Interpretable deep learning for deconvolutional analysis of neural signals

Bahareh Tolooshams[1 2 3 *], **Sara Matias**[1 4 *], **Hao Wu**[1 4], **Simona Temereanca**[5], **Naoshige Uchida**[1 4], **Venkatesh N. Murthy**[1 4], **Paul Masset**[1 4 † ✉], **Demba Ba**[1 2 6 † ✉]

[1]*Center for Brain Science, Harvard University, Cambridge MA, 02138;* [2]*John A. Paulson School of Engineering and Applied Sciences, Harvard University, Cambridge MA, 02138;* [3]*Computing + Mathematical Sciences, California Institute of Technology, Pasadena, CA, 91125;* [4]*Department of Molecular and Cellular Biology, Harvard University, Cambridge MA, 02138;* [5]*Carney Institute for Brain Science, Brown University, Providence, RI, 02906;* [6]*Kempner Institute for the Study of Natural & Artificial Intelligence, Harvard University, Cambridge MA, 02138*

✉ **For correspondence:** masset.neuro@gmail.com, demba@seas.harvard.edu

[*]BT and SM have contributed equally to this work.

[†]DB and PM have jointly supervised this work.

Data availability: Data and code will be deposited in a public repository upon publication.

Funding: BT and DB were in part supported by ARO under Grant W911NF-16-1-0368; this was a collaboration between the US DOD, UK MOD, and UK Engineering and Physical Research Council (EPSRC) under the Multidisciplinary University Research Initiative. This work was supported by NIH grant 5R01DC017311 to NU and VNM. SM was supported by the Human Frontier Science Program (LT000801/2018), the Harvard Brain Science Initiative (Young Scientist Transitions Award) and the Brain and Behavior Research Foundation (NARSAD Young Investigator Grant no.30035). PM was partially supported by a grant from the Harvard Mind Brain Behavior Interfaculty Initiative.

Competing interests: The authors declare no competing interests.

Abstract

The widespread adoption of deep learning to build models that capture the dynamics of neural populations is typically based on "black-box" approaches that lack an interpretable link between neural activity and function. Here, we propose to apply algorithm unrolling, a method for interpretable deep learning, to design the architecture of sparse deconvolutional neural networks and obtain a direct interpretation of network weights in relation to stimulus-driven single neuron activity through a generative model. We characterize our method, referred to as deconvolutional unrolled neural learning (DUNL), and show its versatility by applying it to deconvolve single-trial local signals across multiple brain areas and recording modalities. To exemplify use cases of our decomposition method, we uncover multiplexed salience and reward prediction error signals from midbrain dopamine neurons in an unbiased manner, perform simultaneous event detection and characterization in somatosensory thalamus recordings, and characterize the responses of neurons in piriform cortex. Our work leverages the advances in interpretable deep learning to gain a mechanistic understanding of neural dynamics.

Introduction

Understanding the activity of neurons, both at the single neuron and population levels, in relation to features in the environment and the behaviour of an organism, is a key question in neuroscience. Recent technological advancements and experimental methods have allowed researchers to record from an increasingly large population of identified single neurons using high-throughput electrophysiology or imaging in animals performing complex tasks [1–3]. In such complex environments, external events might unfold in a variety of timescales, which can give rise to neural signals also expressed over different timescales across the population of recorded neurons. Moreover, these neural representations show complex dynamics and differing levels of multiplexing. For example, single neurons across the cortical hierarchy exhibit varying degrees of mixed selectivity to task parameters depending on task structure and demands [4–9]. Neuromodulatory neurons, such as midbrain dopamine neurons, can respond to different environmental and internal vari-

ables [10, 11]. Additionally, single-neuron activity has been proposed to be composed of multiple components required for reward evaluation, such as valence and salience [12, 13].

In order to understand how multiple representations in simultaneously-recorded single neurons enable population level computations, we need fast and reliable methods for decomposing their activity into overlapping and non-overlapping local components/events that can capture important intrinsic heterogeneity in the recorded populations. Here, we develop such a deconvolutional method.

A reasonable deconvolution method ought to meet several requirements. First, the method should be able to be implemented on single instantiations of the neural data without the need for averaging over trials or animals [14–16]. Preferably, it should be applicable to both structured and naturalistic tasks in which there is little or no trial structure [17–20]. Second, it should be flexible with respect to the source signal (e.g., spike count data or a proxy signal such as calcium levels via a fluorescent indicator [21]). Third, the method should utilize an expressive class of mappings from latent representations to data, namely ones that can capture the complexity of neural data. Finally and importantly, the method should be interpretable. By interpretable we mean the existence of 1) a direct mapping between stimuli (internal or external) and latent variables; and 2) a direct mapping between these latent variables, which are effectively parameters of the network, to computational function. In our framework, this concept of interpretability is in place by design, since it is based on a probabilistic generative model that can be interpreted via neural impulse responses [22–24].

Prior work using deep learning has addressed, to varying extents, the first three of these desiderata by extracting a low-dimensional latent space from the neural data through a non-linear deep neural architecture [25–28]. However, they do not provide a direct link between the contributions of single neurons or neuron types and the population level computation, owing to their "black-box" approach typical of deep networks [29, 30]. Our method complements these existing tools, by extracting interpretable impulse-like responses of multiplexed signals from single neurons, which can be further used to characterize heterogeneity and homogeneity across neural populations.

Broadly, interpretability methods can be categorized into two groups [31]: explainable and interpretable deep learning. The former, also called *mechanistically interpretable* deep learning, develops interpretability methods to explain black-box models. For example, in computer vision, saliency maps are constructed to highlight input image pixels that are discriminative with respect to an output decision of a deep neural network [32, 33]. A more generalizable example is Local Interpretable Model-Agnostic Explanations (LIME), a framework for explaining predictions of any black-box model by learning a locally-interpretable model around the prediction of interest [34]. However, this class of models does not make the neural network interpretable in and of itself: the model tries to explain what the network does. First, this means that there is no direct mapping from the embedding to the data: for instance, the explainable model might conclude that the network is optimizing for a feature that is simply correlated with the learning objective of the network, missing the true understanding of the "black-box" system [35]. Second, this approach does not guide the neural network architecture to learn useful representations. That is, the network may perform discrimination based on non-generalizable spurious features. Many of the current methods used in neuroscience fall in this category, and recent work has successfully gained mechanistic insights into neural circuit computations using this approach. Still, such analysis was achieved from a posteriori interpretation and manual tweaking of the network architecture [36]).

In contrast, model-based interpretable deep learning [37] (Figure 1a) is an emerging technique to design deep neural networks that are inherently interpretable. In particular, algorithm unrolling [38], a sub-category of interpretable deep learning, offers deep neural networks whose weights and representations can be directly interpreted as parameters and variables of an underlying generative model [38, 39]. This one-to-one mapping between the neural weights and latent representations of a generative model introduces interpretability. These mappings can be learned using an iterative algorithm optimizing the model [39–41]. Importantly, this generative model does not require detailed assumptions about the data: it provides domain knowledge infor-

mation, without restricting the model's output in such a way that important features of the data would be missed. Following seminal work in algorithm unrolling [39], numerous applications have been developed across several fields, including computational imaging (e.g., super-resolution [42] and image deblurring [43]), medical imaging [44, 45], identification of dynamical systems [46], remote sensing applications (e.g., radar imaging [47]) or source separation in speech processing [48].

Here, we propose a novel framework combining algorithm unrolling with convolutional sparse coding (i.e., dictionary learning), called Deconvolutional Unrolled Neural Learning (DUNL), that fulfills all the above-listed desiderata (Figure 1a). Our method offers a flexible framework to deconvolve single-trial neuronal activity into interpretable and local components. Our source code is flexible, easy to use, and adaptable to various applications by simple modifications of neural network non-linearities and training loss functions, without requiring the user to re-derive an optimization algorithm for their specific application.

To demonstrate the versatility and usefulness of DUNL, we apply it to the deconvolution of neural signals acquired in a wide range of experimental conditions. First, we show that it can deconvolve the salience and reward prediction error (RPE) components of naturally multiplexed reward signals encoded by dopamine neurons in the midbrain. Second, we demonstrate that it can deconvolve cue and outcome components of slow calcium signals recorded from dopamine neurons during associative learning. Third, we show simultaneous event detection and characterization of neural activity from the thalamus in a high signal-to-noise ratio (SNR) setting. Fourth, we demonstrate that in a low SNR setting, we can extract classes of neural responses from the piriform cortex in the presence of random and overlapping odor pulses. Finally, we perform model characterization and compare it to other decomposition methods to show how local interpretability in a limited data regime is an important feature of our deconvolution method.

Results

Sparse deconvolutional learning uncovers structure in single-trial, single-neuron activity

We aim to decompose single-trial neural activity into local impulse-like responses to sparse yet recurring events. We assume that the observed neural activity is the result of a combination of recurring components–kernels of a "dictionary"–whose timing and magnitudes can vary on an event-by-event basis. Thus, we seek to obtain a reconstruction of the neural data by optimizing the model that generates these components or kernels. To achieve this, the neural activity is modeled as a sum of convolutions between these kernels, and their timing and magnitude in response to recurring sparse events. We refer to the vector representing the timing of events and the strength of neural response as a sparse code. Stochasticity in the estimated activity is added by passing this convolved signal through a generative model using a probability distribution of the natural exponential family (e.g., Gaussian, Binomial, and Poisson).

More specifically, we model (Figure 1b) the observations $\boldsymbol{y}^{n,j}$ from neuron n at trial j using the natural exponential family [49, 50] (e.g., Binomial or Poisson for spiking and Gaussian for calcium signals) with distribution mean of $\boldsymbol{\mu}^{n,j}$. We impose a generative model on the nth neuron's mean activity at trial j, $\boldsymbol{\mu}^{n,j}$, and express it as the convolution of K localized kernels $\{\boldsymbol{h}_k^n\}_{k=1}^K$ and sparse codes (representations) $\{\boldsymbol{x}_k^{n,j}\}_{k=1}^K$, along with a background, baseline, measured activity level $a^{n,j}$ (Figure 1b). The convolutional structure enables the identification of local patterns occurring across time. Kernels and codes are interpretable in the following sense. Kernels capture characteristics shared among trials (or neural population, depending on the model design): they characterize the neuron's response to time-sensitive sparse events/stimuli. The nonzero entries of the sparse latent representation $\boldsymbol{x}_k^{n,j}$ represent the time when the event associated with the kernel k occurs in trial j; their amplitude captures the strength of the neural response. In relation to the functional identification of a system, this model characterizes the system in terms of cause-effect relationship: the code captures the timing of a stimulus applied locally in time [51]; the kernel captures the

impulse response of the neuron, whose dynamics are modeled through resistor-capacitor (RC) differential equations [52].

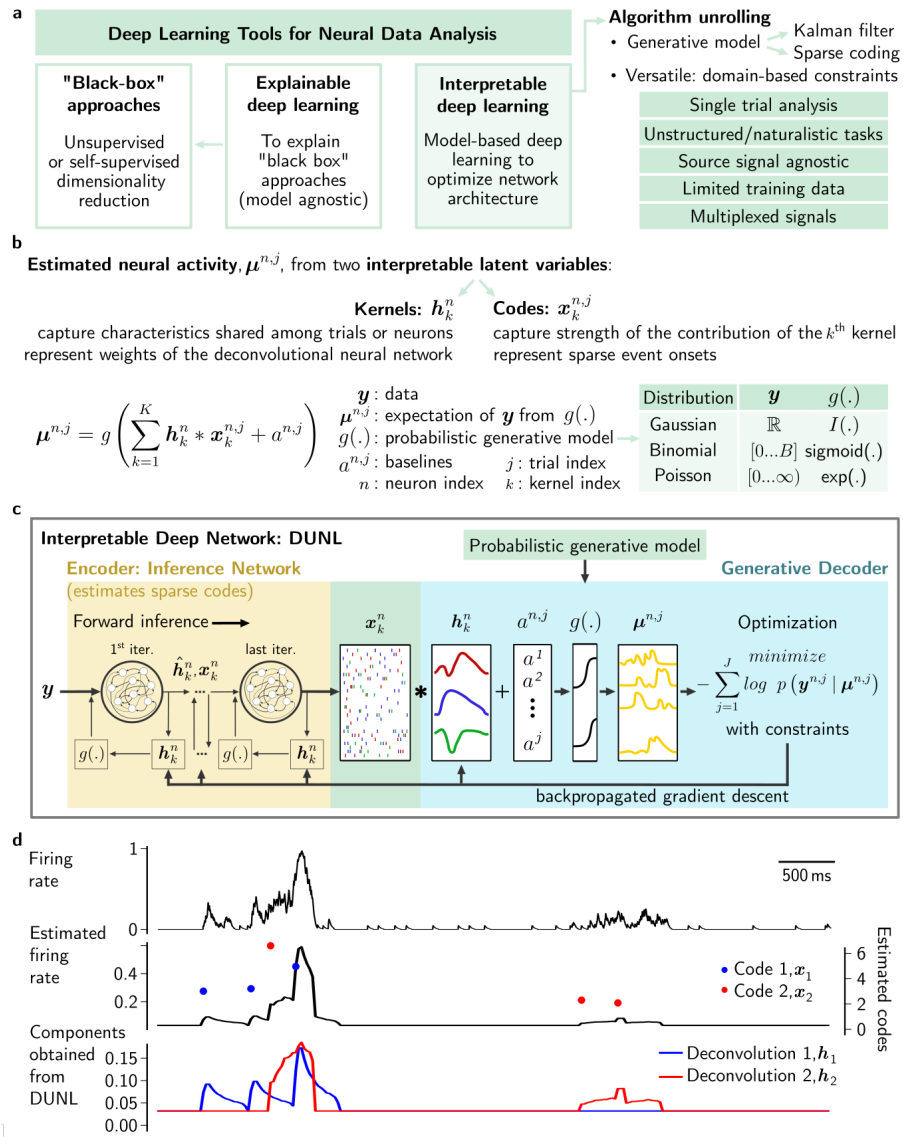


Figure 1. Interpretable deep learning with deconvolutional unrolled neural learning: DUNL. a, Categorization of deep learning tools developed for neural data analysis and advantages of using algorithm unrolling. **b**, Generative model used by DUNL to estimate neural activity as a function of the sum of convolution between kernels and sparse codes. **c**, Schematic representation of DUNL: the deep inference network, whose weights are the estimated kernels, estimates the sparse codes used as an input to the generative decoder. The output of this decoder is used to optimize the network. **d**, Demonstration of DUNL's ability to deconvolve events from unstructured single-trials, where two recurrent events occur locally at random times and with varying amplitudes.

Thus, the kernels in the model are learned fully from data, i.e., they do not obey a user-specified parametric form, and the codes are sparse in time. We learn the kernels and codes by minimizing the negative data log-likelihood $\sum_{j=1}^{J} \log p(\boldsymbol{y}^{n,j} \mid \boldsymbol{\mu}^{n,j})$ regularized/penalized by terms encouraging desired properties on the codes and kernels. We impose a sparsity prior, to promote a few code activations in time, and an optional second-order covariance structure on the codes to cap-

ture dependencies among kernels (e.g., discouraging activation of two event types simultaneously). Where needed, we apply smoothing regularization on the kernels [53, 54].

We map the optimization into an encoder/decoder neural architecture following the algorithm unrolling approach [38] (Figure 1c). We call this framework Deconvolutional Unrolled Neural Learning (DUNL), an application of algorithm unrolling to convolutional dictionary learning [55–57]. The encoder is a deep-structured recurrent convolutional neural network. Unlike sequential deep encoder approaches, this encoder shares the same parameters as the generative model/decoder. The encoder takes single neuron single-trial observation $\boldsymbol{y}^{n,j}$ as input and encodes a set of sparse representations $\{\boldsymbol{x}_k^{n,j}\}_{k=1}^K$. As explained above, this latent code corresponds to event/stimuli onsets and the strength of neural response to the event. The decoder is a shallow network based on the proposed generative model. This decoder maps the estimated time series of sparse representations into a time-series estimate of the mean neural activity. Both the encoder and decoder are characterized by the kernels $\{\boldsymbol{h}_k^n\}_{k=1}^K$ from the generative model (i.e., kernels are weights of the artificial deep neural network and can be trained by backpropagation). Training DUNL involves both a forward pass (inference for codes) and a backward pass (training to learn the kernels, see Supplementary Methods), both of which are parallelizable over neurons and trials.

To demonstrate the applicability of DUNL, we start by applying it to synthetic spiking data. The experiment consists of two event types characterized by local kernels. Within one trial, the events happen 3 times uniformly at random. In this unstructured experiment, the goal is to recover the underlying kernels, as well as the timing and magnitude of the events associated with them, independently of whether these are single events or composed events (superposition of more than one kernel). DUNL successfully decomposes the synthetic neural data into kernels and codes (Figure 1d), and it achieves so in a data-limited regime (see following sections).

To summarize, we introduce a novel framework to recover the statistics of time series data as a sparse superposition of kernels (Figure 1d), that is akin to a convolutional generalization of Generalized Linear Models (GLMs), in which both covariates and kernels are learnable, contrary to GLMs in which the kernels are user-defined and fixed [58]. Importantly, our method outputs a response amplitude for each individual occurrence of an event, a feature that is absent from other encoding methods.

DUNL uncovers salience and value signals from single dopamine neurons

We first apply DUNL to deconvolve multiplexed signals in the responses of dopamine neurons. The activity of dopamine neurons in the midbrain has long been an interest of neuroscientists, both in fundamental and clinical research, given their involvement in motivated behaviours, learning, and multiple physiological functions. A subset of these neurons located in the Ventral Tegmental Area (VTA) has been described as encoding a reward prediction error (RPE) from temporal difference (TD) reinforcement learning algorithms [59–64]. This computation requires the neural representation of the value of rewards in the environment: a transient positive RPE response signals an unexpected increase in the value of the environment. However, reward is a subjective quantity that is non-linearly modulated along multiple dimensions of reward (e.g., probability, size, etc.), and it has been suggested that the reward responses of dopamine neurons multiplex two sequential and overlapping signals [65], the first one carrying information related to the salience of the reward and the second one carrying subjective value information, or utility, of the reward [12]. This distinction is important from a computational point of view, because only the value-like component matches the reward prediction error signal driving learning in TD algorithms.

However, in practice, most studies of dopamine neurons ignore this potential multiplexing by averaging dopamine responses over a single time window following reward delivery [66, 67], or, at best, apply user-defined ad-hoc windows to try to isolate these two contributions [68]. We used DUNL to find, in a data-driven manner, whether the reward responses of dopamine neurons can indeed be decomposed into two components and whether these are differently modulated by reward value.

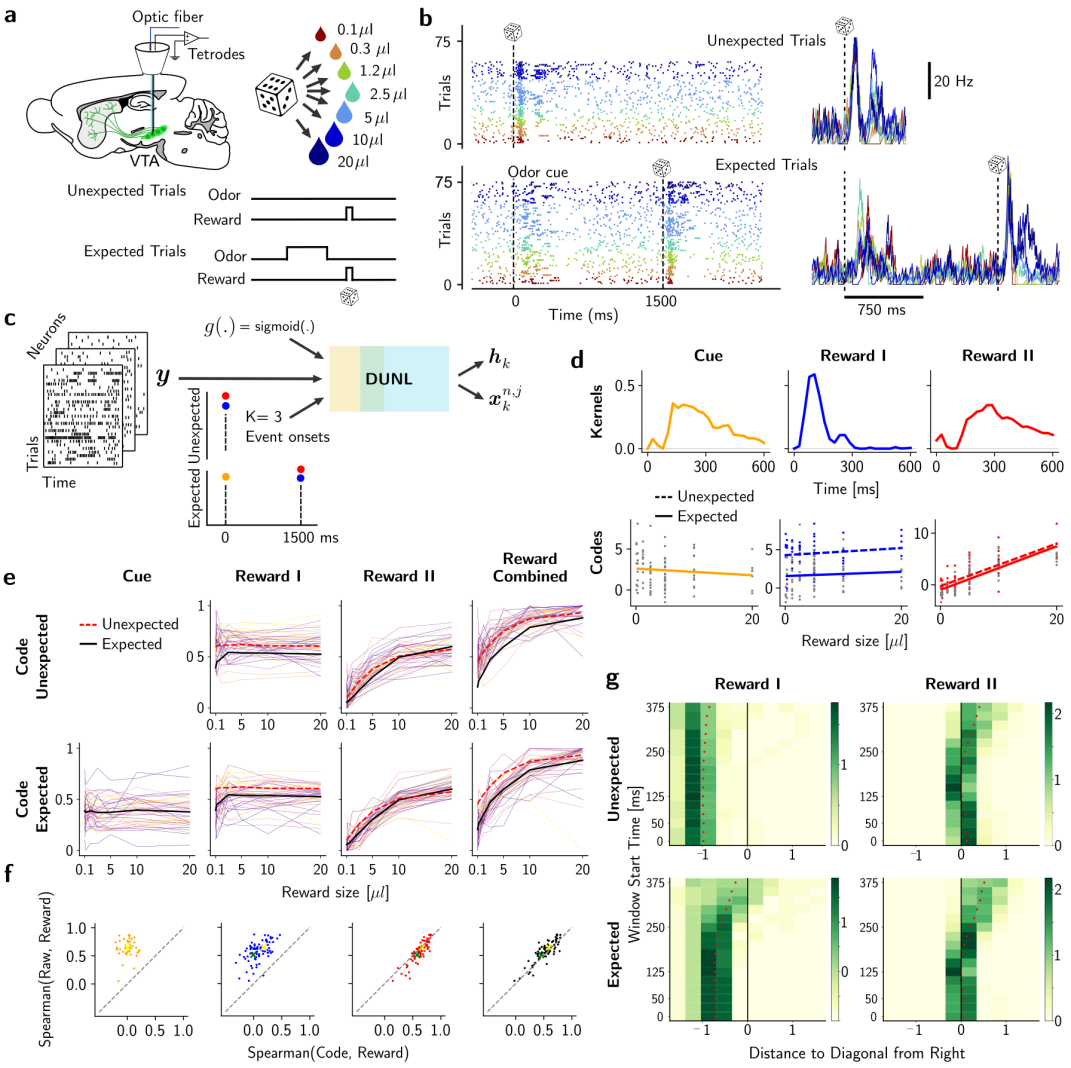


Figure 2. DUNL uncovers salience and value signals from single dopamine neurons' reward responses. **a**, Experimental setup showing optical fiber and tetrode recordings on a sagittal slice of the mouse brain (top left), distribution of reward sizes (top right), and task structure (bottom). **b**, Raster plot (each dot is a spike) from one neuron (left) and corresponding firing rate averaged across trials of the same reward size. **c**, Representation of the input information used to run DUNL in this dataset: neuron activity across time and trials, timing of stimuli, number of kernels to learn, probabilistic generative model. **d**, Learned kernels shared across neurons (top) and inferred code amplitudes for one example neuron (bottom, each dot responds to a single-trial code, the line is the linear regression of the codes over reward sizes). **e**, Diversity of neural encodings (code amplitudes) as a function of reward size for expected and unexpected trials; each line represents one neuron; the black line is the average for expected trials, and the red dashed line is the average for unexpected trials. The lines are normalized per neuron, and the normalization constants are shared across trial types and codes. For non-normalized curves, see Figure S3. **f**, Spearman's rank correlation between codes and reward size vs. the windowed averaged firing rates and reward size within the full 600 ms window. The alignment of the red dots (Reward II) under the diagonal line illustrates that the value-like code is more informative about the reward size (each neuron is represented by two dots (expected and unexpected); the average of all neurons is shown by the marker x (t-test: $p = 0.050$ expected (yellow), $p = 6.19 * 10^{-5}$ unexpected (green)). **g**, Mapping of the Spearman's correlation (its distance from the diagonal) as a function of the window start time for the windowing method. The positive distance corresponds to below the diagonal. Colorbar: normalized probability density function at each bin, such that the integral over the shown range in the x-axis is 1. For experiment results on limited data ($< 8\%$ of current analyses data) see Supplementary materials (Figure S4).

We used electrophysiological data from 40 optogenetically identified dopamine neurons [66, 67] recorded in mice performing a classical conditioning task as part of a previous study [69] (Figure 2a and see Methods). In *"Unexpected"* trials, a reward of varying size (i.e., 0.1 to 20 μl) was delivered without a cue, and in *"Expected"* trials, an odor cue preceded reward delivery by 1.5 s (Figure 2a,b). Although the cue predicted the timing of the reward, it provided no information about its magnitude.

