# SpikeSSL: A Universal Spike Inference Framework with Dynamics-Informed State-Space Layers

**Chenghao Yue**[1,*] **Siming Xing**[1,*] **Shuran Liu**[1] **Angran Li**[2] **Yuanlong Zhang**[1,†]
[1]School of Life Sciences, Tsinghua University, Beijing, China
[2]School of Future Information Innovation, Fudan University, Shanghai, China
{yuech26,xsm25,liusr24}@mails.tsinghua.edu.cn
arli25@m.fudan.edu.cn; yuanlongzhang@tsinghua.edu.cn
[*]Equal contribution. [†]Corresponding author.

## Abstract

Two-photon calcium imaging is a standard tool for recording large neural populations in vivo, yet inferring spikes accurately across the growing diversity of calcium indicators remains an open problem. Existing supervised methods achieve reasonable in-domain accuracy but generalize poorly to unseen indicators, because different indicators induce distinct fluorescence kinetics and signal statistics while existing architectures remain relatively simple generic temporal regressors without dynamics-matched inductive bias. We propose SpikeSSL, a universal spike inference framework whose temporal backbone is a bank of bidirectional IIR state-space layers broadly motivated by calcium dynamics. A multi-modal conditioning encoder maps indicator identity, sampling rate, and trace-level signal statistics into a global conditioning vector that modulates the backbone via Adaptive Layer Normalization, while a heteroscedastic variance head provides calibrated per-frame uncertainty. On a benchmark with five fixed evaluation splits built from 33 public ground-truth datasets, SpikeSSL achieves state-of-the-art performance in both in-domain and zero-shot leave-one-indicator-out settings. We also develop a biophysical simulation pipeline capable of generating paired fluorescence-spike traces with systematically varied kinetic parameters, spike statistics, response nonlinearities, baseline drift, and noise. Using this pipeline, we synthesize approximately 11,000 simulated traces. Augmenting training with these data effectively closes the cross-indicator domain gap and improves zero-shot generalization. Code is publicly available at `https://github.com/detimage123/SpikeSSL`.

## 1 Introduction

Two-photon calcium imaging is a widely used tool for recording neural population activity in vivo [4, 29]. In this modality the observed fluorescence trace is a noisy, nonlinear, and temporally filtered proxy whose relation to the underlying spike train depends heavily on the calcium indicator, optical noise, and cell-type biophysics [22, 16]. Spike deconvolution, the task of recovering the latent spike train from the observed calcium fluorescence, is therefore an essential computational step. Existing spike inference methods broadly fall into two paradigms. Model-based approaches such as OASIS [5] and MLSpike [3] invert explicit generative models of calcium dynamics, but they require indicator-specific kinetic parameters whose optimal values may diverge from measured quantities. Deep learning-based methods such as CASCADE [16] and ENS$^2$ [28] learn the inverse mapping from paired fluorescence-electrophysiology recordings and achieve stronger in-domain performance.

Nevertheless, universal spike inference remains difficult, as models trained on observed indicators can degrade markedly on unseen ones. The central difficulty is that modern calcium indicators induce substantially different fluorescence dynamics. The underlying forward model relating calcium

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

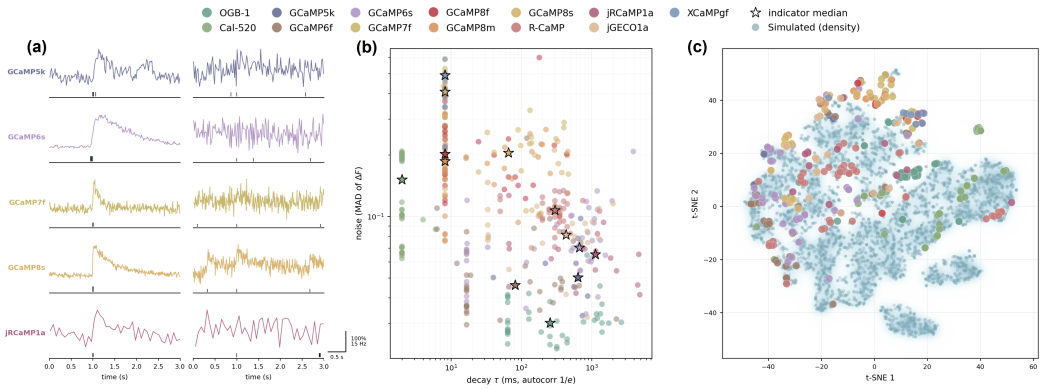



Figure 1: Indicator diversity and coverage gaps. **(a)** Representative fluorescence and spike snippets show markedly different response shapes across calcium indicators and among related GCaMP variants. **(b)** Indicator medians separate in kinetic and signal-statistic space, while simulated traces populate sparse regions between real datasets. **(c)** A t-SNE projection of per-window features shows partially overlapping but systematically displaced indicator domains.

dynamics to fluorescence, parameterized by indicator-specific rise and decay time constants among other factors [23, 3, 2, 26], varies dramatically across widely used sensors. Figure 1(a,b) illustrates both the differences between different indicators and the data variation within the same indicator. Figure 1(c) further shows that the domain gap between indicator families is large in per-window feature space. At the same time, existing CNN-based supervised models treat spike inference as a generic temporal regression task without inductive bias tailored to calcium dynamics. The forward model itself motivates a structurally matched design, as the first-order linear recurrence governing calcium decay shares the mathematical form of scan operations in deep state-space layers [8, 7, 21], with the state-decay coefficient serving as a direct analogue of the indicator time constant. If the network's temporal backbone is built from such recurrences, cross-indicator generalization becomes a more structured problem.

Informed by this observation, we propose SpikeSSL, a universal spike inference framework built on dynamics-informed state-space layers. The temporal backbone consists of bidirectional IIR scan layers spanning a basis of decay rates from fast transient detection to slow baseline tracking. A multi-modal conditioning encoder maps indicator identity, sampling rate, and trace-level signal statistics into a global conditioning vector, which modulates the backbone via Adaptive Layer Normalization as a built-in inductive bias rather than a post-hoc adaptation. In addition, a heteroscedastic variance head provides calibrated per-frame uncertainty and yields wider confidence intervals for out-of-distribution predictions. Although this architecture substantially improves universal inference, the scarcity of paired real recordings still limits cross-indicator generalization. We therefore design a biophysical simulation pipeline that samples spike trains from stochastic point processes and neuron-model dynamics, then passes them through a calcium forward model with varied kinetics, response nonlinearities, depletion, baseline drift, and observation noise. The resulting synthetic traces augment training and successfully bridge the domain gap between real indicator datasets, as illustrated in Figure 1(c).

Our main contributions are as follows.

- We propose SpikeSSL, a universal spike inference framework whose bidirectional IIR state-space layers are broadly motivated by calcium indicator decay dynamics. A multi-modal conditioning encoder maps indicator identity, sampling rate, and trace-level signal statistics into a global conditioning vector to modulate the backbone, together with a heteroscedastic variance head for calibrated uncertainty estimation.
- On a benchmark with five fixed evaluation splits built from 33 public ground-truth datasets, SpikeSSL achieves state-of-the-art performance in both in-domain and zero-shot LOIO settings, establishing its universal capability across diverse experimental conditions.
- We develop a biophysical simulation pipeline that generates synthetic calcium traces with

systematically varied kinetic parameters and firing statistics. We synthesize approximately 11,000 simulated traces covering a wide range of indicator kinetic regimes, effectively closing the cross-indicator gap in zero-shot transfer.

## 2 Related Work

### 2.1 Spike Inference from Calcium Imaging

Spike inference methods divide into model-based and data-driven approaches. Model-based approaches invert explicit calcium dynamics models, including linear deconvolution [25], Bayesian approaches [23, 24, 14], the biophysical MLSpike model [3], and the OASIS active-set method [5]. These methods require indicator-dependent kinetic parameters whose optimal values often diverge from biophysically measured quantities [18]. Data-driven methods, from early SVM approaches [19] to the deep architectures advanced through the Spikefinder benchmark [1], have converged on end-to-end learned representations. CASCADE [16] provides noise-optimized training across 27 datasets and expands coverage via calibrated noise augmentation, but does not vary kinetic parameters across indicators. ENS$^2$ [28] offers a compact 1D U-Net that achieves competitive accuracy with fewer parameters. Existing supervised methods rely on largely generic temporal backbones, and their performance degrades on unseen indicators. The Spikefinder benchmark [1] underscored this pattern, showing that improvements on one indicator rarely transferred to others, and coverage of the CASCADE database [16] remains uneven, with GCaMP8 variants and red-shifted sensors [26, 18] underrepresented relative to established GECIs, leaving models vulnerable to silent degradation on newly developed indicators.

### 2.2 Cross-Domain Generalization

Cross-domain generalization studies how models trained on multiple source domains transfer to unseen targets without target-domain labels [12, 9, 27]. In spike inference, domains are naturally defined by calcium indicators, and the challenge is accurate inference on held-out indicators with kinetics and statistics unseen during training. This problem is exacerbated by the limited diversity of paired ground-truth recordings. Although the database assembled by Rupprecht et al. [16] broadened coverage across indicators and experimental settings, and has since been extended with recordings for newly developed indicators and additional tissue types [18, 17], important gaps remain in indicator kinetic space. Recent work shows that CASCADE trained on one domain, such as cortex or GCaMP6-dominated data, transfers suboptimally to spinal cord recordings or GCaMP8 indicators [17, 18], highlighting the need for stronger cross-indicator generalization. Simulation-based augmentation offers a complementary strategy, where a calibrated forward model can generate paired fluorescence–spike traces spanning a systematic grid of kinetic parameters, extending the training distribution into regimes where real recordings are absent. This direction has received limited exploration in spike inference despite its potential to bridge coverage gaps without additional electrophysiology experiments.

## 3 Methodology

In this section, we present SpikeSSL, a universal spike inference framework with dynamics-informed state-space layers that integrates a bidirectional IIR backbone, multi-modal conditioning, and output heads for spike rate and uncertainty estimation. We then describe the biophysical simulation pipeline used for cross-indicator data augmentation.

### 3.1 Overall Architecture

SpikeSSL is a dynamics-informed network for spike inference whose architecture draws broad motivation from the biophysics of calcium indicators. The model takes a univariate fluorescence trace $y \in \mathbb{R}^T$ together with indicator identity and sampling rate, and produces a per-frame spike rate $\hat{s} \in \mathbb{R}^T_{\geq 0}$ along with calibrated uncertainty $\hat{\sigma}^2 \in \mathbb{R}^T_{>0}$. The trace is first projected to model dimension $D$ via a $1 \times 1$ convolution, while a conditioning encoder maps indicator identity, sampling rate, and trace-level signal statistics through dedicated branches into a global conditioning vector $c \in \mathbb{R}^{64}$.

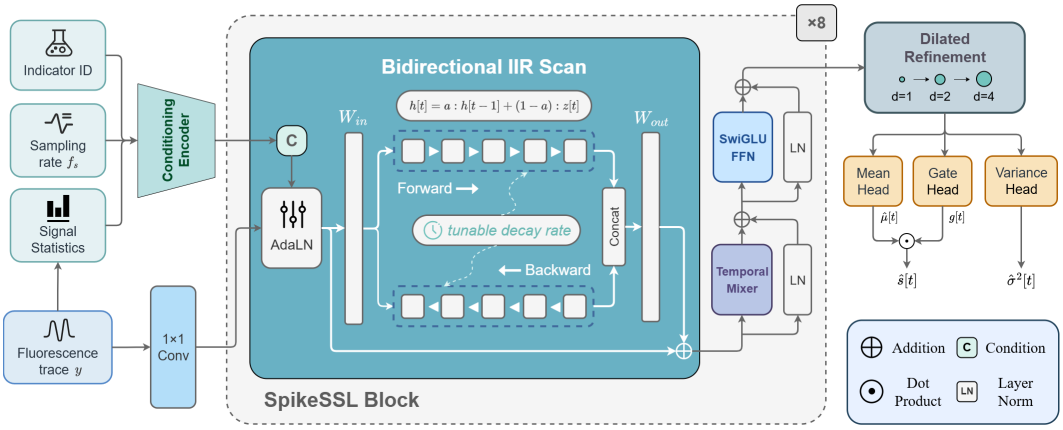



Figure 2: Architecture of SpikeSSL. A fluorescence trace is projected by a pointwise convolution and processed by a stack of SpikeSSL blocks. Each block contains a bidirectional IIR scan with learnable decay rates that capture the exponential decay dynamics of calcium indicators, modulated by AdaLN conditioning on indicator identity, sampling rate, and trace-level signal statistics, followed by a Temporal Mixer for local temporal mixing and a SwiGLU FFN for channel mixing. A dilated refinement module aggregates multi-scale temporal context. Three parallel output heads produce a spike rate estimate, a binary spiking gate, and a per-frame uncertainty estimate.

The projected trace then passes through $N$ stacked blocks, each containing a bidirectional IIR scan with AdaLN conditioning, a Temporal Mixer, and a SwiGLU feed-forward network (FFN). After the stack, a dilated refinement module of three residual convolutional layers with dilation schedule 1, 2, and 4 expands the receptive field from local to medium-range context, complementing the long-range multi-scale decay basis encoded by the IIR layers.

Three output heads operate on the refined features. The mean head predicts the non-negative spike rate $\hat{\mu}[t]$ via a convolutional projection followed by ReLU. A sigmoid gate head $g[t] = \sigma(w_g^\top h[t])$ classifies each frame as spiking or silent, is detached from the mean head's gradient during training, and is trained with a separate binary cross-entropy loss, preventing co-adaptation between the two heads. At inference, $\hat{s}[t] = \hat{\mu}[t]$ if $g[t] > 0.5$ and $\hat{s}[t] = 0$ otherwise, eliminating low-amplitude noise in silent periods. A variance head produces per-frame uncertainty estimates $\hat{\sigma}^2[t]$ as described in Section 3.4.

