single fixed budget across all cells in the population.

Before launching the MCMC chain, each trace is screened using its SNR. For a cell with SNR < 2.0, the sampler immediately returns zero spikes without running a single MCMC sweep to avoid wasting compute time on noisy cells. This threshold was set from the point at which measured inference quality (F_β, CosMIC) falls below a usable level on synthetic data, rather than from the noise floor of the SNR metric itself (see Optimizations).

Spike detection from probabilities

In addition to the continuous-time raw spike times from the post-burn-in sweeps, OMSI builds a per-frame trace by rounding each posterior sample's spike times to the nearest frame, incrementing that frame's count once for every spike that rounds into it (so that multiple spikes from the same sample landing in one frame are each counted, rather than collapsed to a single occupied-frame indicator), and averaging these per-frame counts across all post-burn-in samples. The result is each frame's expected spike count under the posterior (which can exceed 1 for frames where a sample commonly places more than one spike, as can occur during bursts) rather than a probability strictly bounded to $[0, 1]$. We refer to it as the probability trace for brevity, following common usage, but it is more precisely an occupancy-weighted spike count. OMSI offers two methods for converting this posterior into a single point-estimate spike train, selectable by the user. Posterior-consensus calling is the default.

In posterior-consensus calling, each post-burn-in posterior sample is scored by summing the probability trace at that sample's own (rounded) spike times and subtracting a penalty of 0.5 for every spike in excess of the posterior mean spike count. The sample with the highest score is returned directly as the continuous-time spike-time estimate. This score rewards agreement with the marginal, per-frame posterior consensus and penalizes excess spikes relative to the posterior mean count; it is a heuristic criterion rather than an evaluation of each sample's joint posterior probability, so the selected sample is not guaranteed to be the maximum a posteriori (MAP) spike train in the formal sense. Because the returned spike train is nonetheless an actual posterior sample rather than a derived statistic, it requires no additional smoothing, binning, or thresholding, and spike times retain the sub-frame precision of the original sample.

Alternatively, Otsu-threshold calling is available as a binned, smoothed estimate of the same posterior: the probability trace is smoothed with a Gaussian kernel ($\sigma = \max(1.5,\ 0.02 f_s)$ frames) and local maxima are identified using a standard peak-finding algorithm. Their heights are thresholded using an Otsu criterion[24] applied to the empirical distribution of peak amplitudes. The threshold is set below the mean of the noise-peak cluster by five standard deviations, with a minimum value of 0.001. Peaks exceeding this threshold and separated by at least 100 milliseconds are reported as spike events and converted to time in units of seconds.

General metrics

Precision and recall were calculated using the values of true positives (TP), false positives (FP) and false negatives (FN) as

$$\text{Precision} = \frac{TP}{TP+FP} \quad \text{and} \quad \text{Recall} = \frac{TP}{TP+FN}$$

while F_β was calculated as

$$F_\beta = \frac{(1+\beta^2) * P * R}{\beta^2 * P + R} \quad \text{where} \quad \beta = 0.5$$

Signal-to-noise ratio (SNR) was calculated as

$$\text{SNR} = \frac{F_{99} - F_8}{\hat{\sigma}}, \quad \hat{\sigma} = \frac{\text{median}(|\Delta F|)}{0.6745}$$

where F_{99} and F_8 are the 99th and 8th percentiles of the fluorescence trace, and $\hat{\sigma} = \text{median}(|\Delta F|)/0.6745$ is a MAD-based noise estimate. Note that the theoretically correct form for estimating σ from first differences is $\text{median}(|\Delta F|)/(0.6745\sqrt{2})$, because $\text{Var}(\Delta F) = 2\sigma^2$ under independent and identically distributed Gaussian noise. The form without $\sqrt{2}$ is used here and in the noise prior as a slightly conservative estimate. Where precise calibration of a detection threshold is required (as in

OASIS spike detection, see Methodological comparisons), the variance-corrected form is used explicitly. Excess kurtosis was calculated as

$$K_4 = \frac{\mu_4}{\sigma^4} - 3 = \frac{\frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})^4}{\left(\frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})^2\right)^2} - 3$$

Synthetic data generation

Ground-truth spike trains are first drawn on a time grid upsampled 10-fold relative to the population's target frame rate f_s (i.e. at $f_{s,\text{high}} = 10 f_s$). At every high-resolution bin, cell i emits a spike as an independent Bernoulli trial with probability $\lambda_i / f_{s,\text{high}}$, approximating a Poisson process with firing rate λ_i. Firing rates are drawn as $\lambda_i \sim \text{Lognormal}(\log 0.2, 1.0)$ and clipped to $[0.01, 4.0]$ Hz. A randomly chosen 15% of cells are designated "bursty": after each of that cell's primary spikes, with probability 0.6 an additional $n \sim \text{Uniform}\{2, \ldots, 5\}$ spikes are inserted at $2, 4, \ldots, 2n$ high-resolution bins after the primary spike (i.e. spaced $2/f_{s,\text{high}}$ apart), giving burst lengths of 3-6 spikes (the primary spike plus 2-5 extras).

Each cell's noise-free calcium trace is obtained by convolving its high-resolution binary spike train with a double-exponential kernel with fixed rise time $\tau_r = 50$ ms and the decay time τ_d specified for that population, normalized to unit peak height. Every spike therefore contributes the same nominal unit amplitude. No additional per-cell or per-spike amplitude is drawn, and the baseline offset b is fixed at 0 throughout. The convolved trace is resampled from $f_{s,\text{high}}$ down to the population's target frame rate f_s (by decimation when $f_{s,\text{high}}/f_s$ is an integer, otherwise by linear interpolation).

Independent Gaussian noise $\varepsilon \sim N(0, \sigma^2)$ is then added to each cell's clean trace. The noise level σ is set in one of two ways. When a target SNR is specified directly for the population (e.g. the SNR-sensitivity benchmark, Figure 2e), $\sigma = (F_{99} - F_1)/\text{SNR}_{\text{target}}$, where F_{99} and F_1 are the 99th and 1st percentile of the clean trace. Otherwise (the default used for Figure 1, 2b-d, 3a-c, S1a, S1d, and S2), a per-cell target excess kurtosis is instead drawn as $K_{\text{target}} = K_{\min} + \text{Exponential}(\text{scale} = (K_{\max} - K_{\min})/3)$, clipped to $[K_{\min}, K_{\max}]$ (default $[4, 50]$; $[0, 25]$ for Figures 1 and S3), and σ is solved for analytically.