We modeled the data with three non-negative kernels: one to characterize the response to the odor cue, and another two for the reward event (Figure 2c) [12]. DUNL was provided with the timing of the cue and reward events but not the trial types (reward amounts). The goal is to recover the generating kernels, associated with the cue and reward events, given only raw spiking data and the timing of these events. We defined DUNL's inputs such that the kernels would be shared across the population of neurons, but the codes would be individualized for each neuron in single trials (Figure 2c), such that each neuron is characterized by its own decomposition of its estimated firing rate (see example neuron decomposition in Figure S2).

DUNL's output showed that, as expected, the magnitude of the associated code obtained using the kernel for cue responses is essentially invariant to the reward size. More importantly, although we did not instruct DUNL to retrieve salience and value-related components separately, DUNL obtained two reward-related kernels which can be characterized as responding to salience (blue, Reward I) and value (red, Reward II) (Figure 2d). In fact, when we plot the code values for each kernel as a function of reward size we observe that codes corresponding to salience (blue, Reward I) are modulated by expectation (unexpected vs. expected), but almost invariant to the reward size, and codes corresponding to the value (red, Reward II) are strongly positively correlated with reward size, both for individual neurons and across the population average (Figure 2e, and Figure S3 for non-normalized data). In fact, the value code carries more information about the reward size than the firing rates over a traditional ad-hoc window (Figure 2f). Furthermore, combining the two reward kernels (Reward I and II) does not improve the information about the reward size, indicating that the salience-like code does not contribute to value information. We also found that as the ad-hoc window shrinks to exclude the first spike(s) traditionally attributed to salience, the ad-hoc window method improves in the representation of reward size (for Reward II, the best ad-hoc window approximately excludes the first 125 (expected) and 150 (unexpected) ms of data from the reward onset). Still, DUNL's code is more informative of the reward value (Figure 2g).

DUNL's successful decomposition of neural responses to the reward, as opposed to spike counts from ad-hoc windows, indicates that the code amplitudes in single trials from the value kernel is a powerful measure of the neurons' tuning to reward size. Importantly, we also showed that DUNL can successfully perform similar learning/inference in a data-limited regime ($< 8\%$ of the current analyses data, Figure S4). To quantify the quality of our decomposition as a function of the number of trials used for training, we simulated dopamine neurons in the same experimental settings. We found that in our simulated dopamine data, we could recover well-fitted kernels with as little as 14 trials Figure S5). In summary, we showed that DUNL can discover two components in the reward responses of dopamine neurons in a systematic, data-driven approach, recovering a first component that is not modulated by reward size, while the second component is. We note that although the choice of the number of kernels, in this case two for reward events, is a hyperparameter to set a priori, it can be tuned using validation sets. Overall, DUNL will empower future studies to precisely quantify the value-like component as the reward prediction error response of single dopamine neurons in an unbiased manner.

DUNL deconvolves cue, salience, and value signals from single dopamine neurons in two-photon calcium recordings

To demonstrate DUNL's flexibility and applicability to other data modalities beyond spike trains, we next applied DUNL to two-photon calcium imaging data [64, 70]. To this goal, we recorded the activity of 56 dopamine neurons in mice using two-photon calcium imaging with a gradient

refractive index (GRIN) lens (Figure 3a) in a classical conditioning task with the same structure as in the above experiment [67, 69], but with a longer delay between the cue and the reward delivery. In unexpected trials, rewards of different sizes (i.e., 0.3 to 11 μl) were delivered once at a random time. In expected trials, an odor cue was delivered 3 s prior to the reward delivery. There is diversity in the responses of neurons to the cue and the multiple reward deliveries and, in general, we see modulation by expectation and reward size (Figure 3b).

We characterized the neural activity using four kernels (Figure 3c), as follows. The response to the odor cue in expected trials was characterized by one kernel (orange), and the reward response (at the reward onsets) in both unexpected and expected trials was modeled by three kernels: the blue kernel can be freely active with a positive code, while the red/green kernels were positive and their codes were positive and negative, respectively. We coupled red and green kernels such that only one of them is encouraged to be active on each trial. To achieve this, we used structured representation learning (see Methods Section in Supplementary Materials). This structural regularization is motivated to capture the different response dynamics of the calcium signal to increases versus decreases in the underlying firing of the neurons due to the different onset and offset dynamics of the sensor. DUNL's output shows that the blue reward kernel resembles the salience response, and the red/green reward coupled kernels resemble the value response (Figure 3d). The inferred single-trial codes from a single neuron (Figure 3e) and across the population (Figure 3f) show that the salience-like kernel (Reward I) is almost invariant to reward size, while the combination of the value-components kernels (Reward II Coupled) correlates positively with reward size.

To understand this choice of kernel characterization, we looked into the interactions of the chosen kernels in the decomposition of the raw data (Figure S6). In this dataset, we observed that many neurons lack an obvious salience-like response (i.e., an early transient increase of the neural activity that is invariant to the reward size), probably because the cue-related calcium signal has not yet decayed to the baseline, potentially masking the salience response. Due to the calcium sensor's faster onset than offset dynamics, we observed faster salience-contaminated positive responses for high-reward trials, and a very slow negative response for low-reward trials. Given this different temporal dynamics of positive and negative signals, the decomposition of reward signals into only two kernels (salience and value-like) would result in a combination of salience and valence information for both kernels, such that both kernels would be correlated with the reward size.

We computed the Spearman correlation between the Cue code, the Reward I code (blue), the Reward II coupled (red + green) code, as well as all the reward codes combined (blue + red + green) with the reward size. These correlation values were then compared to the ad-hoc approach where the correlation was computed using the 4 s windowed averaged activity at the reward onset. This analysis showed that only when all reward codes (salience-like + value-like) are combined, the codes become more informative of the reward size than the ad-hoc windowing approach (the distribution of points is below the identity line, Figure 3g). This can also be noticed in the average population activity (Figure 3f and Figure S7). Regardless of the window size used for computing the value component of the reward response in traditional approaches, the Reward Combined code is significantly more informative of the reward size than the windowing approach in both unexpected and expected trials (Figure 3h). We attribute this success to the denoising capability of DUNL: it performs deconvolution of the cue response from the reward response, which is important in these slow calcium signals. For further discussion on the limited temporal resolution of these data and the recovery capability of DUNL.

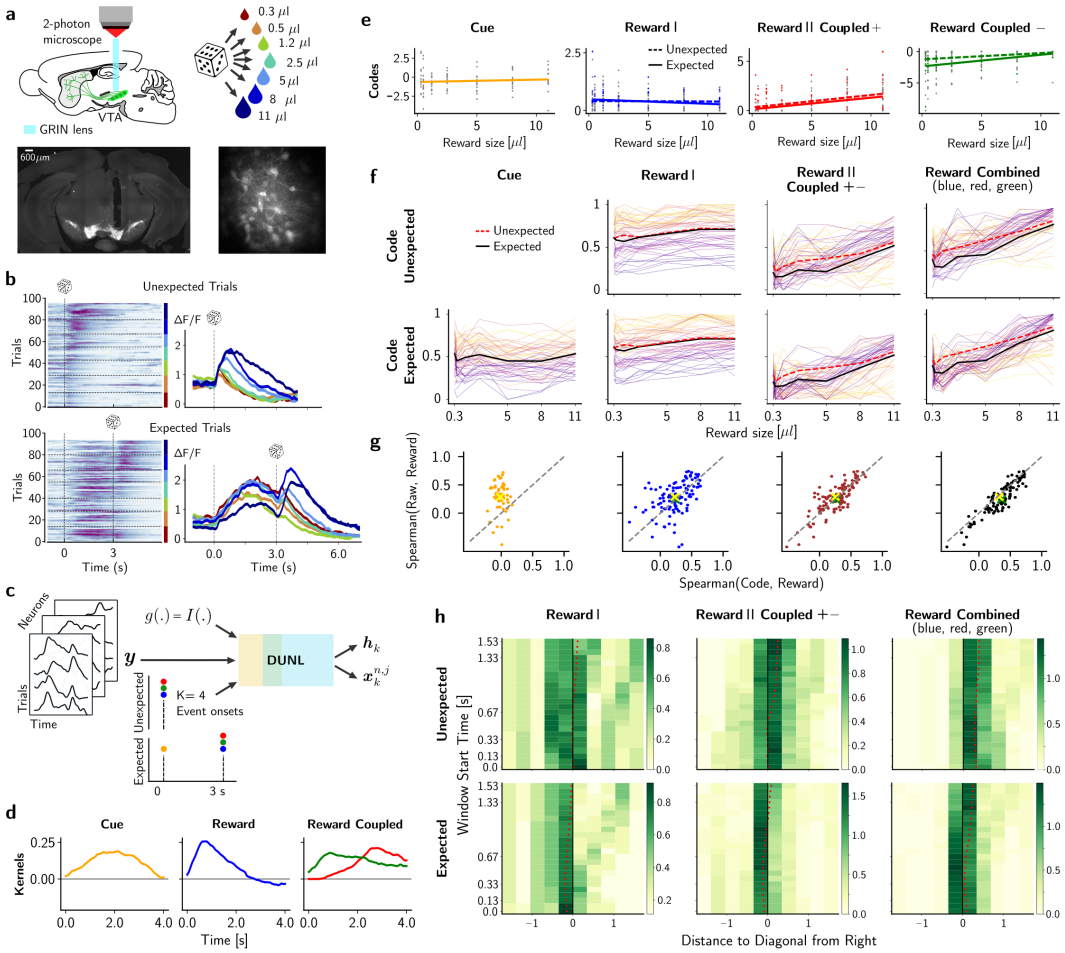


Figure 3. Deconvolution of cue, salience, and value components from dopamine calcium data. a, Experimental setup depicted on a sagittal slice of the mouse brain with dopamine neurons represented in green (top) and histology images showing dopamine neurons expressing the fluorescent calcium indicator GCaMP6m (bottom): coronal slice of the mouse brain showing GRIN lens track over the VTA (left, scalebar: 600μm), and projection image of the field of view obtained during an experimental acquisition under the two-photon microscope showing individual VTA dopamine neurons (right). **b**, Left: Heatmap of time-aligned trials at the reward onset. The trials are ordered from low to high reward size, with a horizontal line separating the different trial types. Right: Averaged time-aligned activity of an example neuron for each reward size. **c**, Inputs used to run DUNL in this dataset: calcium activity across time and trials, timing of stimuli, number of kernels to learn, and probabilistic generative model. **d**, Kernel characterization of cue and reward events. Three kernels were used to estimate the reward response: one for salience (blue) and two non-concurrent kernels for positive or negative value (green and red). **e**, Code amplitude as a function of the reward size for an example neuron (each dot corresponds to the code inferred from single-trial neural activity, these values are fitted by linear regression). **f**, Diversity of neural encodings as a function of reward size for unexpected (top) and expected trials (bottom): each line represents one neuron, the black line shows the average for expected trials, and the red dashed line average for unexpected trials. Activity is normalized per neuron and across trial types, and codes for comparison across subfigures. **g**, Spearman correlation of the codes (x-axis) and the windowed average activity of 4 seconds (y-axis) with respect to the reward sizes: each dot represents one neuron and the average across all neurons are shown by yellow (expected) and green (unexpected) 'x' marker (Reward Combined has $p = 0.008$, and $p = 3.468 * 10^{-9}$ t-test, respectively). The third panel (brown) from the left combines the code from Reward Coupled kernels (positive and negative, depending on the trial). The right panel combines all the reward-related codes (salience-like Reward and value-like Reward-Coupled). **h**, Heatmap of the distance of the yellow (unexpected) and green (expected) 'x' marker in **f**, from the diagonal as a measure of the increased Spearman's correlation between codes and reward size, as the interval chosen for the ad-hoc window is modified: it shrinks from the bottom to the top of the y-axis to gradually exclude the early activities after the onset. Positive values are located below the diagonal. On the right panel, the marker is closest to the diagonal when 0.4 s of activity at the reward onset is excluded in the ad-hoc window approach. Colorbar: normalized probability density function at each bin, such that the integral over each line in the x-axis is 1.

DUNL demonstrates modulation of somatosensory thalamus by whisker velocity using unsupervised simultaneous onset detection and kernel characterization

Owing to its algorithm unrolling foundation, DUNL is a very versatile framework, whose inputs can be adjusted according to the application. In our previous examples, we provided DUNL with the expected number of kernels and expected times of events to guide the learning process. However, this information might be completely omitted, and we can use DUNL to perform simultaneous onset detection and learn local kernels in an unsupervised manner. This approach will be more successful in a high signal-to-noise ratio (SNR) setting.

To demonstrate this, we applied DUNL to electrophysiological recordings from the somatosensory thalamus of rats recorded in response to periodic whisker deflections. The whisker position was controlled by a piezoelectric stimulator using an ideal position waveform [71]. The experiment was designed with trials starting/ending with a 500 ms baseline; in the middle 2000 ms, 16 deflections of the principal whisker were applied, each with a period of 125 ms (Figure 4a). We considered the whisker position to be the stimulus and attributed a particular phase of the whisker position as the event of interest to detect. The goal is to detect the onset of the events, and characterize the neural response to the stimulus using one kernel. In this experiment, onsets of events were unknown, and DUNL looks for up to 18 events in each trial (Figure 4b; the additional 2 events were to adjust for unknown activities outside the known 16 deflection stimuli, e.g., see Figure 4g bottom)). We refer the reader to the Methods section and Table S4 for more information on the unsupervised DUNL method and training.

We divided the data in a training and test set to show that DUNL simultaneously characterizes the shape of neural spiking modulation (Figure 4d,e) and infers (detects) the event onsets at single-trial level (Figure 4g). The learned kernel suggests that the measured neurons encode, and are modulated by, the whisker velocity, a feature found by prior work [72, 73]. Moreover, unlike prior work analyzing these data by averaging the time-aligned trials [49], DUNL does inference on single-trial data (Figure 4g top) with bin spike counts of only 5 ms. The heterogeneity of the inferred code amplitudes (Figure 4g bottom) is indicative of the intrinsic variability of the neural response to the stimulus. This feature is absent in previously published GLM analyses [72, 73], which assume the neural responses are constant across deflections. Figure 4d shows the reconstructed average firing rate and the peristimulus time histogram for one neuron. For event detection, we showed that DUNL performs significantly better than a peak-finding algorithm (applied on the smoothed raster plot) (Figure 4f). This experiment highlights the ability of DUNL to detect event onset while simultaneously characterizing the neural response to the event; this event detection feature is absent in prior GLM frameworks [58].

Characterization of single neurons in an unstructured olfactory experiment using DUNL

Finally, we highlight how DUNL can be used for exploratory data analysis. We applied DUNL to electrophysiological data recorded from the piriform cortex of mice engaged in an olfactory task in which short odor pulses occur at random times across trials, mimicking the statistics of natural odor plumes [74]. In each trial, 50 ms Gamma-distributed odor pulses were delivered. We recorded and isolated 770 neurons from mice's anterior piriform cortex (Figure 5a,b, details of data acquisition and scientific results on this data will be reported fully in another publication). The structure of piriform cortex neural responses to sequences of odor pulses are largely unexplored, and here we use DUNL to characterize them. To model neural responses, we aligned the non-zero elements of the sparse code to the timing of the odor pulses, and spike counts were modeled with a Poisson process (Figure 5c). Each neuron was characterized by one kernel. We learned both the kernels and the code amplitudes for all recorded neurons. Using k-means clustering, we identified 4 clusters for the kernel shapes detected in the population (Figure 5d,e). Three of these neural populations correspond to neurons whose activity increases following an odor pulse, albeit with different dy-

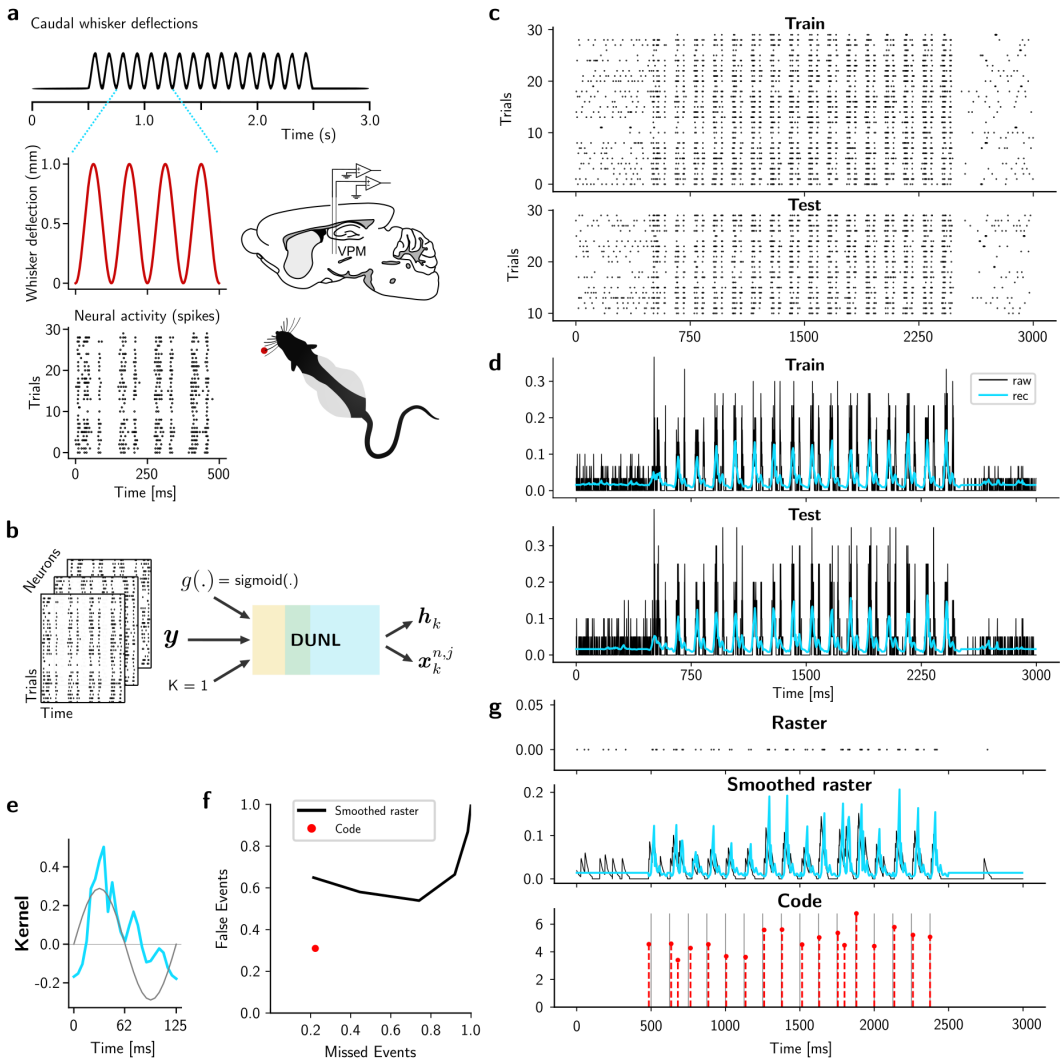


Figure 4. Event detection with DUNL for analysis of spiking data from the somatosensory thalamus. a, Experimental setup [71]: periodic whisker deflections of constant velocity are imposed in each trial (top), resulting in phase-locked neural activity (bottom left) in the VPM region of the thalamus in anesthetized rats (bottom right). **b**, DUNL setup to detect one kernel across the entire population. **c**, Raster plot from one neuron (both train and test trials). Each trial starts/end with 500 ms baseline period; 16 deflections with period of 125 ms are applied to the whisker of the rat for a total of 2000 ms. **d**, Peristimulus time histogram from one neuron (black) and DUNL estimate of the firing rates (blue). **e**, Kernel characterization of the whisker motion (blue). The gray sinusoid is the first derivative of stimulus motion. **f**, Quantification of the miss/false events detected by DUNL. The red dot represents the performance of DUNL when events are detected on single-trials. The black curve shows the performance of a peak-finding algorithm on the smoothed spike trains for a range of thresholds. We used a tolerance of 10 ms (2-time bins) while computing the false/miss events. **g**, spikes in one example trial (top), smoothed spike rate in black, and spike rate estimation in blue (middle), with the inferred code on the detection of 16+2 events in time (bottom). (For more information on the analyzed neurons and stimuli in relation to the original paper collecting the data [71], see supplementary materials).

namics, while the other cluster corresponds to neurons whose activity is inhibited by the olfactory pulses. One can complement this exploratory data analysis using a different number of clusters (Figure S8).

This application demonstrates how any type and shape of kernels can be learned by DUNL, without any assumptions guiding the shape of the kernels. Thus, DUNL can capture a diversity

that may not be recoverable when using a hand-crafted family-of-basis, highlighting the value of non-parametric temporal characterization of neural responses [75].

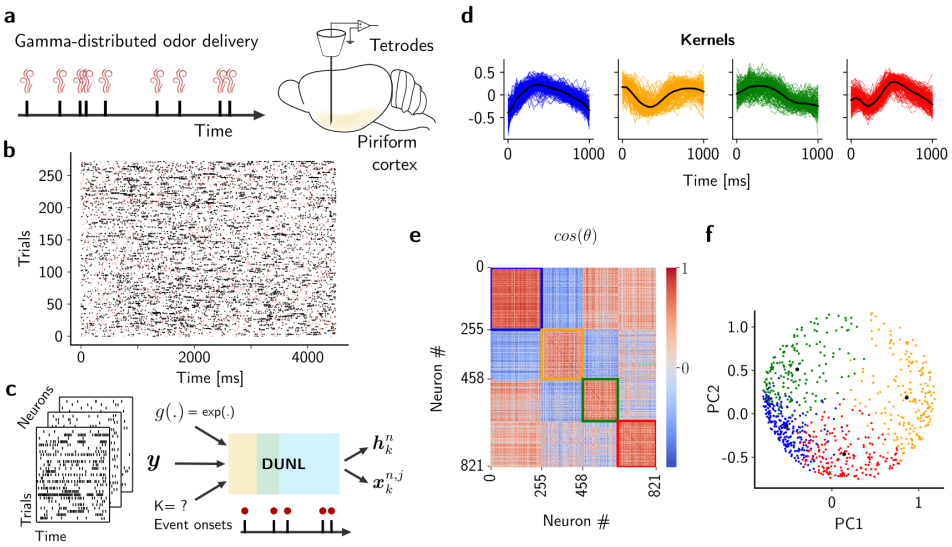


Figure 5. Characterizing the structure of piriform cortex responses with DUNL. a, The experimental task. **b**, Raster plot from one neuron in an unstructured trial-based experiment. The red dots denote 50 ms Gamma-distributed odor pulses, and the black dots represent single spikes. **c**, Schematic of the analysis using DUNL. **d**, Kernel characterization of single neurons. The kernels (zero-mean, normalized) are shown for the four clusters. For K-means with 3 and 5 clusters, see Figure S8. **e**, The cosine similarity matrix corresponding to the clusters of learned kernels. **f**, Visualization of the four clusters of kernels using Principal Component Analysis (principal components 1 and 2). See also Supplementary materials.

Model characterization

To assess the reliability of the results reported here and guide DUNL's end users, we characterized the performance of DUNL on a wide range of simulated data, focusing on the spiking-data modality. The section includes two distinct simulation studies.

Simulation study I: data generation. In scenario I, we focused on a setting where a neuron responds to two different types of events, characterized by two distinct kernels of length 400 ms. In each trial, 3 different events from each kernel can occur. They are unstructured, such that event onsets are chosen uniformly at random, with a minimum distance of 200 ms between two events of the same type. However, events of different types can occur simultaneously, thus convolving their activity (Figure 6a, blue and red events). The strength of the neural responses of the neuron were generated by the Gaussian distribution with mean 50 and variance of 2 for blue events and with mean 55 and variance of 2 for red events. The baseline firing rate was chosen to be 8 Hz.

