S Supplemental Sections

S.1 Optopatch voltage imaging of chronically TTX-treated and unperturbed hPSC-derived neurons

hPSC-derived neurons differentiation was performed as previously described [26, 27], followed by Optopatch voltage imaging also as previously described [28]. Methods are reproduced below for completeness.

hPSC Culture— Human ESCs were maintained on plates coated with Geltrex (Life Technologies, A1413301) in mTeSR Plus medium (StemCell Technologies, 100-1130) and passaged with Accutase (Gibco, A11105). All cell cultures were maintained at 37 °C, 5% CO2.

Neuronal Induction— hPSC-derived neurons were differentiated from an hPSC line (H1/WA01) using combined NGN2 programming with SMAD and WNT inhibition in the presence of mouse astrocytes [26, 27]. On day 0, hPSCs were differentiated in N2 medium (500 mL DMEM/F12 [1:1] [Gibco, 11320-033]), 5 mL Glutamax (Gibco, 35050-061), 7.5 mL sucrose (20%, Sigma, S0389), 5 mL N2 supplement B (StemCell Technologies, 07156) supplemented with SB431542 (10 µM, Tocris, 1614), XAV939 (2 µM, Stemgent, 04-00046), and LDN-193189 (100 nM, Stemgent, 04-0074) along with doxycycline hyclate (2 $\mu g.mL^{-1}$, Sigma, D9891) and Y27632 (5 mM, Stemgent 04-0012). On day 1 and 2 media was changed to N2 medium supplemented with SB431542 (5 µM, Tocris, 1614), XAV939 (1 µM, Stemgent, 04-00046), and LDN-193189 (50 nM, Stemgent, 04-0074) with doxycycline hyclate (2 $\mu g.mL^{-1}$, Sigma, D9891) and Zeocin (1 $\mu g.mL^{-1}$, Invitrogen, 46-059). On Day 3 neuronal precursor cells were passaged with Accutase into Neurobasal media (500 mL Neurobasal [Gibco, 21103-049], 5 mL Glutamax [Gibco, 35050-061], 7.5 mL Sucrose [20%, Sigma, S0389], 2.5 mL NEAA [Corning, 25-0250 Cl]) supplemented with B27 (50x, Gibco, 17504-044), BDNF, CTNF, GDNF (10 $ng.mL^{-1}$, R&D Systems 248-BD/CF, 257-NT/CF, and 212-GD/CF) and doxycycline hyclate (2 $\mu g.mL^{-1}$, Sigma, D9891) in a 24-well format and infected with lentiviral optogenetic constructs (HT076, hSyn Cre-off Archon-TS-darkCitrine-TSx3-ER) at 2 MOI. On day 7, the cells were passaged with Accutase onto 10mm glass coverslip bottom dishes precoated with Geltrex containing a monolayer of mouse cortical astrocytes. Estimated neuron/astrocyte ratio was 1:2 with 80k neurons plated per 10mm dish. Cell were matured in Neurobasal media supplemented with B27 (50x, Gibco, 17504-044), BDNF, CTNF, GDNF (10 $ng.mL^{-1}$, R&D Systems 248-BD/CF, 257-NT/CF, and 212-GD/CF) and doxycycline hyclate (2 $\mu g.mL^{-1}$, Sigma, D9891) with 50% media changes twice a week.

TTX treatment and Optopatch imaging— On Day 35 500 nM TTX (Tocris) was added to the culture media and cultures were returned to the incubator for additional 48 hours. Parallel control cultures were kept in TTX-free media. 10 minutes prior to recording the cultures were washed 3 times in pre-warned recording solution (125 mM NaCl, 2.5 mM KCl, 3 mM CaCl2, 1 mM MgCl2, 15 mM HEPES, 30 mM glucose (pH 7.3) and adjusted to 305–310 mOsm with sucrose) to wash out the TTX. Recordings were obtained in recording solution in the absence of TTX at 23 °C. Cellular activity was recorded on custom-built wide-field microscope equipped with oblique illumination lens and a wide 20x objective. The cells were stimulated with 500 ms blue light (488 nm) at 1 Hz of increasing intensity (20 to 120 mW/cm^2) for 6 seconds, while firing patterns were recorded under continuous red light (635 nm) illumination at 1 kHz.

S.2 Simultaneous BeRST fluorescence voltage imaging and single-cell patch-clamp EP recording experimental procedure

Simultaneous BeRST imaging and single-cell patch-clamp EP recordings were performed as described previously [20]. Methods are reproduced below for completeness.

Cell Culture— All animal procedures were approved by the UC Berkeley Animal Care and Use Committees and conformed to the NIH Guide for the Care and Use of Laboratory Animals and the Public Health Policy.

Rat Hippocampal Neurons— Hippocampi were dissected from embryonic day 18 Sprague Dawley rats (Charles River Laboratory) in cold sterile HBSS (zero Ca2+, zero Mg2+). All dissection products were supplied by Invitrogen, unless otherwise stated. Hippocampal tissue was treated with trypsin (2.5%) for 15 min at 37 °C. The tissue was triturated using fire polished Pasteur pipettes, in minimum essential media (MEM) supplemented with 5% fetal bovine serum (FBS; Thermo Scientific), 2% B-27, 2% 1 M D-glucose (Fisher Scientific) and 1% GlutaMax. The dissociated cells (neurons and glia) were plated onto 12 mm diameter coverslips (Fisher Scientific) pre-treated with PDL at a density of 30-40,000 cells per coverslip in MEM supplemented media (as above). Cells were maintained at 37 °C in a humidified incubator with 5% CO2. At 1 day in vitro (DIV), half of the MEM supplemented media was removed and replaced with FBS-free media to supress glial cell growth (Neurobasal media containing 2% B-27 supplement and 1% GlutaMax). Functional imaging was performed on 8-15 DIV neurons to access neuronal excitability and connectivity across different stages of development. References to biological replicates, or "n," refer to the number of dissections data were collected from.

VoltageFluor/BeRST 1 Stocks and Cellular Loading— For all imaging experiments, BeRST 1 was diluted from a 250 µM DMSO stock solution to 0.1-1µM in HBSS (+Ca2+, +Mg2+, -phenol red). To load cells with dye solution, the media was first removed from a coverslip and then replaced with the BeRST-HBSS solution. The dye was then allowed to load onto the cells for 20 minutes at 37 °C in a humidified incubator with 5% CO2. After dye loading, coverslips were removed from the incubator and placed into an Attofluor cell chamber filled with fresh HBSS for functional imaging at room temperature (20-23 °C).

Voltage Imaging with BeRST— Voltage imaging was performed on an upright AxioExaminer Z-1 (Zeiss) or an inverted Zeiss AxioObserver Z-1 (Zeiss), both equipped with a Spectra-X light engine LED light (Lumencor), and controlled with Slidebook (3i). Images were acquired using a W-Plan-Apo/1.0 NA 20x water immersion objective (Zeiss) or a Plan-Apochromat/0.8 NA 20x air objective (Zeiss). Images (2048 px × 400 px, pixel size: 0.325 µm × 0.325 µm) were collected continuously on an OrcaFlash4.0 sCMOS camera (sCMOS; Hamamatsu) at a sampling rate of 0.5 kHz, with 4×4 binning, and a 631 nm LED (13 mW/mm^2, SpectraX) with a 631/28 nm excitation bandpass. Emission was collected after passing through a quadruple bandpass dichroic (432/38 nm, 509/22 nm, 586/40 nm, 654 nm LP and quadruple bandpass emission filter (430/32 nm, 508/14 nm, 586/30 nm, 708/98 nm).