### 3.2 Dynamics-Informed IIR Backbone

The central design principle of SpikeSSL is to ground the network's temporal processing in the dynamics of calcium indicators. Simple first-order exponential decay is among the most common and general models of calcium dynamics, and although the underlying biophysics can be substantially more complex, this recurrence still provides a useful inductive bias. The IIR scan used in deep state-space layers [8, 7, 21, 10, 6, 15] follows a structurally similar recurrence,

$$h[t] = \alpha \cdot h[t-1] + (1 - \alpha) \cdot z[t], \tag{1}$$

where $\alpha = \sigma(\alpha_{\text{logit}})$ is a per-channel decay rate parameterized via the sigmoid function, and $z$ is the input projected from the trace embedding. The resemblance to exponential calcium decay motivates initializing and conditioning the decay rates to span the physiologically relevant range. By initializing the base logits as $\alpha_{\text{logit}} \sim \text{linspace}(0, 4)$, the model spans a multi-scale bank of decay rates from fast time constants of a few frames to slow trends unfolding over hundreds of frames. This range deliberately covers the full spread of indicator kinetics observed in the benchmark, providing a structured inductive bias that reflects the underlying biophysics rather than leaving the network to discover relevant temporal scales from data statistics alone.

To capture both causal context from past fluorescence and anti-causal context from future decay shape, SpikeSSL employs bidirectional IIR scans. Each layer computes a forward scan $h^{\text{fwd}}$ and a time-reversed backward scan $h^{\text{bwd}}$, concatenates them along the channel dimension, and projects

back to the model dimension,

$$h = W_{\text{out}} \, [\, h^{\text{fwd}} \parallel h^{\text{bwd}} \,] + b_{\text{out}}. \tag{2}$$

### 3.3 Multi-Modal Conditioning

Different indicators exhibit different kinetics, and SpikeSSL must adapt its decay rates accordingly. We achieve this through a multi-modal conditioning encoder, Adaptive Layer Normalization (AdaLN) [13], and per-sample decay rate modulation.

The conditioning vector $c \in \mathbb{R}^{d_c}$ fuses three sources of information,

$$c = \text{MLP}\big( \, e_{\text{ind}} \parallel f_{\text{fs}} \parallel f_{\text{stats}} \, \big), \tag{3}$$

where $e_{\text{ind}}$ is a learned embedding of the discrete indicator identity, $f_{\text{fs}} = \text{MLP}(\log f_s)$ encodes the sampling rate, and $f_{\text{stats}}$ encodes trace-level signal statistics. Each source is independently projected to a shared dimension before concatenation and fusion.

The signal statistics component extracts a 7-dimensional summary vector from the input trace,

$$f_{\text{stats}} = \big[ \, \hat{\sigma}_y, \; \hat{\gamma}_y, \; \hat{\kappa}_y, \; \rho_1, \; \rho_3, \; \rho_5, \; \rho_{10} \, \big], \tag{4}$$

comprising the standard deviation, skewness, excess kurtosis, and normalized autocorrelation at lags 1, 3, 5, and 10. These statistics are computed deterministically with no learnable parameters and serve as an implicit proxy for the indicator's SNR and kinetic properties. Slowly decaying indicators produce high autocorrelation, while noisy or fast indicators exhibit low autocorrelation and high kurtosis. This module enables SpikeSSL to adapt its temporal dynamics even when the indicator type is not specified.

The conditioning vector modulates the IIR sub-block via AdaLN with zero-initialized projection,

$$\text{AdaLN}(x, c) = \text{LN}(x) \odot (1 + \gamma(c)) + \beta(c), \tag{5}$$

where $(\gamma, \beta) = W_c \, c$ is a linear projection initialized to zero, so that at initialization the model reduces to standard Layer Normalization. AdaLN is applied only to the temporal IIR sub-block, while the subsequent convolution and feed-forward sub-blocks use standard normalization. This concentrates the indicator-conditioned modulation on the component that directly encodes decay dynamics.

In addition to AdaLN, the conditioning vector directly shifts the IIR decay logits,

$$\alpha_{\text{eff}} = \sigma\big(\alpha_{\text{logit}} + W_\alpha \, c\big), \tag{6}$$

where $W_\alpha$ is zero-initialized. This allows each sample's effective decay rates to be continuously modulated by its indicator identity and signal properties, enhancing the model's ability to adapt to diverse kinetic regimes.

### 3.4 Heteroscedastic Uncertainty Estimation

SpikeSSL produces a per-frame variance estimate $\hat{\sigma}^2[t]$ alongside the spike rate prediction. The variance head consists of a single convolutional projection from the shared backbone features plus a per-indicator learned bias,

$$\log \hat{\sigma}^2[t] = w_v^\top \, h[t] + b_{\text{ind}}, \tag{7}$$

where $b_{\text{ind}}$ is an indicator-specific scalar bias. Both the projection and bias are zero-initialized. A key emergent property is that the variance output serves as an automatic out-of-distribution detector, as in-domain predictions yield tight confidence intervals while zero-shot predictions on unseen indicators produce inflated variance.

### 3.5 Biophysical Simulation Data Generation

To populate underrepresented regimes while preserving realistic action-potential-driven fluorescence, we augment training with simulated calcium traces generated by an enhanced MLSpike-based forward model [3]. The simulator complements real paired recordings by expanding kinetic coverage rather than replacing empirical data, and samples parameters across ranges covering observed

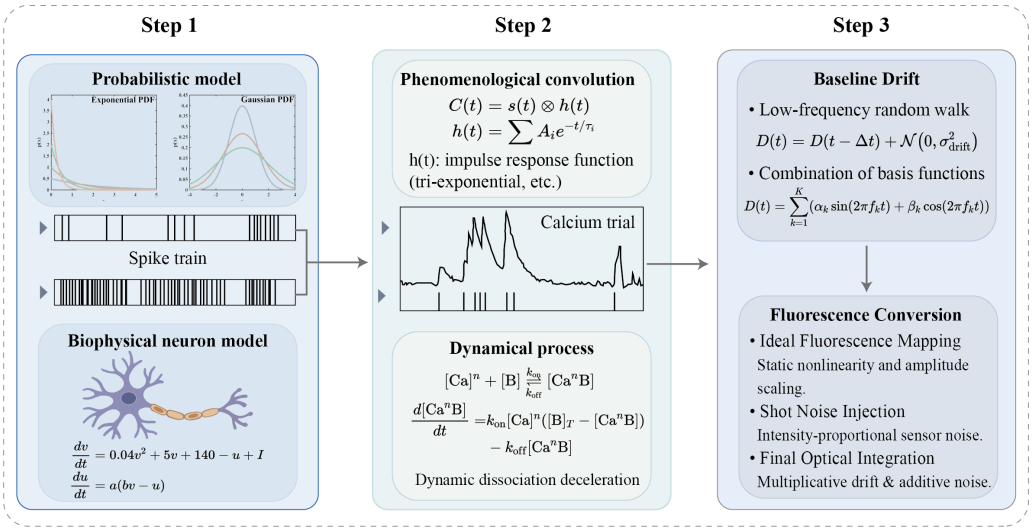



Figure 3: Hierarchical Workflow of the Calcium Data Simulator.

indicators and plausible unseen sensors. As illustrated in Fig. 3, the simulation pipeline contains three components: spike train synthesis, calcium transient generation, and observation noise modeling. For each trial, we sample a recording duration and target firing rate, then generate spike trains with two complementary strategies. One strategy draws spike counts and inter-spike intervals from stochastic point-process models, allowing sparse, irregular, quasi-periodic, and burst-like firing. The other integrates the Izhikevich neuron model [11] to produce spikes from continuous membrane potential dynamics. Together, these strategies provide a broad repertoire of neuronal activity regimes before the fluorescence forward model is applied.

The generated spikes are then passed through a modified calcium simulator. Compared with standard MLSpike, our simulator introduces stochastic amplitude jitter, a recovery variable that captures depletion and interval dependent response scaling, and indicator specific cooperative binding kinetics. Finally, we form observable fluorescence by adding slow baseline drift and Gaussian measurement noise to the modeled calcium response, producing traces with variability in baseline stability and signal quality. These mechanisms broaden the range of firing patterns, response amplitudes, noise conditions, and indicator dynamics beyond the real training set alone, helping bridge held out kinetic regimes during zero shot transfer. *More details are provided in the supplementary material.*

### 3.6 Training Objective

The total loss combines three components, namely a heteroscedastic reconstruction loss, structural penalty terms, and a gate detection loss,

$$\mathcal{L} = \mathcal{L}_{\text{recon}} + \mathcal{L}_{\text{struct}} + \lambda_g \mathcal{L}_{\text{gate}}. \tag{8}$$

The reconstruction loss follows the $\beta$-NLL formulation [20], which precision-weights the per-frame Huber loss by the predicted variance,

$$\mathcal{L}_{\text{recon}} = \frac{1}{\hat{\sigma}^2} \cdot \text{Huber}_\delta(\hat{s}, s) + \tfrac{1}{2} \log \hat{\sigma}^2. \tag{9}$$

High-confidence frames receive amplified gradients while noisy frames are down-weighted, and the entropy term $\frac{1}{2} \log \hat{\sigma}^2$ prevents the degenerate solution $\hat{\sigma}^2 \to \infty$.

The structural penalties impose three asymmetric constraints that are not attenuated by the variance estimate, namely an over-prediction penalty that penalizes $\hat{s}[t] > s[t]$ more heavily than under-prediction, a peak under-prediction penalty that pulls up missed high-amplitude spike events, and a mean bias penalty that prevents systematic offset between predicted and true mean spike rates,

$$\mathcal{L}_{\text{struct}} = \beta_{\text{over}} \cdot \mathcal{L}_{\text{over}} + \beta_{\text{peak}} \cdot \mathcal{L}_{\text{peak}} + \lambda_\mu \cdot \mathcal{L}_{\text{bias}}. \tag{10}$$

The gate loss trains the binary spike-or-silent classifier via cross-entropy, $\mathcal{L}_{\text{gate}} = \text{BCE}(g[t], \mathbf{1}[s[t] > \epsilon])$, with gradients detached from the mean head to prevent co-adaptation.

Table 1: In-domain resluts. All methods are trained and evaluated on the full ground-truth database with per-dataset train and test splits. **Bold** / Underline indicate the 1st / 2nd best results.

| Method | V1-OGB | | | V1-GCaMP6s | | | V1-GCaMP8 | | | Other-RCaMP | | | SC-GCaMP6s | | | Mean | Params |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Corr↑ | Err↓ | Bias↓ | Corr↑ | Err↓ | Bias↓ | Corr↑ | Err↓ | Bias↓ | Corr↑ | Err↓ | Bias↓ | Corr↑ | Err↓ | Bias↓ | Corr↑ | |
| OASIS | 0.298 | 1.157 | −0.481 | 0.332 | 0.973 | −0.883 | 0.493 | 0.878 | −0.659 | 0.504 | 1.054 | −0.508 | 0.365 | 0.975 | −0.964 | 0.398 | — |
| MLSpike | 0.363 | 4.379 | 3.244 | 0.224 | 1.017 | −0.928 | 0.558 | 0.935 | −0.887 | 0.354 | 1.369 | −0.761 | 0.307 | 1.005 | −0.919 | 0.361 | — |
| CASCADE [16] | 0.613 | 1.117 | −0.250 | 0.361 | 1.718 | **0.520** | 0.825 | 3.246 | 3.108 | 0.675 | 1.060 | −0.388 | 0.149 | 1.005 | −0.639 | 0.524 | 34K |
| ENS$^2$ [28] | 0.641 | 1.032 | −0.368 | 0.358 | 2.265 | 1.354 | 0.775 | 1.827 | 1.596 | 0.695 | **0.884** | −0.509 | 0.256 | 1.116 | **−0.154** | 0.545 | 145K |
| SpikeSSL | **0.743** | **0.810** | **−0.067** | **0.579** | **0.945** | −0.741 | **0.915** | **0.565** | **0.017** | **0.731** | 1.269 | **0.234** | **0.736** | **0.764** | −0.704 | **0.741** | 3.9M |

Table 2: Zero-shot LOIO results. Each column group holds out one of the five evaluation splits during training and evaluates on it without target-domain data. **Bold** / Underline indicate the 1st / 2nd best results.

| Method | V1-OGB | | | V1-GCaMP6s | | | V1-GCaMP8 | | | Other-RCaMP | | | SC-GCaMP6s | | | Mean |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Corr↑ | Err↓ | Bias↓ | Corr↑ | Err↓ | Bias↓ | Corr↑ | Err↓ | Bias↓ | Corr↑ | Err↓ | Bias↓ | Corr↑ | Err↓ | Bias↓ | Corr↑ |
| OASIS | 0.298 | 1.157 | −0.481 | 0.332 | 0.973 | −0.883 | 0.493 | 0.878 | −0.659 | 0.504 | 1.054 | −0.508 | 0.365 | 0.975 | −0.964 | 0.398 |
| MLSpike | 0.363 | 4.379 | 3.244 | 0.224 | 1.017 | −0.928 | 0.558 | 0.935 | −0.887 | 0.354 | 1.369 | −0.761 | 0.307 | 1.005 | −0.919 | 0.361 |
| CASCADE | 0.560 | 1.208 | **−0.144** | 0.304 | 2.571 | 1.614 | 0.757 | 3.723 | 3.602 | 0.551 | 1.086 | −0.300 | 0.126 | 1.083 | −0.442 | 0.460 |
| ENS$^2$ | 0.574 | **1.005** | −0.366 | 0.310 | 3.401 | 2.654 | 0.716 | 2.038 | 1.715 | 0.635 | 0.966 | −0.454 | 0.178 | 1.092 | **−0.391** | 0.483 |
| SpikeSSL | **0.650** | 1.979 | 0.675 | **0.509** | **0.944** | **−0.677** | **0.853** | **0.513** | **−0.091** | **0.717** | 1.172 | 0.606 | **0.542** | **0.911** | −0.892 | **0.654** |