Every synthetic population used for a given figure or benchmark condition is drawn independently; no two panels in this manuscript share an identical realized dataset, even where the generative parameters (cell count, duration, frame rate, decay time, kurtosis or SNR range) are identical. One exception is the SNR-sensitivity benchmark (Figure 2e), where a single 50-cell spike-train population is generated once (at a near-noise-free SNR) and then re-noised independently at each tested SNR level, so that only the injected noise differs across that sweep. Table 1 lists the generative parameters for every synthetic population reported in this manuscript.

Figure/panel	N cells	Duration	f_s (Hz)	τ_d (s)	Noise spec.
Fig. 1 (a-g)	500	1200 s	30	1.2	$K \in [0, 25]$
Fig. 2a (sweep count)	50	600 s	30	1.2	$K \in [4, 50]$
Fig. 2b (cell count)	50-3000	300 s	30	1.2	$K \in [4, 50]$
Fig. 2b (duration)	100	300-7200 s	30	1.2	$K \in [4, 50]$
Fig. 2d (τ_d sweep)	50	300 s	30	0.2-2.0	$K \in [4, 50]$
Fig. 2e/3a,b (f_s sweep)	50	300 s	7.5-100	1.2	$K \in [4, 50]$
Fig. 2e (SNR sweep)	50	300 s	30	1.2	SNR direct, [1,100]
Fig. 3c (7.5 vs. 30 Hz)	50	300 s	7.5, 30	1.2	$K \in [4, 50]$
Fig. S1a (support window)	200	2400 s	30	1.2	$K \in [4, 50]$
Fig. S1b (conv. tol.)	100	1200 s	30	1.2	SNR$\sim 2 + \text{Exp}(2)$, [2,20]
Fig. S1c (burn-in tol.)	100	1200 s	30	1.2	SNR$\sim 2 + \text{Exp}(2)$, [2,20]
Fig. S1d (SNR screening)	200	2400 s	30	1.2	$K \in [4, 50]$
Fig. S2a-c (NNLS vs. FOOPSI)	200	300 s	30	1.2	$K \in [4, 50]$
Fig. S3 (OASIS/CASCADE thresh.)	100	600 s	30	1.2	$K \in [0, 25]$

Table 1. Generative parameters for every synthetic population reported in this manuscript. K denotes the excess-kurtosis target range used for noise calibration; where a direct SNR is listed instead, noise was calibrated to the stated SNR rather than to a kurtosis target.

Ground-truth dataset preprocessing and inclusion criteria

For the Allen Institute simultaneous ephys-imaging dataset[8] (Figure 4), raw somatic and neuropil fluorescence (F, F_{neu}) were neuropil-corrected as $F_{\text{corr}} = F - 0.7 F_{\text{neu}}$. A smoothed baseline was estimated by first applying a zero-phase (`filtfilt`), 3rd-order Butterworth low-pass filter with a 5 Hz cutoff to F_{corr}, then taking the 8th percentile of the filtered trace within a sliding 60-second window (a single global 8th percentile was used instead for recordings shorter than 60 s). $\Delta F/F$ was then computed as $(F_{\text{corr}} - F_{\text{baseline}})/\max(F_{\text{baseline}}, 1)$, where the floor of 1 (in raw fluorescence units) prevents division blow-up for near-zero or negative baselines. The low-pass filter is used only to stabilize this baseline estimate and is not applied to the

$\Delta F/F$ trace itself. Ground-truth ephys spike times were assigned to the nearest imaging frame using the recording's per-frame ephys-sample index, restricted to spikes falling within the imaged epoch, and spikes landing in the same frame less than 3 ms apart were deduplicated to one event.

Cells were included in the Figure 4 analysis if their $\Delta F/F$ excess kurtosis was ≥ 0.5; this removed cells with insufficient event-like structure in the fluorescence trace before spike inference was attempted (105 of the recorded cells passed this filter and are reported in Figure 4: 87 high-zoom and 18 low-zoom; 64 with $\tau_d = 0.5$ s (GCaMP6f) and 41 with $\tau_d = 1.2$ s (GCaMP6s); by zoom$\times\tau_d$: 51 high-zoom/GCaMP6f, 13 low-zoom/GCaMP6f, 36 high-zoom/GCaMP6s, 5 low-zoom/GCaMP6s). High- and low-zoom recordings of a given experiment are distinct imaging sessions in the source dataset rather than a single re-analyzed recording, and are not cell-registered against one another; both are retained and are always reported as separate groups (Figure 4b,c,f) rather than pooled or deduplicated. The calcium decay time constant ($\tau_d = 0.5$ or 1.2 s) assigned to each recording follows directly from the sensor (GCaMP6f or GCaMP6s) used in that session.

For the expanded multi-sensor comparison (Figure S4, S5), three source datasets (one each of GCaMP7f, GCaMP8s, and XCaMPgf) were excluded outright because they showed temporal misalignment between the imaging and electrophysiology streams that was not consistent across cells within the same recording. This was identified qualitatively, from all four independently-implemented inference methods performing uniformly poorly on these recordings, rather than from an automated per-cell lag metric. The GCaMP5k, OGB1, and jGECO sensors were also excluded from the final comparison, leaving five sensors reported in Figure S4 (GCaMP6f, GCaMP6s, GCaMP8f, GCaMP8m, jRCaMP). Recordings from inhibitory interneurons were excluded because all four methods compared here are designed and validated on excitatory principal cells, and interneuron firing rates routinely exceed the Nyquist limit of the imaging frame rates used, making individual spike resolution infeasible regardless of inference method.

Methodological comparisons

To fairly compare MCMC methods to OASIS and CASCADE, we set out to get the most accurate continuous-time spike detection from their binned spike rates. We use a peak-based spike detection rather than simple frame-by-frame thresholding. For OASIS, the deconvolved spike amplitude trace s is thresholded at 1.0σ (one noise standard deviation, estimated as the MAD of first differences of the fluorescence trace, $\hat{\sigma} = \text{median}(|\Delta F|)/(0.6745\sqrt{2})$), and a standard peak-finding algorithm returns only local maxima exceeding that height with a minimum inter-peak distance of 5% of the frame rate in frames. For CASCADE, the smoothed spike-probability trace is processed using a similar peak-finding procedure, but with a fixed probability threshold of 0.5, meaning only frames where the model is at least 50% confident a spike occurred are considered, with the same minimum inter-peak distance. This inter-peak constraint in both cases prevents a single slow calcium transient from being counted as multiple spikes. For justification of threshold selection, see Figure S3.