Electrophysiology— For electrophysiological experiments, pipettes were pulled from borosilicate glass (Sutter Instruments, BF150-86-10), with a resistance of 5–8 MΩ, and were filled with an internal solution; (in mM) 115 potassium gluconate, 10 BAPTA tetrapotassium salt, 10 HEPES, 5 NaCl, 10 KCl, 2 ATP disodium salt, 0.3 GTP trisodium salt (pH 7.25, 275 mOsm). Recordings were obtained with an Axopatch 200B amplifier (Molecular Devices) at room temperature. The signals were digitized with a Digidata 1440A, sampled at 50 kHz and recorded with pCLAMP 10 software (Molecular Devices) on a PC. Fast capacitance was compensated in the on-cell configuration. For all electrophysiology experiments, recordings were only pursued if the series resistance in voltage clamp was less than 30 MΩ. For whole-cell, current clamp recordings in hippocampal neurons, following membrane rupture, resting membrane potential was assessed and recorded at $I = 0$ and monitored during the data acquisition.

S.3 CellMincer preprocessing and global feature extraction details

Before a voltage imaging movie $X(t, x, y)$ is received as input to a CellMincer model, our pipeline applies several preprocessing steps to: (1) approximately isolate the background fluorescence; (2) normalize the dynamic range of background-subtracted data prior to denoising; (3) precompute a number of global movie statistics for conditioning the local denoiser. In this section, we detail the data preprocessing and global feature extraction stages of the CellMincer pipeline.

Data preprocessing and trend isolation— Background fluorescence is a dynamic imaging artifact both highly individual to its source dataset and magnitudes larger than the true fluorescence signal, so removing it aids the network in identifying neuron action potentials. To model this background activity separately for each (x, y) pixel, we temporally interpolate each pixel's trace with a low-order polynomial (with a default value of $n_{\text{poly}} = 3$) to obtain the following decomposition:

$$X(t, x, y) = X_{\text{trend}}(t, x, y) + \sigma_{\text{detrended}}\, X_{\text{detrended}}(t, x, y). \tag{5}$$

By design, X_{trend} approximately captures the smooth temporal trend and DC bias offset in the recording, whereas $X_{\text{detrended}}$ represents the normalized residual fluorescence signal. When specified by the user, we obtain the smooth trend fit only from the resting periods (typically the beginning and the end segments of a recording segment). When such resting periods are not included in the recording, we regress over the entire recording and use a lower order polynomial ($n_{\text{poly}} = 1$) to avoid overfitting to the neural activity. We note that normalizing the detrended component by its standard deviation over all pixels and time points, $\sigma_{\text{detrended}}$, allows CellMincer to train over multiple datasets and data sources (see Eq. 5). After denoising such a detrended dataset, CellMincer reports both the output without modification and reconstituted with the original scaling and trend (Eq. 5).

Precomputing global features— After the preprocessing stage of the CellMincer pipeline, we precompute the global features as follows. First, we further decompose $X_{\text{detrended}}(t, x, y)$ into slow and fast components:

$$X_{\text{detrended}}(t, x, y) = X^{\text{slow}}_{\text{detrended}}(t, x, y) + X^{\text{fast}}_{\text{detrended}}(t, x, y), \tag{6}$$

where $X^{\text{slow}}_{\text{detrended}}(t, x, y)$ is the moving average of $X_{\text{detrended}}(t, x, y)$ over a short window. For a 500 Hz recording, we calculate the moving average over 10 frames, corresponding to 20 ms. The goal here is to separate the neural activity into fast transients (e.g. spikes) and slower features (e.g. subthreshold activity). We calculate the same set of global features from the two components independently. We define the general spatially-resolved temporal auto-correlation function as such:

$$\rho[X; \Delta t, \Delta x, \Delta y](x, y) = \frac{1}{T} \sum_{t=1}^{T} X(t, x, y)\, X(t - \Delta t, x - \Delta x, y - \Delta y). \tag{7}$$

The first three global features are: (1) the square root of $\rho[X^{\text{slow}}_{\text{detrended}}; 0, 0, 0](x, y)$, i.e. the pixelwise slow temporal variability; (2) the square root of $\rho[X^{\text{fast}}_{\text{detrended}}; 0, 0, 0](x, y)$, i.e. the pixelwise fast temporal variability; (3) the temporal mean of $X^{\text{slow}}_{\text{detrended}}(t, x, y)$, i.e. the mean pixelwise slow activity. In addition to these, we include 17 other normalized and spatially-resolved auto-correlation functions as follows:

$$\frac{\rho[X^{\text{slow}}_{\text{detrended}}; \Delta t, \Delta x, \Delta y](x, y)}{\rho[X^{\text{slow}}_{\text{detrended}}; 0, 0, 0](x, y) + \epsilon}, \qquad \frac{\rho[X^{\text{fast}}_{\text{detrended}}; \Delta t, \Delta x, \Delta y](x, y)}{\rho[X^{\text{fast}}_{\text{detrended}}; 0, 0, 0](x, y) + \epsilon}, \tag{8}$$

for $\Delta x, \Delta y \in \{-1, 0, 1\}$, $\Delta t \in \{0, 1\}$, and excluding $\Delta x = \Delta y = \Delta t = 0$. Put together, these amount to 37 feature maps. Next, we spatially downsample both $X^{\text{slow}}_{\text{detrended}}$ and $X^{\text{fast}}_{\text{detrended}}$ by a factor of two, such that each image-space pixel corresponds to the average signal over a two native-resolution pixels. We calculate the same set of 37 features maps, and upsample the obtained feature

map by a factor of two back to the original resolution. The rationale is to bring more distant spatially-lagged auto-correlations into a feature map in the native resolution. In principle, this procedure can be repeated multiple times to capture further dilated and averaged auto-correlations. We stop the procedure at the second level, obtaining $F = 2 \times 37 = 74$ spatial feature maps in total which we collect and concatenate into a $F \times H \times W$ tensor. Conveniently, the F channels of this tensor encode a standardized set of spatiotemporal auto-correlations at different lengthscales, which can be used by the model to infer covarying groups of pixels without having access to the full movie.

S.4 CellMincer neural network design, training schedule, and implementation details

The neural network architecture of CellMincer consists of a U-Net which produces deep embeddings of individual frames and a temporal post-processor which reduces a sequence of frame embeddings into a single denoised frame (see Fig. 1b).

Our U-Net design allows for the augmentation of the input frame with our precomputed global features. This augmentation can occur either by concatenating the two before passing it through the U-Net or by repeating this concatenation at each step of the U-Net's contracting path, iteratively downsampling the global features in tandem with the frame embedding (see Fig. 1). We find that this *repeat* global feature augmentation reinforces the network bias toward using the global features, improving downstream performance. In addition, our U-Net implementation is not limited to a specific input dimension (as often required by conventional implementations), as demonstrated by our protocol of training on small imaging crops while using whole frames at evaluation time. This allows the model to train on imaging corpora with mixed dimensions and generalize to arbitrarily sized inputs without needing to dissect the input into uniformly-sized patches. Without padding each convolution layer, our U-Net produces image contraction, so we apply reflection padding to the input to achieve our desired output dimensions.

The temporal post-processor takes as input a short window of frame embeddings from the U-Net, convolves the time dimension, and collapses the feature dimension, producing a single output for each pixel. In this manner, no further spatial entanglement is introduced, so we do not include global features at this computation step (see Fig. 1d).