Table 3: LOIO results with simulation augmentation ("+ Sim."). Training is augmented with biophysically simulated calcium traces. **Bold** / Underline indicate the 1st / 2nd best results.

| Method | V1-OGB | | | V1-GCaMP6s | | | V1-GCaMP8 | | | Other-RCaMP | | | SC-GCaMP6s | | | Mean |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Corr↑ | Err↓ | Bias↓ | Corr↑ | Err↓ | Bias↓ | Corr↑ | Err↓ | Bias↓ | Corr↑ | Err↓ | Bias↓ | Corr↑ | Err↓ | Bias↓ | Corr↑ |
| CASCADE + Sim. | 0.552 | 2.418 | 1.423 | 0.439 | 1.522 | **0.691** | 0.716 | 1.408 | 1.018 | 0.642 | 1.421 | 0.614 | 0.405 | 3.206 | 2.084 | 0.551 |
| ENS$^2$ + Sim. | 0.486 | 1.634 | 0.592 | 0.447 | 2.146 | 1.418 | 0.796 | 1.882 | 1.487 | 0.588 | 1.764 | 0.756 | 0.461 | 2.291 | 1.054 | 0.556 |
| SpikeSSL + Sim. | **0.697** | **1.441** | **0.419** | **0.533** | **0.931** | −0.697 | **0.902** | **0.523** | **0.175** | **0.708** | **0.849** | **−0.505** | **0.678** | **0.771** | **−0.653** | **0.704** |

# 4 Experiments

## 4.1 Datasets

We evaluate on the ground-truth database assembled by Rupprecht et al. [16], comprising 33 datasets with 1,960 paired neuron recordings spanning 11 calcium indicators, two species (mouse and zebrafish), multiple brain regions (V1, CA3, olfactory bulb, telencephalon) and spinal cord, with both excitatory neurons and inhibitory interneurons (PV, SST, VIP). The corpus covers chemical dyes (OGB-1, Cal-520), legacy GECIs (GCaMP5k, 6f, 6s, 7f [2]), next-generation GCaMP8 variants (8f, 8m, 8s [26]), and red-shifted indicators (R-CaMP, jRCaMP1a, jGECO1a). Recent spinal cord recordings [17] introduce a major tissue-type shift from conventional cortical preparations. We additionally augment training with biophysically simulated calcium traces stored in multiple generated batches (Section 3.5 and Appendix A.4) used exclusively during training.

## 4.2 Benchmark and Metrics

We compare SpikeSSL against four baselines spanning the major paradigms, namely OASIS [5], MLSpike [3], CASCADE [16], and ENS$^2$ [28]. The underlying recordings are publicly available, and we evaluate them under five fixed evaluation splits. In the *in-domain* setting, all splits are included during training with per-dataset train and test subsets. In the LOIO setting, five representative experimental conditions, namely V1-OGB, V1-GCaMP6s, V1-GCaMP8, Other-RCaMP, and SC-GCaMP6s, are each held out in turn and evaluated without any target-domain training data. Across this benchmark, indicator decay time constants span from $\sim$50 ms for GCaMP8f to $\sim$600 ms for GCaMP6s, creating fundamentally different inverse problems for each condition.

We evaluate all methods using Pearson correlation between the predicted and ground-truth spike rate [22, 16, 28], computed per neuron and averaged across neurons within each dataset. We additionally report normalized error and mean bias (signed difference in temporal means) to assess systematic over- or under-prediction.

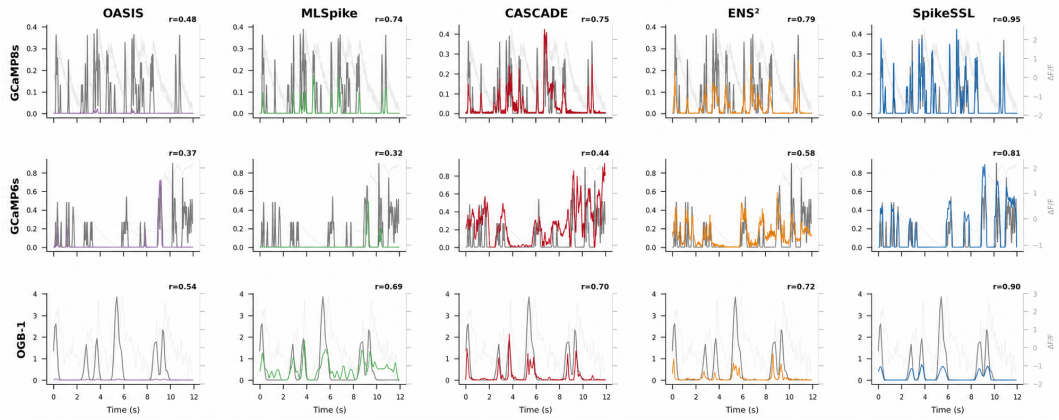



Figure 4: Qualitative LOIO comparison on GCaMP8s, GCaMP6s, and OGB-1 test neurons. Each column shows a different method, and the grey curve denotes the ground-truth smoothed spike rate.

### 4.3 Implementation Details

SpikeSSL uses 8 stacked blocks with model dimension $D$=192 and conditioning dimension $d_c$=64. We train with Adam using an initial learning rate of $10^{-3}$. The model converges within 20 epochs. For the loss hyperparameters, we set the over-prediction penalty $\beta_{\text{over}}$=3.0, peak under-prediction weight $\beta_{\text{under}}$=0.3, Huber threshold $\delta$=0.1, and mean bias weight $\lambda_\mu$=0.5. The gate loss weight $\lambda_g$=5.0 is set deliberately large to suppress false-positive spike predictions in silent periods.

### 4.4 In-Domain Results

Table 1 reports in-domain performance with all evaluation splits included during training. SpikeSSL achieves state-of-the-art performance, substantially outperforming all baselines. These results demonstrate that the dynamics-informed IIR backbone and multi-modal conditioning provide strong inductive bias even in the fully supervised setting where all indicator types are observed during training. The consistently lower error and bias values confirm that SpikeSSL produces well-calibrated spike rate estimates across the full diversity of indicators.

### 4.5 Zero-Shot LOIO Results

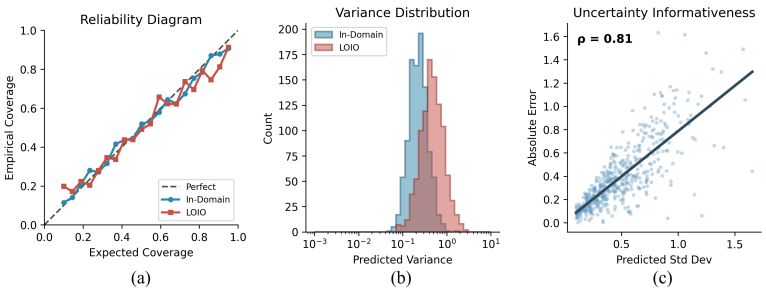
Under the LOIO protocol defined in Section 4.2, cross-indicator generalization remains difficult for all methods, but SpikeSSL achieves the highest mean correlation across all five held-out evaluation splits in Table 2. This result indicates that the IIR backbone and indicator-aware conditioning provide a meaningful structural advantage under domain shift. When training is augmented with biophysically simulated calcium traces, zero-shot performance improves further across all held-out splits in Table 3, with particularly large gains on conditions whose kinetics are sparsely covered by the real training set. Figure 4 shows that SpikeSSL preserves sharper spike structure than the classical baselines on representative unseen indicators.

### 4.6 Ablation Studies

We perform ablations under the LOIO protocol and report mean correlation across all five held-out splits in Table 4. V1 replaces AdaLN with a concatenation-based conditioning scheme, and V2 removes the signal-statistics branch from the conditioning vector. Together they confirm that both conditioning mechanisms contribute to cross-indicator adaptation. V3 replaces the bidirectional IIR module with a standard temporal convolutional block, supporting the value of the dynamics-informed multi-scale backbone. V4 removes the heteroscedastic variance head and $\beta$-NLL reweighting, while V5 removes the binary gate head and its cross-entropy loss, confirming that both heads improve predictive quality beyond their primary roles in uncertainty calibration and spike denoising.

Table 4: Ablation study under LOIO. Mean correlation is averaged over all five held-out evaluation splits.

| Method | Mean Corr (↑) |
|---|---|
| V1 (− AdaLN) | 0.601 |
| V2 (− Stats) | 0.623 |
| V3 (− Bi-IIR) | 0.594 |
| V4 (− Var.) | 0.608 |
| V5 (− Gate) | 0.621 |
| **SpikeSSL** | **0.654** |




(a) (b) (c)

Figure 5: Uncertainty calibration of SpikeSSL. Panel (a) shows reliability curves. Panel (b) compares variance distributions. Panel (c) plots std dev against absolute error.

Table 5: Param-matched control. Scaling fails to close the gap.

| Method | Mean Corr (↑) | #Params |
|---|---|---|
| CASCADE-L | 0.554 | 0.4M |
| CASCADE-H | 0.533 | 4.6M |
| ENS$^2$-L | 0.576 | 2.1M |
| ENS$^2$-H | 0.564 | 4.2M |
| SpikeSSL-L | 0.649 | 7.2M |
| **SpikeSSL** | **0.654** | 3.9M |

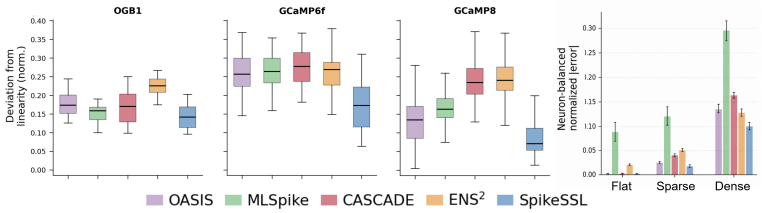



Figure 6: Linearity and activity-regime analysis. First three: deviation from linearity; Last: normalized error in flat, sparse, and dense windows.

### 4.7 Discussion

**Uncertainty Calibration** SpikeSSL's heteroscedastic variance head provides calibrated per-frame uncertainty in addition to point estimates. Figure 5 shows that confidence intervals remain close to empirical coverage, that predicted variance increases under LOIO shift, and that higher predicted standard deviation is associated with larger absolute error. Taken together, these observations suggest that the variance head captures meaningful epistemic uncertainty rather than acting as a generic smoothing term. At the same time, this analysis is limited to the five-split LOIO benchmark considered here, so whether calibration remains equally reliable under more extreme acquisition shifts, such as substantially different noise statistics or sampling regimes, remains an open question.

**Architectural Gains Beyond Parameter Scaling** Table 5 is intended to test whether SpikeSSL's gains can be explained simply by model size. We scale CASCADE and ENS$^2$ to larger parameter budgets and also evaluate a larger SpikeSSL variant. Increasing the baseline parameter counts indeed improves their performance, but they still do not reach SpikeSSL. In contrast, further scaling SpikeSSL does not yield additional gains, suggesting that its advantage comes from the dynamics-informed IIR backbone, indicator-aware conditioning, and uncertainty-aware objective rather than from simply stacking more parameters.

**Linearity and Response Scale of Spike Rate Estimation** A useful spike inference model should keep predicted firing rates close to a linear, correctly scaled transform of the ground truth. Figure 6(a) summarizes this property on OGB-1, GCaMP6f, and GCaMP8, where SpikeSSL shows the smallest deviation from linearity across distinct kinetic regimes. Figure 6(b) further stratifies local 1-s windows into flat, sparse, and dense activity regimes, showing lower error for SpikeSSL during both silent and active periods. These results are consistent with the multi-scale IIR backbone, whose fast and slow channels help preserve response scale across indicator kinetics.

## 5 Conclusion

We presented SpikeSSL, a universal spike inference framework that narrows the cross-indicator generalization gap through two complementary mechanisms. A bidirectional IIR backbone with a multi-scale decay basis spanning the full range of physiologically relevant indicator time constants, combined with multi-modal conditioning, provides a structured inductive bias for the inverse problem. Simulation-augmented training further bridges gaps in the kinetic parameter space of unseen indica-

tors. On five fixed evaluation splits built from public ground-truth datasets, SpikeSSL achieves state-of-the-art performance in both in-domain and zero-shot settings without target-domain ground truth. By providing a single model that generalizes across indicator types without retraining, SpikeSSL can reduce the experimental burden of ground-truth data collection and facilitate the adoption of new calcium indicators in neuroscience laboratories.

## References


[1] Philipp Berens, Jeremy Freeman, Thomas Deneux, Nikolay Chenkov, Thomas McColgan, Artur Speiser, Jakob H Macke, Srinivas C Turaga, Patrick Mineault, Peter Rupprecht, et al. Community-based benchmarking improves spike rate inference from two-photon calcium imaging data. *PLoS Computational Biology*, 14(5):e1006157, 2018.

[2] Tsai-Wen Chen, Trevor J Wardill, Yi Sun, Stefan R Pulver, Sabine L Renninger, Amy Baohan, Eric R Schreiter, Rex A Kerr, Michael B Orger, Vivek Jayaraman, et al. Ultrasensitive fluorescent proteins for imaging neuronal activity. *Nature*, 499(7458):295–300, 2013.

[3] Thomas Deneux, Attila Kaszas, Gergely Szalay, Gergely Katona, Tamás Lakner, Amiram Grinvald, Balázs Rózsa, and Ivo Vanzetta. Accurate spike estimation from noisy calcium signals for ultrafast three-dimensional imaging of large neuronal populations in vivo. *Nature Communications*, 7(1):12190, 2016.