Simulation study I: fitting using DUNL. We trained DUNL with these synthetic data using bin size resolution of 25 ms while the number of trials available for training varies from 25 to 1600 (results in Figure 6b and Figure S9 are from a test set with 500 trials). The number of events in each trial was known, but the timing of the events was unknown to DUNL. DUNL estimated the firing rate of the neuron and deconvolved it into two components, corresponding to each event type. Moreover, the magnitude of the sparse codes inferred by DUNL encoded the local activity of each event (kernel) within the trial (Figure 6b). Lastly, the result held with small kernel recovery error in the limited data regime, i.e., 25 training trials (Figure 6c,d).

Simulation study II: data generation. In this scenario, we restricted ourselves to the setting of a single neuron and a single event to assess how well DUNL can learn the kernel associated with this neuron, as well as its codes. This model characterization empowers end users to assess the reliability of DUNL based on the statistics of their data. We evaluated the performance of DUNL as

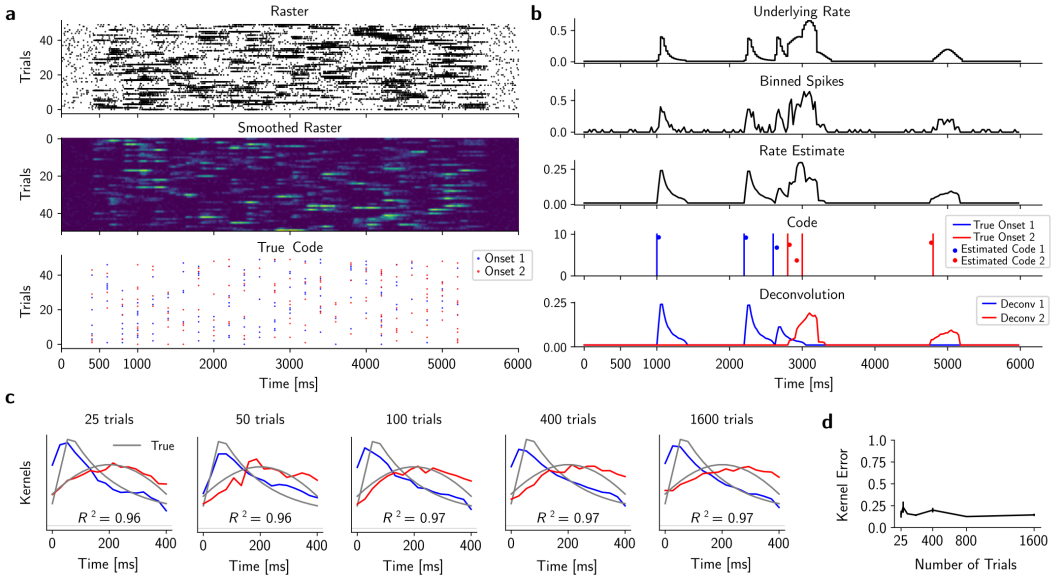


Figure 6. DUNL model characterization with two kernels in an unstructured experiment. a, Test trials for this unstructured synthetic experiment: raster plot of spiking events (top), smoothed raster (middle), and true event onsets (bottom). **b**, Example trial showing original firing rate (top), estimated firing rate by DUNL (middle), estimated codes and kernels (bottom). **c**, Estimated kernels for 25, 50, 100, 400, and 1600 training trials. Gray traces represent the true kernels used to generate the data. The test R^2 fit score, evaluated on the binned spikes, is shown for each training size. **d**, Kernel recovery error (i.e., $\sqrt{1-(\text{cosine similarity})^2}$, as a function of number of training trials. Specifically, we compute cross-correlation to compute the cosine similarity to account for learning shifted kernels. See Figure S9 for more single trials and decomposition examples. The event detection hit rate is, on average, 80.22% and 80.68% for the smallest and largest training datasets with a tolerance of 2 time bins (with a tolerance of 3 time bins, the hit rate is increased to 91.58% and 93.34%, respectively. We report hit rate with tolerance to account for learned shifted kernels (see the Supplementary Methods for more info on shift/delay property of DUNL).

a function of the background firing rate (i.e., $2, 5, 8, 11, 14$, and 17 Hz), the bin-size the model uses to count spikes (i.e., $5, 10, 25$, and 50 ms) (Figure S10), and the number of trials (i.e., $10, 25, 250, 500, 1000$), available to learn the model parameters (i.e., kernels) (Figure S11).

We simulated multiple trials of activity, a subset of which we used for training, and the other for testing. Each trial was 4000 ms long with 1 ms resolution. In each trial, 5 similar events happen uniformly at random with a minimum distance of 200 ms. We assumed the neural response to the event is 500 ms long. We modeled the strength of the neural response using a Gaussian distributed code amplitude of mean 30 with variance 2. Given the code and kernel, the firing rate of the neuron was constructed based on the DUNL generative model using the Binomial distribution. The test set consisted of 100 trials following similar statistics to the training set. For a low background firing rate, a few spikes were observed in each trial (e.g., for 2 Hz, only 29 spikes were observed in one trial, whereas for 17 Hz firing rate, 202 spikes were observed on average in each trial). Hence, learning and inference were challenging when the neuron was very silent.

Simulation study II: fitting using DUNL. We considered two scenarios: a) known timing of events (known support), and b) unknown timing of events with a known number of events (unknown support)(Figure S12). The dopamine (Figures 2 and 3) and olfactory (Figure 5) experiments from earlier sections correspond to *known support* scenario, and the whisker deflection (Figure 4) experiment corresponds to *unknown support*. When the onsets were known, the inference was reduced to estimating the amplitude of the sparse codes and the training was for learning the kernel. When the onsets were unknown, the inference was more challenging: it involved estimating the event onsets in addition to the neural strength response (the code amplitude). In this case, the reliability of learning the kernel was entangled with the reliability of estimating the event onsets.

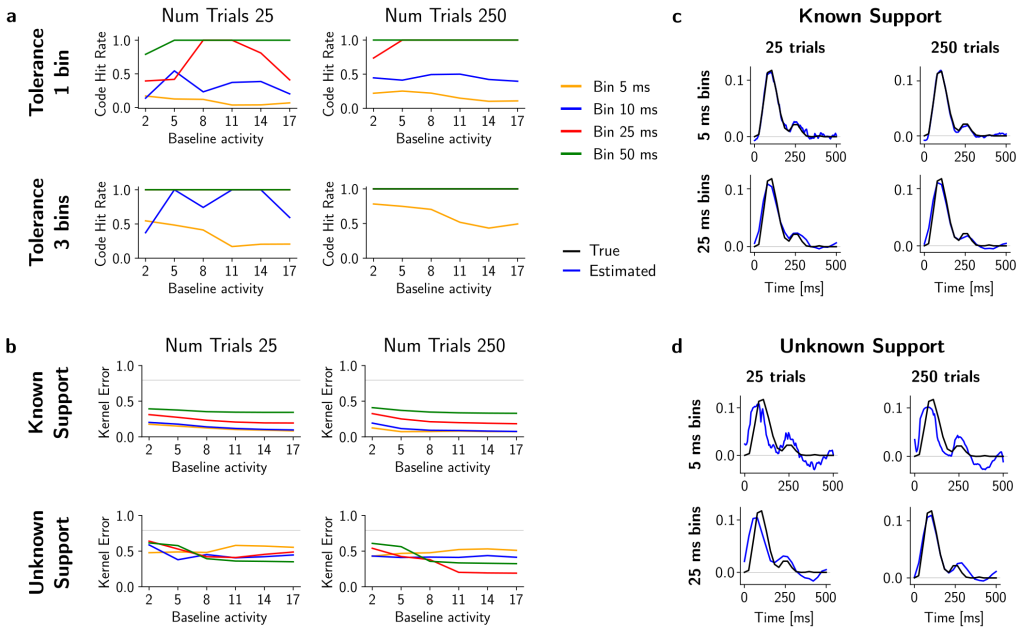


Figure 7. DUNL model characterization with a single kernel. The learning is evaluated through a kernel recovery error (i.e., $\sqrt{1-(\text{cosine similarity})^2}$, and code inference/recovery is evaluated through an event recovery error (i.e., $1-\frac{\text{\#identified events}}{\text{\# total events}}$) when events' timing are unknown. **a**, The hit rate (event onset detection) as a function of baseline activity. Results are shown for 25 and 250 training trials available, as a function of two different tolerance factors. For low tolerance on event detection, increasing the time bin resolution window improves the performance. **b**, Kernel recovery error when support is known (first row) and unknown (second row). For known support, smaller bin size results in better kernel recovery. The increase in kernel recovery error as the bin size increases is due to the decrease in resolution of the kernel. On the other hand, for unknown support scenarios, increasing the bin size is beneficial as it helps to have better event detection (shown in a). **c**, True (black) and learned (blue) kernels when event onsets are known. Kernels are shown for bin sizes of 5 and 25 ms, and 25 and 250 training trials. **d**, Similar to c but for the case when the onsets of events are unknown, hence estimated. See Figures S10 to S12 for more detailed info.

We showed that when the event onsets are known (Figure 7b known event onsets), DUNL's kernel recovery is relatively robust to the baseline firing rate and can successfully be achieved with few trials (e.g., 25 trials). In this setting, high-temporal resolution (e.g., 5 ms bin size) should be used, regardless of the size of the data. If data are very limited (e.g., 25 trials), increasing the bin size slightly (e.g., 5 ms to 10 ms) is important to implicitly learn a smoother kernel (Figure 7c) (we note that one can also tune the kernel smoothing hyperparameter in the DUNL training framework for better results with very limited data). When the event onsets are unknown (Figure 7b unknown event onsets), the bin-size imposes a limit on how well the kernel can be learned (Figure 7d). This challenge comes from the fact that the lower the bin size, the harder the event detection (Figure 7a). We recommend using as large as possible bin sizes that match a user's tolerance for event detection errors. In summary, the higher the number of trials, the higher the firing rate, and the larger the bin-size, the better DUNL's ability to learn kernels and infer event onsets. Our analyses can help practitioners explore in which regime their experimental data lies and assess which parameters of the model can be recovered from the data.

Comparison with other decomposition methods

DUNL is a versatile deconvolutional method that can extract directly interpretable latent representations from time-series data. Its main strengths are its ability to learn multiple local kernels within single trials, either in a supervised or unsupervised manner, and the capacity to do so in a limited

data regime. This is achieved through its learnable network architecture, implemented using algorithm unrolling. To emphasize how our framework fills a gap in the space of functionalities of other decomposition methods, we compared DUNL with other frameworks (baselines), namely: 1) dimensionality reduction methods, such as Principal Component Analysis (PCA), Non-negative Matrix Factorization (NMF), and trial-based Poisson GLM regression [58]; 2) the deep learning framework for latent factor analysis via dynamical systems (LFADS) [25]. We first performed comparison analysis with LFADS over full-length trial simulated data to demonstrate the ability of DUNL to perform local characterization and deconvolution. Second, we showed how the set of bases and coefficients offered by each baseline fails to offer interpretability and the salience/value decomposition of interest in the dopamine spiking data. In the latter, we applied the methods on local windowed-data to focus on the capability of their set of bases. Accordingly, we compared with NMF as opposed to convNMF/seqNMF [76].

We started by using a synthetic dataset to compare DUNL with LFADS [25], a deep learning framework for inferring latent factors from single-trial neural activity. We note that LFADS fits the data at the same resolution as the original rate, while DUNL uses a bin size of 25 ms to model the firing rate. We challenged the interpretability of LFADS in a simple scenario: the trials of the experiment were time-aligned and structured, with two different types of events occurring in single trials. The events were non-overlapping. We evaluated the interpretability of DUNL and LFADS from their ability to deconvolve the single-trial neural activity into two traces, each corresponding to one underlying event type.

We generated such trials for a training dataset and a test dataset (Figure 8a). Both DUNL and LFADS were able to estimate the underlying firing rate of the simulated neuron (Figure 8b,c,d).

DUNL is able to recover the underlying kernels (Figure 8e), and has R^2 fit score, of $0.89, 0.95, 0.97, 0.95, 0.96, 0.97, 0.97$, for training scenarios with $25, 50, 100, 200, 400, 800, 1600$ examples, respectively (the score is evaluated on the binned spikes). LFADS has R^2 score of 0.999 on the test set. Despite the good fit, LFADS: a) finds factors that span the entire trial duration, lacking the locality provided by DUNL, and b) fails to deconvolve the neural activities excited by events of different types from one another (Figure 8f). Overall, unlike DUNL, which provides a functional relation between kernels and firing rates of neurons via a probabilistic generative model, LFADS inference is based on a recurrent neural network, whose encoder and decoder are not tied to one another, thus lacking a direct link between spiking data and the latent factors. At last, we note that DUNL is a convolutional framework, i.e., it can analyze trials of various lengths. However, LFADS can only run on trials of similar length.

To further demonstrate how latent variables from other decomposition methods might not capture interpretable convolved contributions to the neural activity, we applied classical and deep dimensionality reduction methods to the dopamine spiking data. The result showed that Principal Component Analysis (PCA), and Non-negative Matrix Factorization (NMF) can be used to extract components from windowed data (600 ms starting from the reward onset) (Figure S13a,c). However, the results suggest that even if both PCA and NMF fit the data well, neither of them offers the salience/value interpretability that DUNL provides. The coefficients extracted from each trial and the Spearman's rank correlation between each neuron's coefficients across all trials are not aligned with the decomposed codes from DUNL (Figure S13b,d). For PCA, this is due to the dissimilarity of the learned kernels to what we know from the salience and value responses. For NMF, the kernels are semi-similar to the learned kernels in DUNL (non-negative kernels are learned in the spiking scenario). However, the NMF coefficients are not capable of capturing the dip in neural responses, due to their non-negativity constraint, resulting in a lower Spearman's rank correlation to the reward sizes.

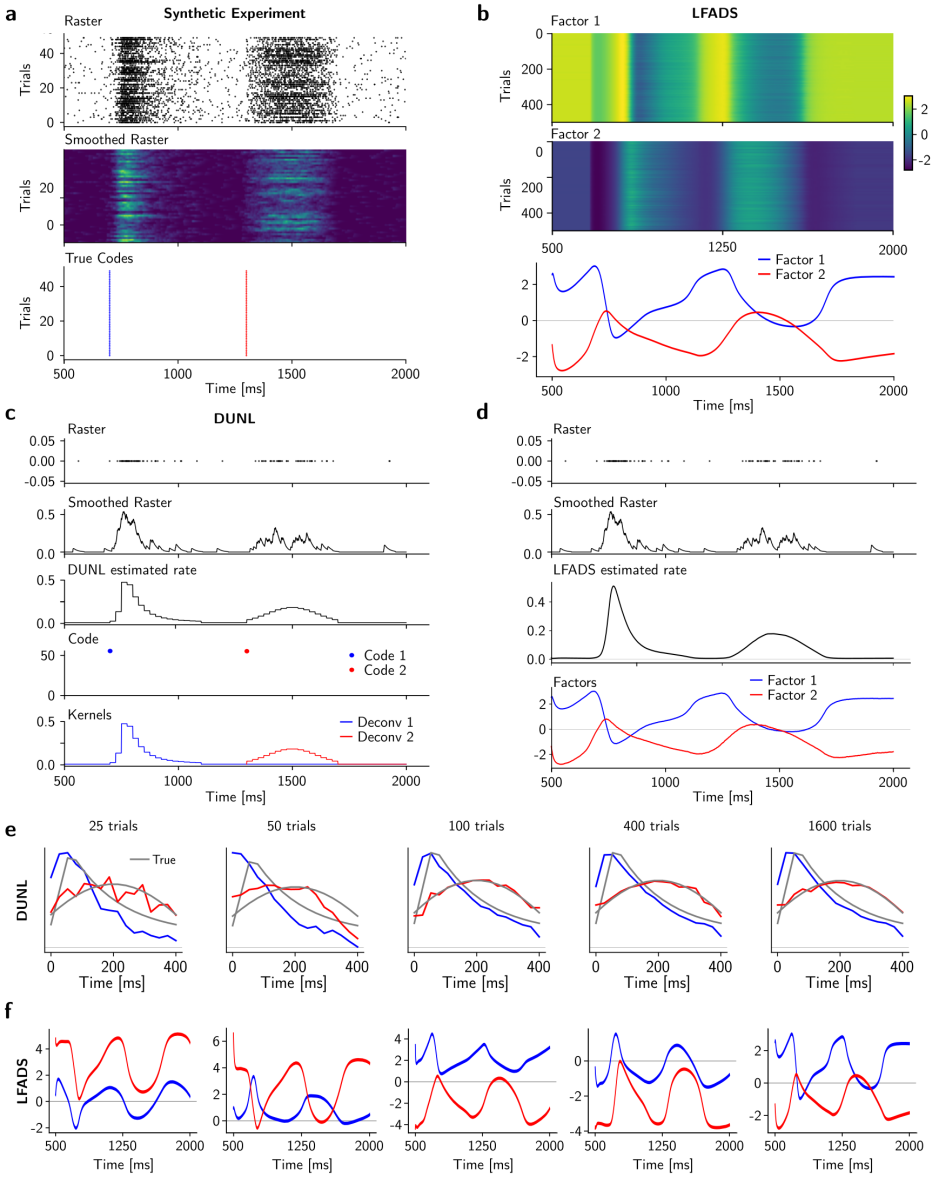


Figure 8. Comparison of DUNL with LFADS. DUNL finds local components in single trials. **a**, Experimental setting implemented for simulated spiking neural activity: this is a multiple trial-based experiment in which two events occur in single trials (top and middle). The event onsets for the two events in each trial of the test set (bottom). **b**, LFADS estimates two factors for this dataset, which span the entire duration of the trial. **c**,**d**, In an example trial (raster and smoothed rate on top), DUNL estimates the firing rate of this simulated neuron in 25 ms bins (**c**, middle) while LFADS estimates the firing rate in the original rate (**d**, middle). DUNL finds local codes for the two kernels (**c**, bottom), while LFADS finds two kernels that span the entire trial duration (**d**, bottom). **e**, Kernels found by DUNL using 25, 50, 100, 400 or 1600 trials for training (compared to the true underlying kernels used to generate the data in gray). **f**, Average LFADS factors across the training scenarios.

Moreover, we compared LFADS with DUNL using the dopamine spiking data. Since LFADS's factors cover the entire trial duration, we applied it to windowed spiking data time-aligned to the reward onset. Specifically, we used only 600 ms of each trial, starting from the reward onset. Using two factors, we found that despite LFADS's success in estimating the firing rate of neurons in each trial ($R^2 = 0.999$ on the test set), the learned factors lack salience-value interpretability (Figure S13e). The comparison of the Spearman's rank correlation analysis on the LFADS factors and DUNL's codes with respect to the reward response (Figure S13f) further supports the absence of

salience-like characterization in the LFADS method: both factors incorporate value information. These effects could be due to the more expressive architecture of LFADS that overfits the data at the expense of a parsimonious and interpretable description (See Figure S13g-h in Supplementary materials for LFADS analysis in the limited data-regime).

Finally, we applied GLM with a family of basis functions [58] using a similar time-bin resolution of 25 ms as DUNL to the dopamine spiking experiment. We guide the framework to fit the data using a set of bases at the onset of the reward for a duration of 600 ms. We show results from two scenarios, each with a different set of bases, i.e., non-linear raised cosines and raised cosines. We found that although GLM with pre-defined bases has smooth fitting curves, it cannot deconvolve single-trials into components that are interpretable from the perspective of salience and value in this experiment (Figure S14). We argue that this is primarily due to the dissimilarity of the pre-defined bases to the kernels learned by DUNL. One may use the DUNL kernels within the GLM framework to perform the deconvolution of interest, thus taking advantage of the interpretablility of learned kernels in place of pre-defined bases. However, in the absence of a kernel-learning framework, such interpretable kernels are unknown a *priori*.

Discussion

The technical and computational developments of the last decade have enabled the acquisition of increasingly large datasets of neural data during behaviour, through the use of high-throughput electrophysiology and two-photon calcium imaging in animals performing complex tasks [2, 3]. This trend has shifted the focus of many neural analyses from the characterization of single neurons to the analysis of emergent, dynamic properties of simultaneously-recorded neural populations [77]. Both of these approaches are important for understanding how neural computations lead to behaviour. In fact, as we increasingly study more cognitive variables that enable complex behaviour, such as learning, decision-making, evidence integration, or cognitive maps, the field must have more capacity to investigate how different neuron types, or heterogeneity within a more or less homogeneous population, support dynamic population-level computations in stochastic environments. With new technologies enabling neuroscience research to grow into more naturalistic and unstructured settings, closer to the natural environments inhabited by animals [78], tools that bridge the activity and properties of single neurons with their population and circuit-level computations during these unconstrained behaviours are of utmost importance.

Here, we introduced the use of unrolled dictionary learning-based neural networks [55–57] to deconvolve multiplexed components of neural data that are relatable to human-interpretable latent variables. This is a technique, based on algorithm unrolling [38, 39], to design an interpretable deep neural network. Our method, DUNL (Deconvolutional Unrolled Neural Learning), fulfills important desiderata of a decomposition method: it can be implemented in single instantiations of the neural data, it can be trained with a limited dataset, it is flexible in regard to the source signal, it generates a mapping between data and latent variables, and, importantly, from these latent variables to human-interpretable variables. This is achieved through the use of a generative model that guides the architecture of the inference deep neural network during the optimization process (Figure 1). This method is a deconvolutional method that can look for and encode local overlapping events within a single trial (e.g., cue and reward components in the dopamine experiments or multiple deflections in the whisker experiment), while Principal Component Analysis (PCA), Non-negative Matrix Factorization (NMF), and Latent Factor Analysis via Dynamical System (LFADS) can only offer components/factors covering the entire duration of a trial (for this reason, we apply them on windowed data aligned at the onset of the events of interests). Our work, while sharing the statistical nature of previous methods based on optimization using generalized linear models (GLM) [58, 75], goes beyond them by a) learning kernels (covariates) from the data and b) using deep learning and backpropagation for data fitting, such that the typical response function of neurons and their amplitudes to multiple events in single trials are directly obtained from the network

weights and latent representations.

Our method owes its efficiency to the combination of algorithm unrolling with sparse coding to provide temporal structure to the analysis of the neural data. Exogenous stimuli, behaviour, and neural activity all share time as a fundamental variable, and sparse coding has a rich history in neuroscience as an interpretable theory of early sensory processing in many brain regions and systems [79, 80]. By adding temporal structure to sparse coding, we obtain an expressive artificial neural network that can deconvolve single-trial neuronal activity into interpretable components, because they correlate with exogenous stimuli and/or behaviour. First, the obtained latent representations are aligned with time and, second, they are sparse. Our method's deconvolution of neural response components can be seen as resulting in an input/output characterization of the functional properties of a system (neuron(s) in this case). The appeal of approaches such as GLMs comes from the fact that, in some sense, they provide such input/output descriptions of neural responses. Signal processing theory [52] has well-established links between such descriptions and computations (e.g., differential equations). The GLM-like statistical nature of our models linking latent, learnable, representations and neuronal data, together with their translation, via algorithm unrolling, into interpretable deep-learning architectures, leads to a powerful approach for analyzing single-trial neural data that satisfies the desiderata put forth.

Our method expands the techniques, available to neuroscientists for analysis of neural data, such as NMF, PCA, sparse coding, GLM, and deep neural networks [25–28, 81–84]. In particular, our method is useful when multiplexed signals are encoded by individual neurons or populations, and when detection of events is needed in high SNR settings. Our method does not intrinsically impose constraints on the basis/kernels, such as orthogonality as in PCA, or non-negativity as in NMF. It is also not limited by a user-defined set of basis functions, and it outputs a measure of the intensity of the response on an event-by-event basis.

Importantly, constraints and regularization can be easily added to the optimization problem if these are useful (such as enforcing or discouraging the co-activation of certain kernels, non-negativity, etc.). Moreover, our method has the ability to learn local characteristics from time-series, which is missing in LFADS. The user can choose whether to learn individual kernels for each neuron, or shared kernels among neurons with individualized code values, and our model can be trained with very few trials: its computational efficiency is provided by the sparsity constraint. Thus, our framework is easier to use, customize, and train than previous methods [25].