Through optimization trials, we determined an Adam optimizer with standard momentum parameters ($\beta_1 = 0.9, \beta_2 = 0.999$) was most effective for training CellMincer. We applied a cosine-annealed learning rate with linear warmup [29], parameterized at $\eta_{\max} = 10^{-4}$. To increase the diversity of imaging used to train our model, we configured our training samples to consist of small 62×62 crops padded with 30 pixels on each side, striking a balance between the minibatch diversity and the training signal that comes from each entry in the minibatch. With this configuration, we were able to maximize GPU utilization by training on minibatches of 20 samples per GPU (reduced to 10 samples for our largest model variant). We found that 50,000 training iterations generally led to sufficient model convergence when using a training set of limited size (1-5 recordings). More investigation is needed to determine whether a longer training period is needed to make full use of a larger training set.

In the course of CellMincer's development, we explored a series of variations on its architecture and training schemes, some of which are reported in detail in Sec. S.6. Of those omitted from our results, we considered single U-Net architectures that combined spatial and temporal processing, either by using a 3D U-Net to model time or by concatenating the frame sequence within the feature dimension. While our *time as features* model was computationally more efficient, we could not reach the expressivity and performance afforded by our current two-stage design. We also experimented with the choice of learning rate schedule, opting for an empirically validated cosine annealing with 10% linear warmup [29]. This schedule mitigates early instability while the model performance is highly variant while improving convergence near the end of training. In addition, we briefly explored the use of stochastic weight averaging [30], and discovered that it had a paradoxically negative impact on performance. Our hypothesis is that the contours of our model's loss landscape are highly nonconvex so that an averaging of local optima removes us from

the optimal parameter manifold.

We implemented CellMincer as a CLI tool in the PyTorch Lightning framework, which offers ease of scalability with multi-GPU training. To offer a sense of training costs and runtime, a CellMincer model trained on a single dataset using one NVIDIA Tesla T4 GPU for a standard 50,000 iterations would take 12-16 hours to finish, while a larger operation using 26 datasets and 4 GPUs may take 6-7 days. In practice, we found that training a fresh model to denoise a new dataset is not necessary. For instance, in our end-to-end hypothesis testing experiment (Sec. S.10), we simply used a model previously trained on a different voltage imaging corpus (recorded under similar conditions) and did not involve any of the datasets in the experiment. Denoising a typical dataset with CellMincer, by contrast, takes no more than a few minutes on any GPU setup.

S.5 Simulating realistic voltage imaging datasets using Optosynth

In order to optimize the architecture and hyperparameters of a data denoising technique and study the impact of various design choices on the bottom line denoising performance, one needs noiseless or high-SNR *ground truth* data. To generate such ground truth recordings and their noisy realizations, we developed a physics-based simulation companion software called *Optosynth* in which we aimed to carefully model salient aspects of the phenomenology of voltage imaging. We briefly describe the key steps involved in Optosynth simulations as follows (see Supplementary Fig. S 1 for a graphical overview).

Data procurement and preprocessing— We used Allen SDK [31] to access Allen Brain Atlas data, and procured 485 neurons from mouse primary visual cortex (VISp) from the Allen cell types database with paired morphology and EP data (Patch-seq) [32–34]. We minimally preprocess reconstructed morphology data as follows. We scale the image-space pixel from the original 0.1144 µm/pixel by a factor of 10 to 1.144 µm/pixel, representative of the typical magnification of voltage imaging experiments. We project the 3D morphology into a 2D binary mask. Allen morphology reconstructions only provide the location and radius of the soma. We use this information to generate a synthetic soma shape circumscribed by a random Fourier curve with 3 frequency components and a wiggle amplitude of no more than 20% of the soma radius. We also minimally preprocess the EP data by truncating the sweeps to 2 sec in duration, amounting to 0.5 sec of resting recording, 1s of electrical stimulation, followed by another 0.5 sec of resting recording. For each neuron, we keep track of the stimulation current amplitude (pA) and the membrane potential (mV) time series.

Setting up the simulation— The Optosynth simulation configuration is used to generate a manifest for the experiment. The key user input is the number of desired stimulation segments for the full experiment, and is given to the tool as a list of $[I_{\min}, I_{\max}]$ stimulation amplitude intervals. Based on this specification, we filter the pool of available Patch-seq neurons to the subset of neurons that have at least one sweep within each stimulation interval. Relaxing the stimulation current to fall within an interval (in contrast with matching a specific value) allows more flexibility in choosing neurons for simulation, because not all Patch-seq neurons have received the same set of stimulation amplitudes.

Simulating the neuron fluorescence density— For each neuron, we precompute a spike decay map $\alpha(\mathbf{r})$, a delay map $\tau(\mathbf{r})$, and a fluorescent reporter spatial density map $\rho(\mathbf{r})$ as follows:

$$\alpha(\mathbf{r}) = \exp\left(-\frac{||\mathbf{r}-\mathbf{r}_{\text{soma}}||}{\ell_{\text{decay}}}\right), \tag{9a}$$

$$\tau(\mathbf{r}) = -\frac{||\mathbf{r}-\mathbf{r}_{\text{soma}}||}{v_{\text{prop}}}, \tag{9b}$$

$$\rho^{(i)}(\mathbf{r}) \sim f_{\text{clamp}}\left[\text{GP}(0, \ell_{\text{reporter}}); \rho^{(\text{min})}_{\text{reporter}}, \rho^{(\text{max})}_{\text{reporter}}\right], \qquad (i = 1, \ldots, N_{\text{neurons}}). \tag{9c}$$

Here, $\mathbf{r}$ is a 2D position vector, $\mathbf{r}_{\text{soma}}$ refers to the soma location of the neuron of interest, ℓ_{decay} is a specified signal decay lengthscale, v_{prop} is the action potential propagation velocity [3, 35].