[4] Winfried Denk, James H Strickler, and Watt W Webb. Two-photon laser scanning fluorescence microscopy. *Science*, 248(4951):73–76, 1990.

[5] Johannes Friedrich, Pengcheng Zhou, and Liam Paninski. Fast online deconvolution of calcium imaging data. *PLoS Computational Biology*, 13(3):e1005423, 2017.

[6] Daniel Y Fu, Tri Dao, Khaled K Saab, Armin W Thomas, Atri Rudra, and Christopher Ré. Hungry hungry hippos: Towards language modeling with state space models. In *International Conference on Learning Representations*, 2023.

[7] Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2024.

[8] Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured state spaces. In *International Conference on Learning Representations*, 2022.

[9] Ishaan Gulrajani and David Lopez-Paz. In search of lost domain generalization. In *International Conference on Learning Representations*, 2021.

[10] Ankit Gupta, Albert Gu, and Jonathan Berant. Diagonal state spaces are as effective as structured state spaces. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

[11] Eugene M. Izhikevich. Simple model of spiking neurons. *IEEE Transactions on Neural Networks*, 14(6): 1569–1572, 2003.

[12] Krikamol Muandet, David Balduzzi, and Bernhard Schölkopf. Domain generalization via invariant feature representation. In *International Conference on Machine Learning*, pages 10–18, 2013.

[13] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4195–4205, 2023.

[14] Eftychios A Pnevmatikakis, Josh Merel, Ari Pakman, and Liam Paninski. Bayesian spike inference from calcium imaging data. In *2013 Asilomar Conference on Signals, Systems and Computers*, pages 349–353. IEEE, 2013.

[15] Michael Poli, Stefano Massaroli, Eric Nguyen, Daniel Y Fu, Tri Dao, Stephen Baccus, Yoshua Bengio, Stefano Ermon, and Christopher Ré. Hyena hierarchy: Towards larger convolutional language models. In *International Conference on Machine Learning*, 2023.

[16] Peter Rupprecht, Stefano Carta, Adrian Hoffmann, Mayumi Echizen, Antonin Blot, Alex C Kwan, Yang Dan, Sonja B Hofer, Kazuo Kitamura, Fritjof Helmchen, and Rainer W Friedrich. A database and deep learning toolbox for noise-optimized, generalized spike inference from calcium imaging. *Nature Neuroscience*, 24(9):1324–1337, 2021.

[17] Peter Rupprecht, Wei Fan, Steve J Sullivan, Fritjof Helmchen, and Andrei D Sdrulla. Spike rate inference from mouse spinal cord calcium imaging data. *Journal of Neuroscience*, 45(18), 2025.

[18] Peter Rupprecht, Márton Rózsa, Xusheng Fang, Karel Svoboda, and Fritjof Helmchen. Spike inference from calcium imaging data acquired with gcamp8 indicators. *bioRxiv*, pages 2025–03, 2025.

[19] Takuya Sasaki, Naoya Takahashi, Norio Matsuki, and Yuji Ikegaya. Fast and accurate detection of action potentials from somatic calcium fluctuations. *Journal of Neurophysiology*, 100(3):1668–1676, 2008.

[20] Maximilian Seitzer, Arash Tavakoli, Dimitrije Antic, and Georg Martius. On the pitfalls of heteroscedastic uncertainty estimation with probabilistic neural networks. In *International Conference on Learning Representations*, 2022.

[21] Jimmy T.H. Smith, Andrew Warrington, and Scott Linderman. Simplified state space layers for sequence modeling. In *International Conference on Learning Representations*, 2023.

[22] Lucas Theis, Philipp Berens, Emmanouil Froudarakis, Jacob Reimer, Miroslav Román Rosón, Tom Baden, Thomas Euler, Andreas S Tolias, and Matthias Bethge. Benchmarking spike rate inference in population calcium imaging. *Neuron*, 90(3):471–482, 2016.

[23] Joshua T Vogelstein, Brendon O Watson, Adam M Packer, Rafael Yuste, Bruno Jedynak, and Liam Paninski. Spike inference from calcium imaging using sequential monte carlo methods. *Biophysical Journal*, 97(2):636–655, 2009.

[24] Joshua T Vogelstein, Adam M Packer, Timothy A Machado, Tanya Sippy, Baktash Babadi, Rafael Yuste, and Liam Paninski. Fast nonnegative deconvolution for spike train inference from population calcium imaging. *Journal of Neurophysiology*, 104(6):3691–3704, 2010.

[25] Emre Yaksi and Rainer W Friedrich. Reconstruction of firing rate changes across neuronal populations by temporally deconvolved $Ca^{2+}$ imaging. *Nature Methods*, 3(5):377–383, 2006.

[26] Yan Zhang, Márton Rózsa, Yajie Liang, Daniel Bushey, Ziqiang Wei, Jihong Zheng, Daniel Reep, Gerard Joey Broussard, Arthur Tsang, Getahun Tsegaye, et al. Fast and sensitive gcamp calcium indicators for imaging neural populations. *Nature*, 615(7954):884–891, 2023.

[27] Kaiyang Zhou, Ziwei Liu, Yu Qiao, Tao Xiang, and Chen Change Loy. Domain generalization: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(11):4396–4429, 2022.

[28] Zhanhong Zhou, Hei Matthew Yip, Katya Tsimring, Mriganka Sur, Jacque Pak Kan Ip, and Chung Tin. Effective and efficient neural networks for spike inference from in vivo calcium imaging. *Cell Reports Methods*, 3(5), 2023.

[29] Warren R. Zipfel, Rebecca M. Williams, and Watt W. Webb. Nonlinear magic: Multiphoton microscopy in the biosciences. *Nature Biotechnology*, 21(11):1369–1377, 2003.

# A Supplementary Material

This appendix consolidates the technical details and extended validation in one continuous supplement. Appendix A.1 expands the conditioning encoder, dilated refinement module, Temporal Mixer, and SwiGLU FFN. Appendix A.2 documents the recording sets and fixed evaluation splits, Appendix A.4 specifies the biophysical simulation pipeline, and Appendix A.5 details benchmark construction. The remaining subsections provide additional analyses of conditioning, domain breadth, data scaling, ablations, adaptation regimes, learned dynamics, simulation, and robustness.

## A.1 Supplementary architecture modules

This subsection expands three modules that are named in the main text but not visualized in Figure 2, namely the conditioning encoder, the dilated refinement module, and the Temporal Mixer and SwiGLU FFN sub-blocks within each SpikeSSL block.

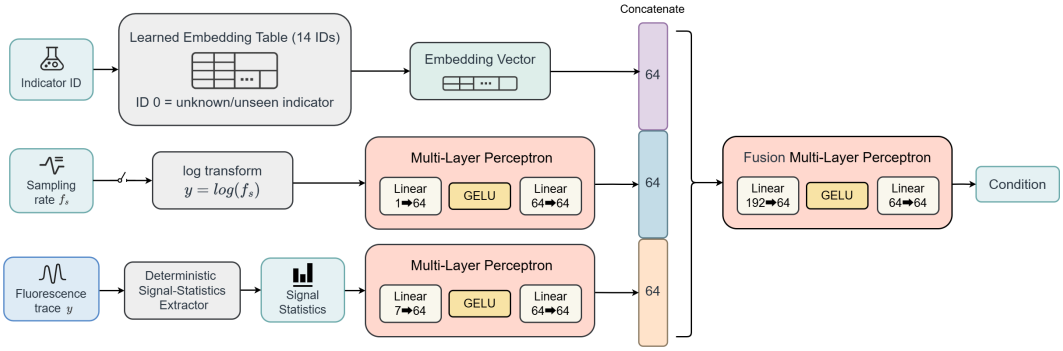



Figure 7: Conditioning encoder architecture. Three modality-specific branches independently encode their inputs before fusing them into a common representation. The indicator embedding branch maps discrete indicator identity to a 64-dimensional learned vector, where ID 0 is reserved for unknown or unseen indicators. The sampling-rate branch encodes $\log f_s$ through a two-layer MLP. The signal-statistics branch encodes the 7-dimensional descriptor comprising standard deviation, skewness, excess kurtosis, and autocorrelation at lags 1, 3, 5, and 10 through a two-layer MLP. The three 64-dimensional outputs are concatenated and fused by a $192 \to 64 \to 64$ MLP to produce the global conditioning vector $c \in \mathbb{R}^{64}$.

### A.1.1 Conditioning encoder internals

The conditioning encoder maps three modality-specific inputs through dedicated branches into the global conditioning vector $c \in \mathbb{R}^{64}$. The indicator branch uses a learned embedding table over 14 IDs, where ID 0 is reserved for unknown or unseen indicators and the remaining IDs cover named indicators including OGB-1, GCaMP6f, GCaMP6s, GCaMP8 variants, R-CaMP, jRCaMP1a, jGECO1a, and XCaMPgf, each projected to 64 dimensions by a linear layer. The sampling-rate branch maps $\log f_s$ through a two-layer MLP of hidden width 64, falling back to $\log 30$ when the frame rate is unavailable. The signal-statistics branch maps a 7-dimensional descriptor comprising standard deviation, skewness, excess kurtosis, and autocorrelation at lags 1, 3, 5, and 10 through a separate two-layer MLP of width 64. The three 64-dimensional branch outputs are concatenated and fused by a final $192 \to 64 \to 64$ MLP to produce $c$, keeping each modality isolated until the fusion stage so that indicator-specific metadata and data-driven statistics contribute through clearly separated pathways.

### A.1.2 Dilated refinement module

After the stack of conditioned SpikeSSL blocks and a final LayerNorm, the model applies a residual dilated convolution stack. With the default setting of three refinement layers, the dilation schedule is 1, 2, and 4, where each layer consists of a 1D convolution with kernel size 3 and matching dilation-dependent padding, followed by batch normalization, GELU, dropout, and a residual addition, expanding the temporal receptive field without downsampling. Whereas the bidirectional

IIR scan captures long-range indicator-specific decay, the refinement stack provides complementary short-to-medium-range local correction capacity at full temporal resolution, functioning as a lightweight post-backbone smoother and sharpener.

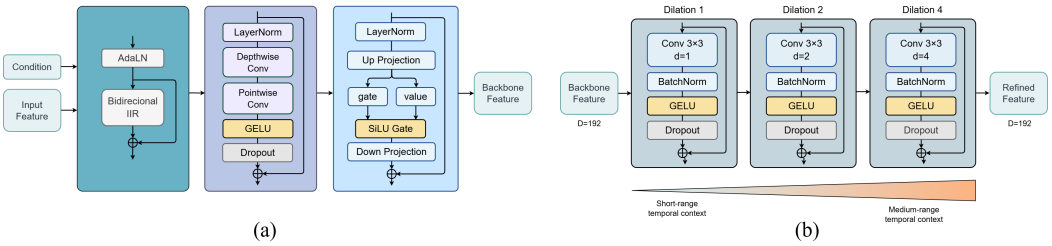



Figure 8: **(a)** SpikeSSL block sub-modules. The Temporal Mixer applies a depthwise 1D convolution (kernel width 7) followed by a pointwise convolution, GELU activation, and a residual connection, providing local temporal mixing without indicator-conditioned normalization. The SwiGLU FFN projects to twice the channel width, splits into gate and value tensors, computes $\mathrm{SiLU}(\mathrm{gate}) \odot \mathrm{value}$, and projects back to model dimension $D$, followed by a residual connection. **(b)** Dilated refinement module comprising three residual layers with dilation factors 1, 2, and 4 that successively expand the temporal receptive field at full resolution, each containing a kernel-3 1D convolution, batch normalization, GELU, and dropout.

### A.1.3 Temporal Mixer and SwiGLU FFN inside each block

Each SpikeSSL block contains three sequential sub-modules. The first is the AdaLN-conditioned bidirectional IIR layer described in the main text. The second and third are deliberately left unconditioned. The Temporal Mixer applies LayerNorm to the block state, followed by a depthwise 1D convolution of kernel width 7, a pointwise 1D convolution, GELU activation, dropout, and a residual addition, providing local temporal mixing without an additional indicator-conditioned normalization path.

The SwiGLU FFN applies LayerNorm and projects the state to twice the expanded width, splits the projection into gate and value tensors, computes $\mathrm{SiLU}(\mathrm{gate}) \odot \mathrm{value}$, projects back to model dimension $D$, and adds the result residually. With the default configuration $D{=}192$ and expansion factor 2, the intermediate width is 384. AdaLN modulation is confined to the IIR branch, where indicator-specific decay kinetics have the clearest physical interpretation, while the Temporal Mixer and SwiGLU FFN serve as standard local and channel mixers.

## A.2 Benchmark data composition and split definitions

Table 6 provides an overview of all benchmark datasets. Specifically, Table 7 categorizes the ground-truth recording sets by indicator family. Furthermore, Table 8 lists the five fixed evaluation splits.

The recording-level inventory makes the source of benchmark heterogeneity explicit: the corpus combines chemical dyes and genetically encoded indicators across species, anatomical regions, induction methods, and sampling rates rather than treating an indicator as a single acquisition setup. Frame rates span approximately 2.5–500 Hz, and the recordings cover cortical, hippocampal, olfactory, and spinal preparations collected by multiple laboratories. Consequently, the benchmark tests variation in acquisition and biological context in addition to nominal sensor identity. Recording-level separation also prevents overlapping windows from the same acquisition from being divided between training and evaluation.