For methodological comparisons, we used OASIS 0.2.1, CASCADE 2.0, and CaImAn 0.7.

All methods were run on a basic lab workstation with a 12th Gen Intel Core i5-12600K (10 physical cores: 6 performance + 4 efficiency, 16 logical threads), 32 GB RAM, an Nvidia GeForce RTX 3060 with 12 GB vRAM, and Ubuntu 24.04.4.

Data availability

All code for OMSI is publicly available from https://github.com/ucsb-goard-lab/OMSI. Data for ground truth comparisons are publicly available through[8,15].

References

1. Peron, S., Freeman, J., Iyer, V., Guo, C. & Svoboda, K. A cellular resolution map of barrel cortex activity during tactile behavior. *Neuron* **86**, 783–799, DOI: 10.1016/j.neuron.2015.03.027 (2015).
2. Huber, D. *et al.* Multiple dynamic representations in the motor cortex during sensorimotor learning. *Nature* **484**, 473–478, DOI: 10.1038/nature11039 (2012).
3. Sofroniew, N. J., Flickinger, D., King, J. & Svoboda, K. A large field of view two-photon mesoscope with subcellular resolution for in vivo imaging. *eLife* **5**, e14472, DOI: 10.7554/eLife.14472 (2016).
4. Stringer, C., Pachitariu, M., Steinmetz, N., Carandini, M. & Harris, K. D. High-dimensional geometry of population responses in visual cortex. *Nature* **571**, 361–365, DOI: 10.1038/s41586-019-1346-5 (2019).
5. Yu, C.-H., Stirman, J. N., Yu, Y., Hira, R. & Smith, S. L. Diesel2p mesoscope with dual independent scan engines for flexible capture of dynamics in distributed neural circuitry. *Nat Commun* **12**, 6639, DOI: https://doi.org/10.1038/s41467-021-26736-4 (2021).
6. Peters, A. J., Chen, S. X. & Komiyama, T. Emergence of reproducible spatiotemporal activity during motor learning. *Nature* **510**, 263–267, DOI: 10.1038/nature13235 (2014).

7. Marks, T. D. & Goard, M. J. Stimulus-dependent representational drift in primary visual cortex. *Nat. Commun.* **12**, 5169, DOI: 10.1038/s41467-021-25436-3 (2021).
8. Huang, L. *et al.* Relationship between simultaneously recorded spiking activity and fluorescence signal in gcamp6 transgenic mice. *eLife* **10**, e51675, DOI: https://doi.org/10.7554/eLife.51675 (2021).
9. Scheuss, V., Yasuda, R., Sobczyk, A. & Svoboda, K. Nonlinear [ca2+] signaling in dendrites and spines caused by activity-dependent depression of ca2+ extrusion. *The J. Neurosci.* **26**, 8183–8194, DOI: 10.1523/JNEUROSCI.1962-06.2006 (2006).
10. Tian, L. *et al.* Imaging neural activity in worms, flies and mice with improved gcamp calcium indicators. *Nat. Methods* **6**, 875–881, DOI: 10.1038/nmeth.1398 (2009).
11. Chen, T.-W. *et al.* Ultrasensitive fluorescent proteins for imaging neuronal activity. *Nature* **499**, 295–300, DOI: 10.1038/nature12354 (2013).
12. Akerboom, J. *et al.* Optimization of a gcamp calcium indicator for neural activity imaging. *The J. Neurosci.* **32**, 13819–13840, DOI: 10.1523/JNEUROSCI.2601-12.2012 (2012).
13. Pologruto, T. A., Yasuda, R. & Svoboda, K. Monitoring neural activity and [ca2+] with genetically encoded ca2+ indicators. *The J. Neurosci.* **24**, 9572–9579, DOI: 10.1523/JNEUROSCI.2854-04.2004 (2004).
14. Friedrich, J., Zhou, P. & Paninski, L. Fast online deconvolution of calcium imaging data. *PLOS Comput. Biol.* **13**, e1005423, DOI: https://doi.org/10.1371/journal.pcbi.1005423 (2017).
15. Rupprecht, P. *et al.* A database and deep learning toolbox for noise-optimized, generalized spike inference from calcium imaging. *Nat. Neurosci.* **24**, 1324–1337, DOI: https://doi.org/10.1038/s41593-021-00895-5 (2021).
16. Vogelstein, J. T. *et al.* Spike inference from calcium imaging using sequential monte carlo methods. *Biophys. J.* **97**, 636–655, DOI: https://doi.org/10.1016/j.bpj.2008.08.005 (2009).
17. Pnevmatikakis, E. A., Merel, J., Pakman, A. & Paninski, L. Bayesian spike inference from calcium imaging data, DOI: https://doi.org/10.48550/arXiv.1311.6864 (2013). 1311.6864.
18. Pnevmatikakis, E. A. *et al.* Simultaneous denoising, deconvolution, and demixing of calcium imaging data. *Neuron* **20**, 285–299, DOI: https://doi.org/10.1016/j.neuron.2015.11.037 (2016).
19. Vogelstein, J. T. *et al.* Fast nonnegative deconvolution for spike train inference from population calcium imaging. *J. Neurophysiol.* **104**, 3691–3704, DOI: 10.1152/jn.01073.2009 (2010). PMID: 20554834, https://doi.org/10.1152/jn.01073.2009.
20. Lam, S. K., Pitrou, A. & Seibert, S. Numba: a llvm-based python jit compiler. In *Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC*, LLVM '15, DOI: 10.1145/2833157.2833162 (Association for Computing Machinery, New York, NY, USA, 2015).
21. Moritz, P. *et al.* Ray: A distributed framework for emerging AI applications. *CoRR* **abs/1712.05889** (2017).
22. Giovannucci, A. *et al.* Caiman: An open source tool for scalable calcium imaging data analysis. *eLife* **8**, e38173, DOI: https://doi.org/10.7554/eLife.38173 (2019).
23. Reynolds, S., Abrahamsson, T., Sjöström, P. J., Schultz, S. R. & Dragotti, P. L. Cosmic: A consistent metric for spike inference from calcium imaging. *Neural Comput.* **30**, 2726–2756, DOI: https://doi.org/10.1162/neco_a_01114 (2018).
24. Otsu, N. A threshold selection method from gray-level histograms. *IEEE Transactions on Syst. Man, Cybern.* **9**, 62–66, DOI: https://doi.org/10.1109/TSMC.1979.4310076 (1979).