To show the versatility of our model, we applied it to a diversity of experimental settings. First, we deconvolved multiplexed components of the reward response of dopamine neurons acquired using electrophysiology and using calcium imaging, to show that our methods are source-agnostic. Our results illustrate the challenge of measuring neural activity with sensors whose dynamics are slower than the dynamics of the signals encoded in single neurons, and our ability to deconvolve slow calcium responses to odor cues from the reward response: these signals become artificially convolved by the calcium sensor, but DUNL can recover them. We also showed, in these datasets, that our method provides more interpretability than alternative dimensionality reduction methods, such as PCA, NMF, and LFADS, even if their combined components fit the original data very well. Our results show that the inferred value-like code is more informative about the reward size than traditional ad-hoc window activity averaging, opening up the possibility of a more precise characterization of dopamine neuron's heterogeneity. Second, we used DUNL to simultaneously detect a kernel and the timing of events in a high SNR setting. Neurons from the sensory thalamus have a stereotyped response to whisker deflections, which can be detected by DUNL with minimal input. This goes beyond previous analysis using GLMs, which performed averaging over trials as well as windowed analysis over whisker deflections and did not provide an event-by-event measure of the amplitude of the response to an individual whisker deflection in a single trial. Finally, we used DUNL to find the kernels of individual neurons from the piriform cortex in response to randomly delivered odor plumes. This application shows how it can be used for exploratory data analysis, namely to cluster different types of neural responses.

To conclude, we point out that the unrolling framework can be extended to provide interpretable latent representations under other regimes, besides the sparsity one used here. More complex generative models can be used, for instance, Kalman filtering-based neural networks [38, 46], 2D filters with other constraints, or group sparsity [85]. Our work is a first step towards leveraging the advances in interpretable deep learning to gain a mechanistic understanding of neural dynamics and underlying computations.

Acknowledgment

We thank all members of the Uchida, Murthy, and Ba laboratories for helpful discussions, particularly Jacob Zavatone-Veth, Adam S. Lowet, and Andrew H. Song for feedback on the manuscript. We thank Gil Costa and Scidraw.io for the rat schematic (doi.org/10.5281/zenodo.3926343). We thank the funding agencies that supported our work: the US DOD, UK MOD, and UK Engineering and Physical Research Council (EPSRC) for the ARO Grant (W911NF-16-1-0368 to DB), the NIH (grant 5R01DC017311 to NU and VNM), the Human Frontier Science Program (LT000801/2018 to SM), the Harvard Brain Science Initiative (Young Scientist Transitions Award to SM); the Brain and Behavior Research Foundation (NARSAD Young Investigator Grant no.30035 to SM); and the Harvard Mind Brain Behavior Interfaculty Initiative (to PM).

Author contributions

Author contributions are summarized in the table.

Table 1. Author contributions

Authors	BT	SM	HW	ST	NU	VM	PM	DB
Study conception	✓	✓					✓	✓
Methodology	✓							✓
Formal analysis	✓							
Investigation: performed *in silico* experiments	✓							
Investigation: performed *in vivo* experiments		✓	✓	✓				
Data curation	✓	✓	✓	✓				
Writing - initial draft and final manuscript	✓	✓					✓	✓
Writing - critical review and revision	✓	✓		✓	✓	✓	✓	✓
Writing - data presentation	✓	✓						
Supervision							✓	✓

References

1. Jun, J. J., Steinmetz, N. A., Siegle, J. H., Denman, D. J., Bauza, M., Barbarits, B., Lee, A. K., Anastassiou, C. A., Andrei, A., Aydın, Ç., *et al.* Fully integrated silicon probes for high-density recording of neural activity. *Nature* **551,** 232–236 (2017).
2. Steinmetz, N. A., Koch, C., Harris, K. D. & Carandini, M. Challenges and opportunities for large-scale electrophysiology with Neuropixels probes. *Current opinion in neurobiology* **50,** 92–100 (2018).
3. Zong, W., Obenhaus, H. A., Skytøen, E. R., Eneqvist, H., de Jong, N. L., Vale, R., Jorge, M. R., Moser, M.-B. & Moser, E. I. Large-scale two-photon calcium imaging in freely moving mice. *Cell* **185,** 1240–1256.e30 (2022).
4. Rigotti, M., Barak, O., Warden, M. R., Wang, X.-J., Daw, N. D., Miller, E. K. & Fusi, S. The importance of mixed selectivity in complex cognitive tasks. *Nature* **497,** 585–590 (2013).

5. Hirokawa, J., Vaughan, A., Masset, P., Ott, T. & Kepecs, A. Frontal cortex neuron types categorically encode single decision variables. *Nature* **576,** 446–451 (2019).

6. Zhang, A. & Zador, A. M. Neurons in the primary visual cortex of freely moving rats encode both sensory and non-sensory task variables. *PLoS Biology* **21,** e3002384 (2023).

7. Dubreuil, A., Valente, A., Beiran, M., Mastrogiuseppe, F. & Ostojic, S. The role of population structure in computations through neural dynamics. *Nature Neuroscience,* 1–12 (2022).

8. Langdon, C. & Engel, T. A. Latent circuit inference from heterogeneous neural responses during cognitive tasks. *Preprint.* https://www.biorxiv.org/content/10.1101/2022.01.23.477431v1 (2022).

9. Kaufman, M. T., Benna, M. K., Rigotti, M., Stefanini, F., Fusi, S. & Churchland, A. K. The implications of categorical and category-free mixed selectivity on representational geometries. *Current Opinion in Neurobiology* **77,** 102644 (2022).

10. Cox, J. & Witten, I. B. Striatal circuits for reward learning and decision-making. *Nature Reviews Neuroscience* **20,** 482–494 (2019).

11. Lee, R. S., Sagiv, Y., Engelhard, B., Witten, I. B. & Daw, N. D. A feature-specific prediction error model explains dopaminergic heterogeneity. *Preprint.* https://www.biorxiv.org/content/10.1101/2022.02.28.482379v2 (2023).

12. Schultz, W. Dopamine reward prediction-error signalling: a two-component response. *Nature reviews neuroscience* **17,** 183–195 (2016).

13. Matsumoto, H., Tian, J., Uchida, N. & Watabe-Uchida, M. Midbrain dopamine neurons signal aversion in a reward-context-dependent manner. *eLife* **5,** e17328 (2016).

14. Williams, A. H. & Linderman, S. W. Statistical neuroscience in the single trial limit. *Current Opinion in Neurobiology* **70,** 193–205. ISSN: 0959-4388 (2021).

15. Williams, A., Degleris, A., Wang, Y. & Linderman, S. Point process models for sequence detection in high-dimensional neural spike trains. *Advances in neural information processing systems* **33,** 14350–14361 (2020).

16. Czanner, G., Eden, U. T., Wirth, S., Yanike, M., Suzuki, W. A. & Brown, E. N. Analysis of between-trial and within-trial neural spiking dynamics. *Journal of neurophysiology* **99,** 2672–2693 (2008).

17. Gomez-Marin, A., Paton, J. J., Kampff, A. R., Costa, R. M. & Mainen, Z. F. Big behavioral data: psychology, ethology and the foundations of neuroscience. *Nature neuroscience* **17,** 1455–1462 (2014).

18. Pearson, J. M., Watson, K. K. & Platt, M. L. Decision making: the neuroethological turn. *Neuron* **82,** 950–965 (2014).

19. Dennis, E. J., El Hady, A., Michaiel, A., Clemens, A., Tervo, D. R. G., Voigts, J. & Datta, S. R. Systems neuroscience of natural behaviors in rodents. *Journal of Neuroscience* **41,** 911–919 (2021).

20. Miller, C. T., Gire, D., Hoke, K., Huk, A. C., Kelley, D., Leopold, D. A., Smear, M. C., Theunissen, F., Yartsev, M. & Niell, C. M. Natural behavior is the language of the brain. *Current Biology* **32,** R482–R493 (2022).

21. Grienberger, C. & Konnerth, A. Imaging calcium in neurons. *Neuron* **73,** 862–885 (2012).

22. Von Helmholtz, H. *Handbuch der physiologischen Optik* (Voss, 1867).

23. Friston, K. A theory of cortical responses. *Philosophical transactions of the Royal Society B: Biological sciences* **360,** 815–836 (2005).

24. Neisser, U. Cognitive Psychology, Appleton-Century-Crofts, New York, 1967. *The functions and nature of imagery (ed. PW Sheehan). Academic Press, New York. Novick, R., and Lazar,* 955–61 (1967).

25. Pandarinath, C., O'Shea, D. J., Collins, J., Jozefowicz, R., Stavisky, S. D., Kao, J. C., Trautmann, E. M., Kaufman, M. T., Ryu, S. I., Hochberg, L. R., *et al.* Inferring single-trial neural population dynamics using sequential auto-encoders. *Nature methods* **15,** 805–815 (2018).

26. Zhu, F., Grier, H. A., Tandon, R., Cai, C., Agarwal, A., Giovannucci, A., Kaufman, M. T. & Pandarinath, C. *A deep learning framework for inference of single-trial neural population dynamics from calcium imaging with subframe temporal resolution* tech. rep. (Nature Publishing Group, 2022).

27. Keshtkaran, M. R., Sedler, A. R., Chowdhury, R. H., Tandon, R., Basrai, D., Nguyen, S. L., Sohn, H., Jazayeri, M., Miller, L. E. & Pandarinath, C. A large-scale neural network training framework for generalized estimation of single-trial population dynamics. *Nature Methods* **19,** 1572–1577 (2022).

28. Schneider, S., Lee, J. H. & Mathis, M. W. Learnable latent embeddings for joint behavioural and neural analysis. *Nature.* ISSN: 1476-4687 (2023).

29. Castelvecchi, D. Can we open the black box of AI? *Nature News* **538,** 20 (2016).

30. Carvalho, D. V., Pereira, E. M. & Cardoso, J. S. Machine learning interpretability: A survey on methods and metrics. *Electronics* **8,** 832 (2019).

31. Doshi-Velez, F. & Kim, B. Towards a rigorous science of interpretable machine learning. *Preprint.* https://arxiv.org/abs/1702.08608 (2017).

32. Itti, L., Koch, C. & Niebur, E. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on pattern analysis and machine intelligence* **20,** 1254–1259 (1998).

33. Simonyan, K., Vedaldi, A. & Zisserman, A. Deep inside convolutional networks: Visualising image classification models and saliency maps. *Preprint.* https://arxiv.org/abs/1312.6034 (2013).

34. Ribeiro, M. T., Singh, S. & Guestrin, C. *"Why should i trust you?" Explaining the predictions of any classifier* in *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining* (2016), 1135–1144.

35. Rudin, C. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence* **1,** 206–215 (2019).

36. Maheswaranathan, N., McIntosh, L. T., Tanaka, H., Grant, S., Kastner, D. B., Melander, J. B., Nayebi, A., Brezovec, L. E., Wang, J. H., Ganguli, S., *et al.* Interpreting the retinal neural code for natural scenes: From computations to neurons. *Neuron* (2023).

37. Shlezinger, N., Whang, J., Eldar, Y. C. & Dimakis, A. G. Model-based deep learning. *Preprint.* https://arxiv.org/abs/2012.08405 (2020).

38. Monga, V., Li, Y. & Eldar, Y. C. Algorithm unrolling: Interpretable, efficient deep learning for signal and image processing. *IEEE Signal Processing Magazine* **38,** 18–44 (2021).

39. Gregor, K. & LeCun, Y. *Learning fast approximations of sparse coding* in *Proceedings of the 27th international conference on international conference on machine learning* (2010), 399–406.

40. Wang, T., Rudin, C., Doshi-Velez, F., Liu, Y., Klampfl, E. & MacNeille, P. A bayesian framework for learning rule sets for interpretable classification. *The Journal of Machine Learning Research* **18,** 2357–2393 (2017).

41. Wang, F. & Rudin, C. *Falling Rule Lists* in *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics* **38** (San Diego, California, USA, 2015), 1013–1022.

42. Wang, Z., Liu, D., Yang, J., Han, W. & Huang, T. *Deep networks for image super-resolution with sparse prior* in *Proceedings of the IEEE international conference on computer vision* (2015), 370–378.

43. Schuler, C. J., Hirsch, M., Harmeling, S. & Schölkopf, B. Learning to Deblur. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **38,** 1439–1451 (2016).

44. Dardikman-Yoffe, G. & Eldar, Y. C. Learned SPARCOM: unfolded deep super-resolution microscopy. *Optics express* **28,** 27736–27763 (2020).

45. Hosseini, S. A. H., Yaman, B., Moeller, S., Hong, M. & Akçakaya, M. Dense Recurrent Neural Networks for Accelerated MRI: History-Cognizant Unrolling of Optimization Algorithms. *IEEE Journal of Selected Topics in Signal Processing* **14,** 1280–1291 (2020).

46. Revach, G., Shlezinger, N., Ni, X., Escoriza, A. L., Van Sloun, R. J. & Eldar, Y. C. KalmanNet: Neural network aided Kalman filtering for partially known dynamics. *IEEE Transactions on Signal Processing* **70,** 1532–1547 (2022).

47. Tolooshams, B., Mulleti, S., Ba, D. & Eldar, Y. C. *Unfolding Neural Networks for Compressive Multichannel Blind Deconvolution* in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2021), 2890–2894.

48. Wang, Z.-Q., Roux, J. L., Wang, D. & Hershey, J. R. End-to-end speech separation with unfolded iterative phase reconstruction. *Preprint.* https://arxiv.org/abs/1804.10204 (2018).

49. Tolooshams, B., Song, A., Temereanca, S. & Ba, D. *Convolutional dictionary learning based auto-encoders for natural exponential-family distributions* in *Proceedings of the 37th International Conference on Machine Learning* (eds III, H. D. & Singh, A.) **119** (PMLR, 2020), 9493–9503.

50. McCullagh, P. & Nelder, J. A. *Generalized linear models* (Routledge, 2019).

51. Glasgow, N. G., Chen, Y., Korngreen, A., Kass, R. E. & Urban, N. N. A biophysical and statistical modeling paradigm for connecting neural physiology and function. *Journal of Computational Neuroscience,* 1–20 (2023).

52. Marmarelis, V. *Analysis of physiological systems: The white-noise approach* (Springer Science & Business Media, 2012).

53. Park, M. & Pillow, J. W. Receptive field inference with localized priors. *PLoS computational biology* **7,** e1002219 (2011).

54. Aoi, M. C. & Pillow, J. W. Scalable Bayesian inference for high-dimensional neural receptive fields. *Preprint.* https://www.biorxiv.org/content/early/2017/11/01/212217 (2017).

55. Tolooshams, B., Dey, S. & Ba, D. *Scalable convolutional dictionary learning with constrained recurrent sparse auto-encoders* in *2018 IEEE 28th International Workshop on Machine Learning for Signal Processing (MLSP)* (2018), 1–6.

56. Tolooshams, B., Song, A., Temereanca, S. & Ba, D. *Convolutional dictionary learning based auto-encoders for natural exponential-family distributions* in *International Conference on Machine Learning* (2020), 9493–9503.

57. Tolooshams, B. & Ba, D. E. Stable and Interpretable Unrolled Dictionary Learning. *Transactions on Machine Learning Research* (2022).

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

59. Schultz, W., Dayan, P. & Montague, P. R. A neural substrate of prediction and reward. *Science* **275,** 1593–1599 (1997).

60. Bayer, H. M. & Glimcher, P. W. Midbrain dopamine neurons encode a quantitative reward prediction error signal. *Neuron* **47,** 129–141 (2005).

61. Hart, A. S., Rutledge, R. B., Glimcher, P. W. & Phillips, P. E. Phasic dopamine release in the rat nucleus accumbens symmetrically encodes a reward prediction error term. *Journal of Neuroscience* **34,** 698–704 (2014).

62. Sutton, R. S. & Barto, A. G. *Reinforcement learning: An introduction* (MIT press, 2018).

63. Kim, H. R., Malik, A. N., Mikhael, J. G., Bech, P., Tsutsui-Kimura, I., Sun, F., Zhang, Y., Li, Y., Watabe-Uchida, M., Gershman, S. J. & Uchida, N. A Unified Framework for Dopamine Signals across Timescales. *Cell* **183,** 1600–1616 (2020).

64. Amo, R., Matias, S., Yamanaka, A., Tanaka, K. F., Uchida, N. & Watabe-Uchida, M. A gradual temporal shift of dopamine responses mirrors the progression of temporal difference error in machine learning. *Nature Neuroscience* **25,** 1082–1092 (2022).

65. Schultz, W. Behavioral theories and the neurophysiology of reward. *Annu. Rev. Psychol.* (2006).

66. Cohen, J. Y., Haesler, S., Vong, L., Lowell, B. B. & Uchida, N. Neuron-type-specific signals for reward and punishment in the ventral tegmental area. *nature* **482,** 85–88 (2012).

67. Eshel, N., Bukwich, M., Rao, V., Hemmelder, V., Tian, J. & Uchida, N. Arithmetic and local circuitry underlying dopamine prediction errors. *Nature* **525,** 243–246 (2015).

68. Dabney, W., Kurth-Nelson, Z., Uchida, N., Starkweather, C. K., Hassabis, D., Munos, R. & Botvinick, M. A distributional code for value in dopamine-based reinforcement learning. *Nature* **577,** 671–675 (2020).

69. Eshel, N., Tian, J., Bukwich, M. & Uchida, N. Dopamine neurons share common response function for reward prediction error. *Nature neuroscience* **19,** 479–486 (2016).

70. Engelhard, B., Finkelstein, J., Cox, J., Fleming, W., Jang, H. J., Ornelas, S., Koay, S. A., Thiberge, S. Y., Daw, N. D., Tank, D. W., *et al.* Specialized coding of sensory, motor and cognitive variables in VTA dopamine neurons. *Nature* **570,** 509–513 (2019).

71. Temereanca, S., Brown, E. N. & Simons, D. J. Rapid Changes in Thalamic Firing Synchrony during Repetitive Whisker Stimulation. *Journal of Neuroscience* **28,** 11153–11164 (2008).

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

73. Ba, D., Temereanca, S. & Brown, E. N. Algorithms for the analysis of ensemble neural spiking activity using simultaneous-event multivariate point-process models. *Frontiers in computational neuroscience* **8,** 6 (2014).

74. Ackels, T., Erskine, A., Dasgupta, D., Marin, A. C., Warner, T. P., Tootoonian, S., Fukunaga, I., Harris, J. J. & Schaefer, A. T. Fast odour dynamics are encoded in the olfactory system and guide behaviour. *Nature* **593,** 558–563 (2021).

75. Dowling, M., Zhao, Y. & Park, I. M. Non-parametric generalized linear model. *Preprint.* https://arxiv.org/abs/2009.01362 (2020).

76. Mackevicius, E. L., Bahle, A. H., Williams, A. H., Gu, S., Denisenko, N. I., Goldman, M. S. & Fee, M. S. Unsupervised discovery of temporal sequences in high-dimensional datasets, with applications to neuroscience. *Elife* **8,** e38471 (2019).

77. Cunningham, J. P. & Yu, B. M. Dimensionality reduction for large-scale neural recordings. *Nature neuroscience* **17,** 1500–1509 (2014).

78. Mathis, A., Mamidanna, P., Cury, K. M., Abe, T., Murthy, V. N., W, M. M. & Bethge, M. DeepLabCut: markerless pose estimation of user-defined body parts with deep learning. *Nature Neuroscience* **21,** 1281–1289 (2018).

79. Olshausen, B. A. & Field, D. J. Sparse coding of sensory inputs. *Current opinion in neurobiology* **14,** 481–487 (2004).

80. Jortner, R. A., Farivar, S. S. & Laurent, G. A simple connectivity scheme for sparse coding in an olfactory system. *Journal of Neuroscience* **27,** 1659–1669 (2007).

81. Lee, D. D. & Seung, H. S. Learning the parts of objects by non-negative matrix factorization. *Nature* **401,** 788–791 (1999).

82. Olshausen, B. A. & Field, D. J. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature* **381,** 607–609 (1996).

83. Schwartz, O., Pillow, J. W., Rust, N. C. & Simoncelli, E. P. Spike-triggered neural characterization. *Journal of vision* **6,** 13–13 (2006).

84. Kobak, D., Brendel, W., Constantinidis, C., Feierstein, C. E., Kepecs, A., Mainen, Z. F., Qi, X.-L., Romo, R., Uchida, N. & Machens, C. K. Demixed principal component analysis of neural population data. *elife* **5,** e10989 (2016).

85. Theodosis, E., Tolooshams, B., Tankala, P., Tasissa, A. & Ba, D. On the convergence of group-sparse autoencoders. *Preprint.* https://arxiv.org/abs/2102.07003 (2021).

86. Song, A. H., Tolooshams, B. & Ba, D. Gaussian Process Convolutional Dictionary Learning. *IEEE Signal Processing Letters* **29,** 95–99 (2022).

87. Hastie, T., Tibshirani, R. & Wainwright, M. *Statistical learning with sparsity: the lasso and generalizations* (CRC press, 2015).

88. Elad, M. *Sparse and redundant representations: from theory to applications in signal and image processing* (Springer Science & Business Media, 2010).

89. Olshausen, B. A. & Field, D. J. Sparse coding with an overcomplete basis set: A strategy employed by V1? *Vision research* **37,** 3311–3325 (1997).

90. Tootoonian, S. & Lengyel, M. A dual algorithm for olfactory computation in the locust brain. *Advances in neural information processing systems* **27** (2014).

91. Hromádka, T., DeWeese, M. R. & Zador, A. M. Sparse representation of sounds in the unanesthetized auditory cortex. *PLoS biology* **6,** e16 (2008).

92. Zavatone-Veth, J. A., Masset, P., Tong, W. L., Zak, J., Murthy, V. N. & Pehlevan, C. *Neural Circuits for Fast Poisson Compressed Sensing in the Olfactory Bulb* in *Thirty-seventh Conference on Neural Information Processing Systems* (2023).

93. Cleary, B., Cong, L., Cheung, A., Lander, E. S. & Regev, A. Efficient Generation of Transcriptomic Profiles by Random Composite Measurements. *Cell* **171,** 1424–1436.e18. ISSN: 0092-8674 (2017).

94. Cleary, B., Simonton, B., Bezney, J., Murray, E., Alam, S., Sinha, A., Habibi, E., Marshall, J., Lander, E. S., Chen, F., *et al.* Compressed sensing for highly efficient imaging transcriptomics. *Nature Biotechnology,* 1–7 (2021).

95. Chatterji, N. S. & Bartlett, P. L. Alternating minimization for dictionary learning: Local Convergence Guarantees. *Preprint,* 1–26. https://arxiv.org/abs/1711.03634 (2017).

96. Daubechies, I., Defrise, M. & De Mol, C. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Communications on Pure and Applied Mathematics* **57,** 1413–1457 (2004).

97. Blumensath, T. & Davies, M. E. Iterative thresholding for sparse approximations. *Journal of Fourier analysis and Applications* **14,** 629–654 (2008).

98. Tolooshams, B., Dey, S. & Ba, D. Deep Residual Autoencoders for Expectation Maximization-Inspired Dictionary Learning. *IEEE Transactions on Neural Networks and Learning Systems* **32,** 2415–2429 (2021).

99. Siegle, J. H., López, A. C., Patel, Y. A., Abramov, K., Ohayon, S. & Voigts, J. Open Ephys: an open-source, plugin-based platform for multichannel electrophysiology. *Journal of neural engineering* **14,** 045003 (2017).

100. Pachitariu, M., Steinmetz, N. A., Kadir, S. N., Carandini, M. & Harris, K. D. Fast and accurate spike sorting of high-channel count probes with KiloSort. *Advances in neural information processing systems* **29** (2016).

101. Bäckman, C. M., Malik, N., Zhang, Y., Shan, L., Grinberg, A., Hoffer, B. J., Westphal, H. & Tomac, A. C. Characterization of a mouse strain expressing Cre recombinase from the 3' untranslated region of the dopamine transporter locus. *genesis* **44,** 383–390 (2006).

102. Daigle, T., Madisen, L., Hage, T., Valley, M., Knoblich, U., Larsen, R., Takeno, M., Huang, L., Gu, H., Larsen, R., Mills, M., Bosma-Moody, A., Siverts, L., Walker, M., Graybuck, L., Yao, Z., Fong, O., Garren, E., Lenz, G., Chavarah, M., Pendergraft, J., Harrington, J., Hirokawa, K., Harris, J., McGraw, M., Ollerenshaw, D., Smith, K., Baker, C., Ting, J., Sunkin, S., Lecoq, J., Lin, M., Boyden, E., Murphy, G., da Costa, N., Waters, J., Li, L., Tasic, B. & Zeng, H. A suite of transgenic driver and reporter mouse lines with enhanced brain cell type targeting and functionality. *Cell* **174,** 465–480.e22 (2 2018).