Table 6: Overview of all ground truth recording sets.

| Set | Calcium indicator | Induction method | Animal model | Anatomical region | Frame rate (Hz) | $N$ | Source paper |
|---|---|---|---|---|---|---|---|
| 1 | OGB-1 | Acute injection | Mouse | V1 | 11.3 | 11 | Theis et al., 2016 |
| 2 | OGB-1 | Injection + tg(tdTomato-CaMKII$\alpha$) | Mouse | V1 | 15.6 | 16 | Kwan and Dan, 2012 |
| 3 | Cal-520 | Acute injection | Mouse | S1 | 500.0 | 8 | Tada et al., 2014 |
| 4 | OGB-1 | Acute injection | Zebrafish | pDp | 7.7 | 15 | Rupprecht et al., 2021 |
| 5 | Cal-520 | Acute injection | Zebrafish | pDp | 7.8 | 5 | Rupprecht et al., 2021 |
| 6 | GCaMP6f | tg(NeuroD) | Zebrafish | aDp | 30.0 | 8 | Rupprecht et al., 2021 |
| 7 | GCaMP6f | tg(NeuroD) | Zebrafish | dD | 30.0 | 10 | Rupprecht et al., 2021 |
| 8 | GCaMP6f | tg(NeuroD) | Zebrafish | OB | 30.0 | 9 | Rupprecht et al., 2021 |
| 9 | GCaMP6f | AAV | Mouse | V1 | 60.1 | 11 | Chen et al., 2013 |
| 10 | GCaMP6f | tg(Emx1) | Mouse | V1 | 160.1 | 23 | Huang et al., 2019 |
| 11 | GCaMP6f | tg(Cux2) | Mouse | V1 | 158.3 | 25 | Huang et al., 2019 |
| 12 | GCaMP6s | tg(tetOs) | Mouse | V1 | 151.6 | 6 | Huang et al., 2019 |
| 13 | GCaMP6s | tg(Emx1) | Mouse | V1 | 157.5 | 26 | Huang et al., 2019 |
| 14 | GCaMP6s | AAV | Mouse | V1 | 60.1 | 7 | Chen et al., 2013 |
| 15 | GCaMP6s | AAV | Mouse | V1 | 59.1 | 9 | Theis et al., 2016 |
| 16 | GCaMP6s | AAV | Mouse | V1 | 59.1 | 9 | Theis et al., 2016 |
| 17 | GCaMP5k | AAV | Mouse | V1 | 50.0 | 9 | Akerboom et al., 2012 |
| 18 | R-CaMP1.07 | tg(Grik4-cre) + AAV | Mouse | CA3 | 20.0 | 4 | Schoenfeld et al., 2021 |
| 19 | R-CaMP1.07 | AAV | Mouse | S1 | 15.0 | 9 | Bethge et al., 2017 |
| 20 | jRCaMP1a | AAV | Mouse | V1 | 15.0 | 10 | Dana et al., 2016 |
| 21 | jRGECO1a | AAV | Mouse | V1 | 29.8 | 11 | Dana et al., 2016 |
| 22 | OGB-1 | Injection + tg(GFP-GIN) | Mouse | V1 (SST) | 15.6 | 5 | Kwan and Dan, 2012 |
| 23 | OGB-1 | Injection + tg(tdTomato-PV) | Mouse | V1 (PV) | 15.6 | 7 | Kwan and Dan, 2012 |
| 24 | GCaMP6f | tg(PV-cre) + AAV | Mouse (in vitro) | V1 (PV) | 26.6 | 13 | Khan et al., 2018 |
| 25 | GCaMP6f | tg(SOM-cre) + AAV | Mouse (in vitro) | V1 (SST) | 26.6 | 17 | Khan et al., 2018 |
| 26 | GCaMP6f | tg(VIP-cre) + AAV | Mouse (in vitro) | V1 (VIP) | 26.6 | 11 | Khan et al., 2018 |
| 27 | GCaMP6f | tg(tdTomato-PV) + AAV | Mouse | V1 (PV) | 30.0 | 4 | Rupprecht et al., 2021 |
| 28 | XCaMPgf | AAV | Mouse | V1 | 122 | 8 | Zhang et al., 2023 |
| 29 | GCaMP7f | AAV | Mouse | V1 | 122 | 22 | Zhang et al., 2023 |
| 30 | GCaMP8f | AAV | Mouse | V1 | 122 | 36 | Zhang et al., 2023 |
| 31 | GCaMP8m | AAV | Mouse | V1 | 122 | 42 | Zhang et al., 2023 |
| 32 | GCaMP8s | AAV | Mouse | V1 | 122 | 39 | Zhang et al., 2023 |
| 33 | Mixed (GCaMP7f /8/ XCaMPgf) | AAV | Mouse | V1 (Interneurons) | 122 | 17 | Zhang et al., 2023 |
| 40 | GCaMP6s | tg(VGlut2-Cre) | Mouse | Spinal cord | $\sim$2.5 | 21 | Rupprecht et al., 2025 |
| 41 | GCaMP6s | tg(Viaat-Cre) | Mouse | Spinal cord | $\sim$2.5 | 23 | Rupprecht et al., 2025 |

Table 7: Ground-truth recording sets used in the benchmark.

| Category | Indicator | Recording sets | $N$ |
|---|---|---|---|
| Chemical dye | OGB-1 | DS01, DS02, DS04, DS22, DS23 | 64 |
| Chemical dye | Cal-520 | DS03, DS05 | 13 |
| Legacy GECI | GCaMP5k | DS17 | 9 |
| Fast GCaMP6 | GCaMP6f | DS06–DS11, DS24–DS27 | 124 |
| Slow GCaMP6 | GCaMP6s | DS12–DS16 | 57 |
| GCaMP7 | GCaMP7f | DS29 | 22 |
| GCaMP8 variants | GCaMP8f, GCaMP8m, GCaMP8s | DS30, DS31, DS32 | 117 |
| Red-shifted GECI | R-CaMP | DS18, DS19 | 13 |
| Red-shifted GECI | jRCaMP1a | DS20 | 9 |
| Red-shifted GECI | jGECO1a | DS21 | 11 |
| Other GECI | XCaMPgf | DS28 | 8 |
| Interneurons | GCaMP7f, GCaMP8f, GCaMP8m, GCaMP8s, XCaMPgf | DS33 | 17 |
| Spinal cord (excit.) | GCaMP6s | DS40 | 21 |
| Spinal cord (inhib.) | GCaMP6s | DS41 | 23 |

Grouping these recordings by indicator family shows how the 33 source datasets contribute complementary kinetic regimes. The fixed splits below then isolate five representative transfer settings while preserving recording-level separation.

Table 8: Fixed evaluation splits. Source-side recordings are excluded from training under LOIO; evaluation recordings are held out in all settings.

| Split | Held-out under LOIO | Eval. recordings | Contrast |
|---|---|---|---|
| OGB-1 | DS01, DS22, DS23 | DS02, DS04 | Chemical-dye transfer |
| GCaMP6s | DS12, DS13, X-DS11, X-DS12 | DS14, DS15, DS16 | Slow-decay GECI |
| GCaMP8 variants | DS30, DS31 | DS32 | GCaMP8f/8m→8s |
| Red-shifted | DS18, DS19, DS20 | DS21 | Cross-indicator transfer |
| Spinal cord | DS41 | DS40 | Brain-to-spinal-cord |

### A.3 Loss function ablation under `+sim` LOIO

Table 9 compares four loss configurations on the five `+sim` LOIO splits. Adding each auxiliary term consistently improves mean correlation, and the full three-component setup achieves the highest score on every split.

Table 9: Mean Pearson correlation under `+sim` LOIO for SpikeSSL trained with progressive loss additions. All runs use identical data and hyperparameters, and only the active loss terms differ. **Bold**: best per column.

| Loss configuration | V1-OGB | V1-GCaMP6s | V1-GCaMP8 | Other-RCaMP | SC-GCaMP6s | Mean |
|---|---|---|---|---|---|---|
| Huber regression only | 0.527 | 0.445 | 0.861 | 0.703 | 0.622 | 0.632 |
| + variance head ($\beta$-NLL) | 0.536 | 0.461 | 0.872 | 0.712 | 0.641 | 0.644 |
| + gate BCE | 0.540 | 0.467 | 0.876 | 0.717 | 0.651 | 0.650 |
| Full (+ mean-rate) | **0.547** | **0.476** | **0.886** | **0.725** | **0.668** | **0.660** |

### A.4 Biophysical simulation pipeline specification

#### A.4.1 Spike Time Simulation

To construct realistic spike timings, we implemented two distinct strategies, namely drawing inter-spike intervals from predefined probability distributions and simulating membrane potential dynamics via the Izhikevich neuron model.

In the first strategy, the total spike count for each trial is determined by a target mean firing rate $\lambda_{\text{spike}}$ and a randomly generated recording duration $T$. Three stochastic point-process models cover distinct firing regimes. For standard stochastic firing, the spike count follows a Poisson distribution $N \sim \text{Poisson}(\lambda_{\text{spike}} T)$ and inter-spike intervals (ISIs) are drawn independently. A Gamma distribution $\Delta t_k \sim \Gamma(\kappa, \theta)$ generalizes this to model the relative refractory period, reducing to the standard exponential when $\kappa = 1$ and penalizing short ISIs when $\kappa > 1$. For quasi-periodic firing, a linear interpolation introduces stochastic perturbations into a uniform spike train, $\Delta t_k = (1-r)\cdot T_{\text{period}} + r \cdot T_{\text{rand}}$, where $T_{\text{period}} = 1/\lambda$ and $r \in [0, 1]$ controls the degree of randomness. For bursty firing, burst onsets follow a Poisson process with rate $\lambda_{\text{burst}}$, the spike count per burst follows $n_i \sim \text{Poisson}(\mu_{\text{burst}})$, and intra-burst ISIs are generated by stochastic jitter around a sub-millisecond mean interval.

In the second strategy, we numerically integrate the Izhikevich neuron model over a duration $T$ to efficiently generate biophysically realistic spike trains. By capturing the rapid depolarization and slow recovery of the membrane potential, this model faithfully reproduces complex firing dynamics. To reflect physiological stochasticity, the driving current $I(t)$ is formulated as a multi-scale composite synaptic drive:

$$I(t) = I_{\text{base}} + I_{\text{slow}}(t) + I_{\text{med}}(t) + I_{\text{fast}}(t),$$

where $I_{\text{base}}$ denotes the baseline current profile, while the $slow$, $med$, and $fast$ terms represent the respective frequency components of the synaptic perturbations.The temporal evolution of the fast membrane potential $v$ and the slow recovery variable $u$ under this driving current is governed by the following system of ordinary differential equations:

$$\frac{dv}{dt} = 0.04v^2 + 5v + 140 - u + I,$$
$$\frac{du}{dt} = a(bv - u).$$

Upon reaching the spike apex ($v \geq 30mV$), the state variables are instantaneously reset: $v \leftarrow c$ and $u \leftarrow u + d$. The dimensionless parameter set $(a, b, c, d)$ governs these intrinsic dynamics, enabling the simulation of distinct neuronal subclasses, including Regular Spiking (RS), Intrinsically Bursting (IB), Chattering (CH), and Fast Spiking (FS) neurons.

### A.4.2 Fluorescence Signal Simulation

In the MLSpike framework, the normalized intracellular calcium concentration $c(t)$ is modeled as a first-order linear system driven by the spike train $s(t) = \sum_k \delta(t - t_k)$, which is a set of Dirac functions placed at spike times $t_k$. The calcium level decays exponentially with a time constant $\tau$, governed by the differential equation:

$$\frac{dc}{dt} = -\frac{1}{\tau}c + s(t).$$

However, the assumption of a constant unitary response deviates from actual biological processes and contradicts experimentally observed data. To address this, we define the input $s(t)$ as a sequence of weighted discrete events, where the calcium influx elicited by the $k$-th spike at time $t_k$ is scaled by a dynamic weight $w(t_k)$. Consequently, the dynamics of $c(t)$ are modified to:

$$\frac{dc}{dt} = -\frac{1}{\tau}c + s(t) * w(t),$$

where the dynamic weight is formulated as:

$$w(t) = R(t) \cdot \max(0, 1 + \epsilon_k).$$

Here, $\epsilon_k \sim \mathcal{N}(0, \sigma_A^2)$ represents the stochastic amplitude jitter drawn from a normal distribution, reflecting the natural variability of calcium influx per spike. The variable $R(t) \in [0, 1]$ serves as a depletion variable representing the fraction of the available calcium response pool. When an action potential occurs at $t = t_k$, the available pool is instantaneously depleted by a constant proportion $d$ (the depletion factor):

$$R(t_k^+) = R(t_k) \cdot (1 - d),$$

where $R(t_k^+)$ denotes the instantaneous value of $R(t)$ immediately after the spike occurs at time $t_k$. During the inter-spike intervals ($t \neq t_k$), the temporal evolution is a continuous exponential recovery towards its maximum capacity of 1:

$$\frac{dR}{dt} = \frac{1}{\tau_{rec}}(1 - R(t)),$$

where $\tau_{rec}$ is the recovery time constant.