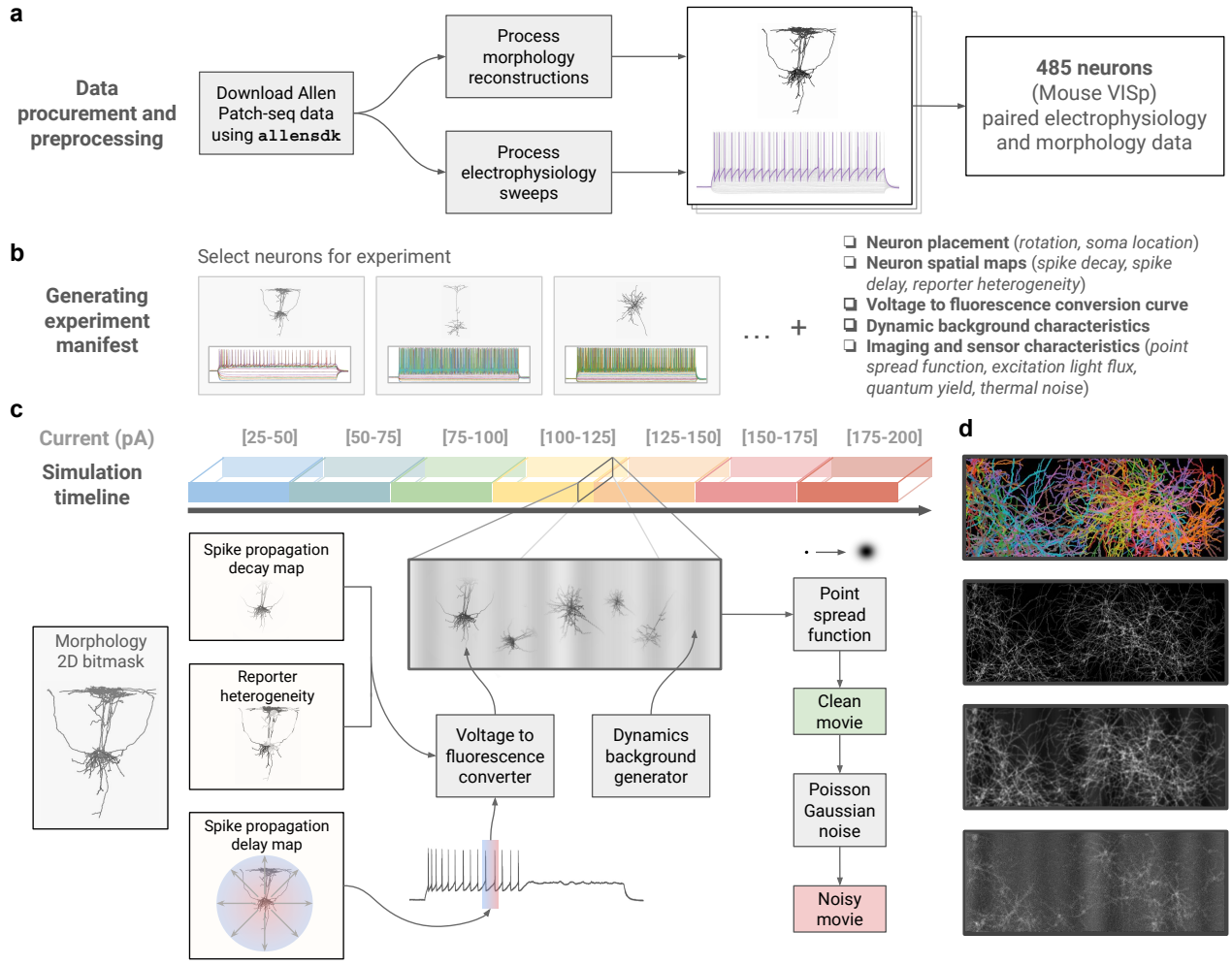

Figure S 1: Overview of Optosynth voltage imaging simulation environment. (a) Single-neuron paired morphology and EP data downloaded from Allen Brain Atlas; (b) Generating experiment manifest, including selecting neurons and sweeps for each segment of the experiment, and random sampling and precomputing various simulation accessories; (c) Schematic illustration of the generation process of a movie frame: depending on the position of a pixel on a given neuron, an action potential wavefront propagation delay is read off from the precomputed delay map and is used to select the appropriately delayed timepoint on the EP voltage trace. The voltage value is converted to fluorescence amplitude in combination with the precomputed reporter heterogeneity and spike decay maps. This process is repeated within an efficient vectorized algorithm for all pixels for a given neuron and for all other neurons in the simulation. A background frame is generated and added to the total fluorescence amplitude map generated by the neurons. A point spread function (Gaussian blur) is applied to the total fluorescence map to generate a *clean movie* frame. The application of pixelwise Poisson-Gaussian noise with specified parameters (thermal noise strength, quantum yield) generates a *noisy movie* frame. This process is repeated for each frame in the stimulation segment and for all other segments in the simulation. (d) From top to bottom: (1) neuronal masks juxtaposed in different colors; (2) a simulated frame before the addition of background and PSF; (3) the same frame after the addition of background and PSF; (4) the same frame after the addition of Poisson-Gaussian noise.

We sample $\rho(\mathbf{r})$ from a Gaussian process with zero mean and an isotropic Gaussian kernel with lengthscale ℓ_{reporter}. The f_{clamp} function linearly rescales the dynamic range of the randomly generated density map to the specified lower and upper limit, $\rho^{(\text{min})}_{\text{reporter}}$ and $\rho^{(\text{max})}_{\text{reporter}}$, respectively. We randomly place each neuron on the imaging field, together with a random rotation, displacement, and scale factors. To this end, we take the initial binary mask of the i'th neuron and apply a general isotropic affine transformation on the mask to obtain the final mask on the imaging field, which we refer to as $\mathsf{M}^{(i)}(\mathbf{r}) \in [0, 1]$. The precomputed spatial maps and the geometric placements

on the imaging field are held constant for all stimulation segments in the experiment, as well as potentially other simulated trials involving the same neurons.

The spatial fluorescence density emanating from a pixel at position $\mathbf{r}$ at time t emanating from the i'th neuron, $F^{(i)}(\mathbf{r}, t)$, is calculated as follows:

$$V_{\mathrm{LP}}^{(i)}(t) = \mathcal{F}_{\mathrm{LP}}\left[V^{(i)}(t); f_{\mathrm{LP}}\right], \tag{10}$$

$$V^{(i)}(\mathbf{r}, t) = \alpha(\mathbf{r})\, V_{\mathrm{LP}}^{(i)}(t - \tau(\mathbf{r})) + (1 - \alpha(\mathbf{r}))\, V_0, \tag{11}$$

$$F^{(i)}(\mathbf{r}, t) = \mathsf{M}^{(i)}(\mathbf{r})\, \rho^{(i)}(\mathbf{r})\, \frac{F_\infty}{1 + \exp\left[-\beta\left(V^{(i)}(\mathbf{r}, t) - V_{\mathrm{rise}}\right)\right]}, \tag{12}$$

First, we apply a low-pass Fourier filter $\mathcal{F}_{\mathrm{LP}}$ with cutoff frequency f_{LP} on the patch-clamp EP recording of i'th neuron, $V^{(i)}(t)$, to obtain $V_{\mathrm{LP}}^{(i)}(t)$. This provision is to simulate the lagged response of one's choice of fluorescent reporter to voltage transients. To obtain the effective spatial membrane potential of the neuron, $V^{(i)}(\mathbf{r}, t)$, we linearly admix the resting potential $V_0 \approx -70\,\mathrm{mV}$ and the appropriately time-delayed and low-passed EP recording, see Eq. 11. Finally, the fluorescence density is obtain by converting the spatial membrane potential to fluorescence and multiplying by the precomputed reporter density and neuron mask, see Eq. (12). We use a sigmoid function to convert voltage to fluorescence, and set the sigmoid slope $\beta \ll 1$ to work mostly in the linear regime. The two parameters F_∞ and V_{rise} are automatically determined by the user's specification of two points on the voltage conversion curve, (V_1, F_1) and (V_2, F_2).

Simulating the background fluorescence density— Typical voltage imaging recording is often accompanied by a source background fluorescence, including the static autofluorescence from the cell culture, and a dynamic slowly-varying component due to convection currents associated with sample heating by the stimulation laser. We simulate the background noise by first sampling a set of patterns from a zero-mean Gaussian process with anisotropic Gaussian kernels:

$$B_{\mathrm{static}}(\mathbf{r}) \sim \mathrm{GP}(0; \ell_{\mathrm{static}}^{(x)}, \ell_{\mathrm{static}}^{(y)}), \tag{13a}$$

$$B_{\mathrm{dynamic}}^{(k)}(\mathbf{r}) \sim \mathrm{GP}(0; \ell_{\mathrm{dynamic}}^{(x)}, \ell_{\mathrm{dynamic}}^{(y)}), \qquad (k = 1, \ldots, K), \tag{13b}$$

where $\ell_{\mathrm{static}}^{(x,y)}$ and $\ell_{\mathrm{dynamic}}^{(x,y)}$ denote the spatial variation lengthscales of static and dynamic background patterns along the x and y axes, and K is the number of dynamic patterns. For each dynamic pattern, we additionally sample a slowly-varying temporal amplitude:

$$a^{(k)}(t) \sim \mathrm{GP}(0; f_{\mathrm{dynamic}}^{-1}), \tag{14}$$

where f_{dynamic} is the dynamic background variation frequency. We finally compose the full background as follows:

$$B(\mathbf{r}, t) = f_{\mathrm{clamp}}\left[B_{\mathrm{static}}(\mathbf{r}); \rho_{\mathrm{static}}^{(\mathrm{min})}, \rho_{\mathrm{static}}^{(\mathrm{max})}\right] + f_{\mathrm{clamp}}\left[\frac{1}{K}\sum_{k=1}^{K} a^{(k)}(t)\, B_{\mathrm{dynamic}}^{(k)}(\mathbf{r}); \rho_{\mathrm{dynamic}}^{(\mathrm{min})}, \rho_{\mathrm{dynamic}}^{(\mathrm{max})}\right]. \tag{15}$$