103. Pachitariu, M., Stringer, C., Dipoppa, M., Schröder, S., Rossi, L. F., Dalgleish, H., Carandini, M. & Harris, K. D. Suite2p: beyond 10,000 neurons with standard two-photon microscopy. *Preprint.* https://www.biorxiv.org/content/early/2017/07/20/061507 (2017).

104. Keemink, S. W., Lowe, S. C., Pakan, J. M. P., Dylda, E., van Rossum, M. C. W. & Rochefort, N. L. FISSA: A neuropil decontamination toolbox for calcium imaging signals. *Scientific Reports* **8,** 3493 (2018).

Supplementary Methods - Methods

Notation

Scalars are denoted by non-bold-lower-case a. Vectors and matrices are denoted by bold-lower-case $\boldsymbol{a}$ and upper-case letters $\boldsymbol{A}$, respectively. We let $n = 1, \ldots, N$ index neurons, and $j = 1, 2, \ldots, J$ the number of trials. We assume that the time series representing the activity from neuron n at trial j comprises T measurements, which we denote $\boldsymbol{y}^{n,j}$. We denote the full measurement tensor by $\boldsymbol{Y}$ with $T \times J \times N$ dimensions (time/bin measurements, number of trials, number of neurons). We denote the convolution operator by $*$ and its transpose operator (correlation) by $\star$. Finally, we use superscript T for transpose of a matrix $\boldsymbol{A}^{\mathsf{T}}$.

Data Distribution

For spiking data, the spikes at each trial are binned at B ms resolution. Hence, each entry of $\boldsymbol{y}^{n,j}$ represents a spike count ranging from 0 to B. We model the observations using the natural exponential family [49, 50], i.e., $\boldsymbol{y}^{n,j} \sim \text{Poisson}(\boldsymbol{\mu}^{n,j})$ and $\boldsymbol{y}^{n,j} \sim \text{Binomial}(B, \boldsymbol{\mu}^{n,j})$, where $\boldsymbol{\mu}^{n,j}$ models the mean of the distribution for neuron n at trial j. For continuous-valued data, such as Calcium fluorescence data, we model the time series $\boldsymbol{y}^{n,j} \in \mathbb{R}^T$ as a Gaussian distribution with mean $\boldsymbol{\mu}^{n,j}$. We construct the data log-likelihood of the natural exponential family as [49, 50]

$$\log p(\boldsymbol{y}^{n,j} \mid \boldsymbol{\mu}^{n,j}) = g^{-1}(\boldsymbol{\mu}^{n,j})^{\mathsf{T}} \boldsymbol{y}^{n,j} + f(\boldsymbol{y}^{n,j}) - V(\boldsymbol{\mu}^{n,j}), \tag{1}$$

where condition of $\mu^{n,j}$, we assume the entries of $\boldsymbol{y}^{n,j}$ are independent. The functions g (i.e., inverse link), f and V depend on the particular choice of distribution (see Table S1).

Table S1. Natural exponential family data log-likelihood specifications.

	$\boldsymbol{y}$	V($\boldsymbol{z}$)	$g(\cdot)$
Gaussian	$\mathbb{R}$	$\boldsymbol{z}^{\mathsf{T}}\boldsymbol{z}$	$\boldsymbol{I}(\cdot)$
Binomial	$[0 \ldots B]$	$-\mathbf{1}^{\mathsf{T}} \log(\mathbf{1} - \boldsymbol{z})$	sigmoid$(\cdot)$
Poisson	$[0 \ldots \infty)$	$\mathbf{1}^{\mathsf{T}}\boldsymbol{z}$	$\exp(\cdot)$

Generative Model

We follow the perspective of analysis-by-synthesis [24] and Bayesian generative modelling [53].

For each neuron n, we impose a generative model on the neuron's activity (i.e., the firing rate in the spiking setting) and model it as a function of a baseline mean activity level $a_{n,j}$ and a set of K localized kernels $\{\boldsymbol{h}_k^n\}_{k=1}^K$ characterizing the neuron's response to events that occur sparsely in time. We let the sparse vector $\boldsymbol{x}_k^{n,j}$ encode the onsets of events associated with the kernel k in trial j: its nonzero entries represent the times when events occur, and their amplitude the strength of the contribution of the k-th kernel to the neuron's response. Similar to $\boldsymbol{y}^{n,j}$, the entries of the sparse code $\boldsymbol{x}_k^{n,j} \in \mathbb{R}^{T-T_h+1}$ and the filter $\boldsymbol{h}_k^n \in \mathbb{R}^{T_h}$ are both indexed across time. Mathematically, we can express this *convolutional sparse coding* model as follows

$$\boldsymbol{\mu}^{n,j} = g\left(\sum_{k=1}^{K} \boldsymbol{h}_k^n * \boldsymbol{x}_k^{n,j} + a^{n,j}\right) \tag{2}$$

Although the model results in an estimate of each neuron's firing rate on a trial basis, the kernels capture characteristics that are shared among trials and can be distinct across neurons or shared across the neural population. At times, we may use the terminology dictionary element to refer to the kernels. For the scenario where we share the kernels across neurons, we simplify the dictionary notation to $\boldsymbol{h}_k$.

Smooth Sparse Deconvolutional Learning

Optimization

Given the set of observations from all trials $\{\boldsymbol{y}^{n,j}\}_{j=1}^J$ for each neuron n, we learn the kernels and codes by minimizing the negative log-likelihood with a sparse prior on the codes, i.e.,

$$\begin{aligned} \min_{\{\boldsymbol{h}_k^n\}_{k=1}^K, \{\boldsymbol{x}_k^{n,j}\}_{k=1,j=1}^{K,J}} & \sum\nolimits_{j=1}^{J} -\log p(\boldsymbol{y}^{n,j} \mid \{\boldsymbol{h}_k^n, \boldsymbol{x}_k^{n,j}\}_{k=1}^K) + \sum\nolimits_{k=1}^{K} \lambda_k^n \|\boldsymbol{x}_k^{n,j}\|_1 + \frac{\beta_k^n}{T_h} \|\nabla_t \boldsymbol{h}_k^n[t]\|_2^2 \\ \text{subject to} \quad & \|\boldsymbol{h}_k^n\|_2 = 1 \text{ for } k = 1, \dots, K \end{aligned} \tag{3}$$

where λ_k^n controls the sparsity of the codes (i.e. the frequency of onsets in time) for the kernel (event-type) k and neuron n. Moreover, β_k^n controls the smoothness of the kernels, achieved by regularizing the first derivative of the kernel with respect to time samples t [54, 86]. We call the above optimization smooth sparse deconvolutional learning (SSDL).

SSDL is a variant of dictionary learning, also referred to as sparse coding, and has widespread application outside of neuroscience. Dictionary learning is widely known in statistics and signal processing communities [87, 88]. The sparse coding model was initially introduced by Olshausen and Field [89] to model early layers of visual processing. Prior works used sparse coding for modeling neural connectivity and dynamics of early sensory systems [79, 80, 90–92]. Moreover, for imaging transcriptomics, the model is used to learn representations of gene expression [93, 94].

Alternating minimization

Equation (3) is a bi-convex optimization problem and can be solved by an iterative alternating-minimization algorithm [95]. Letting l denote its iterations, the algorithm alternates between a *sparse-coding* step, that computes an estimate of the codes $\boldsymbol{x}_k^{n,j(l)}$ given an estimate $\boldsymbol{h}_k^{n(l)}$ of the dictionary, and a *dictionary-update* step, that uses this new estimate of the codes to obtain refined estimates $\boldsymbol{h}_k^{n(l+1)}$ of the kernels. Mathematically, we can express the two steps as follows

$$\boldsymbol{x}_k^{n,j(l)} = \underset{\boldsymbol{x}_k^{n,j}}{\arg\min} \; -\log p(\boldsymbol{y}^{n,j} \mid \{\boldsymbol{h}_k^{n(l)}, \boldsymbol{x}_k^{n,j}\}_{k=1}^K) + \sum\nolimits_{k=1}^{K} \lambda_k^n \|\boldsymbol{x}_k^{n,j}\|_1 \quad \text{for} j = 1, \dots, J \tag{4}$$

$$\begin{aligned} \{\boldsymbol{h}_k^{n(l+1)}\}_{k=1}^K = \underset{\{\boldsymbol{h}_k^n\}_{k=1}^K}{\arg\min} & \sum\nolimits_{j=1}^{J} -\log p(\boldsymbol{y}^{n,j} \mid \{\boldsymbol{h}_k^n, \boldsymbol{x}_k^{n,j(l)}\}_{k=1}^K) + \frac{\beta_k^n}{T_h} \|\nabla_t \boldsymbol{h}_k^n[t]\|_2^2 \\ \text{subject to} \quad & \|\boldsymbol{h}_k^n\|_2 = 1 \text{ for } k = 1, \dots, K \end{aligned} \tag{5}$$

Compared to classical sparse coding, both our sparse-coding and dictionary-update steps have convolutional structure. Intuitively, the convolutional structure of our model enables the identification of patterns that occur *across* time.

Sparse coding step

The sparse coding step can be solved using the iterative shrinkage-thresholding algorithm (ISTA) [96, 97]. One iteration of this proximal gradient descent algorithm proceeds as follows

$$\begin{aligned} \boldsymbol{x}_{k,r}^{n,j} &= \mathcal{S}_{\alpha\lambda_k^{n,j}}\left(\boldsymbol{x}_{k,r-1}^{n,j} + \alpha\nabla_{\boldsymbol{x}_{k,r-1}^{n,j}} \log p(\boldsymbol{y}^{n,j} \mid \{\boldsymbol{h}_k^{n(l)}, \boldsymbol{x}_{k,r-1}^{n,j}\}_{k=1}^K)\right) \\ &= \mathcal{S}_{\alpha\lambda_k^{n,j}}\left(\boldsymbol{x}_{k,r-1}^{n,j} + \alpha\boldsymbol{h}_k^{n,j} \star (\boldsymbol{y}^{n,j} - g(\sum_{u=1}^{K} \boldsymbol{h}_u^n * \boldsymbol{x}_{u,r-1}^{n,j} + a^{n,j}))\right), \end{aligned} \tag{6}$$

where the so-called shrinkage operator $\mathcal{S}_b(z) \triangleq \text{sign}(z)\max(|z| - b, 0)$ is a nonlinear, sparsifying, thresholding operation, and r denotes the sparse coding iteration r. For non-negative sparse coding, i.e., when the entries of the sparse code can only take a non-negative value, the shrinkage operation $\mathcal{S}(z)$ reduces to the celebrated $\text{ReLU}_b(z) = (z - b) \cdot \mathbf{1}_{z \geq b}$ nonlinearity. The converged code estimate from the iterative update in (7) is a minimizer of the sparse coding step (6). In applications where the onsets of events are known (i.e., code support is given), we apply an additional indicator function of events $\boldsymbol{e}_k^{n,j}$ at every iteration. Thus, the iterative updates compute estimates of the strength of k-th kernel contribution to neural activity at known event-onset times.

$$\boldsymbol{x}_{k,r}^{n,j} = \boldsymbol{e}_k^{n,j} \cdot \mathcal{S}_{\alpha\lambda_k^{n,j}}\left(\boldsymbol{x}_{k,r-1}^{n,j} + \alpha\boldsymbol{h}_k^{n,j} \star (\boldsymbol{y}^{n,j} - g(\sum_{u=1}^{K} \boldsymbol{h}_u^n * \boldsymbol{x}_{u,r-1}^{n,j} + a^{n,j}))\right) \tag{7}$$

Dictionary learning step

We use gradient-based methods to update the dictionary. In its simpler form, the update is of stochastic projected gradient descent.

$$\boldsymbol{h}_k^{n(l+1)} = \mathcal{P}\left(\boldsymbol{h}_k^{n(l)} - \eta\nabla_{\{\boldsymbol{h}_k^{n(l)}\}_{k=1}^K}\left(-\log p(\boldsymbol{y}^{n,j} \mid \{\boldsymbol{h}_k^{n(l)}, \boldsymbol{x}_k^{n,j(l)}\}_{k=1}^K) + \frac{\beta_k^n}{T_h}\|\nabla_t \boldsymbol{h}_k^{n(l)}[t]\|_2^2\right)\right) \tag{8}$$

where $\mathcal{P}(\boldsymbol{z}) = \boldsymbol{z}/\|\boldsymbol{z}\|_2$ performs a norm projection, and η is the learning rate.

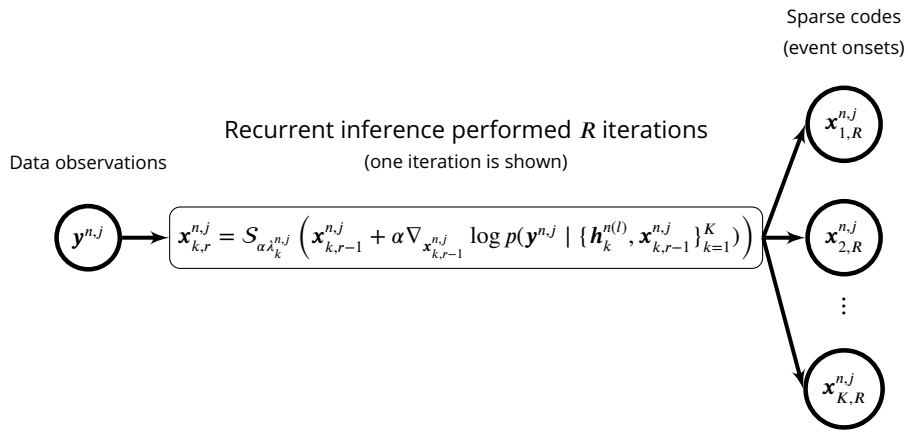
Interpretable deconvolutional unrolled neural learning (DUNL)

Inference network

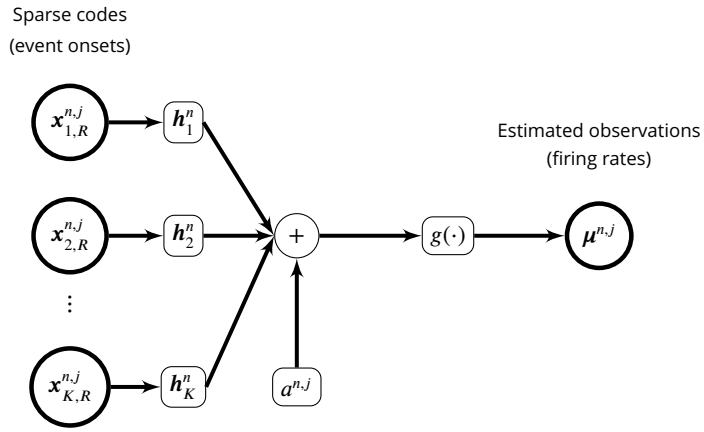
The alternating minimization procedure explained above can be mapped into an encoder/decoder neural architecture. Specifically, we use algorithm unrolling [38] to map the sparse coding step (4) into an encoder. This is similar to the network architectures proposed in [49, 98] for dictionary learning. In this architecture, each sparse coding iteration (6) is interpreted as one layer of a neural network with a particular recurrent convolutional structure and shrinkage or ReLU non-linearity. We refer to this encoder as an inference network that maps the single-neuron, single-trial time series $\boldsymbol{y}^{n,j}$, into estimates of the time series sparse codes $\{\boldsymbol{x}_k^{n,j}\}_{k=1}$, encoding event onsets and their contribution to explain the data (see Figure S1a).

Generative decoder of DUNL

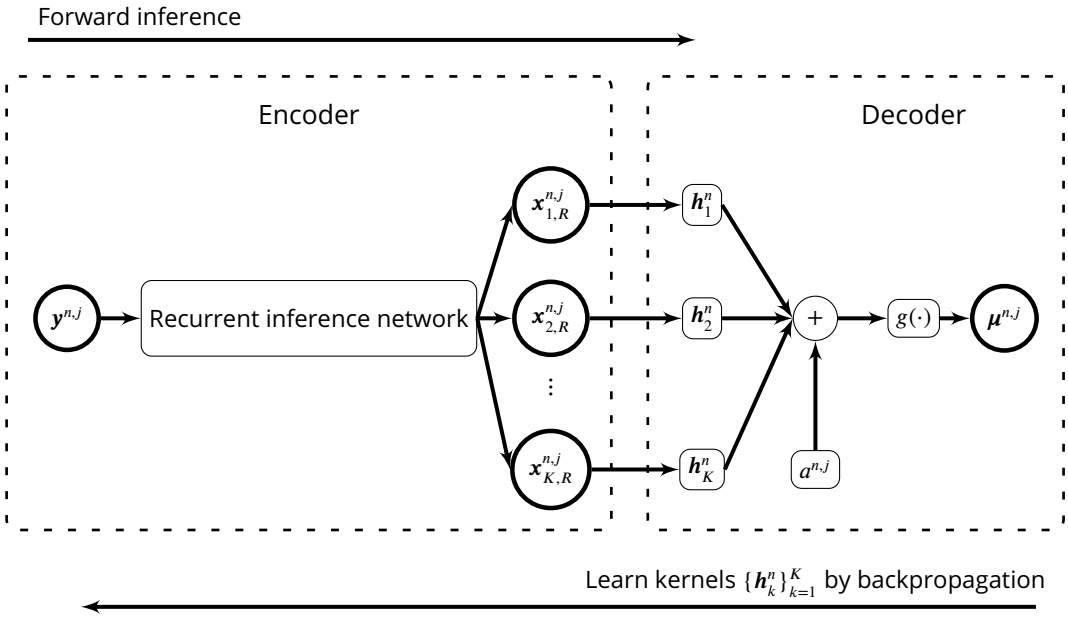
Given the codes from the inference network, we construct a decoder based on the generative model (2) (Figure S1b). This decoder maps the estimated time series of sparse codes from a given neuron into a time-series observation estimate (e.g., a time series representing firing rate in the case of spiking data).



(a) Forward inference network for sparse coding. Given a choice of distribution from the exponential family, this recurrent encoder uses unfolded iterations of the sparse-coding step from dictionary learning to map the time series of observations $\boldsymbol{y}^{n,j}$ to time series representing estimates of the sparse codes associated with each kernel. Given an estimate of the kernels $\{\boldsymbol{h}_k^n\}_{k=1}^K$, this network computes an estimate of their corresponding event onsets and amplitudes.



(b) Generative decoder.



(c) Encoder and decoder architecture of deconvolutional unrolled neural learning.

Figure S1. Deconvolutional unrolled neural learning (DUNL).

Deconvolutional Unrolled Neural Learning (DUNL)

We combine the inference network and the generative decoder to construct an interpretable network which can be trained by backpropagation. Training lets us learn the kernels $\{\boldsymbol{h}_k^n\}_{k=1}^K$ that characterize the neural response to events coded by $\{x_k^{n,j}\}_{k=1}^K$. As detailed in the introduction, The interpretability of this network is two-fold: the network trainable parameters are directly related to the kernels $\{\boldsymbol{h}_k^n\}_{k=1}^K$, and the encoder latent representation corresponds to the event onsets and their strengths.

Training this network involves both a forward pass (inference) and a backward pass (training to learn the dictionary), both of which are embarrassingly parallelizable over neurons and trials. Therefore, the interpretation of the sparse-coding and dictionary-update steps as a network enables to seamlessly take advantage of the parallelism offered by GPUs.

Structured representation

Motivated by biological constraints/prior knowledge, we may want to impose structure on the codes in addition to sparsity. Calcium fluorescence, for instance, does not encode electrical activity linearly: the signal exhibits different dynamics when the firing rate of a neuron increases than when it decreases. Even though the dynamics of the underlying firing rate might be the same, the measured calcium signal will be different, but these dynamics can be captured through the addition of structured representations in our framework. Consider a version of our model for fluorescence data, with one kernel for each neuron. This model could not capture such a nonlinear relationship because both positive (increased activity above baseline) and negative (decreased activity) codes would need to use the same filter/kernel. To overcome this challenge, we can introduce an additional filter and a prior on the codes of both filters that prevent the co-occurrence of event onsets, i.e., that prevents both filters from contributing to neural activity at the same onset times. As we will demonstrate in our analyses of fluorescence data from dopamine neurons, such priors allow us to capture the nonlinear relation between activity level and fluorescence. We can either enforce such latent structure on the codes or learn it by incorporating an additional term into the original optimization. Mathematically, our modified optimization solves

$$\min_{\{\boldsymbol{h}_k^n\}_{k=1}^K, \{\boldsymbol{x}_k^{n,j}\}_{k=1,j=1}^{K,J}} \sum\nolimits_{j=1}^{J} -\log p(\boldsymbol{y}^{n,j} \mid \{\boldsymbol{h}_k^n, \boldsymbol{x}_k^{n,j}\}_{k=1}^K) + \sum\nolimits_{k=1}^{K} \lambda_k^n \|\boldsymbol{x}_k^{n,j}\|_1 + \frac{1}{2}\beta^n \boldsymbol{x}^{n,j\top} \boldsymbol{Q} \boldsymbol{x}^{n,j} \quad \text{subject to} \quad \|\boldsymbol{h}_k^n\|_2 = 1 \text{ for } k = 1, \dots, K \tag{9}$$

where $\boldsymbol{x}^{n,j} = [\boldsymbol{x}_1^{n,j\top}, \boldsymbol{x}_2^{n,j\top}, \dots, \boldsymbol{x}_K^{n,j\top}]^\top$, and $\boldsymbol{Q} \in \mathbb{R}^{K(T-T_k+1)\times K(T-T_k+1)}$ is a symmetric matrix with block structure. For example, given two kernels ($K = 2$) when codes are non-negative, $\boldsymbol{Q} = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{I} \\ \boldsymbol{I} & \boldsymbol{0} \end{bmatrix}$ enforces a structure such that the kernels $\boldsymbol{h}_1$ and $\boldsymbol{h}_2$ are discouraged to get activated simultaneously. Variations of such latent regularization are $\tilde{\boldsymbol{x}}^{n,j\top} \boldsymbol{Q} \tilde{\boldsymbol{x}}^{n,j}$ where $\boldsymbol{Q} \in \mathbb{R}^{K\times K}$, and $\tilde{\boldsymbol{x}}_k^{n,j}$ captures the energy of the code $\boldsymbol{x}_k^{n,j}$ (e.g., $\|\boldsymbol{x}_k^{n,j}\|_2, \|\boldsymbol{x}_k^{n,j}\|_2^2, \|\boldsymbol{x}_k^{n,j}\|_1$, etc.). Although we treat $\boldsymbol{Q}$ as a hyperparameter, it can be related and is proportional to the negative inverse covariance matrix of the code $\boldsymbol{x}^{n,j}$, hence it can be learned. Overall, this regularization modifies the recurrent inference network to

$$\boldsymbol{x}_{k,r}^{n,j} = \mathcal{S}_{\alpha\lambda_k^{n,j}} \left(\boldsymbol{x}_{k,r-1}^{n,j} + \alpha \boldsymbol{h}_k^{n,j} \star (\boldsymbol{y}^{n,j} - g(\sum_{u=1}^{K} \boldsymbol{h}_u^n * \boldsymbol{x}_{u,r-1}^{n,j} + a^{n,j}) - \alpha\beta^n \sum_{v=1}^{K} \boldsymbol{Q}_k \boldsymbol{x}_{v,r}^{n,j}) \right) \tag{10}$$

where $\boldsymbol{Q}_k$ is the kth column block of $\boldsymbol{Q}$. Our source code is written such that this regularization can be enforced not at every unrolled layer but with an occurring period. In addition to the residuals, the kernel codes are now interconnected to one another through $\boldsymbol{Q}$. From (10), we see that the amplitudes of $\boldsymbol{x}_{k,r}^{n,j}$ is damped when $\boldsymbol{Q}_{k,v}$ is positive, and $\boldsymbol{x}_{v,r}^{n,j}$ has high activity.

Network parameters

In this section, we explain the network parameters and the effect of each in the training. Additionally, we specify which parameters are hyperparameters (i.e., to be set by the user), which are

Algorithm 1: Iterative power method to approximate unrolled step size α.