Acknowledgments

We thank Dr. Spencer Smith, Dr. Will Redman, Dr. Liam Paninski, and Filip Tomaska for helpful discussions and feedback.

Funding

This work was supported by NIH grants F31EY037558 (DMM) and R01NS121919 (MJG).

Author contributions statement

DMM conceived of the project. MJG supervised all aspects of the project. DMM wrote the algorithm and analyzed results. DMM and MJG wrote the manuscript.

Competing interests

The authors declare no competing interests.

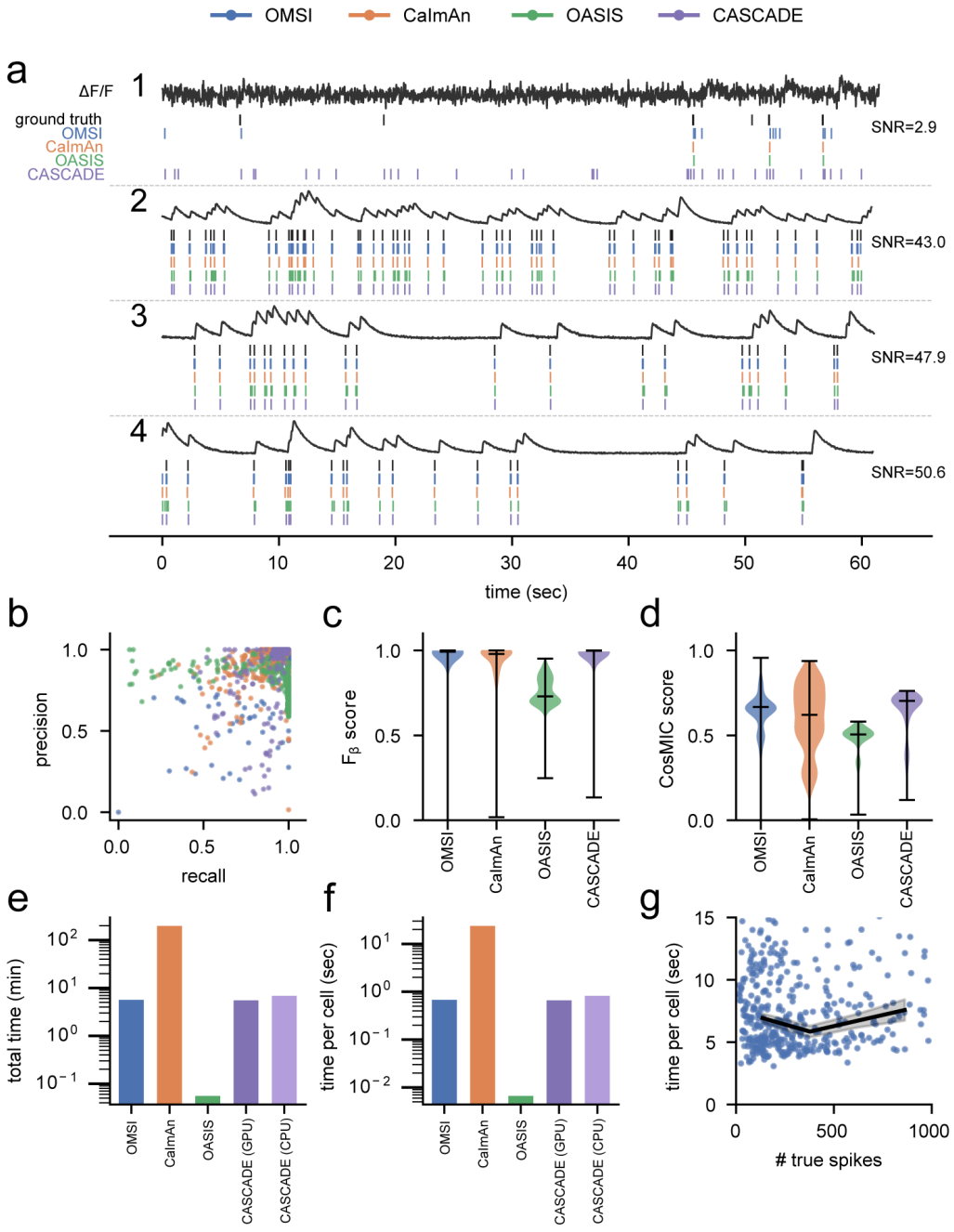


Figure 1. Fixed-parameter benchmarks on simulated data. **(a)** True calcium trace for example simulated cells and each method's predictions as a raster. Ground truth in black, OMSI in blue, CaImAn MCMC in orange, CASCADE in purple, and OASIS in green. **(b)** Recall and precision for each method. Each point is a cell, N=500. **(c)** F_β scores ($\beta = 0.5$). **(d)** Same as (c) for CosMIC scores. **(e)** Total time to run inference on 500 cells. **(f)** Same as (e) for mean time per cell. **(g)** Time per cell for OMSI as a function of the number of total true spikes for each cell. Each point is a cell. Running median with standard error is shown. Cells with t=0 seconds, which had a kurtosis value below threshold, are excluded from the plot.