Generating the clean and noisy recordings— The total fluorescence density is the sum of fluorescence density associated with the neurons and the background:

$$F(\mathbf{r}, t) = \sum_{i=1}^{N_{\mathrm{neurons}}} F^{(i)}(\mathbf{r}, t) + B(\mathbf{r}, t). \tag{16}$$

We interpret the fluorescence density $F(\mathbf{r}, t)$ as the density of excitable fluorophores per voxel, such that the reporter fluorescence laser converts $F(\mathbf{r}, t)$ to an emitted photon count per imaging interval via a conversation factor Q. To account for point spreading due to imaging optics, we further convolve $F(\mathbf{r}, t)$ with a normalized Gaussian point spread function (PSF) with lengthscale ℓ_{PSF} to obtain the clean fluorescence density:

$$\lambda(\mathbf{r}, t) = \int \mathrm{d}\mathbf{r}'\, \mathrm{PSF}(\ell_{\mathrm{PSF}})(\mathbf{r}')\, F(\mathbf{r} - \mathbf{r}', t). \tag{17}$$

The clean recording is given as:

$$X_{\text{clean}}(\mathbf{r}, t) = R\, Q\, \lambda(\mathbf{r}, t) + \Delta_{\text{dc}}, \tag{18}$$

where R is the sensor voltage gain per absorbed photon, Q is the photon emission rate per excited fluorophore, and Δ_{dc} is a dc offset (characteristic of typical sensor readouts). The noisy recordings are obtained by applying a Poisson-Gaussian noise and quantizing to integer counts:

$$X_{\text{noisy}}(\mathbf{r}, t) \sim \left\lfloor R\,\text{Poisson}\left[Q\,\lambda_{\text{clean}}(\mathbf{r}, t)\right] + \text{Gaussian}\left(0, \sigma_{\text{sensor}}\right) + \Delta_{\text{dc}} \right\rfloor. \tag{19}$$

Note that the SNR is implicitly controlled by the two parameters Q (number of photons per fluorophore) and σ_{sensor} (sensor noise).

To generate multiple simulated recordings from the same neurons, we simply repeat the simulation process multiple times while keeping the experiment manifest constant (including choice of Patch-seq neurons, as well as their geometries and precomputed spatial maps).

S.6 Optimizing CellMincer network architecture and training schedule using Optosynth-simulated datasets

We performed an extensive hyperparameter exploration to study the role of various modeling choices and identify optimal settings. Simulated data produced by Optosynth was an ideal setting for optimization experiments and ablation studies because of the access to ground truth imaging and its capacity to simulate a range of imaging conditions. The ground truth imaging enabled the direct computation of performance metrics for model evaluation, while the simulation versatility allowed us to test the model on data *imaged* at various SNR without the overhead associated with real-world data collection. Using Optosynth, we generated five datasets under the same neuron density and SNR conditions, three of which were allocated to the training set while the other two were set aside for testing.

Because a complete grid search was not feasible, we chose a baseline configuration that produced a viable model and varied the parameters around this baseline one at a time. Some choices, such as the use of an Adam optimizer, the learning rate scheme, and the number of training iterations, were decided in the baseline model and do not appear in our optimization experiments. We determined variations on each of the other hyperparameters of interest to apply to our baseline model and trained a CellMincer model with each of the resulting configuration variants. These models were subsequently used to denoise both the training and testing datasets, and we computed the distribution of PSNR gain over each frame within the active stimulation periods. The distributions of these PSNR gains and all of our model variants are summarized in Supplementary Fig. S 2.

Our key finding was that the inclusion of global features produced a dominating gain in PSNR. On both training data and unseen test data, CellMincer models incorporating global features exhibited an additional 5 dB gain in PSNR over the baseline model (i.e. a striking 3-fold increase in SNR), in contrast with other architectural variations that yielded 0-1.5 dB over the baseline (Supplementary Fig. S 2a-c). Indeed, we found that the baseline model performance (which did not include the global features) was highly sensitive to changes in the temporal window length (i.e. the denoising context size), with longer windows significantly improving performance. It is evident that both modifications to this baseline model address the limited temporal context it is provided, of which global features is by far the more effective and computationally efficient option. To determine the architectural settings that best synergize with the inclusion of global features, we performed this optimization experiment again with a parallel set of CellMincer variations, all of which included *repeat* global features (Supplementary Fig. S 2d-f). While most of the model performance variation remained consistent with our results in the first iteration, we observed that the performance gain induced by larger temporal contexts plateaued at a window length of 13 frames, comparable in size to our baseline window of 9 frames. This further supports our hypothesis that very long temporal windows can be exploited by a model without global features as a compensatory measure, and by explicitly including global features, we removed the need for