Input: Input size T, initial estimate or randomly initialized $\{\boldsymbol{h}_k\}_{k=1}^{K}$, the inverse link function $g(\cdot)$

Initialize: $\boldsymbol{x}^{(0)} = [\boldsymbol{x}_1^{\mathsf{T}(0)}, \boldsymbol{x}_2^{\mathsf{T}(0)}, \dots, \boldsymbol{x}_K^{\mathsf{T}(0)}]^{\mathsf{T}}$ using Normal distribution

Repeat: $m = 0, 1, \dots, M-1$

$\boldsymbol{x}^{(m)} = \boldsymbol{x}^{(m)} / \|\boldsymbol{x}^{(m)}\|_2$

$\boldsymbol{x}^{(m+1)} = \boldsymbol{H}^{\mathsf{T}} g(\boldsymbol{H}\boldsymbol{x}^{(m)})$

Output: $\|\boldsymbol{x}^{(M)}\|_2$

learned during training, and which are estimated per inference.

Unrolled step size α: This is the step size inside the unrolled network. For stability purposes, $\alpha < 1/\sigma_{\max}(\boldsymbol{H})$, where $\sigma_{\max}$ is the maximum singular value, $\boldsymbol{H} = [\boldsymbol{H}_1|\boldsymbol{H}_2|\cdots|\boldsymbol{H}_K]$, and $\boldsymbol{H}_k$ is the linear Toeplitz matrix corresponding to the convolution kernels $\boldsymbol{h}_k$. An upper bound on the step size α can be approximated by the iterative power method shown in Algorithm 1.

Sparse regularizer λ: This is the regularization parameter to enforce sparsity on the latent representation estimated at the encoder. This parameter can be set to 0 or a small value when the event onsets are known and enforced by the indicator $\boldsymbol{e}_k^{n,j}$ at the unrolled iterations. However, its presence is crucial in the absence of known support (event onsets).

Baseline $a^{n,j}$: Our model assumes that there is a baseline activity constant over time in each trial for each neuron. This can be estimated by taking the mean activity at the beginning of each trial prior to the appearance of events of interest, followed by the link function $g^{-1}(\cdot)$.

Unrolled layers R: The inference network (Figure S1a) is equivalent to the optimization (4) when $R \to \infty$. However, given computational limitation R is finite. We recommend setting R on the order of 100 when the code support is known, and 1000 when the support is not known.

Evaluation

We note that in simultaneous learning of the kernels and sparse codes in DUNL, it is possible to fit the very same neural firing rate with a right/left-shifted kernel in time along with a left/right-shifted sparse code. Indeed, for kernels with decaying ends to baseline, this can frequently happen and the two rate models are equivalent. Thus, when we evaluate DUNL, we account for this by using cross-correlation as a metric for kernel recovery, and by using a tolerance when computing the event detection hit rate.

Supplementary Methods - Training

Dopamine spiking experiment

There are $N = 40$ optogenetically identified dopamine neurons. Across neurons, the number of trials ranges from $J = 121$ to $J = 302$. For surprise trials, we analyze the data from 1 s before the reward onset and 2.1 s after the onset. Similarly, for expected trials, we consider the data from 1 s before the cue onset up to 0.6 s after the reward onset. For expected trials, the reward is delivered after 1.5 s from the cue onset. We refer the reader to [69] for more information on data acquisition.

We use the Binomial distribution with time-bin resolution of 25 ms for data modeling. We set $K = 3$ to learn three non-negative kernels shared across all neurons and all trials; one to characterize the neural response to the cue, and the other two to characterize salience and value for the reward prediction error responses. Each kernel is 600 ms long in time, and the baseline firing rate $a_{n,j}$ is estimated for a single trial using the 1 s data prior to the event onset (bins with estimated baseline lower than 0.001 are set to 0.001 for stability purposes prior passing through the log link function. Given the learned kernels, from each neuron at each trial, we infer three codes to identify the neural strength response to cue ($k = 1$) and reward ($k = 2, 3$).

In addition to the norm projection $\mathcal{P}(\cdot)$, we apply element-wise $\text{ReLU}_0(\cdot)$ projection after every

Table S2. Parameters for dopamine spiking experiment.

Data			
Sampling rate	1 ms	Trial length	[121, 302]
Number of neurons	40	Number of Trials	[60-156]
Total number of neurons	40	Total number of examples	8,786
Code		**Kernel**	
Non-negativity	False	Non-negativity	True
Sparse regularizer λ (network)	0	Normalization	True
Sparse regularizer λ (loss)	0	Numbers	3
Code support knowledge	True	Length	600 ms (24)
Code Q regularization	False	Smoother	False
Code Q regularization matrix	-	Smoother penalty	-
Code Q regularization period	-	Initialization	Random Normal
Q regularization scale	-	Share among neurons	True
Q regularization norm type	-		
Top k sparsity	-		
Top k period	-		
Adam optimizer		**Other network parameters**	
Number of epochs	15	Model distribution	Binomial
Batch size	32	Time bin resolution	25 ms
Learning rate	0.01	Unrolling non15lin	Shrinkage
Learning rate decay	False	Unrolling number	100
Learning rate decay step	-	Unrolling mode	FISTA
Adam eps	0.001	Unrolling alpha	0.1
Backpropagation type	Truncated		
Truncated iterations	10		

backpropagation (kernel updates) to enforce kernel non-negativity. To enforce the known support for each kernel, the indicator vector for cue code $\boldsymbol{e}_1^{n,j}$ is set to 1 at the cue onset. Similarly, $\boldsymbol{e}_2^{n,j}$ and $\boldsymbol{e}_3^{n,j}$ are set to 1 at the reward onset for each neuron n at each trial j (the event indicators are zero at other time-points). The data, model, and training parameters are summarized in Table S2. For the data-limited scenario, only 685 out of $8{,}786$ total number of trials are used for training; in this case, the kernel smoother penalty is set to 0.0005.

Dopamine calcium experiment

The data is captured from 3 different sessions, each with $N = 6$, 20, and 30 neurons. Data, model, and training information are summarized in Table S3. Below, we explain the modeling in detail.

Given the continuous domain of calcium imaging, we model the data using Gaussian distribution. We learn $K = 5$ kernels; one kernel to characterize the neural activity in response to the odor cue without reward (we call this regret), another kernel for the odor cue in the expected trial prior to the appearance of the reward, and three kernels to model RPEs. Specifically, for RPEs, we use one kernel with non-negative code to model salience, two kernels to model value (one with non-negative and another with non-positive code). We note that we do not enforce any other constraint for the kernels to explicitly model salience or value; the decomposition is natural upon training. Each kernel is 4 s long in time. The baseline firing rate is estimated from the 1 s data interval prior to the first event onset for every trial.

To attribute the kernels to the specific event of interest, we set the indicator vector for cue regret code $\boldsymbol{e}_1^{n,j}$ to 1 at the cue onset on regret trials, and zero, otherwise. Similarly, $\boldsymbol{e}_2^{n,j}$ is set to 1

Table S3. Parameters for dopamine calcium experiment. For code non-negativity, -1,1,2 are for negative, positive, and two-sided, respectively. For kernel non-negativity flag, 0 is for negative/positive, and 1 is for positive.

Data			
Sampling rate	15 Hz	Trial length	4.53 - 15.2 s
Number of neurons	{6, 20, 30}	Number of Trials	{252, 299, 195}
Total number of neurons	56	Total number of examples	13,342
Code		**Kernel**	
Non-negativity	[2,2,1,1,-1]	Non-negativity	[0,0,0,1,1]
Sparse regularizer λ (network)	0	Normalization	True
Sparse regularizer λ (loss)	0	Numbers	5
Support knowledge	True	Length	4 s (60)
Q regularization	True	Smoother	False
Q regularization matrix	see text	Smoother penalty	-
Q regularization period	1	Initialization	Random Normal
Q regularization scale	2.5	Share among neurons	True
Q regularization norm type	2		
Top k sparsity	-		
Top k period	-		
Adam optimizer		**Other network parameters**	
Number of epochs	15	Model distribution	Gaussian
Batch size	8	Time bin resolution	1 ms
Learning rate	0.01	Unrolling nonlin	Shrinkage
Learning rate decay	False	Unrolling number	100
Learning rate decay step	-	Unrolling mode	FISTA
Adam eps	0.001	Unrolling alpha	0.1
Backpropagation type	Truncated		
Truncated iterations	10		

on expected trials at the cue onset, and zero, otherwise. This attribute kernel $\boldsymbol{h}_1$ characterizes the neural response to cue in the absence of a reward, and $\boldsymbol{h}_2$ represents the neural response to cue in expected trials. Furthermore, $\boldsymbol{e}_3^{n,j}$, $\boldsymbol{e}_4^{n,j}$, and $\boldsymbol{e}_5^{n,j}$ are all 1-sparse for trials with reward, and they are non-zero at the reward onset.

We use the structured representation optimization formulation described earlier to discourage codes $\boldsymbol{x}_4$ and $\boldsymbol{x}_5$ to be active at the same time. We use $\tilde{\boldsymbol{x}}^{n,j\top}\boldsymbol{Q}\tilde{\boldsymbol{x}}^{n,j}$ regularization variation with $\tilde{\boldsymbol{x}}^{n,j} \in \mathbb{R}^5$ capturing the amplitude of the non-zero entry of each code, i.e.,

$$\tilde{\boldsymbol{x}}^{n,j\top} = [\|\boldsymbol{x}_1^{n,j\top}\|_2, \|\boldsymbol{x}_2^{n,j\top}\|_2, \|\boldsymbol{x}_3^{n,j\top}\|_2, \|\boldsymbol{x}_4^{n,j\top}\|_2, \|\boldsymbol{x}_5^{n,j\top}\|_2]. \tag{11}$$

Moreover, we set $\boldsymbol{Q}_{4,5} = \boldsymbol{Q}_{5,4} = 2.5$ and other entries of $\boldsymbol{Q}$ is set to 0.

Whisker thalamus spiking experiment

The training and modeling parameters of the whisker spiking experiment are summarized in Table S4. The original data contains data from 17 pairs of neurons and their activities in response to three types of stimuli. Considered neurons are from pair/neuron of $1/2, 2/1, 4/1, 5/1, 6/1, 8/2, 10/2, 16/2, 17/1$ excluding non-responsive neurons or those with very low signal-to-noise ratio [71]. Additionally, the neural characterization is done for stimulus number 3, where the deflection velocity is constant.

In our analysis, we characterized the response of neurons to the stimuli by one kernel. For more re-fined characterization, one may choose to learn two kernels; prior work discussed that

Table S4. Parameters for whisker thalamus experiment.

Data			
Sampling rate	1 ms	Trial length	4000
Number of neurons	10	Number of Trials	(25 train, 25 test)
Total number of neurons	10	Total number of examples	(250 train, 250 test)
Code		**Kernel**	
Non-negativity	True	Non-negativity	False
Sparse regularizer λ (network)	0.03	Normalization	True
Sparse regularizer λ (loss)	0.03	Numbers	1
Support knowledge	False	Length	125 ms (25)
Q regularization	False	Smoother	True
Q regularization matrix	-	Smoother penalty	0.003
Q regularization period	-	Initialization	Stimuli velocity
Q regularization scale	-	Share among neurons	True
Q regularization norm type	-		
Top k sparsity	18		
Top k period	10		
Adam optimizer		**Other network parameters**	
Number of epochs	120	Model distribution	Binomial
Batch size	30	Time bin resolution	5 ms
Learning rate	0.01	Unrolling nonlin	Shrinkage
Learning rate decay	False	Unrolling number	800
Learning rate decay step	-	Unrolling mode	FISTA
Adam eps	0.001	Unrolling alpha	0.5
Backpropagation type	Truncated		
Truncated iterations	0		

each whisker cycle could evoke two response types (one that encodes the caudal direction whisker movement and another that captures the rostral direction movement on the way back to the neutral whisker position) [71].

Olfactory experiment

An experimental session consists of ≈ 250 trials. In each trial, a custom device delivered 50 ms odor pulses of the same peak concentration to the animal's nose at a Poisson-distributed pulse rate between 0.5-4 pulse/s for 5 s. Neural activity in the animal's anterior piriform cortex was recorded with a custom-built 32-channel tetrode drive at a 30 kHz sampling rate using the Open Ephys recording system [99]. The data are downsampled to 1 ms resolution for analysis. Single-unit spiking activities were isolated using Kilosort2 [100]. We isolated 5-40 single units in each recording session. At the end of each session, the entire bundle of tetrodes was lowered by 40 μm to obtain a new set of neurons for the subsequent session. We recorded $C = 770$ neurons during $S = 17$ behavioural sessions from 3 mice. The data and model parameters for this experiment are summarized in Table S5.

The clustering analysis is based on K-means. Figure S8 shows K-means with 3, 4, 5, 6 clusters. We have used 90% of the neurons at random (repeated 40 times) to compute the adjusted random index (ARI); on average, ARI is 0.94, 0.96, 0.95, 0.77, for 3, 4, 5, 6 clusters, respectively. We observed similar clustering effect using spectral clustering with a Radial Basis Function (RBF) kernel.

Table S5. Parameters for olfaction experiment.

Data			
Sampling rate	reduced to 1 ms	Trial length	4500 ms
Number of neurons	770	Number of Trials	≈ 250
Total number of neurons	-	Total number of examples	-
Code		**Kernel**	
Non-negativity	True	Non-negativity	False
Sparse regularizer λ (network)	varies	Normalization	True
Sparse regularizer λ (loss)	0	Numbers	1
Support knowledge	True	Length	1000 ms (20)
Q regularization	False	Smoother	True
Q regularization matrix	-	Smoother penalty	0.1
Q regularization period	-	Initialization	Aligned raster
Q regularization scale	-	Share among neurons	False
Q regularization norm type	-		
Top k sparsity	-		
Top k period	-		
Adam optimizer		**Other network parameters**	
Number of epochs	500	Model distribution	Poisson
Batch size	Full-batch	Time bin resolution	50 ms
Learning rate	0.01	Unrolling nonlin	Shrinkage
Learning rate decay	False	Unrolling number	100
Learning rate decay step	-	Unrolling mode	FISTA
Adam eps	0.001	Unrolling alpha	0.1
Backpropagation type	Full		
Truncated iterations	-		

Simulated model characterization experiment

Table S6 summarized all the modeling and training parameters for simulation on DUNL characterization.

Simulated dopamine spiking experiment

The dopamine spiking simulation follows closely the data from the dopamine spiking real experiment. Table S7 summarizes the parameters of this experiment.

Simulated structured spiking experiment with non-overlapping events

This section summarizes the information on the data used for the comparison of DUNL and LFADS in their ability to capture local characteristics from single trials. Table S8 summarizes the parameters of this experiment.

Simulated unstructured spiking experiment with overlapping events

This section summarizes the information on the experiment demonstrating the ability of DUNL to detect and locally characterize events appearing at random. In this experiment, there are two types of events, each happening three times in a trial. While there is a 200 ms minimum distance between events of the same type, events of different types are allowed to fully overlap. Table S9 summarizes the parameters of this experiment.

Table S6. Parameters for simulated model characterization experiment. The information in the parenthesis are for different time bin resolution scenario of (5 ms, 10 ms, 25 ms, 50 ms).

Data			
Sampling rate	1 ms	Trial length	4000
Number of neurons	1	Number of Trials	25, 50, 100, 250, 500
Total number of neurons	1	Total number of examples	25, 50, 100, 250, 500
Code		**Kernel**	
Non-negativity	True	Non-negativity	False
Sparse regularizer λ (network)	0.03	Normalization	True
Sparse regularizer λ (loss)	0.03	Numbers	1
Support knowledge	-	Length	500 ms (100, 50, 20, 10)
Q regularization	False	Smoother	True
Q regularization matrix	-	Smoother penalty	(0.2, 0.01, 0.004, 0.0002)
Q regularization period	-	Initialization	Sinusoidal shape
Q regularization scale	-	Share among neurons	True
Q regularization norm type	-		
Top k sparsity	5		
Top k period	10		
Adam optimizer		**Other network parameters**	
Number of epochs	15-100	Model distribution	Binomial
Batch size	128	Time bin resolution	(5, 10, 25, 50) ms
Learning rate	0.01	Unrolling nonlin	Shrinkage
Learning rate decay	False	Unrolling number	800
Learning rate decay step	-	Unrolling mode	FISTA
Adam eps	0.001	Unrolling alpha	0.25
Backpropagation type	Truncated		
Truncated iterations	20		

Supplementary Methods - Two-photon Calcium Imaging Data Acquisition

Surgeries

Stereotaxic viral injections and GRIN lens implantation: Surgeries were performed under aseptic conditions. Mice were anesthetized with isoflurane (1–2 at 0.5–1 L.min^{-1}), and local anesthetic (lidocaine (2%)/bupivacaine (0.5%) 1:1 mixture, subcutaneous (s.c.)) was applied at the incision site. Analgesia (buprenorphine for pre-operative treatment, 0.1 mg.kg^{-1}, intraperitoneal (i.p.); ketoprofen for post-operative treatment, 5 mg.kg^{-1}, i.p.) was administered for 3 days after surgery. A custom-made head plate was placed on the well-cleaned and dried skull with adhesive cement (C&B Metabond, Parkell) containing a small amount of charcoal powder. To express the calcium indicator GCaMP in dopamine neurons, AAV5-CAG-FLEX-GCaMP7f (1.8 × 10^{13} particles per ml) was injected unilaterally in the VTA (300 nl, bregma - 3.0 mm AP, 0.5 mm ML, 4.6 mm DV from dura) in two DAT-Cre mice (*Slc6a3*$^{tm1.1(cre)Bkmn}$, Jackson Laboratory, 006660) [101] respectively. A third mouse was a double transgenic resulting from crossing DAT-Cre with Ai148D (B6.Cg-Igs7$^{tm148.1(tetO-GCaMP6f,CAG-tTA2)Hze/J}$, Jackson Laboratory, 030328) [102] for expression of GCaMP6f in dopamine neurons. The injection was done at a rate of approximately 20 nl.min^{-1} for a total of 300 nl using a manual plunger injector (Narishige). For both DAT-Cre and DAT-Cre;Ai148 double transgenic mice, a GRIN lens (0.6 mm in diameter, 7.3 mm length; 1050-004597, Inscopix) was slowly inserted above the VTA after insertion and removal of a 25-gauge needle. The implants were secured with C&B Metabond adhesive cement (Parkell) and dental acrylic (Lang Dental).

Table S7. Parameters for simulated dopamine spiking experiment.

Data			
Sampling rate	1 ms	Trial length	3100 ms
Number of neurons	40	Number of Trials	[14 - 300]
Total number of neurons	40	Total number of examples	[560 - 1200]
Code		**Kernel**	
Non-negativity	False	Non-negativity	True
Sparse regularizer λ (network)	0	Normalization	True
Sparse regularizer λ (loss)	0	Numbers	3
Support knowledge	True	Length	600 ms (24)
Q regularization	False	Smoother	False
Q regularization matrix	-	Smoother penalty	-
Q regularization period	-	Initialization	Random Normal
Q regularization scale	-	Share among neurons	True
Q regularization norm type	-		
Top k sparsity	-		
Top k period	-		
Adam optimizer		**Other network parameters**	
Number of epochs	200	Model distribution	Binomial
Batch size	2	Time bin resolution	25 ms
Learning rate	0.01	Unrolling nonlin	Shrinkage
Learning rate decay	False	Unrolling number	100
Learning rate decay step	-	Unrolling mode	FISTA
Adam eps	0.001	Unrolling alpha	0.1
Backpropagation type	Truncated		
Truncated iterations	10		

Behavioral training and testing protocol

Mice were water-deprived in their home cage 1–2 days before the start of behavioral training, two or more weeks after surgery. During water deprivation, each mouse's weight was maintained above 85% of its original value. Mice were habituated to the head-fixed setup by receiving water every 4 s (6 μl drops) for 3 days, after which association between odors and outcomes started. A mouse lickometer (1020, Sanworks) was used to measure licking as infrared beam breaks. Water valves (LHDA1233115H, The Lee Company) were calibrated, and a custom-made olfactometer based on a valve driver module (1015, Sanworks) and a valve mount manifold (LFMX0510528B and LHDA1221111H valves, The Lee Company) was used for odor delivery. All components were controlled through a Bpod state machine (1027, Sanworks). Odors were diluted in mineral oil (Sigma-Aldrich) at 1:10, and 30 μl of each diluted odor was placed inside a syringe filter (2.7-μm pore size, 6823-1327, GE Healthcare). Odorized air was further diluted at 1:10 and delivered at $1{,}000$ ml.min^{-1}. Odors used for each association were randomly assigned from the following list of odors: isoamyl acetate, p-cymene, ethyl butyrate, (+)-carvone, ($\pm$)-citronellal, α-ionone, L-fenchone. One of these odors was associated with a distribution of reward sizes while a second odor was not paired with any outcome (nothing). For the rewarded odor, after a 2-s trace period, a reward was delivered whose size was taken randomly from a uniform distribution of the following sizes: 0.3, 0.5, 1.2, 2.5, 5.0, 8.0, 11.0 μl. Variable-size non-cued rewards taken from the same distribution were also delivered throughout the sessions. Mice completed one session per day.

Table S8. Parameters for simulated structured spiking experiment for comparison of DUNL with LFADS for their ability to learn local characterization from data.

Data			
Sampling rate	1 ms	Trial length	2000 ms
Number of neurons	1	Number of Trials	25 - 1600
Total number of neurons	1	Total number of examples	25 - 1600
Code		**Kernel**	
Non-negativity	True	Non-negativity	False
Sparse regularizer λ (network)	0.1	Normalization	True
Sparse regularizer λ (loss)	0.1	Numbers	2
Support knowledge	False	Length	400 ms (16)
Q regularization	False	Smoother	True
Q regularization matrix	-	Smoother penalty	0.015
Q regularization period	-	Initialization	Random Normal
Q regularization scale	-	Share among neurons	True
Q regularization norm type	-		
Top k sparsity	1		
Top k period	10		
Adam optimizer		**Other network parameters**	
Number of epochs	125-1500	Model distribution	Binomial
Batch size	128	Time bin resolution	25 ms
Learning rate	0.01	Unrolling nonlin	Shrinkage
Learning rate decay	False	Unrolling number	800
Learning rate decay step	-	Unrolling mode	FISTA
Adam eps	0.001	Unrolling alpha	0.25
Backpropagation type	Truncated		
Truncated iterations	5		

Image acquisition

Imaging was performed using a custom-built two-photon microscope. The microscope was equipped with a diode-pumped, mode-locked Ti:sapphire laser (Mai-Tai, Spectra-Physics). All imaging was done with the laser tuned to 920 nm. Scanning was achieved using a galvanometer and an 8-kHz resonant scanning mirror (adapted confocal microscopy head, Thorlabs). Laser power was controlled using a Pockels Cell (ConOptics 305 with M302RM driver). The average beam power used for imaging was 40–120 mW at the tip of the objective (Plan Fluorite ×20, 0.5 NA, Nikon). Fluorescence photons were reflected using two dichroic beamsplitters (FF757-Di01-55×60 and FF568-Di01-55×73, Semrock), were filtered using a bandpass filter (FF01-525/50-50, Semrock), and were collected using GaAsP photomultiplier tubes (H7422PA-40, Hamamatsu), whose signal was amplified using transimpedance amplifiers (TIA60, Thorlabs). Microscope control and image acquisition were done using ScanImage 4.0 (Vidrio Technologies). Frames with 512×512 pixels were acquired at 15 Hz. Synchronization between behavioral and imaging acquisitions were achieved by triggering microscope acquisition in each trial to minimize photobleaching using a mechanical shutter (SC10, Thorlabs).

Data pre-processing

Acquired images were pre-processed in the following manner. (1) Movement correction was performed using phase correlation image registration implemented in Suite2P[103]. (2) Region-of-interest (ROI) selection was performed manually in FIJI from the mean and standard deviation pro-

Table S9. Parameters for simulated unstructured spiking experiment for detection and deconvolution of two event types.