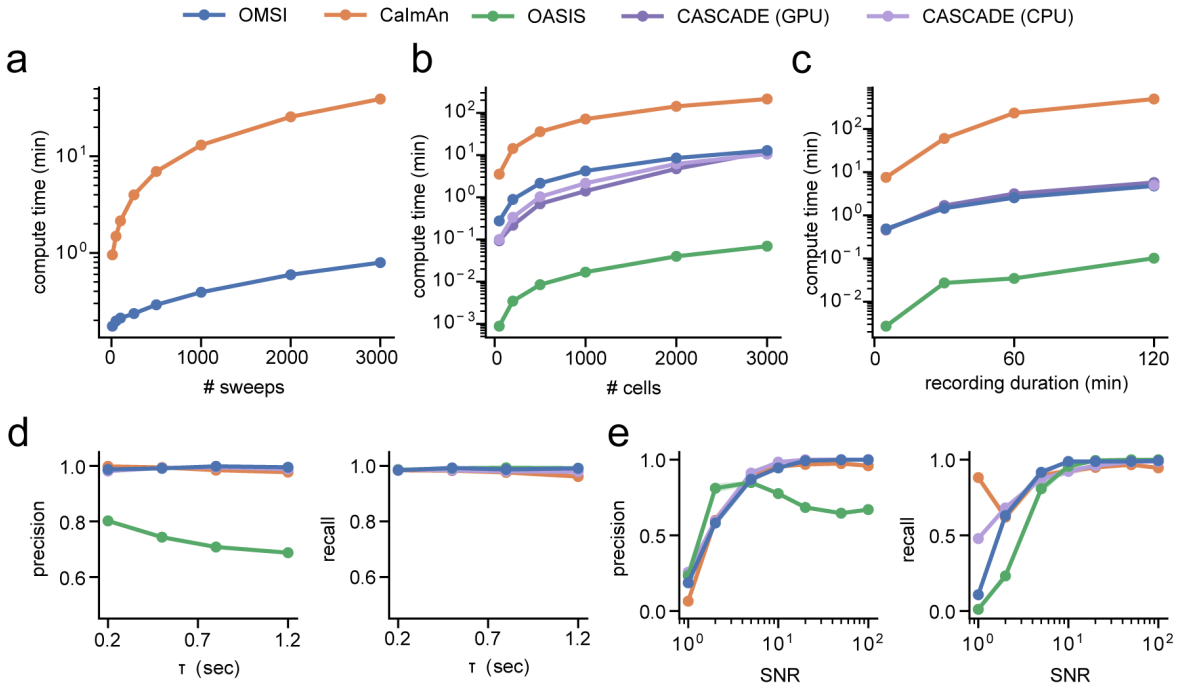


Figure 2. Swept-parameter benchmarks on simulated data. **(a)** Compute time as a function of the number of sweeps for the two MCMC implementations, **(b)** number of cells for all methods, and **(c)** minutes of recording duration. Each point along the line is a separately fit model to a separate dataset, but each method is used to infer spikes from identical calcium traces. **(d)** Precision and recall as a function of mean decay time constant (τ). **(e)** Same as (d) for binned signal-to-noise ratio (SNR).

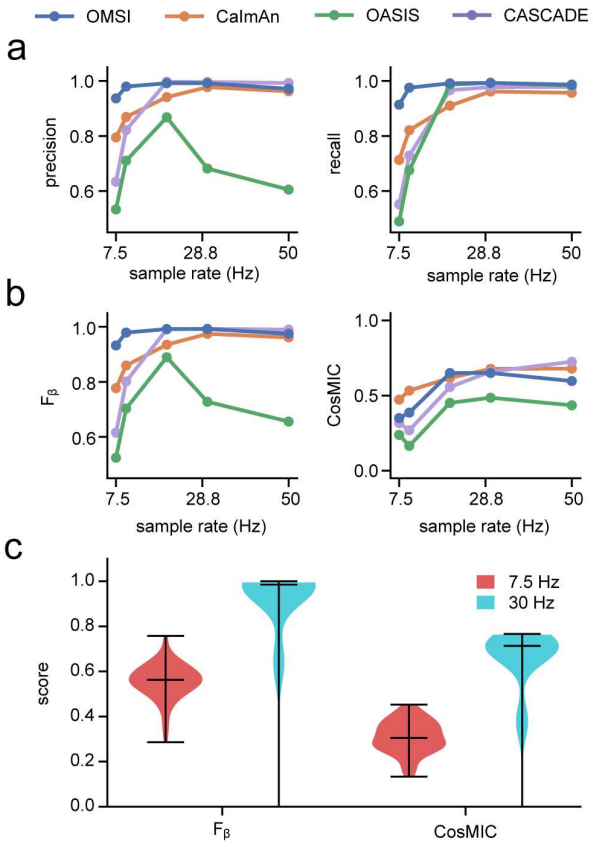


Figure 3. Swept-parameter benchmarks on simulated data. **(a)** Precision and recall as functions of sample rates for simulated data. **(b)** Same as (a) for F_β. **(c)** F_β and CosMIC scores for CASCADE on datasets recorded at 7.5 Hz and 30 Hz. For each condition, the model was pre-trained on datasets at the sample rate they were tested on. However, the 7.5 Hz dataset was a downsampled version of 30 Hz data, as no true 7.5 Hz ground truth datasets exist.