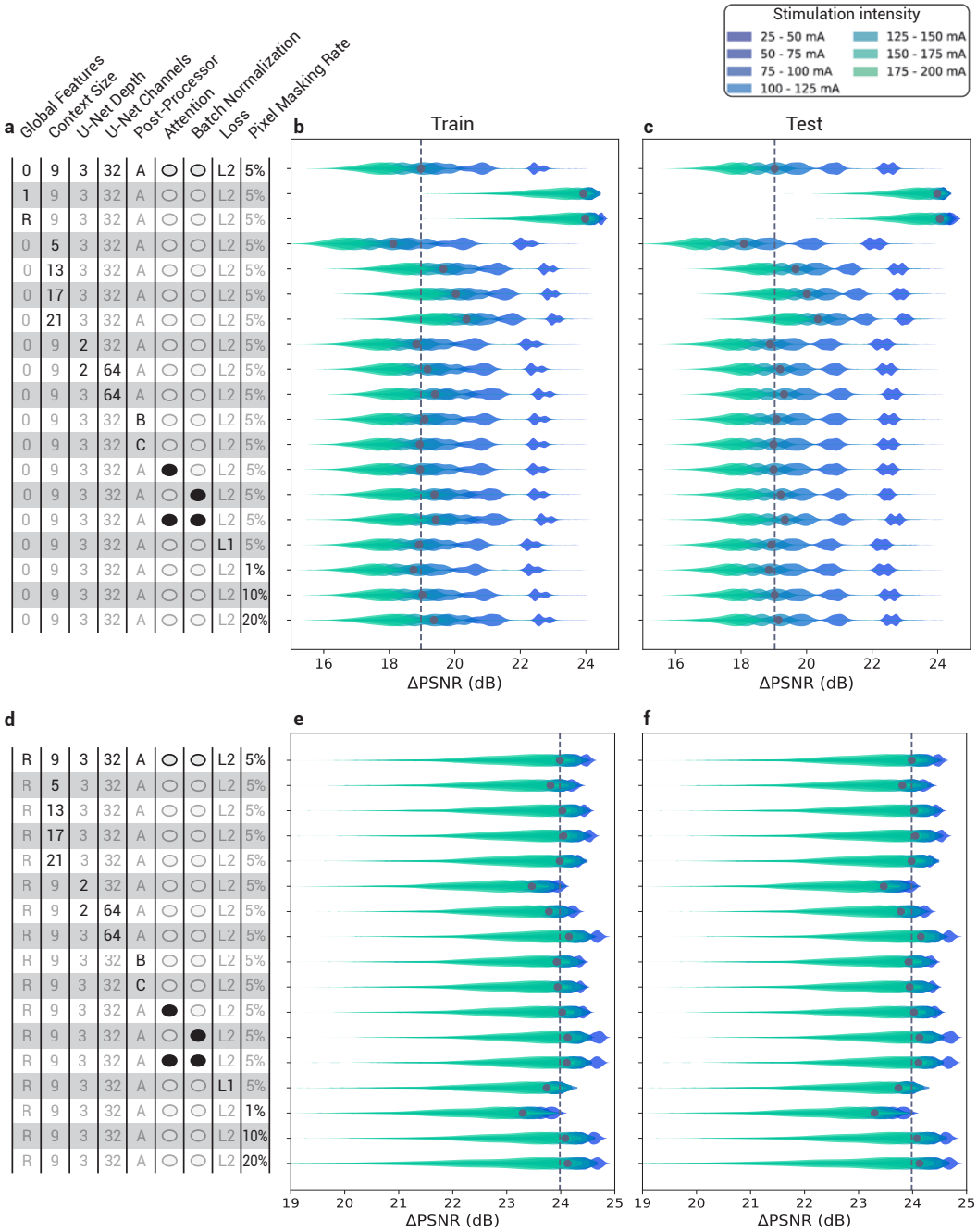

Figure S 2: CellMincer hyperparameter settings and their resulting models' performance on Optosynth data. Each model was evaluated on both its training data (b, e) and unseen test data (c, f). (a)-(c) Initial series of experiments using no global features as a baseline. (d)-(f) Followup iteration of experiments using repeated global features as a baseline. The global features setting determines whether the precomputed global features is not used (0), used to augment the U-Net input only at the beginning (1), or used to augment repeatedly at every contracting path step (R). The included temporal post-processor variants refers to the architecture of the ultimate multilayer perceptron component: $C \to C/2 \to C$ (A), $C \to C \to C/2 \to 1$ (B), and $C \to C \to C \to 1$ (C). The architectural variants are ordered in increasing complexity. The pixel masking setting refers to the Bernoulli parameter used to decide whether each pixel is masked, a sampling process repeated for each training iteration. The second set of experiments adjusts the original baseline model to use a conditional U-Net with repeated global features.

long contexts. The result is a network architecture that requires comparatively fewer input frames for denoising, allowing it to denoise datasets in a fraction of the computational time needed by comparable deep learning architectures like DeepCAD.

S.7 Metrics for evaluating denoising performance on Optosynth simulated datasets

Peak signal-to-noise ratio (PSNR), measured in decibels, is a metric of similarity between a clean image and its noisy realization and is defined as:

$$\mathrm{PSNR}\left[X_{\mathrm{clean}}, X_{\mathrm{noisy}}\right] = 10\,\log_{10}\left(\frac{I_{\max}}{\frac{1}{WH}\sum_{x=1}^{W}\sum_{y=1}^{H}\left[X_{\mathrm{clean}}(x,y) - X_{\mathrm{noisy}}(x,y)\right]^2}\right), \tag{20}$$

where $I_{\max}$ is the maximum possible value for the signal intensity. Many of our results are reported in PSNR *gain*, in which we use the PSNR between the raw noisy data and the clean data as a baseline. Reporting the results in terms of PSNR gain is more meaningful and comparable across different settings as it does not depend on $I_{\max}$. Structural similarity index measure (SSIM) is another metric describing the perceived quality of noisy digital images. We chose to reports on results in terms of PSNR given the flexibility it affords (e.g. the ability to be restricted over arbitrary spatial regions such as foreground or background pixels), as well as its wide adoption in the fluorescence imaging community.

S.8 Procuring fluorescence intensity traces and aligning to joint electrophysiology data

Our analyses of single-neuron traces are contingent on identifying representative ROIs over which the fluorescent signal is averaged, which is also a common practice in the field. To determine neuronal ROIs, we select a small set of seed pixels that belong to a neuron's soma and calculate cross-correlations between these pixels and every other pixel in the raw recording. On the resulting cross-correlation map, we choose a manually tuned threshold that captures the soma region and apply the watershed algorithm to add spatial continuity. The resulting ROI is then used to compute traces for the raw recording as well as its denoised counterparts.

Our ROI-extracted fluorescence trace can be interpreted as a noisy affine transformation of the neuron's electrophysiological activity. In the absence of a calibration dataset, we rely on an optimization-based approach to align the obtained fluorescence traces (in arbitrary units) to the EP data (in mV). We first remove the trend from the imaging trace by median filtering with a moving one-second window (which is short enough to correct for pipette movement but long enough to retain the actual signal). We also removed high-frequency jitter from the patch-clamp EP by applying a Savitzky-Golay cubic filter with a 51-point window. These post-filter signals are more easily imposed over one another following several linear transformations. By matching corresponding peaks in both signals, the intensity trace can be transferred onto the EP timescale. This allows us to evaluate intervals between peaks in absolute time and to downsample the EP signal with interpolation. We then find the affine transformation of our intensity trace that minimizes L2 error with the EP signal, producing an aligned voltage imaging trace. Our subsequent analyses center on evaluating the reconstruction quality of these aligned traces. Refer to Fig. 3b for the results of this alignment method.

S.9 Metrics for evaluating denoising performance on real voltage imaging with paired EP

Quantifying residual noise power using short-time Fourier transform— To compute the spectral noise power of a residual aligned fluorescence trace, we apply a short-time Fourier transform (STFT) parameterized by window length 64 and overlap 48. While voltage imaging,

which exhibits a relative downsampling factor of 100, cannot fully capture the underlying EP signal, the peaks of spiking events are particularly high-magnitude points of uncertainty, making them ill-suited for this analysis. To remove spiking events from consideration, we exclude time intervals in which the EP signal's total spectral power exceeds a chosen threshold. For the remaining time intervals, we convert the spectral powers to noise intensity (dB) and average them in each frequency bin to produce the average noise at that frequency. By repeating this process for each denoised signal and the raw signal, we can compute the reductions in noise intensity, yielding the results shown in Fig. 3d. A typical spectrogram is also shown here in Supplementary Fig. S 3, in which we clearly notice spiking events as and the attenuated high-frequency noise in the inter-spike interval in the CellMincer-denoised results. In contrast, the spectrogram of the raw data is visually akin to that of white noise.