Subsequent to the influx of free calcium, the observable fluorescence signal is fundamentally determined by the concentration of calcium-bound indicator molecules. The binding between calcium ions and indicator molecules [B] obeys mass-action kinetics, as illustrated by the reaction:

$$[\mathrm{Ca}]^n + [\mathrm{B}] \underset{k_{\mathrm{off}}}{\overset{k_{\mathrm{on}}}{\rightleftharpoons}} [\mathrm{Ca}^n\mathrm{B}],$$

where $k_{\mathrm{on}}$ and $k_{\mathrm{off}}$ are the association and dissociation rates, respectively, and $n$ is the Hill coefficient representing the degree of cooperativity. This formulation can accommodate both genetically encoded calcium indicators (GECIs, e.g., GCaMP, where $n > 1$) and synthetic chemical dyes (e.g., OGB-1, where $n = 1$). According to the law of mass action, the rate of change for the bound product concentration is given by:

$$\frac{d\left[Ca^nB\right]}{dt} = k_{\mathrm{on}}\left[Ca\right]^n\left[B\right] - k_{\mathrm{off}}\left[Ca^nB\right].$$

By defining the normalized bound indicator fraction $p(t)$ as:

$$p = \frac{1}{\gamma}\frac{\left[Ca^nB\right] - \left[Ca^nB\right]_0}{\left[B\right]_T - \left[Ca^nB\right]_0},$$

the temporal evolution of $p(t)$ under standard constant-rate assumptions is governed by the following parameterized differential equation:

$$\frac{dp}{dt} = \frac{1}{\tau_{on}}\left[1 + \gamma((c_0 + c)^n - c_0^n)\right]\left(\frac{(c_0 + c)^n - c_0^n}{1 + \gamma((c_0 + c)^n - c_0^n)} - p\right),$$

where $\tau_{on}$ is the binding time constant at baseline, $c_0$ is the normalized resting calcium level, and $\gamma$ is the saturation parameter limiting the maximum bound fraction.

To extend this standard formulation and mechanistically characterize the conformational locking of the indicator molecules, we relax the constant-rate assumption for $k_{\text{off}}$. Specifically, the dissociation rate is modeled as a dynamically decelerating variable that explicitly depends on the instantaneous occupancy fraction $p(t)$:

$$k_{\text{off}}(t) = k_{\text{off},0} \cdot \max\left(0.1, 1 - \alpha \cdot p(t)\right),$$

where $k_{\text{off},0}$ is the basal dissociation rate and $\alpha$ determines the deceleration magnitude.

Next, we apply a phenomenological polynomial mapping to the dynamically derived variable $p(t)$. This transformation yields the nonlinearly scaled bound fraction $p_{\text{final}}(t)$:

$$p_{\text{final}}(t) = \begin{cases} c_3 p(t)^3 + c_2 p(t)^2 + c_1 p(t), & \text{if } p(t) \leq 1 \\ 1 + \kappa \cdot \left(p(t) - 1\right), & \text{if } p(t) > 1 \end{cases}$$

where the constraint $c_1 = 1 - c_2 - c_3$ ensures gain conservation at $p = 1$, and $\kappa = 3c_3 + 2c_2 + c_1$ represents the tangent slope at $p = 1$, facilitating a smooth linear extrapolation that guarantees numerical stability outside the standard dynamic range.

Furthermore, to account for the slow continuous baseline drifts inherent to realistic scenarios, we model the baseline fluorescence $B(t)$ as a Wiener process:

$$\frac{dB(t)}{dt} = \eta \cdot \xi_B(t),$$

where $\eta$ scales the amplitude of the baseline drift and $\xi_B(t)$ denotes standard Gaussian white noise.

Ultimately, the observed fluorescence intensity $F(t)$ is formulated as the integrated effect of the drifting baseline, the dynamic bound indicator concentration, and the measurement noise:

$$F(t) = B(t)\left(1 + A \cdot p(t)\right) + \sigma \cdot \xi_F(t),$$

where $A$ represents the maximum fluorescence transient amplitude, and $\sigma \cdot \xi_F(t)$ accounts for the additive Gaussian noise with variance $\sigma^2$.

### A.5 Detailed benchmark construction

The benchmark uses fixed recording-set partitions rather than random neuron-level resampling. In the in-domain setting, the source-side recordings listed in Table 8 are included during training and the corresponding evaluation recordings are held out for testing. In the LOIO setting, the source-side recordings from the target evaluation family are removed from training, while the evaluation recordings remain unchanged. All other real source-domain recordings remain available for training, so the only controlled difference between LOIO runs is the held-out indicator or tissue family.

The five splits in Table 8 are the evaluation units behind the in-domain and LOIO benchmark tables in the main paper. The held-out evaluation recordings are fixed at the recording-set level, which prevents different methods from being compared under different neuron-level samples.

Before training and testing, each fluorescence trace is windowed into 128-frame segments with 30 frames of left context, 30 frames of right context, and overlap size 2. We use raw fluorescence as the input representation, do not apply offline denoising, and do not resample traces before evaluation. When simulation augmentation is enabled, simulated traces are appended to the training pool only and are never included in any evaluation split.

### A.6 Additional results and analyses

#### A.6.1 Representative transfer functions

Figure 9 indicates that the dominant failure mode is not a uniform offset but an indicator-dependent distortion of gain. For OASIS and MLSpike, the predicted rate increases too slowly once the true rate enters the moderate and high regime, which is consistent with the strong saturation visible in the curves. CASCADE and ENS$^2$ recover more dynamic range, but their mappings still bend away

from the identity and exhibit broader inter-neuron variability, indicating that rate calibration remains unstable across indicators. SpikeSSL stays closer to the identity over a wider operating range and maintains a narrower spread across neurons, suggesting that its advantage comes from preserving response gain more faithfully rather than from isolated corrections at a few firing-rate values.

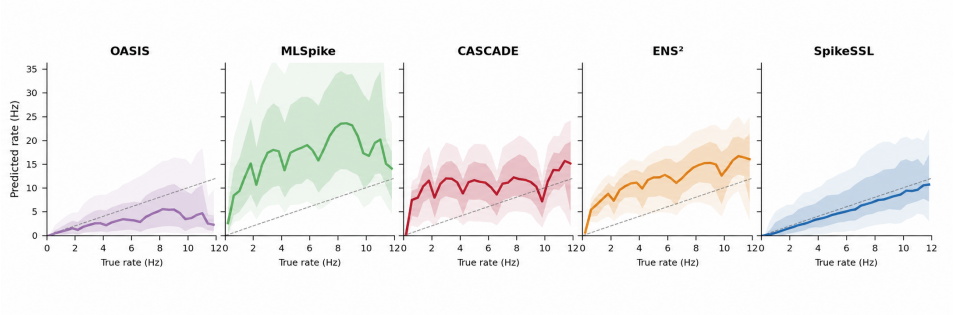



Figure 9: Representative transfer functions for five commonly used indicators. The shaded region denotes inter-neuron variability, and the dashed diagonal indicates the identity mapping. Indicators are selected because they span distinct kinetic regimes.

### A.6.2 Long-trial LOIO visualizations

To complement the per-window correlation scores reported in the main benchmark, we provide long-trial visualisations that span the full recording duration. We include one representative neuron from each of the six held-out calcium indicators in the LOIO evaluation, namely the chemical dye OGB-1, the early and improved fast green GECIs GCaMP5k and GCaMP7f, the slow green GECI GCaMP6s, the red-shifted GECI jGECO1a, and the next-generation fast green GECI GCaMP8f. For every neuron we display the recorded fluorescence trace together with spike-rate predictions from all six methods over the entire recording, with the smoothed ground-truth spike rate overlaid in grey for reference. The Pearson correlation coefficient $r$ reported in each panel is computed over the full trace. These visualisations corroborate the in-domain and LOIO benchmarks in the main paper, and additionally show that SpikeSSL with simulation augmentation tracks both isolated bursts and prolonged elevated firing across indicators with markedly different kinetics, while the remaining gap that simulation augmentation closes is most pronounced for indicators whose kinetics deviate substantially from the supervised training pool, such as jGECO1a and GCaMP5k.

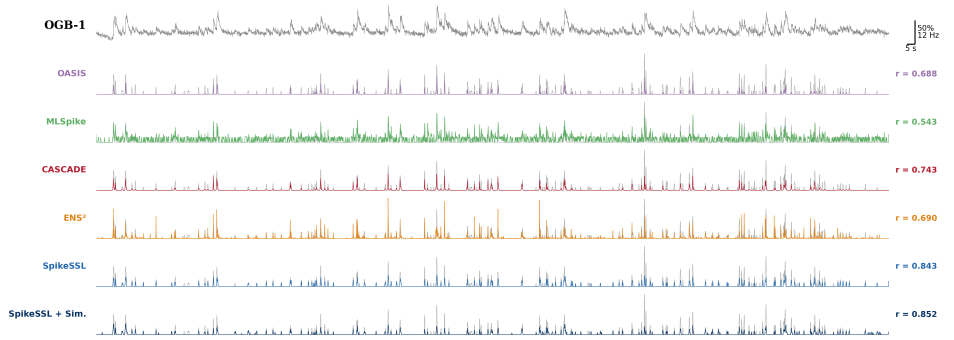



Figure 10: LOIO spike inference on a representative OGB-1 neuron (chemical dye with slow calcium decay kinetics). The top row shows the recorded $\Delta F/F$ trace. Subsequent rows display spike-rate estimates produced by OASIS, MLSpike, CASCADE, ENS$^2$, SpikeSSL, and SpikeSSL+Sim., each overlaid with the smoothed ground-truth spike rate (grey). The Pearson correlation coefficient $r$ is computed over the full recording duration.

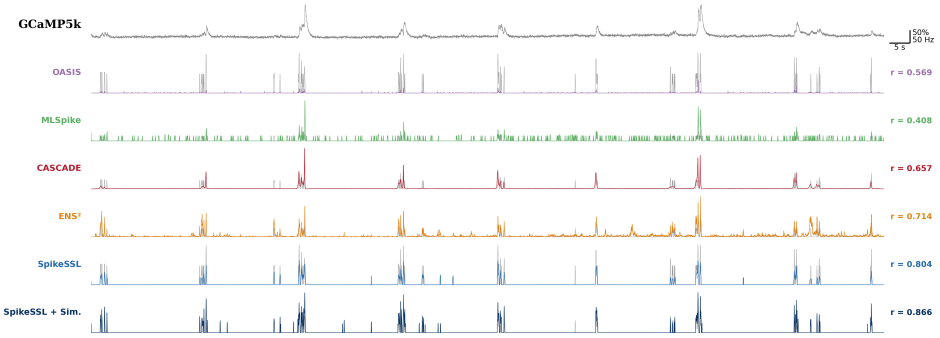



Figure 11: LOIO spike inference on a representative GCaMP5k neuron (an early-generation fast green GECI). Panel layout follows Figure 10.

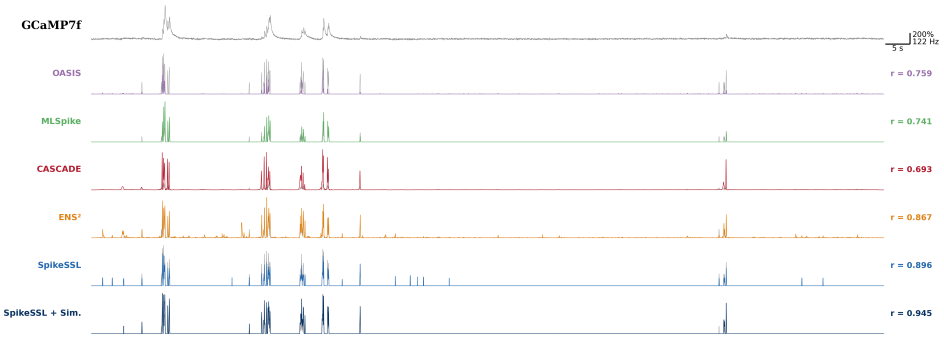



Figure 12: LOIO spike inference on a representative GCaMP7f neuron (an improved fast green GECI with enhanced sensitivity). Panel layout follows Figure 10.

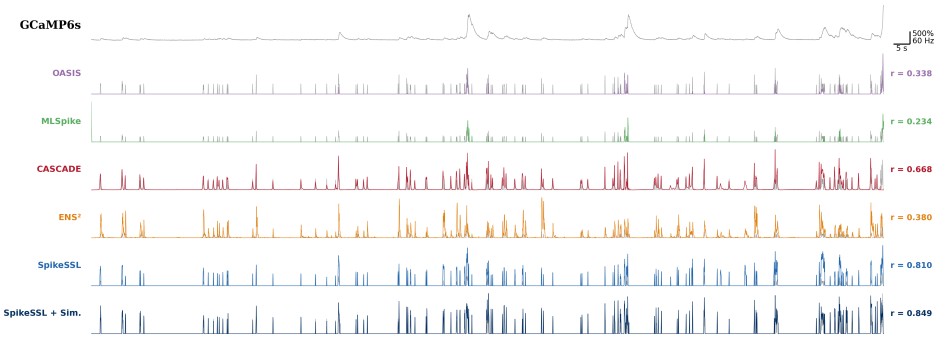



Figure 13: LOIO spike inference on a representative GCaMP6s neuron (a slow GECI recorded in a low-activity regime). Panel layout follows Figure 10.

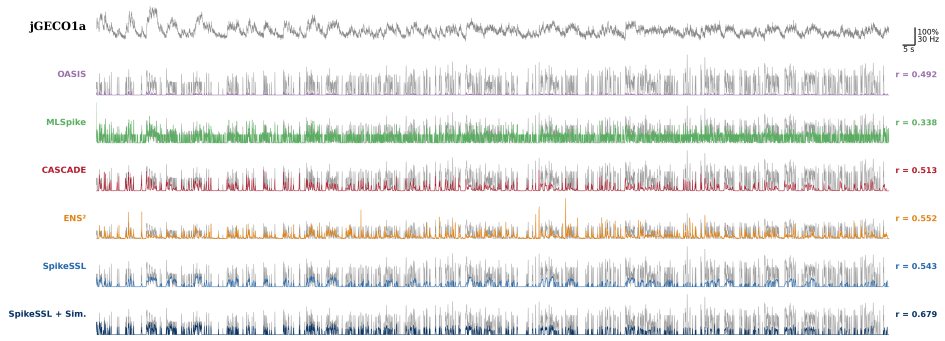



Figure 14: LOIO spike inference on a representative jGECO1a neuron (a red-shifted GECI with distinct spectral and kinetic properties). Panel layout follows Figure 10.