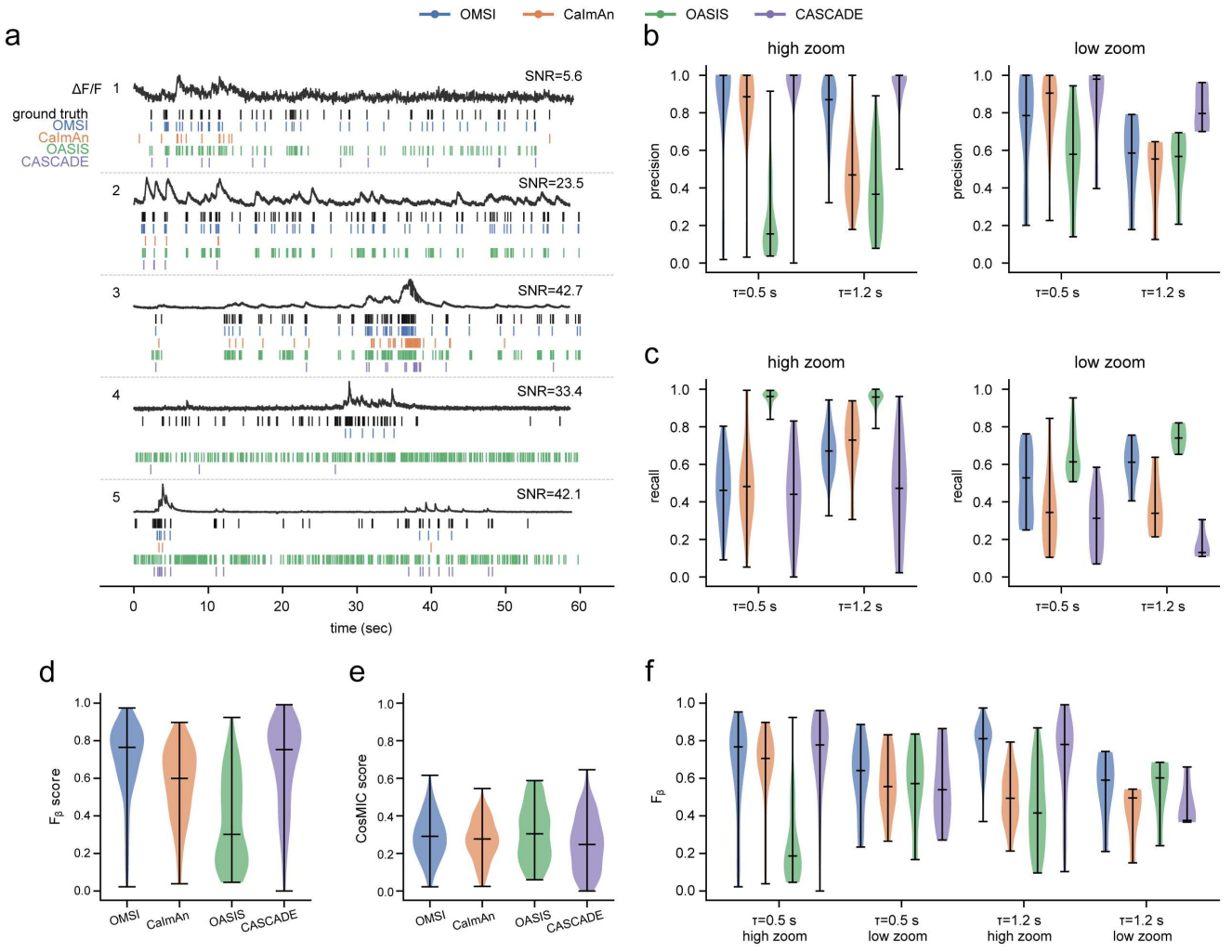


Figure 4. Evaluation on Allen Brain Institute ground-truth data. **(a)** Calcium traces and spike raster for five example cells along with the inference of each method. **(b)** Precision in high (left) and low (right) zoom conditions. **(c)** Same as (b) for recall. **(d)** F_β scores on all cells in the dataset, pooling across recording conditions and τ_d values. **(e)** Same as (d) for CosMIC. **(f)** F_β grouped by decay time constant (τ_d), sensor, and zoom condition.

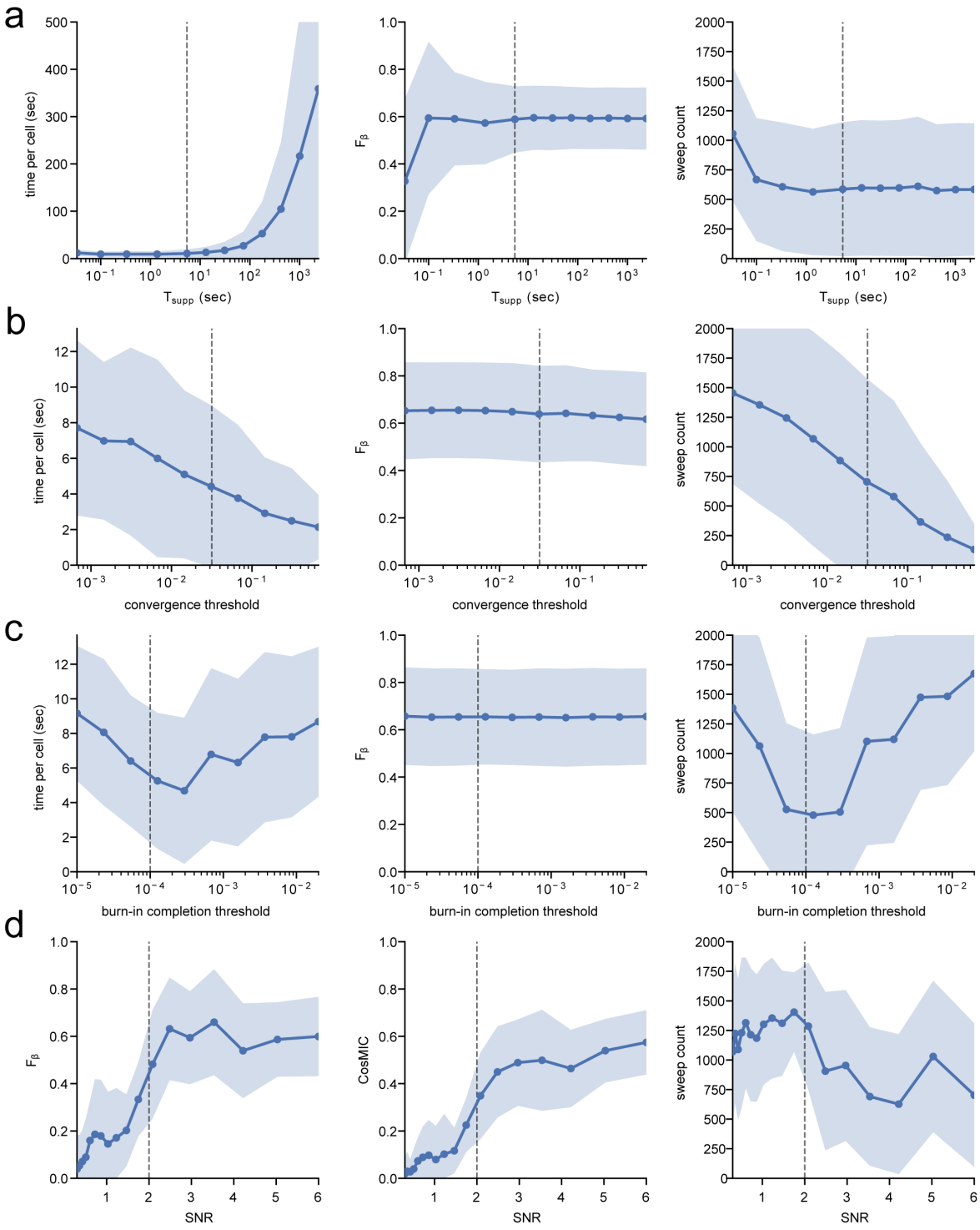


Figure S1. Characterization of optimization parameters. **(a)** Compute time per cell (left), accuracy measured via F_β (middle), and sweep count (right) as functions of the likelihood-evaluation support window duration (T_{supp}). Vertical line indicates default value. **(b)** Markov chain convergence threshold: compute time per cell (left), accuracy measured via F_β (middle), and sweep count (right) as functions of convergence threshold stringency. Vertical line indicates default threshold. **(c)** Burn-in completion threshold: compute time per cell (left), accuracy measured via F_β (middle), and sweep count (right) as functions of burn-in threshold stringency. Vertical line indicates default threshold. The left column in (a)-(c) illustrates the trade-off between threshold stringency and per-cell compute time, while the right column shows the corresponding number of sweeps run, with the chosen thresholds positioned to minimize inference time while maintaining convergence reliability. **(d)** Accuracy measured by F_β (left) and CosMIC (middle), and sweep count (right) as functions of the signal-to-noise ratio (SNR) of cells. Vertical line indicates default threshold.