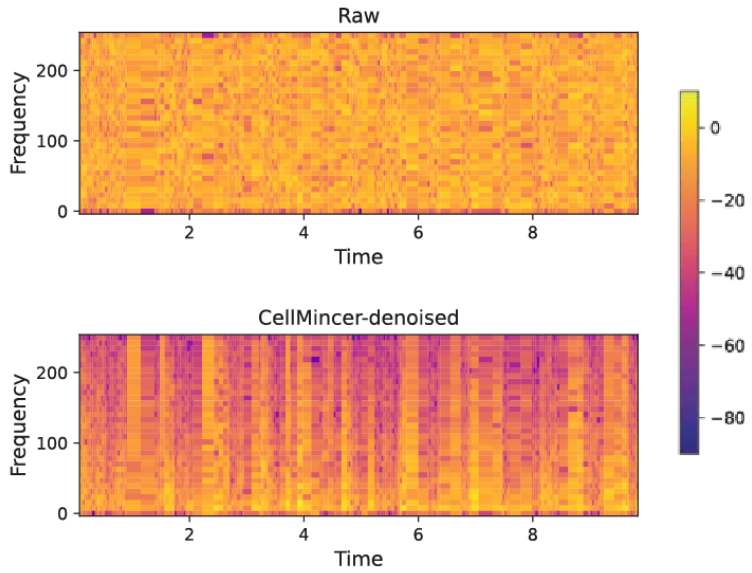

Figure S 3: A sample spectral power map (dB) of a residual voltage imaging recording before and after denoising with CellMincer.

Prominence-based peak calling— We can formulate the notion of quantifying signal reconstruction quality as a peak-calling problem. We consider peaks in the EP signal and classify them by their prominence, as most spikes exceed 20 mV in prominence while peaks in the subthreshold activity fall below 10 mV. A visualization of prominence as a signal peak feature is shown in Supplementary Fig. S 4. Let $X_{\mathrm{EP}}(t)$ and $X_{\mathrm{VI}}(t)$ refer to the filtered and aligned traces derived from the patch-clamp EP and the voltage imaging (VI) respectively, and let $S_p(X)$ be the set of peak time-points in signal $X(t)$ thresholded above a certain prominence p. We express our problem as an evaluation of similarity between $S_p(X_{\mathrm{EP}})$ and $S_{p'}(X_{\mathrm{VI}})$ for $p' \simeq p$. As our set elements comprise points along a continuous time interval, we need to adapt the notions of precision and recall to allow fuzzy matching of nearby peaks. More explicitly, we define:

$$\mathrm{Prec}(X_{\mathrm{EP}}, X_{\mathrm{VI}}; p, p') = \frac{|\{t_{\mathrm{VI}} \in S_{p'}(X_{\mathrm{VI}}) : \exists t_{\mathrm{EP}} \in S_p(X_{\mathrm{EP}})\,\mathrm{s.t.}\,|t_{\mathrm{VI}} - t_{\mathrm{EP}}| < \Delta t\}|}{|S_{p'}(X_{\mathrm{VI}})|}, \tag{21a}$$

$$\mathrm{Rec}(X_{\mathrm{EP}}, X_{\mathrm{VI}}; p, p') = \frac{|\{t_{\mathrm{EP}} \in S_p(X_{\mathrm{EP}}) : \exists t_{\mathrm{VI}} \in S_{p'}(X_{\mathrm{VI}})\,\mathrm{s.t.}\,|t_{\mathrm{VI}} - t_{\mathrm{EP}}| < \Delta t\}|}{|S_p(X_{\mathrm{EP}})|}. \tag{21b}$$

We calculate the F_1-score as usual:

$$F_1(X_{\mathrm{EP}}, X_{\mathrm{VI}}; p, p') = 2 \times \frac{\mathrm{Prec}(\ldots) \times \mathrm{Rec}(\ldots)}{\mathrm{Prec}(\ldots) + \mathrm{Rec}(\ldots)}. \tag{22}$$

As our definitions suggest, we determine peaks to be correctly called when a corresponding peak occurs within a Δt separation. In our evaluations, we set $\Delta t = 2\,\mathrm{ms}$, corresponding to a one-

frame discrepancy in 500 Hz voltage imaging. To introduce tolerance around particular choices of prominence thresholding, we further define:

$$F_1^*(X_{\mathrm{EP}}, X_{\mathrm{VI}}; p) = \max_{p' \in [p-\Delta p, p+\Delta p]} F_1(X_{\mathrm{EP}}, X_{\mathrm{VI}}; p, p'), \tag{23}$$

and we set $\Delta p = 0.5\,\mathrm{mV}$. We average this quantity over bins of prominences p to produce Fig. 3e, which is the ultimate result of this analysis.

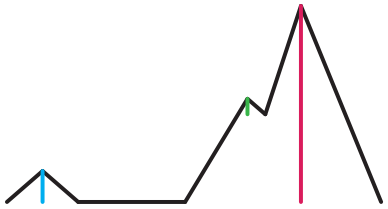

Figure S 4: A visualization of the prominence attribute in a simplified signal with three peaks. A moderate prominence threshold would exclude transient peaks produced by noise fluctuations (green), and a larger threshold would exclude subthreshold activity (blue), leaving only the true action potentials (red).

S.10 Method for segmenting and spike-counting voltage imaging datasets

We extract single-neuron segments from each raw movie and its denoised counterpart using a PCA/ICA-based approach [22]. Similar to CellMincer's data preprocessing step, our first step is to enhance pixel-pixel correlations by detrending the movie, leaving only the signal component stemming from neural activity. Through experimentation with various signal filters, we found that a rolling circle filter parameterized with width 3.2 frames (6.4 ms) and height 16 (fluorescence a.u.), followed by a threshold filter to collapse values between ±15 (fluorescence a.u.) to zero, was effective at isolating spiking events in the denoised data. We could not achieve similar success using various combinations of filters on the raw data, so we left those datasets unchanged. Using movie pixels as samples and time frames as features, PCA reduces the dimension of each pixel from $N_{\mathrm{time}} \sim 10^4$ to $N_{\mathrm{PCA}} = 200$. These components correspond to groups of covarying pixels but not necessarily to individual neurons. For this reason, ICA is used to subsequently unmix the principal components into an independent component set containing our neuron segment candidates. To detect neurons of varying signal strengths, we use a range of parameterizations $N_{\mathrm{ICA}} \in \{10, 20, 50, 100\}$ to produce the components from which our neuron segments are selected through manual filtering, including a careful study of ICA spatial components together with the associated temporal trace around spike-like events.

Following the extraction of these segments, we compute inner products between a segment mask and each movie frame and apply median filtering with window length of 51 frames to generate its corresponding trace. From these traces, we identify spiking events through manual filtering. This process is aided by a peak-finding algorithm to determine spike candidates and a segmented movie visualization to resolve ambiguous spiking sources. These spikes are tallied across stimulation intensities, and statistical separation between the spiking activity of unperturbed and chronically TTX-treated neurons is computed with a Wilcoxon rank sum test.

It should be noted that through the process of computationally and manually filtering segments, candidates with no discernible activity were excluded from analysis, even though they may have been real neurons which failed to express either the light-gated ion channel or fluorescent reporter. This distinction was considerably more apparent in the denoised movies, potentially causing some neurons to be discarded from the denoised count and included in the raw count despite the overall neuron count favoring the denoised data.