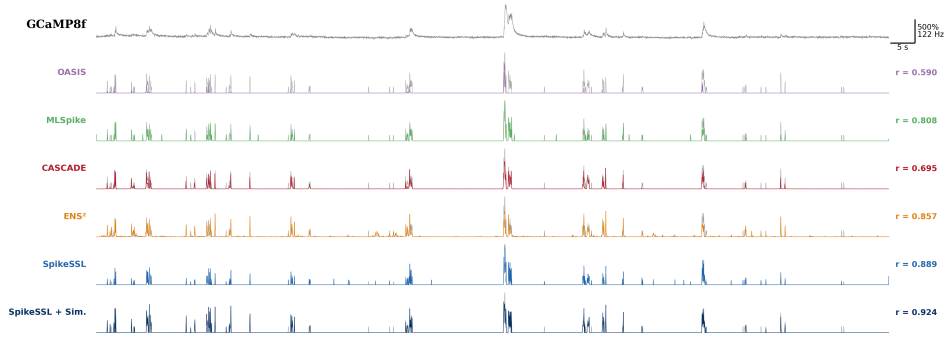



Figure 15: LOIO spike inference on a representative GCaMP8f neuron (a next-generation fast GECI with high signal-to-noise ratio). Panel layout follows Figure 10.

The following subsections present additional controls, robustness studies, and mechanistic analyses that complement the main results. All experiments use the fixed recording-set partitions described in Appendix A.5. Unless stated otherwise, correlation denotes Pearson correlation of predicted and ground-truth spike rates, higher is better, and error and absolute bias are lower is better.

### A.7 Test-Time Conditioning Uses No Target-Domain Pool

The seven trace statistics are computed independently from the current input window. Thus, LOIO inference uses one unlabeled input window, no target-domain pool, no target labels, and no parameter update. Pooling statistics across multiple target windows does not improve performance (Table 10), supporting that the method is per-window conditional inference rather than few-shot adaptation.

Table 10: Effect of pooling unlabeled target windows for trace statistics under LOIO.

| | Per-window | Pool 5 | Pool 20 | Pool all |
|---|---|---|---|---|
| Mean correlation | **0.654** | 0.649 | 0.650 | 0.650 |

LOIO uses the reserved unknown indicator code (ID=0), never a held-out dye label. Removing the discrete-ID branch changes mean LOIO correlation from 0.654 to 0.647. In contrast, in-domain performance falls from 0.741 to 0.703 without the ID branch, and shuffling the observed ID at inference gives 0.578. Therefore, the ID chiefly captures regularities of observed domains, while zero-shot transfer is supported by sampling rate, input-window statistics, and the trace itself (Table 11).

Table 11: Indicator-ID intervention. The LOIO baseline uses the unknown ID=0.

| Setting | V1-OGB | V1-GCaMP6s | V1-GCaMP8 | Other-RCaMP | SC-GCaMP6s | Mean |
|---|---|---|---|---|---|---|
| LOIO baseline (ID=0) | **0.650** | **0.509** | **0.853** | **0.717** | **0.542** | **0.654** |
| LOIO without ID branch | 0.646 | 0.500 | 0.849 | 0.708 | 0.532 | 0.647 |
| In-domain full model | **0.743** | **0.579** | **0.915** | **0.731** | **0.736** | **0.741** |
| In-domain shuffled ID | 0.587 | 0.457 | 0.791 | 0.619 | 0.437 | 0.578 |
| In-domain without ID branch | 0.700 | 0.546 | 0.881 | 0.694 | 0.694 | 0.703 |

For completeness, the structural loss uses valid-frame mask $w_t \in \{0,1\}$, $Z = \sum_t w_t$, $Q = \{t : s_t \geq q_{0.9}(s)\}$, and Huber loss $h_\delta$:

$$L_{\text{over}} = Z^{-1} \sum_t w_t \max(\hat{s}_t - s_t, 0)^2, \tag{11}$$

$$L_{\text{peak}} = |Q|^{-1} \sum_{t \in Q} h_\delta(\max(s_t - \hat{s}_t, 0)), \tag{12}$$

$$L_{\text{bias}} = \left( Z^{-1} \sum_t w_t \hat{s}_t - Z^{-1} \sum_t w_t s_t \right)^2, \tag{13}$$

$$L_{\text{struct}} = \beta_{\text{over}} L_{\text{over}} + \beta_{\text{peak}} L_{\text{peak}} + \lambda_\mu L_{\text{bias}}. \tag{14}$$

We use $\beta_{\text{over}} = 3.0$, $\beta_{\text{peak}} = 0.3$, $\lambda_\mu = 0.5$, $\delta = 0.1$, and $q = 0.9$.

### A.8 Breadth of the Universal Benchmark

The benchmark aggregates ground-truth recordings from multiple laboratories, sensors, species, brain regions, cell types, and acquisition protocols rather than a single experimental setup (Table 6). LOIO removes every recording of the held-out family and applies no target-domain update. To characterize the resulting domain variation, we measured isolated-spike rise time, decay time, and SNR in the public recordings. Large within-family dispersion and systematic cross-family shifts are evident in Table 12. These event-level summaries are distinct from the baseline-aware exponential $\tau$ analysis in Table 19.

Table 12: Isolated-spike kinetic and SNR dispersion in public recordings. Times are seconds.

| Indicator family | $n$ | Rise $\mu \pm \sigma$ | Decay $\mu \pm \sigma$ | SNR $\mu \pm \sigma$ |
|---|---|---|---|---|
| Cal-520 | 6 | 0.086±0.085 | 0.253±0.094 | 5.533±2.295 |
| GCaMP5k | 36 | 0.079±0.095 | 0.073±0.058 | 3.934±0.904 |
| GCaMP6f | 907 | 0.061±0.071 | 0.119±0.120 | 5.570±4.358 |
| GCaMP6s | 436 | 0.104±0.095 | 0.177±0.173 | 5.713±3.011 |
| GCaMP6s spinal | 180 | 0.112±0.164 | 0.155±0.199 | 5.210±2.906 |
| GCaMP7f | 57 | 0.030±0.025 | 0.070±0.058 | 6.125±4.398 |
| GCaMP8 variants | 2,967 | 0.018±0.020 | 0.083±0.060 | 6.417±4.017 |
| Interneurons mixed | 102 | 0.037±0.056 | 0.050±0.044 | 3.886±0.936 |
| OGB-1 | 330 | 0.094±0.073 | 0.193±0.155 | 7.141±9.075 |
| R-CaMP | 260 | 0.082±0.069 | 0.235±0.201 | 10.831±11.666 |
| Unknown | 619 | 0.096±0.122 | 0.138±0.143 | 5.689±4.283 |
| XCaMPgf | 37 | 0.028±0.031 | 0.060±0.044 | 4.511±2.445 |
| jGECO1a | 96 | 0.066±0.056 | 0.126±0.092 | 6.110±3.671 |
| jRCaMP1a | 56 | 0.146±0.166 | 0.261±0.198 | 11.627±26.422 |

For example, rise-time standard deviation exceeds the corresponding mean for GCaMP6f, GCaMP8 variants, and XCaMPgf. Across families, mean decay ranges from 0.050 s to 0.261 s and mean SNR from 3.886 to 11.627. We use "universal" in this operational sense: a single model transfers across this diverse, multi-laboratory benchmark without target labels or retraining.

### A.9 Data Scaling and Statistical Robustness

Increasing the amount of real data, synthetic data, or indicator-family coverage improves mean LOIO correlation (Table 13). These controls indicate that data quantity is the primary limitation in the present benchmark.

Table 13: Data-scaling study under fixed LOIO evaluation.

| Factor | Metric | Scale | | | | |
|---|---|---|---|---|---|---|
| Real-recording fraction | Correlation | 25%: 0.564 | 50%: 0.614 | 75%: 0.637 | 100%: **0.654** | |
| | Error | 25%: 1.791 | 50%: 1.534 | 75%: 1.133 | 100%: 1.123 | |
| | Absolute bias | 25%: 1.187 | 50%: 1.081 | 75%: 0.583 | 100%: 0.588 | |
| Synthetic-data fraction | Correlation | 0%: 0.654 | 25%: 0.667 | 50%: 0.679 | 75%: 0.692 | 100%: **0.704** |
| | Error | 0%: 1.123 | 25%: 1.118 | 50%: 1.113 | 75%: 1.108 | 100%: 1.103 |
| | Absolute bias | 0%: 0.588 | 25%: 0.564 | 50%: 0.539 | 75%: 0.514 | 100%: 0.490 |
| Indicator families | Correlation | 4: 0.543 | 8: 0.583 | 13: **0.654** | | |
| | Error | 4: 1.140 | 8: 1.204 | 13: 1.123 | | |
| | Absolute bias | 4: 0.871 | 8: 0.811 | 13: 0.588 | | |

Across seeds $\{42, 2719, 8888, 1337, 2025\}$, mean LOIO correlation is $0.660 \pm 0.016$, close to the seed-42 result of 0.654; in-domain correlation is $0.739 \pm 0.004$, close to 0.741. Table 14 reports all per-split results. Mean absolute bias is used in the final column.

Table 14: Five-seed robustness results. Values are mean ± standard deviation.

| | LOIO | | | In-domain | | |
|---|---|---|---|---|---|---|
| Split | Corr. | Error | Abs. bias | Corr. | Error | Abs. bias |
| V1-OGB | 0.666±0.030 | 1.096±0.513 | 0.309±0.574 | 0.729±0.017 | 0.875±0.062 | 0.131±0.180 |
| V1-GCaMP6s | 0.507±0.022 | 0.951±0.008 | 0.780±0.069 | 0.575±0.009 | 0.948±0.005 | 0.779±0.031 |
| V1-GCaMP8 | 0.864±0.013 | 0.531±0.015 | 0.054±0.213 | 0.917±0.005 | 0.487±0.044 | 0.036±0.059 |
| Other-RCaMP | 0.724±0.013 | 1.141±0.157 | 0.445±0.267 | 0.748±0.019 | 1.025±0.137 | 0.149±0.130 |
| SC-GCaMP6s | 0.539±0.018 | 0.889±0.046 | 0.840±0.082 | 0.728±0.009 | 0.825±0.036 | 0.784±0.050 |
| Mean | **0.660±0.016** | 0.922±0.116 | 0.486±0.074 | **0.739±0.004** | 0.832±0.021 | 0.376±0.069 |

### A.10 Ablations

Removal and cumulative build-up studies were repeated across the same five seeds. The bidirectional IIR backbone, multi-modal conditioning, gate head, and variance head each contribute to the full model (Table 15).

Table 15: Five-seed removal and cumulative build-up ablations under LOIO.

| Removal variant | Corr. | Error | Abs. bias | Build-up stage | Corr. | Error | Abs. bias |
|---|---|---|---|---|---|---|---|
| −AdaLN | 0.604±0.014 | 1.276±0.103 | 0.456±0.061 | 1D U-Net backbone | 0.481±0.022 | 1.173±0.101 | 0.788±0.088 |
| −statistics | 0.626±0.012 | 1.213±0.091 | 0.556±0.073 | Transformer backbone | 0.484±0.020 | 1.197±0.104 | 0.618±0.074 |
| −bidirectional IIR | 0.597±0.016 | 1.377±0.120 | 0.719±0.086 | Bidirectional IIR backbone | 0.539±0.018 | 1.191±0.101 | 0.757±0.083 |
| −variance head | 0.611±0.013 | 1.243±0.112 | 1.005±0.105 | + indicator ID | 0.550±0.017 | 1.167±0.096 | 0.696±0.079 |
| −gate | 0.624±0.011 | 1.153±0.084 | 0.402±0.057 | + sampling rate | 0.554±0.016 | 1.175±0.098 | 0.672±0.077 |
| Full SpikeSSL | **0.660±0.016** | **0.922±0.116** | 0.486±0.074 | + signal statistics | 0.576±0.015 | 1.145±0.091 | 0.703±0.081 |
| | | | | + AdaLN | 0.594±0.014 | 1.153±0.087 | 0.696±0.076 |
| | | | | + gate | 0.612±0.013 | 1.117±0.082 | 0.665±0.071 |
| | | | | + variance head | **0.660±0.016** | **0.922±0.116** | **0.486±0.074** |

### A.11 Universal, Specific, and Few-Shot Regimes

For each of 13 indicator families, the specific-model comparison holds out one recording and trains on the remaining recordings of that indicator. Universal models pool all indicator families and use exactly the same held-out recording for evaluation. SpikeSSL benefits from pooling, unlike CASCADE (Table 16).

Table 16: Per-indicator comparison of specific and universal training.

| Indicator | CASCADE-specific | SpikeSSL-specific | CASCADE-universal | SpikeSSL-universal |
|---|---|---|---|---|
| OGB-1 | 0.663 | **0.760** | 0.583 | 0.728 |
| Cal-520 | **0.721** | 0.633 | 0.647 | 0.693 |
| GCaMP5k | 0.687 | 0.520 | 0.688 | **0.695** |
| GCaMP6f | 0.500 | **0.626** | 0.416 | 0.562 |
| GCaMP6s | 0.262 | 0.435 | 0.279 | **0.628** |
| GCaMP7f | 0.776 | 0.508 | 0.412 | **0.816** |
| GCaMP8f | 0.753 | 0.928 | 0.815 | **0.933** |
| GCaMP8m | 0.799 | **0.939** | 0.809 | 0.928 |
| GCaMP8s | 0.884 | **0.927** | 0.767 | 0.912 |
| R-CaMP | 0.309 | 0.555 | 0.644 | **0.691** |
| jRCaMP1a | 0.689 | 0.538 | 0.662 | **0.769** |
| jGECO1a | 0.736 | 0.613 | 0.672 | **0.790** |
| XCaMPgf | 0.797 | 0.537 | 0.446 | **0.833** |
| Mean | 0.660 | 0.655 | 0.603 | **0.768** |

For few-shot adaptation, 20 target-domain neurons are set aside as a support pool and all remaining traces form the evaluation set. For $k > 0$, models are fine-tuned from the $k = 0$ universal LOIO weights using $k$ labeled support neurons. Fine-tuning first exceeds the zero-shot value at $k = 5$ and reaches 0.719 at $k = 20$ (Table 17).