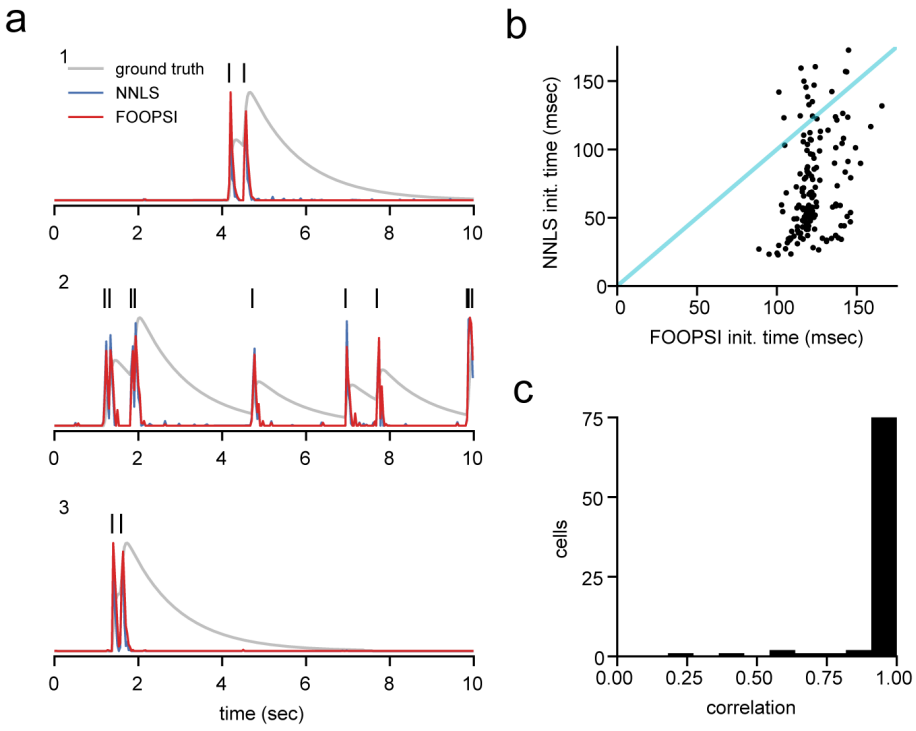


Figure S2. Initialization via NNLS versus FOOPSI. **(a)** Non-negative least squares (NNLS) and FOOPSI initializations for three example synthetic cells. Normalized spike probability is shown for each method alongside ground truth spike raster (above) and ground truth synthetic calcium trace. **(b)** Compute time for the two methods on a population of 200 synthetic cells. **(c)** Correlation between NNLS and FOOPSI initialization traces.

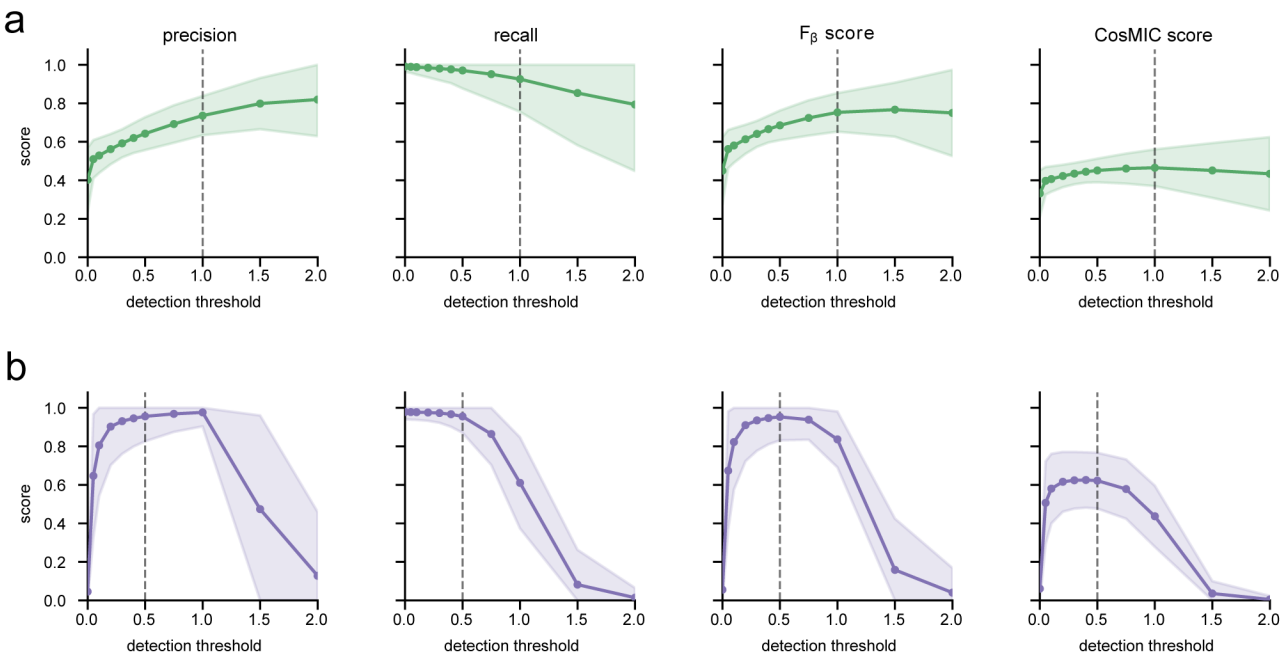


Figure S3. Isolating continuous-time spikes from binned predictions. **(a)** Precision, recall, F_β, and CosMIC scores for a simulated population using a range of thresholds to identify continuous-time spikes from binned output for OASIS. **(b)** Same as (a) for CASCADE.