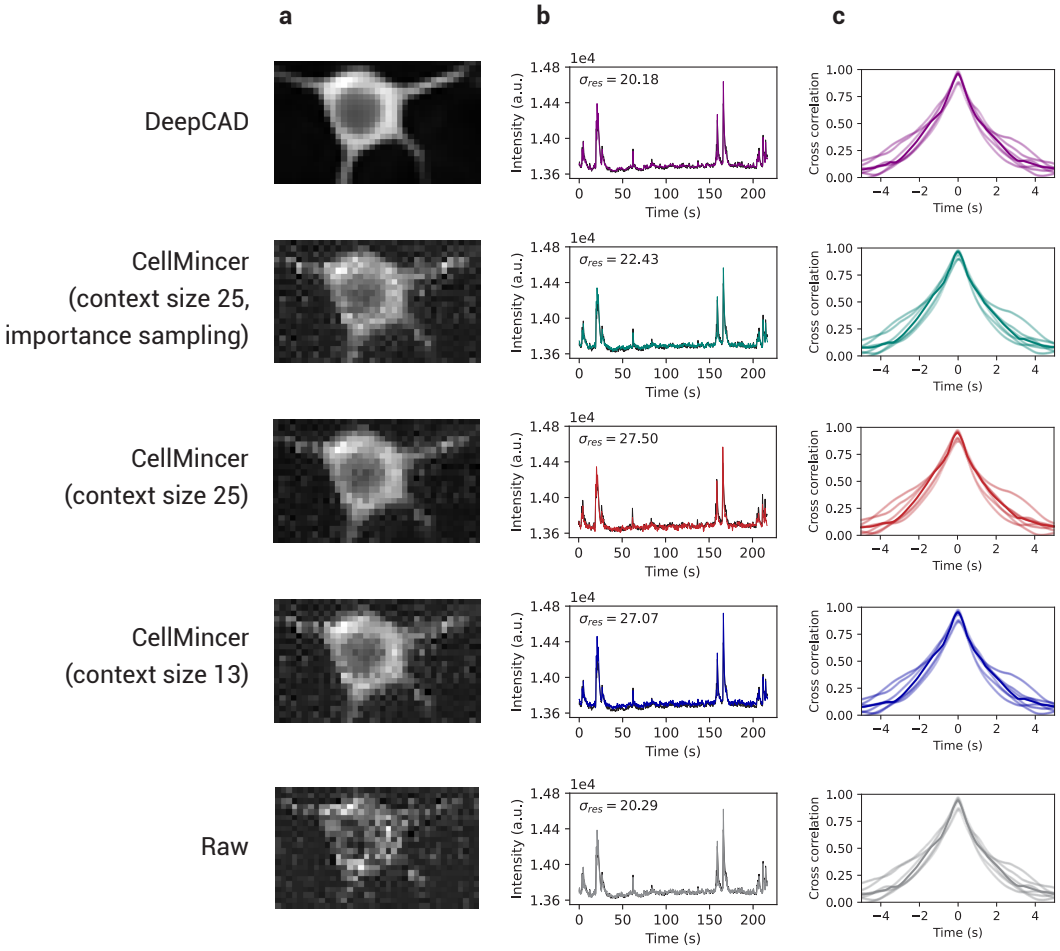

Figure S 5: Benchmarking experiments for several configurations of CellMincer and DeepCAD on calcium imaging. In addition to the CellMincer configuration tuned on Optosynth data, we tested two variants which incorporated a significantly longer context size of 25 frames (up from 13 frames), as well as an implementation of importance sampling to increase the frequency of data crops with discernible neural activity in training batches. (a) Sample denoised frame, zoomed in on a select neuron. (b) Averaged intensity trace over sample frame. The standard deviation of the residual signal with respect to the aligned high-SNR trace is labeled. Note that σ_{res} was computed over the residual signal following an alignment transformation. This transformation was optimized for the alignment to raw data, which explains the raw signal's surprisingly low residual variance. (c) Cross-correlations between the denoised trace and high-SNR trace.

S.11 Limitations of CellMincer vs. DeepCAD on calcium imaging datasets

To first demonstrate an application of the DeepCAD model in its native problem domain, we conducted a limited experiment to compare the efficacy of CellMincer and DeepCAD on calcium imaging. We trained CellMincer and DeepCAD on a training set of seven low-SNR calcium imaging datasets and compared the outputs with their corresponding high-SNR recordings. The results of this experiment, shown in Supplementary Fig. S 5, include multiple iterations of CellMincer configurations that incorporate adaptations to calcium imaging data domain. One adaptation significantly increases the temporal context length to more closely resemble the architecture of DeepCAD, while a second version applies the importance sampling approach used to more efficiently sample active regions of the recording (see Discussion). For the sample neuron presented in Supplementary Fig. S 5a, we found that DeepCAD produced a discernibly cleaner image of the neuron than both CellMincer and the raw data. We then averaged the intensity over each image in column a to produce corresponding single-neuron traces, which we plotted in column b. Overlaid onto these plots is the trace derived from the high-SNR recording aligned to its raw

low-SNR counterpart. Ignoring the residual spread of our raw alignment, which is already minimized by construction, we found that while DeepCAD's trace conforms comparatively well to the high-SNR recording, our importance sampling strategy significantly reduced the performance gap between CellMincer and DeepCAD on calcium imaging. This experiment warrants future research in a more versatile self-supervised training approach that allows denoising both slowly-varying (calcium imaging) and rapidly-varying (voltage imaging) functional imaging modalities.

Optosynth Parameter Name	Symbol	Type	Default Value
`OptosynthSpecs` parameters			
`width`	W	`int`	512
`height`	H	`int`	128
`sampling_rate`	N/A	`float`	500
`duration_per_segment`	N/A	`float`	2.0
`scale_factor`	N/A	`float`	1.0
`min_neuron_fluorescence_scale_factor`	N/A	`float`	0.1
`max_neuron_fluorescence_scale_factor`	N/A	`float`	1.0
`SyntheticNeuronSpecs` parameters			
`dendritic_backprop_velocity`	v_{prop}	`float`	1e+4
`dendritic_backprop_decay_lengthscale`	ℓ_{decay}	`float`	20.0
`min_reporter_density`	$\rho^{(\text{min})}_{\text{reporter}}$	`float`	1.0
`max_reporter_density`	$\rho^{(\text{max})}_{\text{reporter}}$	`float`	10.0
`reporter_density_var_lengthscale`	ℓ_{reporter}	`float`	2.0
`ephys_lowpass_freq`	f_{LP}	`float`	250
`BackgroundFluorescenceSpecs` parameters			
`dynamic_n_components`	K	`int`	20
`dynamic_x_lengthscale`	$\ell^{(x)}_{\text{dynamic}}$	`float`	10
`dynamic_y_lengthscale`	$\ell^{(y)}_{\text{dynamic}}$	`float`	100
`dynamic_temporal_frequency`	f_{dynamic}	`float`	100
`dynamic_min_total_fluorophore_density`	$\rho^{(\text{min})}_{\text{dynamic}}$	`float`	0.0
`dynamic_max_total_fluorophore_density`	$\rho^{(\text{max}}_{\text{dynamic}}$	`float`	0.5
`static_x_lengthscale`	$\ell^{(x)}_{\text{static}}$	`float`	5.0
`static_y_lengthscale`	$\ell^{(x)}_{\text{static}}$	`float`	5.0
`static_min_total_fluorophore_density`	$\rho^{(\text{min})}_{\text{static}}$	`float`	0.0
`static_max_total_fluorophore_density`	$\rho^{(\text{max}}_{\text{static}}$	`float`	0.1
`VoltageToFluorescenceSpecs` parameters			
`beta`	β	`float`	0.01
`v1`	V_1	`float`	-100
`f1`	F_1	`float`	0.4
`v2`	V_2	`float`	50
`f2`	F_2	`float`	1.0
`CameraSpecs` parameters			
`dc_offset`	Δ_{dc}	`float`	500
`gaussian_noise_std`	σ_{sensor}	`float`	10
`psf_lengthscale`	ℓ_{PSF}	`float`	0.25
`readout_per_photon`	R	`float`	2.2
`photon_per_fluorophore`	Q	`float`	50

Table S 1: Optosynth parameters and their default values.