Table 17: Few-shot fine-tuning from zero-shot LOIO pretrained weights.

| $k$ | V1-OGB | V1-GCaMP6s | V1-GCaMP8 | Other-RCaMP | SC-GCaMP6s | Mean |
|---|---|---|---|---|---|---|
| 0 | 0.622 | 0.466 | 0.843 | 0.696 | 0.493 | 0.624 |
| 1 | 0.488 | 0.324 | 0.766 | 0.649 | 0.606 | 0.567 |
| 3 | 0.509 | 0.389 | 0.888 | 0.683 | 0.545 | 0.603 |
| 5 | 0.521 | 0.412 | 0.895 | 0.718 | 0.648 | 0.639 |
| 10 | 0.621 | 0.459 | 0.862 | 0.721 | 0.714 | 0.675 |
| 20 | 0.707 | 0.548 | 0.892 | 0.725 | 0.722 | 0.719 |

### A.12 Disentangling Indicator and Acquisition Shifts

We construct mini-domain evaluations that hold an indicator fixed while varying one additional factor, excluding other shift types from each data pool. In the species split, GCaMP6f mouse V1 recordings (DS09–11 and X-DS09/10) transfer to zebrafish recordings (DS06–08). The cell-type split fixes mouse V1 and GCaMP6f while changing the labeled population. The tissue split fixes GCaMP6s while transferring from brain to spinal cord. SpikeSSL gives the best correlation in all three cases (Table 18).

Table 18: Mini-domain shifts with fixed indicator.

| | GCaMP6f species | | | GCaMP6f cell type | | | GCaMP6s tissue | | |
|---|---|---|---|---|---|---|---|---|---|
| Method | Corr. | Error | Bias | Corr. | Error | Bias | Corr. | Error | Bias |
| OASIS | 0.191 | 0.982 | -0.945 | 0.266 | 0.985 | -0.974 | 0.365 | 0.975 | -0.964 |
| MLSpike | 0.085 | 1.130 | -0.706 | 0.098 | 1.015 | -0.897 | 0.307 | 1.005 | -0.919 |
| CASCADE | 0.540 | 1.304 | 0.539 | 0.574 | 0.977 | -0.144 | 0.226 | 1.083 | -0.442 |
| ENS$^2$ | 0.476 | 1.266 | 0.229 | 0.584 | **0.898** | -0.680 | 0.278 | 1.092 | -0.391 |
| SpikeSSL | **0.636** | **0.952** | -0.442 | **0.673** | 0.931 | -0.798 | **0.542** | **0.911** | -0.592 |

### A.13 Dynamics Readouts of the Conditioned IIR Bank

We compare four complementary quantities in Table 19. Reference $\tau$ is a catalogue decay time, measured $\tau$ is fitted from isolated real fluorescence responses at 30 Hz using a joint baseline and exponential fit, and $\|W_\alpha\|_F$ and ACF $\tau$ are extracted from a separately trained single-indicator SpikeSSL model. $W_\alpha$ maps the condition vector to shifts of the channel-wise $\alpha$ bank. Its Frobenius norm measures the strength of that influence, not a time constant. The norms vary across indicators, while

the three kinetically similar GCaMP8 variants have similar values. ACF $\tau$ is positively associated with measured $\tau$ (Spearman $\rho = 0.886$, $p = 5.6 \times 10^{-5}$), although not in fixed proportion. These complementary readouts support indicator-dependent dynamics rather than a fixed shared kernel.

Table 19: Kinetic measurements and learned dynamics readouts.

| Indicator | Ref. $\tau$ (ms) | $\|W_\alpha\|_F$ | Measured $\tau$ (ms) | ACF $\tau$ (ms) |
|---|---|---|---|---|
| GCaMP8f | 60 | 5.35 | 175 | 145 |
| GCaMP8s | 70 | 5.94 | 314 | 165 |
| GCaMP8m | 80 | 6.11 | 136 | 132 |
| GCaMP7f | 100 | 0.60 | 223 | 154 |
| XCaMPgf | 120 | 0.39 | 179 | 152 |
| GCaMP6f | 180 | 5.44 | 212 | 140 |
| Cal-520 | 200 | 0.41 | 878 | 178 |
| GCaMP5k | 250 | 0.50 | 547 | 163 |
| R-CaMP | 280 | 0.25 | 535 | 171 |
| jRCaMP1a | 280 | 0.74 | 756 | 163 |
| jGECO1a | 300 | 0.81 | 572 | 162 |
| GCaMP6s | 600 | 4.64 | 912 | 192 |
| OGB-1 | 750 | 2.59 | 854 | 191 |

### A.14 Simulation: Benefits and Failure Modes

We repeated simulation-augmented LOIO across five seeds and characterized the target gap in the exact conditioning space of SpikeSSL: log sampling rate, standard deviation, skewness, excess kurtosis, and autocorrelation at lags 1, 3, 5, and 10. After robust scaling, we compute a 512-projection sliced Wasserstein-1 distance. $\Delta W_1$ is the real-only target gap minus the equal-sampled real-plus-simulation gap, so a positive value denotes improved alignment. AC averages the four autocorrelation terms and shape averages standard deviation, skewness, and kurtosis.

Table 20: Simulation gain and alignment analysis.

| Split | No-sim corr. | +Sim corr. | $\Delta$ corr. | $\Delta$SWD | $\Delta W_1$ AC | $\Delta W_1$ shape | $\Delta W_1$ kurt. |
|---|---|---|---|---|---|---|---|
| V1-OGB | 0.666±0.030 | 0.704±0.012 | +0.038 | +0.013 | -0.111 | +0.008 | -0.132 |
| V1-GCaMP6s | 0.507±0.022 | 0.531±0.008 | +0.024 | +0.065 | +0.197 | -0.031 | +0.094 |
| V1-GCaMP8 | 0.864±0.013 | 0.914±0.015 | +0.050 | -0.090 | -0.124 | -0.157 | -0.260 |
| Other-RCaMP | 0.724±0.013 | 0.712±0.009 | -0.012 | +0.018 | +0.110 | -0.051 | -0.180 |
| SC-GCaMP6s | 0.539±0.018 | 0.677±0.010 | +0.138 | +0.126 | +0.187 | +0.065 | +0.202 |
| Mean | **0.660±0.016** | **0.708±0.011** | **+0.048** | | | | |

SC-GCaMP6s obtains the largest gain, alongside the largest overall, temporal, and shape gap reductions. Other-RCaMP is a diagnostic non-uniform case: simulation improves autocorrelation alignment but worsens shape, especially kurtosis, consistent with incomplete coverage of red-indicator nonlinear statistics. Its $-0.012$ change is within five-seed variability, so it does not establish harmful negative transfer. Conversely, GCaMP8 improves despite a larger marginal SWD, consistent with label-aware synthetic supervision regularizing the inverse mapping beyond a label-agnostic distributional metric. These results motivate target-relative weighting or rejection of synthetic windows when conditional dynamics are mismatched.

### A.15 Threshold, Artifact, and Runtime Sensitivity

The inference gate threshold was swept with the trained model fixed. The default $\tau = 0.5$ maximizes mean correlation, the task's primary metric; normalized error is smallest at $\tau = 0.6$, and absolute bias reaches its minimum at $\tau = 0.7$ (Table 21).

Table 21: Gate-threshold sensitivity. Each entry is mean across five LOIO splits.

| Threshold | 0.2 | 0.3 | 0.4 | 0.5 | 0.6 | 0.7 | 0.8 |
|---|---|---|---|---|---|---|---|
| Correlation | 0.628 | 0.651 | 0.654 | **0.654** | 0.648 | 0.640 | 0.628 |
| Normalized error | 1.454 | 1.322 | 1.214 | 1.123 | **1.046** | 1.256 | 1.392 |
| Absolute bias | 0.831 | 0.739 | 0.659 | 0.588 | 0.525 | **0.464** | 0.483 |

Table 22 characterizes two common non-biological perturbations. Per-window statistics do not make SpikeSSL invariant to artifacts: baseline drift is the dominant degradation, while sparse motion is milder. Standard motion correction and baseline detrending are recommended before inference.

Table 22: LOIO robustness to baseline drift and sparse motion artifacts.

| Condition | V1-OGB | V1-GCaMP6s | V1-GCaMP8 | Other-RCaMP | SC-GCaMP6s | Mean |
|---|---|---|---|---|---|---|
| Clean | 0.650 | 0.509 | 0.853 | 0.717 | 0.542 | **0.654** |
| Baseline drift | 0.636 | 0.392 | 0.605 | 0.665 | 0.478 | 0.555 |
| Sparse motion artifacts | 0.618 | 0.466 | 0.835 | 0.702 | 0.500 | 0.624 |
| Drift + motion | 0.620 | 0.409 | 0.673 | 0.684 | 0.463 | 0.570 |

SpikeSSL is designed primarily for offline analysis because the bidirectional IIR block uses bounded future context within each window. Nevertheless, sliding-window inference avoids holding an entire recording in memory. On an A100, one forward pass is approximately 29 times faster than the arrival of a 30-Hz frame window, so overlapping-window processing can keep pace with many acquisition settings while remaining noncausal within each window.

## NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The abstract and the Introduction state our three main contributions (the SpikeSSL architecture with dynamics-informed IIR backbone and multi-modal conditioning, the biophysical simulation augmentation pipeline, and the in-domain plus zero-shot LOIO benchmark), and each is supported by the experimental results in the Experiments section and the appendices.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: The Discussion at the end of the Experiments section explicitly notes that uncertainty calibration is only validated on the five-split LOIO benchmark and may not transfer to more extreme acquisition shifts, that the simulation pipeline relies on indicator-specific kinetic priors, and that ground-truth coverage of newly developed indicators (e.g., GCaMP8 variants and red-shifted sensors) remains uneven.

   Guidelines:

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

judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

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

   Answer: [N/A]

   Justification: The paper does not present formal theorems or proofs; the contribution is an architecture, a simulation pipeline, and an empirical benchmark.

   Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: The Methodology section fully specifies the SpikeSSL architecture, the Implementation Details paragraph reports the optimizer, learning rate, model dimensions, loss weights, and training schedule, and the appendices document the benchmark splits, simulation pipeline, and evaluation protocol. All datasets used are publicly available.

   Guidelines:

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

(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

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

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

   Answer: [Yes]

   Justification: All ground-truth datasets used are publicly available (the CASCADE database [16], the GCaMP8 recordings [26, 18], and the spinal cord set [17]). The code, training scripts, and benchmark configurations will be released in a public repository upon acceptance; an anonymized version is provided as supplemental material.

   Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: The Experiments section (Datasets, Benchmark and Metrics, Implementation Details) reports the five fixed evaluation splits, the optimizer (Adam, $\beta_1$=0.9, $\beta_2$=0.99, lr= $10^{-3}$), model dimensions, loss-term hyperparameters, and the training schedule. Additional details on the benchmark and the simulation pipeline are provided in the appendices.

   Guidelines:

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

7. **Experiment statistical significance**

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

Answer: [Yes]

Justification: Appendix A.9 reports mean $\pm$ standard deviation across five training seeds $\{42, 2719, 8888, 1337, 2025\}$ for SpikeSSL on every fixed in-domain and LOIO split, together with multi-seed ablations and simulation-augmented LOIO results.

Guidelines:

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

8. **Experiments compute resources**

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

Answer: [Yes]

Justification: All training runs use a single NVIDIA A100 GPU and converge within roughly 20 epochs. The model has 8 SpikeSSL blocks with model dimension $D{=}192$, which fits comfortably in 40 GB of GPU memory; full hyperparameters and runtime information are reported in the Implementation Details paragraph of the Experiments section and the appendix.

Guidelines:

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

9. **Code of ethics**

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

Answer: [Yes]

Justification: We have reviewed the NeurIPS Code of Ethics and the work conforms to it. The paper uses only previously published, properly licensed neuroscience datasets and does not involve human subjects, deployable identification systems, or other components flagged by the Code of Ethics.

Guidelines:

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

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: The proposed method is a basic-research tool for inferring spike trains from calcium imaging in animal-model neuroscience. Its primary positive impact is improved analysis of neural recordings, which can accelerate systems-neuroscience research. We are not aware of any direct negative societal applications; the method does not process data from human subjects and produces no content that could be misused for surveillance, identification, or generative misuse.

    Guidelines:

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

11. **Safeguards**

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

    Answer: [N/A]

    Justification: The released model is a domain-specific spike inference network trained on publicly available calcium-imaging recordings from animal experiments and poses no identifiable misuse risk.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.

- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: All datasets and baseline implementations are publicly available and are properly cited (CASCADE [16], OASIS [5], MLSpike [3], ENS$^2$ [28], GCaMP8 and red-shifted indicator recordings [26, 18], spinal cord set [17]). License and term-of-use information for each asset is summarized in the appendix and respected throughout.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [Yes]

    Justification: We will release the SpikeSSL training code, the simulation pipeline, and the trained checkpoints. Documentation, configuration files, and instructions to reproduce all reported numbers are provided alongside the code in the supplemental material.

    Guidelines:

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

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

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

    Answer: [N/A]

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

Guidelines:

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

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

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

    Answer: [N/A]

    Justification: The paper does not involve research with human subjects; all experimental data come from previously published animal-model recordings collected and released by their original authors.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

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

    Answer: [N/A]

    Justification: LLMs were used only to assist with writing and editing the manuscript and do not contribute to the core methodology, model design, simulation pipeline, or experimental results.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.