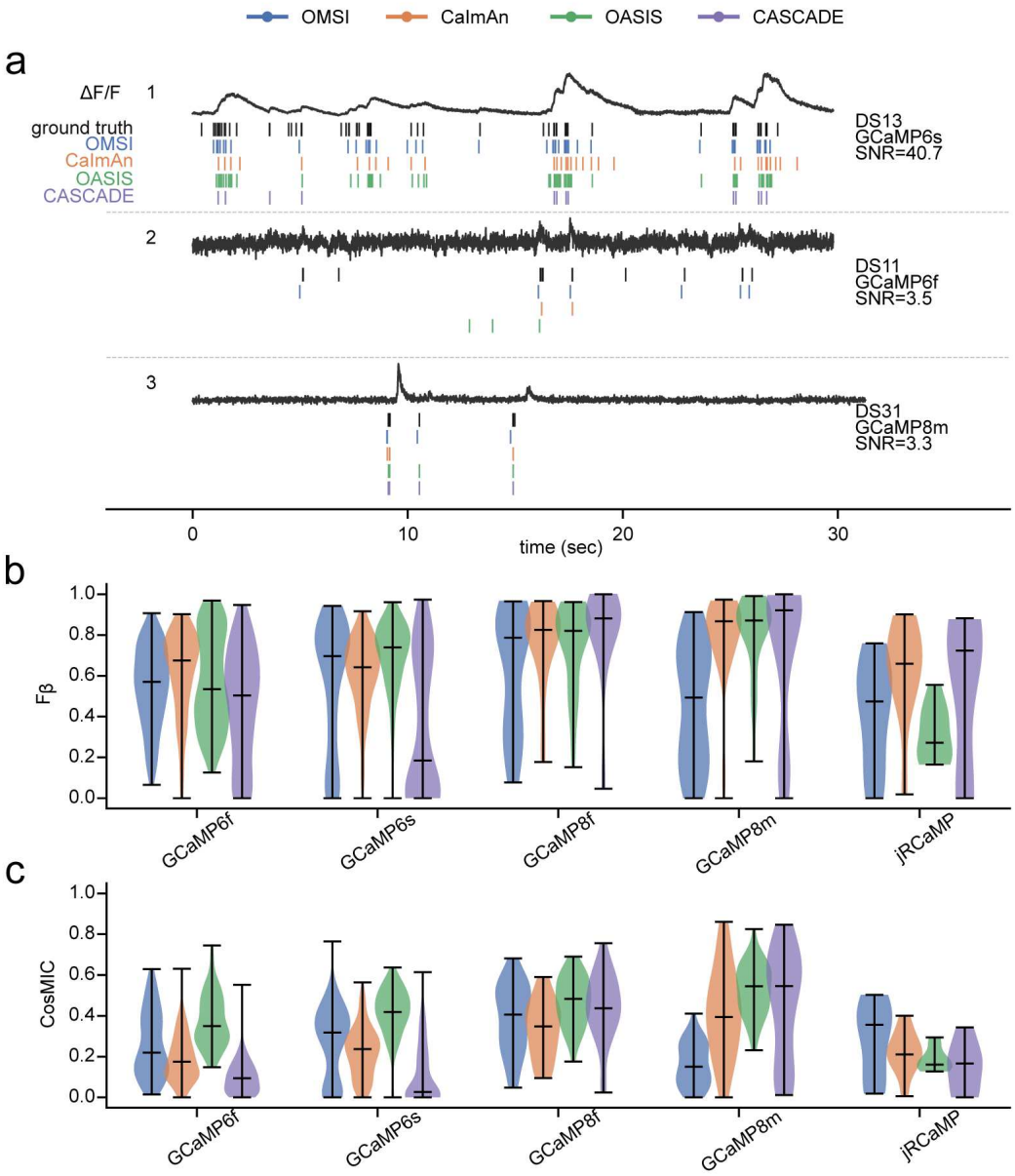


Figure S4. Evaluation on ground truth data with a range of fluorescent sensors. **(a)** Raster for four example cells. **(b)** F_β scores for each sensor (GCaMP6f, GCaMP6s, GCaMP8f, GCaMP8m, jRCaMP). N=[212,103,42,36,9] **(c)** Same as (b) for CosMIC scores.

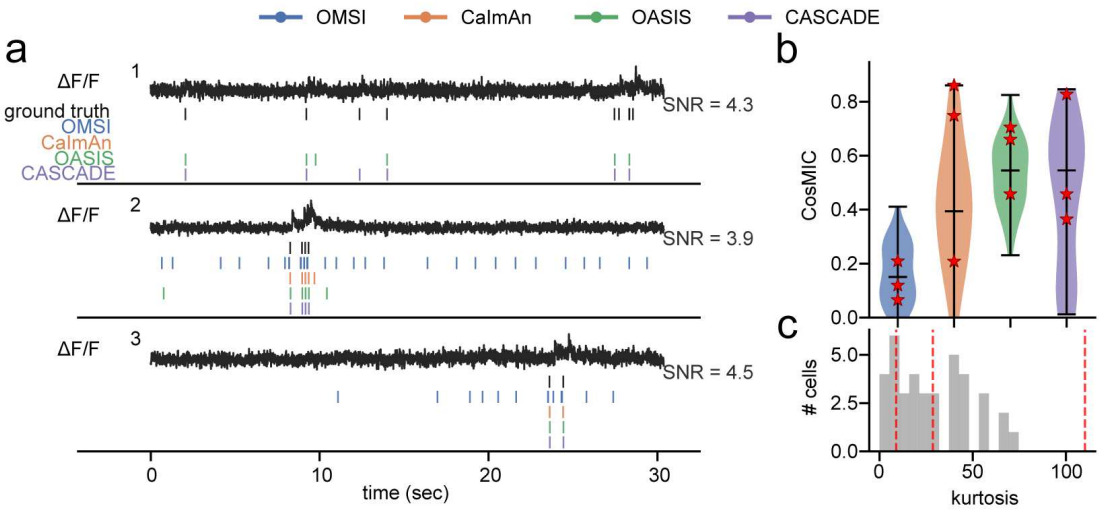


Figure S5. Evaluation for six example ground-truth cells for two GCaMP8 sensors. **(a)** Calcium traces and spike predictions for six cells of GCaMP8m. **(b)** Distribution of CosMIC scores in which red stars indicate the position of the three example cells along that distribution. **(c)** Distribution of excess kurtosis values in the population, with vertical red lines indicating the position of the three example cells at left.