Data			
Sampling rate	1 ms	Trial length	6000 ms
Number of neurons	1	Number of Trials	25 - 1600
Total number of neurons	1	Total number of examples	25 - 1600
Code		**Kernel**	
Non-negativity	True	Non-negativity	False
Sparse regularizer λ (network)	0.1	Normalization	True
Sparse regularizer λ (loss)	0.1	Numbers	2
Support knowledge	False	Length	400 ms (16)
Q regularization	False	Smoother	True
Q regularization matrix	-	Smoother penalty	0.015
Q regularization period	-	Initialization	Random Normal
Q regularization scale	-	Share among neurons	True
Q regularization norm type	-		
Top k sparsity	3		
Top k period	10		
Adam optimizer		**Other network parameters**	
Number of epochs	200-2000	Model distribution	Binomial
Batch size	128	Time bin resolution	25 ms
Learning rate	0.01	Unrolling nonlin	Shrinkage
Learning rate decay	False	Unrolling number	800
Learning rate decay step	-	Unrolling mode	FISTA
Adam eps	0.001	Unrolling alpha	0.25
Backpropagation type	Truncated		
Truncated iterations	5		

jections of a subset of frames from the entire acquisition, as well as a movie of the frames used to build those projections. (3) Neuropil decontamination was performed with FISSA[104] using four regions around each ROI. The neuropil decontaminated fluorescent signal was then filtered with a 12 point Gaussian kernel with 0.6875 standard deviation. Drift along the session was corrected using the running maximum of the running minimum of a 120 s time window. Then $\Delta F/F_0$ was calculated as $\Delta F/F(t) = \frac{F(t)-F_0(t)}{F_0(t)}$ using as F_0 the 6th running percentile in a window of 40 s. This fluorescent trace was used for further data processing and analysis in the network.

Supplementary Figures

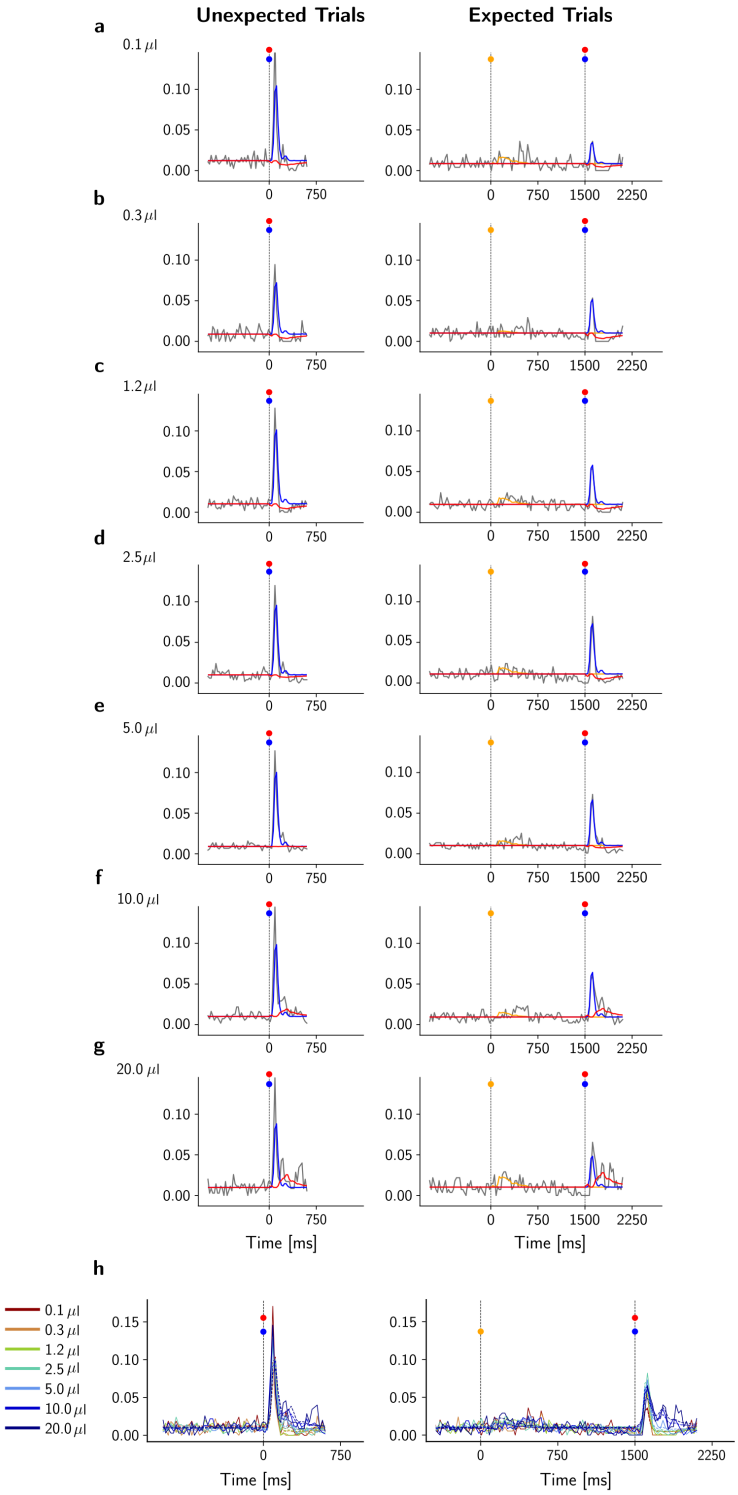


Figure S2. Decomposition of a single dopamine neuron spiking activity. Averaged trial activity for each reward size (**a**-**g**), for unexpected (left) and expected (right) trials (gray traces), were decomposed into Reward I (salience-like, blue) and Reward II (value-like, red) components. The salience kernel contributes in rate estimation of the burst right after the reward onset, and the value kernel contributes in representation of the spikes appearing with around a 100 ms delay. The dip in the neural activity for low reward amount is captured by a negative value code and highlights a negative RPE. **h**, Summary of mean neural response for each trial type and reward size (solid lines) and the corresponding DUNL-reconstructed activity trace (dashed lines).

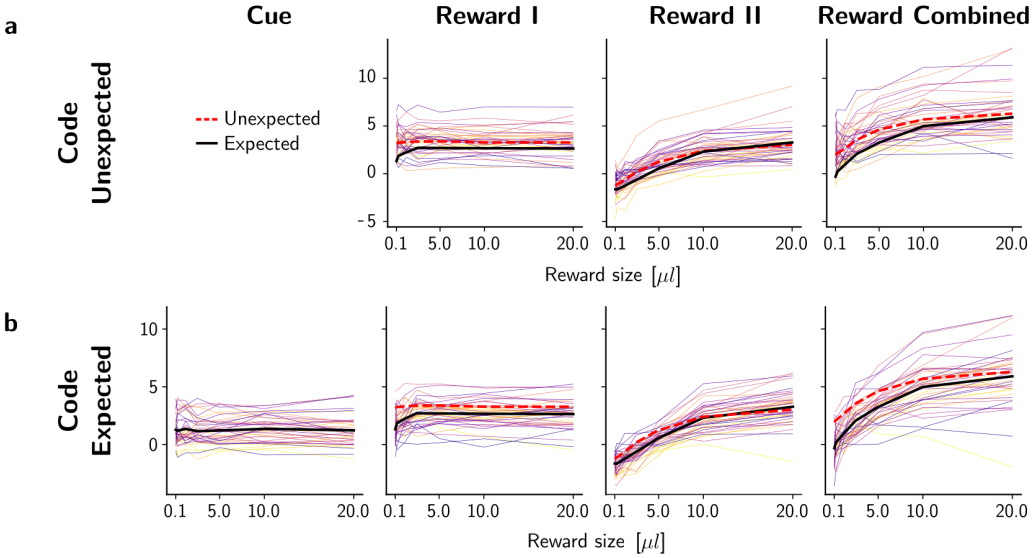


Figure S3. Additional analysis for dopamine spiking data results (Figure 2). Neural code amplitudes as a function of reward size for unexpected (a) and expected trials (b): each line represents one neuron. Compared to Figure 2g, the curves are not normalized here.

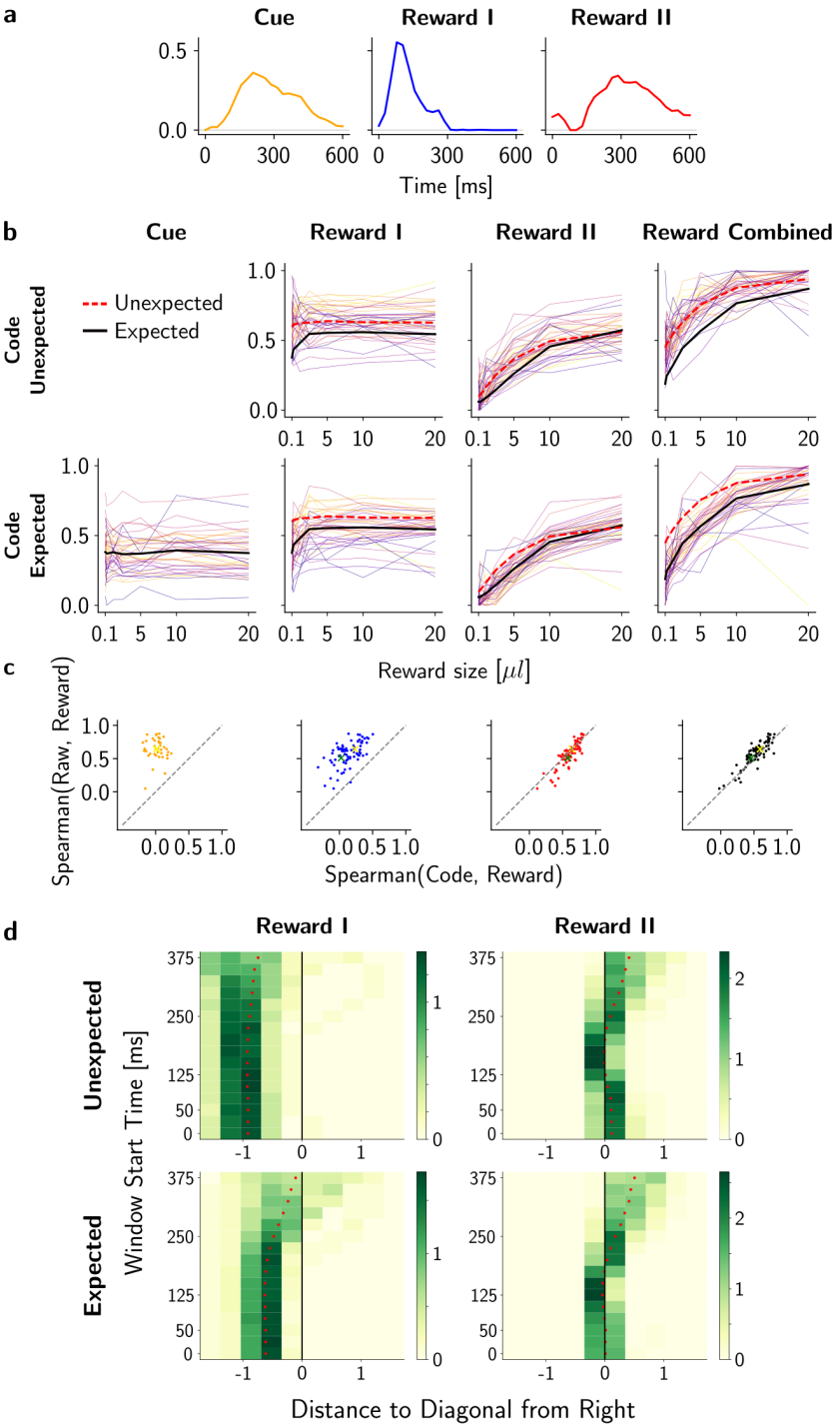


Figure S4. Result of training DUNL with a limited number of trials from dopamine spiking data [69]: **a**, Learned kernels shared across neurons. **b**, Neural code amplitudes as a function of reward size; the figure demonstrates diversity of neural encodings with each line corresponding to one neuron. **c**, Spearman's rank correlation between codes and reward size (x-axis) vs. the windowed average firing rates and reward sizes (y-axis). **d**, Histogram of distance of dots from the diagonal in Spearman's rank correlation from **c**; positive distance means below the diagonal and colorbar shows the normalized probability density function at each bin, such that the integral over the shown range in x-axis is 1.

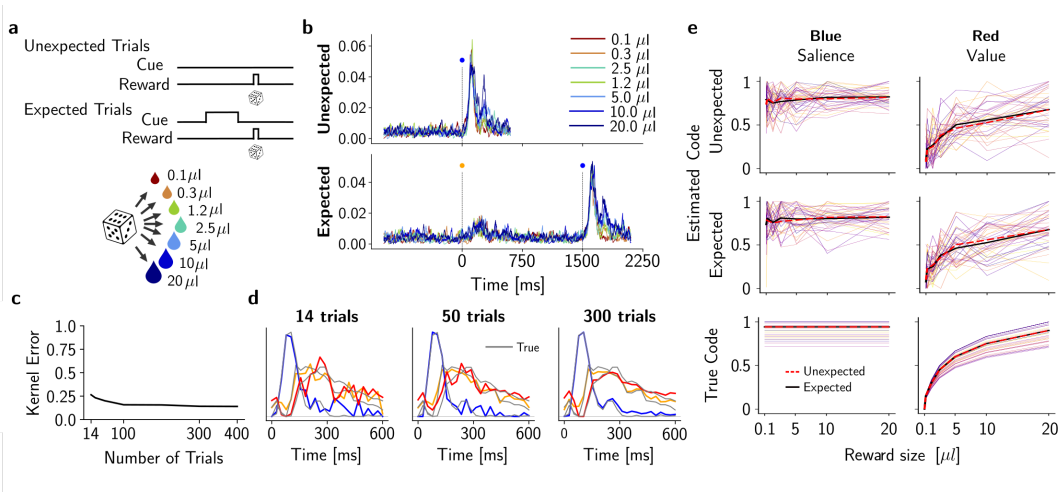


Figure S5. Analysis of kernel quality with the number of trials in simulated dopamine data: **a**, The experiment setup used to generate the data. **b**, PSTH of simulated neurons over each trial type. **c**, The kernel recovery error (i.e., $\sqrt{1-(\text{cosine similarity})^2}$. **d**, Visualization of the learned kernels in color (the true underlying kernels are shown in gray). **e**, DUNL's code estimates as a function of reward sizes (top), and the true underlying code (bottom).

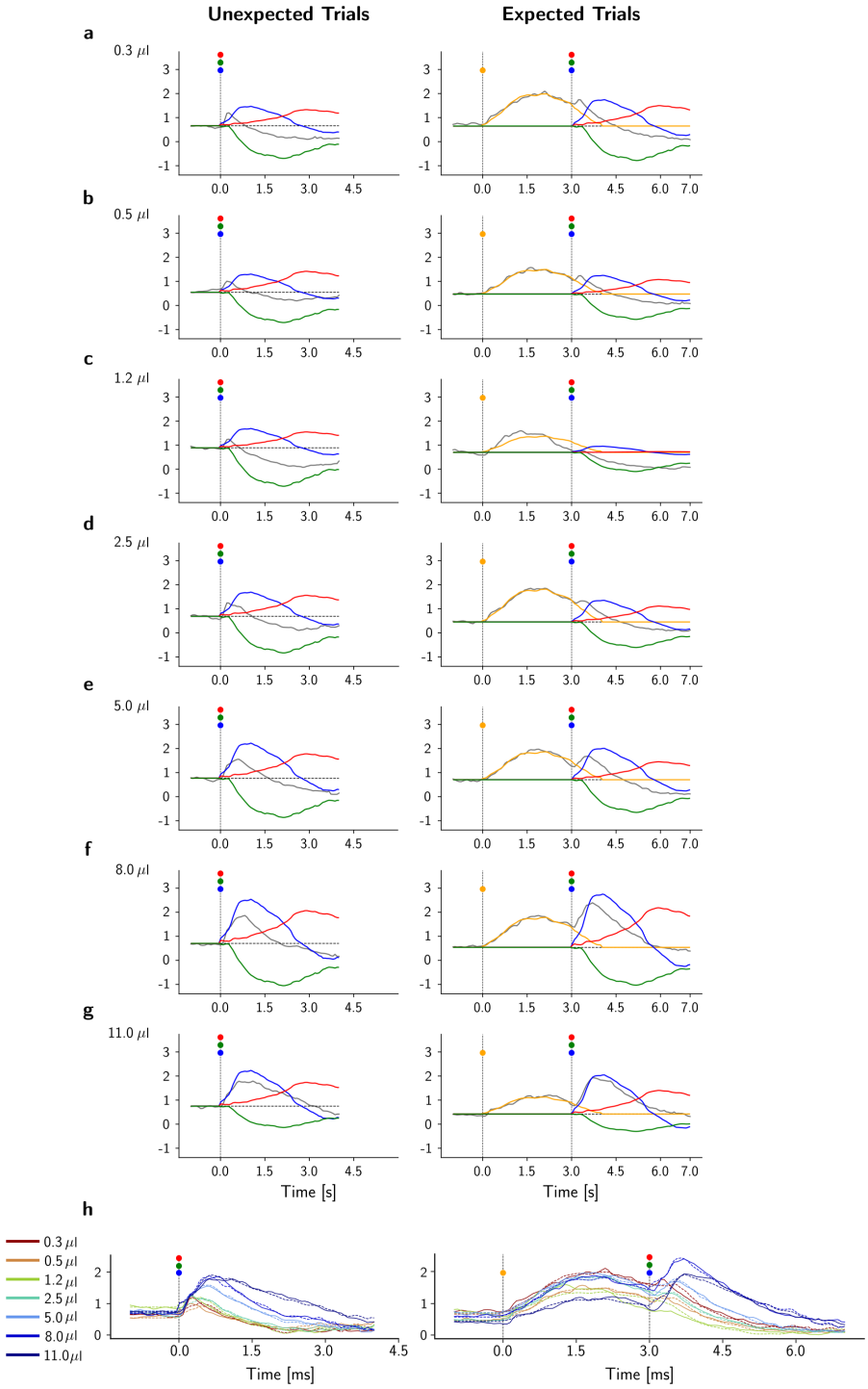


Figure S6. DUNL decomposition of responses from one dopamine neuron recorded using two-photon calcium imaging across reward sizes (**a**-**g**) in unexpected (left) and expected (right) trials. The trial average raw data (gray) and its reconstruction in 4 kernels. Blue models the salience response, red models a positive response for value, and green represents a negative activity for value. **h**, Summary of mean neural response for each trial type and reward size (solid lines) and the corresponding DUNL-reconstructed activity trace (dashed lines).

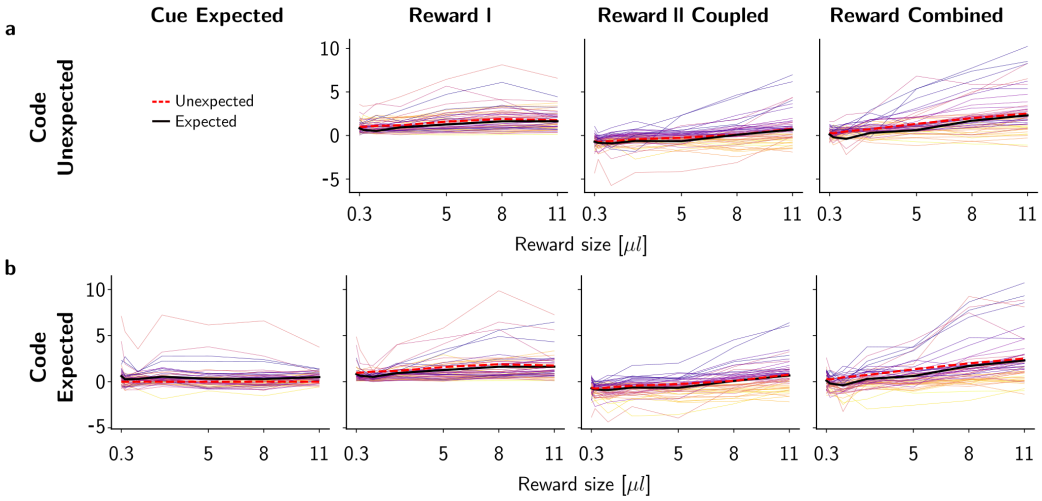


Figure S7. Additional analysis for dopamine calcium signals results (Figure 3). Neural code amplitudes as a function of reward amounts for unexpected (a) and expected trials (b): each line represents one neuron. Compared to Figure 3f, the curves are not normalized here.

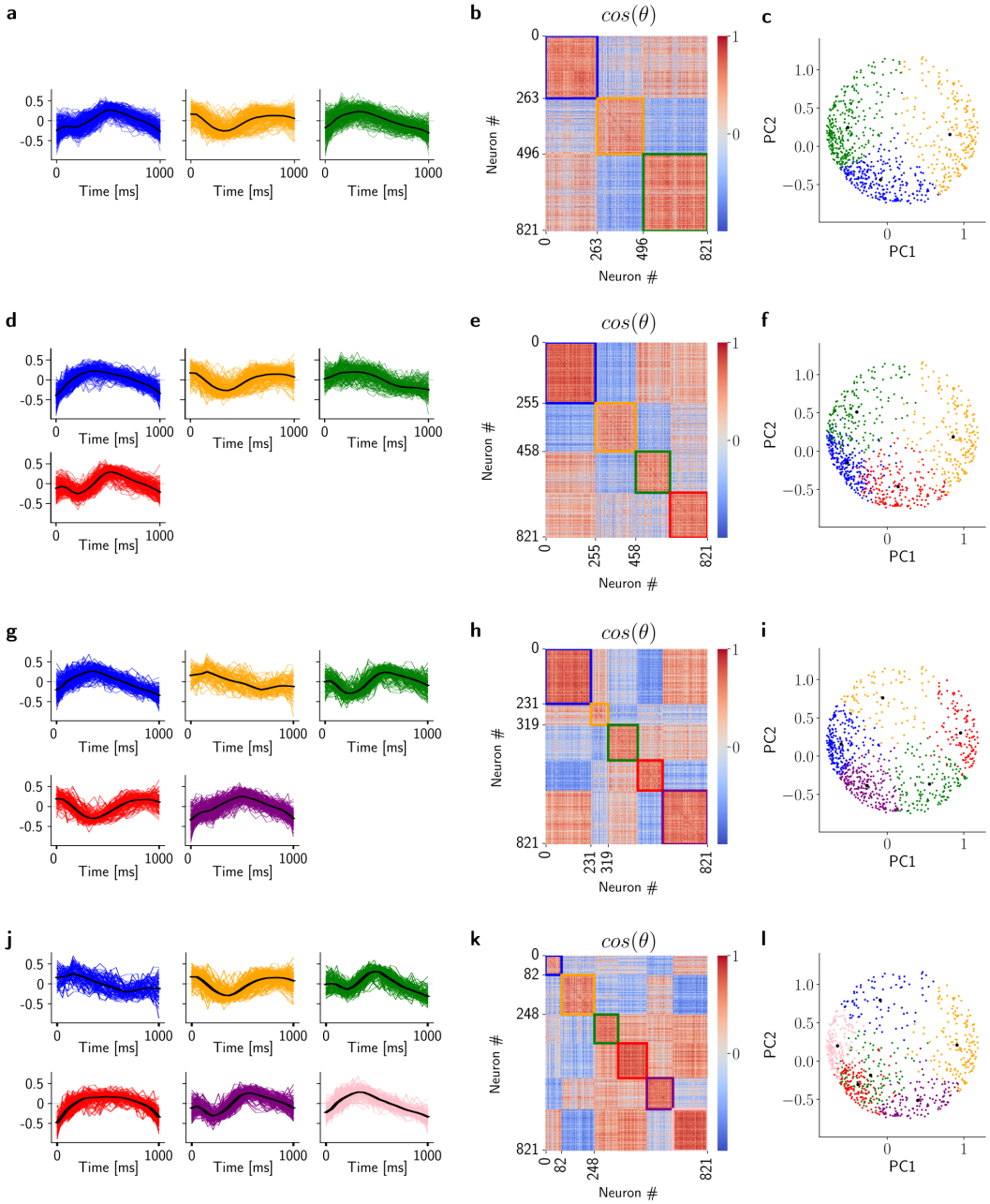


Figure S8. k-means clustering on the DUNL kernels obtained from the piriform cortex neural recordings: 3 (**a**), 4 (**d**), 5 (**g**), and 6 (**j**) clusters; their corresponding similarity matrices using cosine distance (**b**, **e**, **h**, **k**, respectively); and clusters vizualized on the first versus second principal components (**c**, **f**, **i**, **k**, respectively). We observed similar clustering results when using spectral clustering.

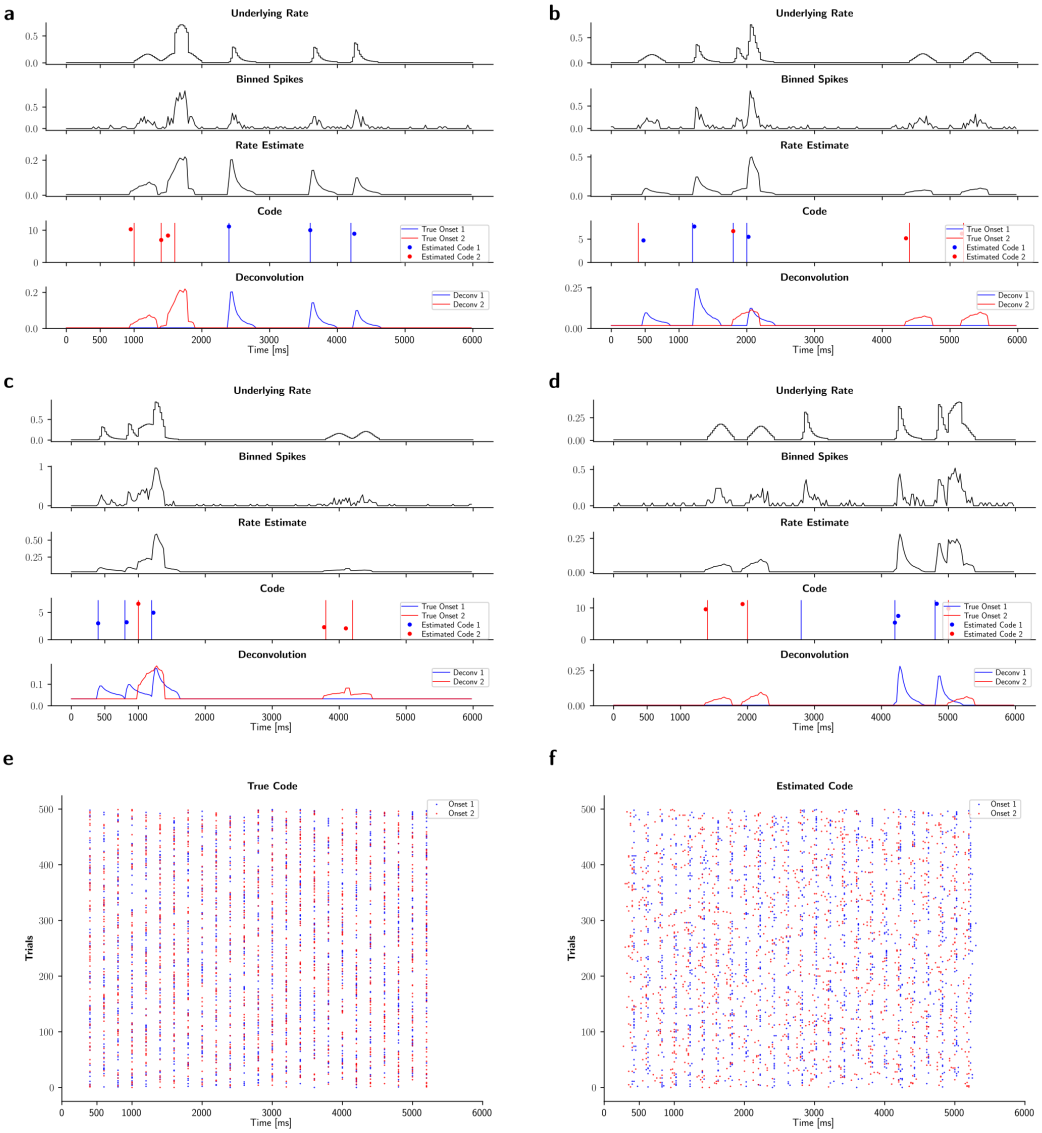


Figure S9. Model characterization with 2 kernels (blue and red). **a-d**, Rate estimation and decomposition of 4 example trials. **e**, The underlying code onsets from both kernels across trials. **f**, The estimated code onsets by DUNL.

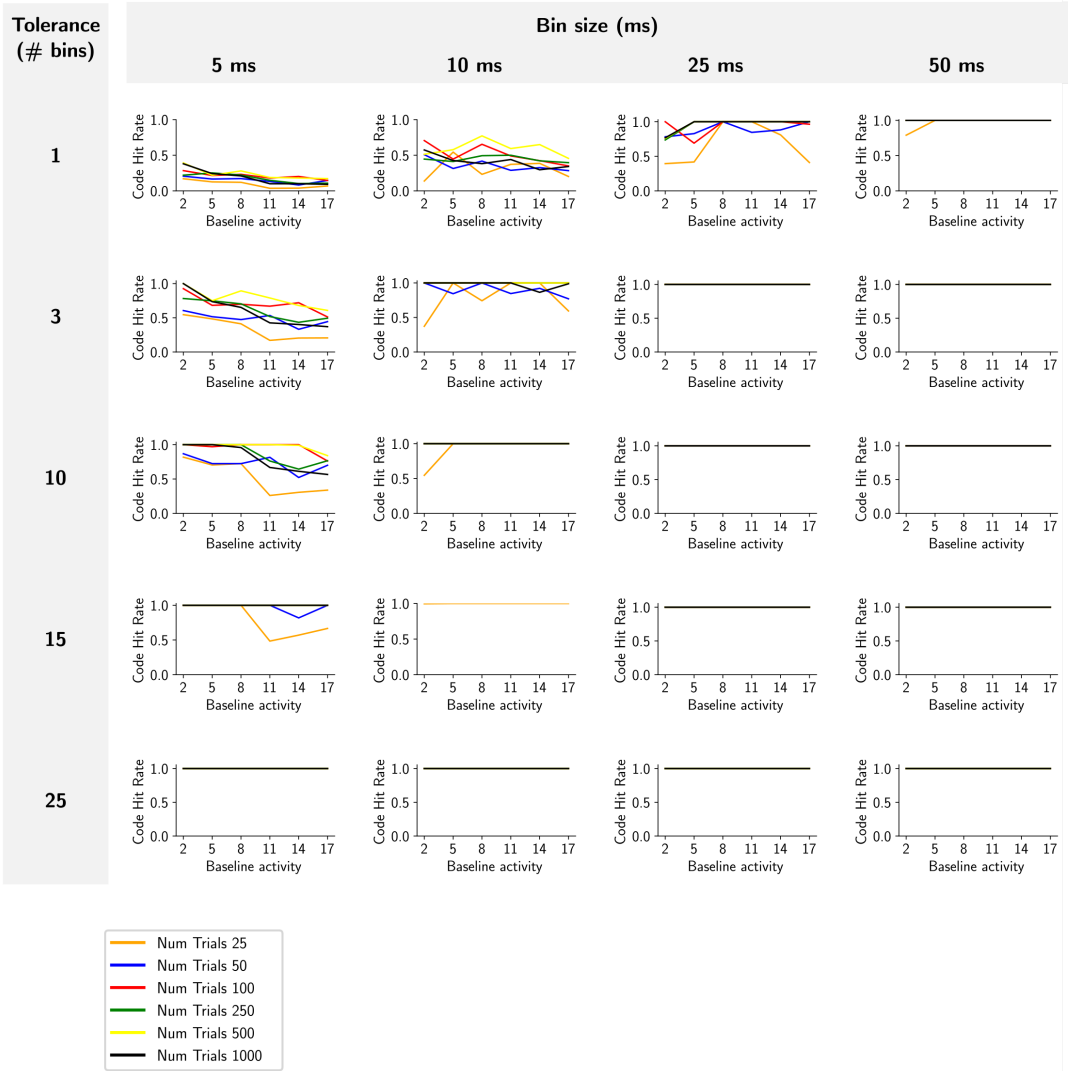


Figure S10. Model characterization. Code hit rate for event identification as a function of bin-size (columns) and time-tolerance (rows).

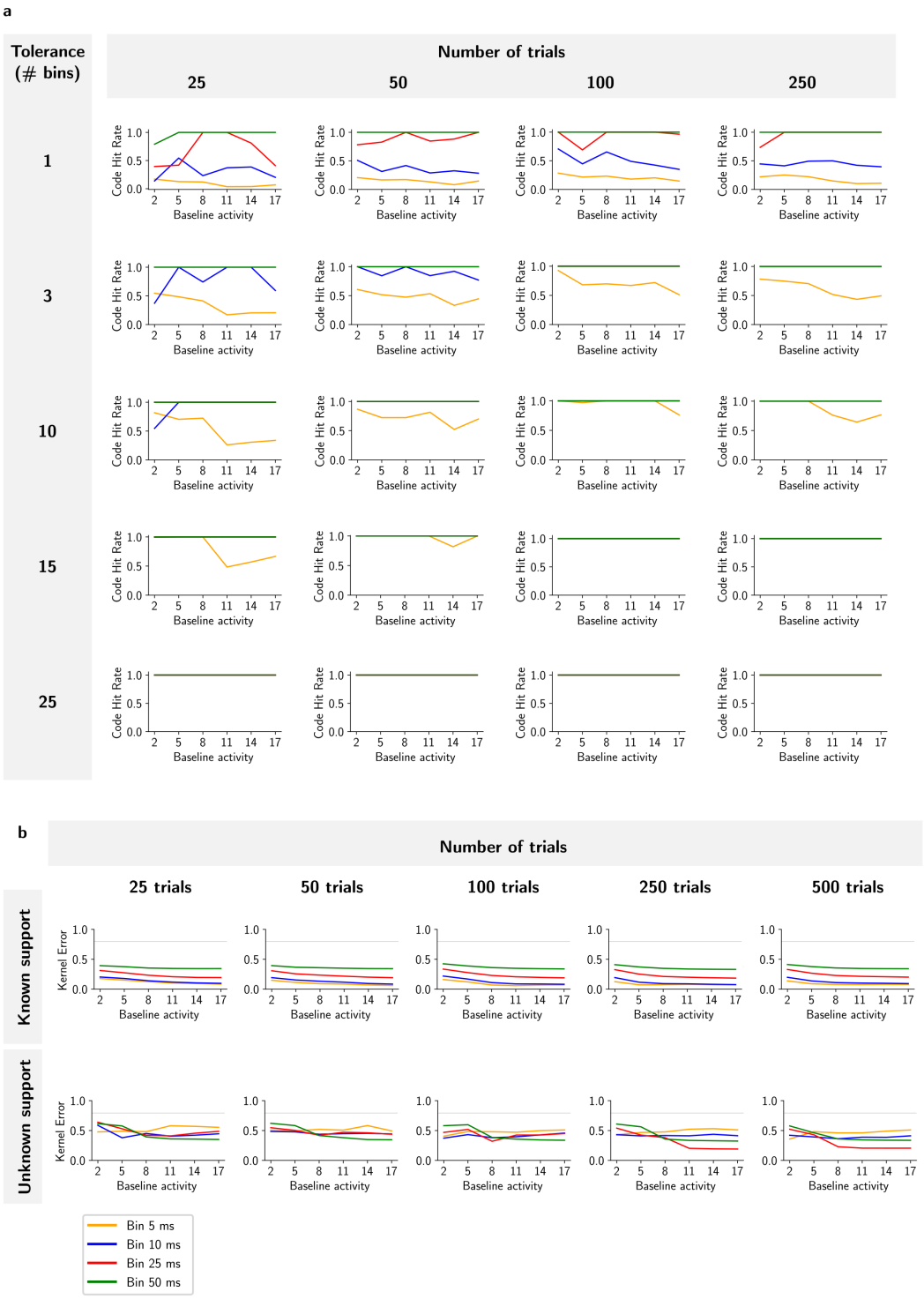


Figure S11. a, Code hit rate for event identification as a function of the number of trials (columns) and time-tolerance (rows). **b**, Kernel recovery error as a function of number of trials available for training for known support (top) and unknown support (bottom) scenarios.

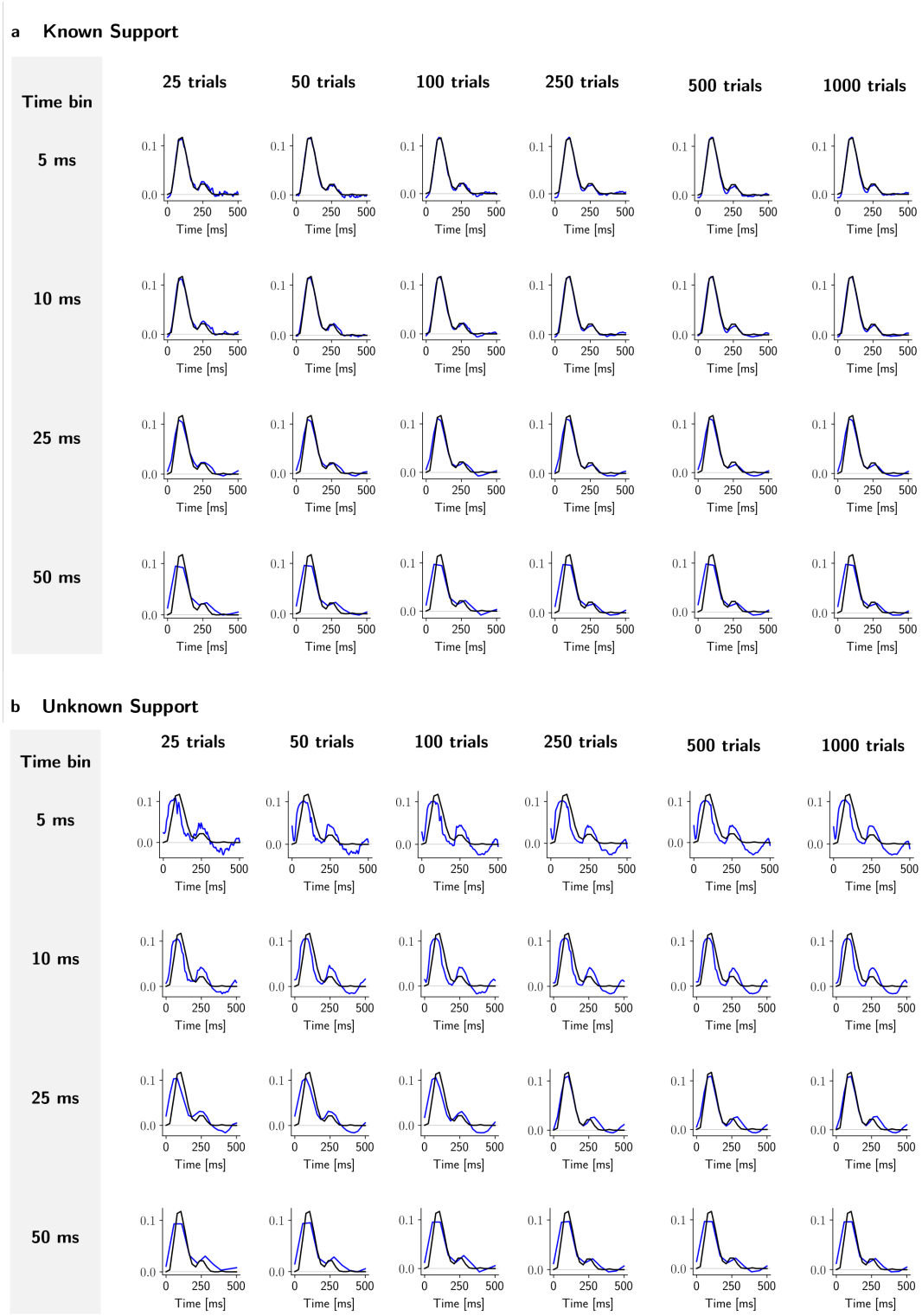


Figure S12. Kernel visualization as a function of trials (columns) and bin-size (rows) when the event onsets (support) are known **a** and unknown **b**.

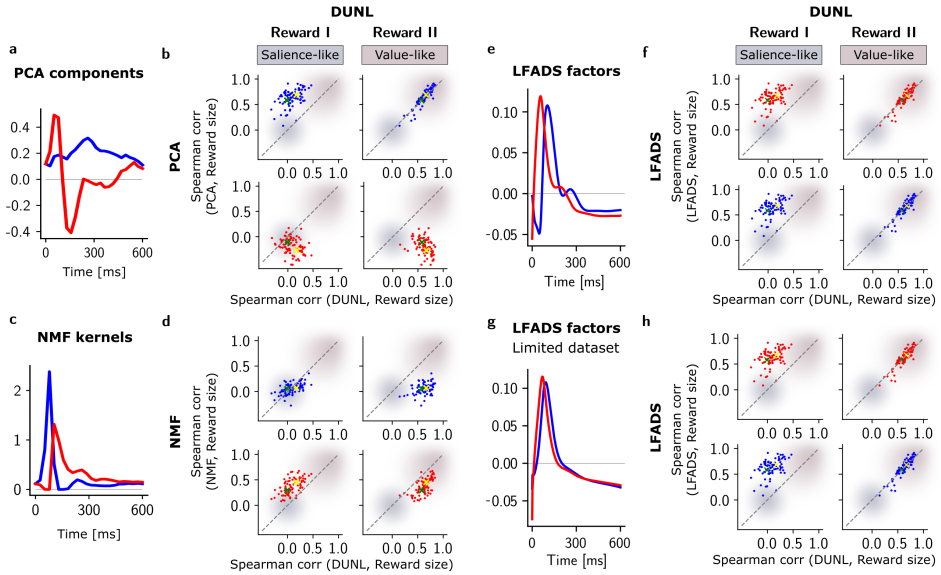


Figure S13. Comparison of DUNL with classical dimensionality reduction (a-d) and a deep learning framework for the dopamine spiking data (e-h). These methods are applied on windowed data of size 600 ms starting from the reward onset from dopamine spiking dataset (Figure 2). PCA is applied to standardized data, NMF is applied to the raw binned data, and LFADS is applied to the raw data. **a-b** PCA. **a**, PCA Kernels with PC1 in blue and PC2 in red color. **b**, Scatter plot of Spearman's rank correlation of DUNL codes for the Reward I (salience-like) and Reward II kernels (value-like) (left and right, respectively) in the x-axis and of Spearman's rank correlation of PCs and reward size on the y-axis. Salience-prone region for both methods is shaded in gray-blue (low correlation with reward size) while Value-prone region for both methods is shaded in mountbatten pink (larger values for the correlation with reward-size). The blue PC contains value information similar to DUNL's Reward II but the red PC contains salience and an anti-correlation with value. **c-d** NMF. **c**, NMF Kernels. **d**, Scatter plot of Spearman's rank correlation of DUNL codes for the Reward I and Reward II kernels (left and right, respectively) in the x-axis and of Spearman's rank correlation of NMF coefficients and reward size on the x-axis. The blue kernel is salience-like, and the red kernel is value-like, but DUNL's Reward II kernel still outperforms the red NMF kernel at representing value. **e-f** LFADS. **e**, The average of two factors over the dataset learned by LFADS (the factors are zero-mean and normalized for visualization purposes). **f**, Spearman's rank correlation of DUNL codes and reward size (x-axis) in comparison to Spearman's rank correlation of the temporal average of the LFADS factors and reward size (y-axis). The comparison of Spearman correlations from DUNL and LFADS shows that both LFADS factors capture similar statistics, only similar to the Reward II kernel in DUNL (value-like); LFADS fails to deconvolve the reward response into salience and value. **g-h** Same as **e-f** for LFADS run on the limited dataset using $< 8\%$ of the data (same dataset as Figure S4). Compared to the full dataset scenario, Spearman correlation results hold; however, certain details on the factors such as the local bump around 200 ms is not captured.

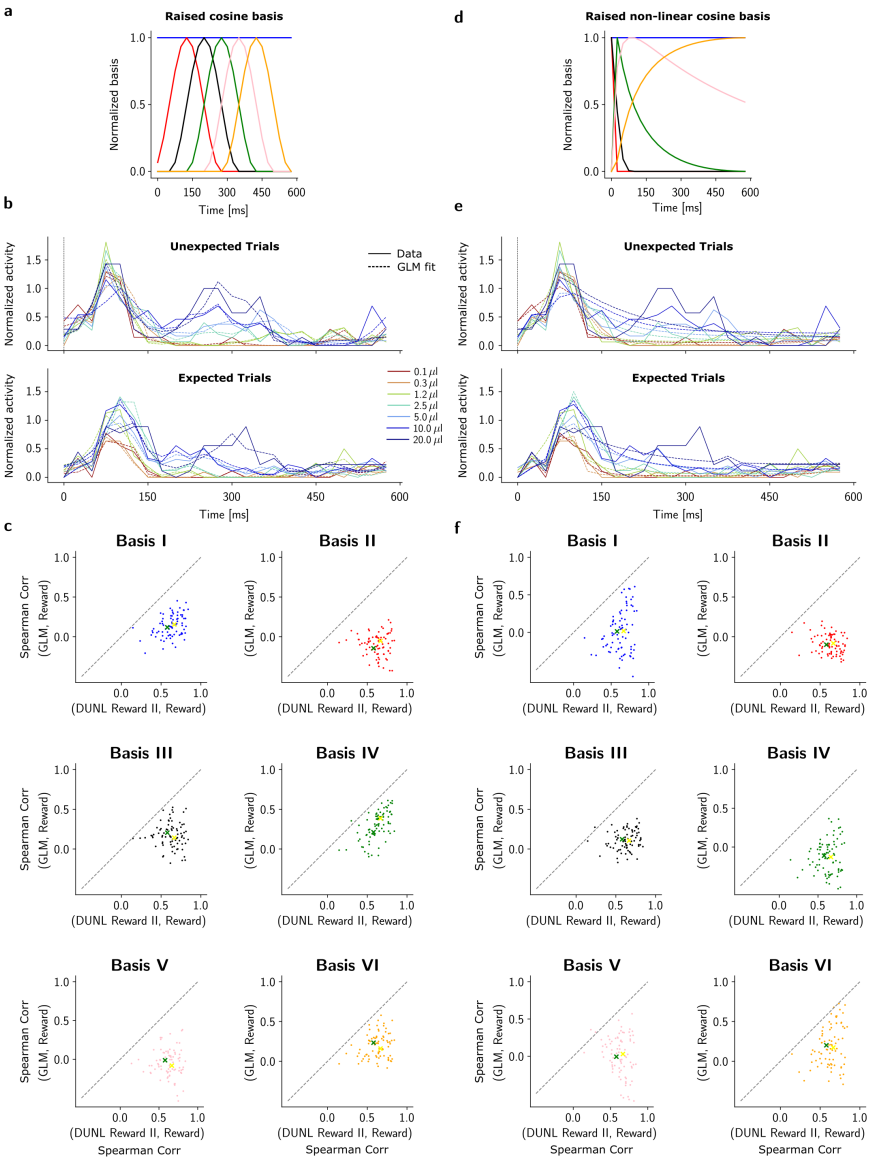


Figure S14. Comparison of DUNL with GLM [58], performing Poisson GLM regression using a set of pre-defined family of basis functions, for the dopamine spiking data. Similar to the comparison with the dimensionality reduction, the methods are applied on windowed data of size 600 ms starting from the reward onset from dopamine spiking data (Figure 2). **a-c**, Raised cosine basis case. **a**, Raised cosine bases (normalized bases with 0/1 min/max are shown). **b**, An example of trial reconstruction by GLM averaged over trial types. **c**, The Spearman's rank correlation of DUNL value code and reward size (x-axis) in comparison to Spearman's rank correlation of coefficients of each of the GLM basis and reward size (y-axis). The comparison shows that neither of the bases is representative of value response; The predefined bases do not offer interpretability from the point of view of deconvolving the reward response into salience and value. The yellow and green markers show the average across *unexpected* and *expected* trials. **d-f**, Nonlinear raised cosine basis case. **d**, Nonlinear raised cosine bases (normalized bases with 0/1 min/max are shown). The first (blue) bases represents the constant bias term. **e**, An example of trial reconstruction by GLM averaged over trial types using the raise cosines. **f**, The Spearman's rank correlation of DUNL value code and reward size (x-axis) in comparison to Spearman's rank correlation of coefficients of each of the GLM bases and reward size (y-axis) for the nonlinear raise cosine case. The presence of dots below the diagonal line indicates that the value code offered by DUNL is a better representative of the reward amount. Overall, this emphasizes the lack of interpretability of GLM with pre-defined family of basis functions within the context of deconvolving the single-trial spiking data into interpretable components.