# *QuantFormer*: Learning to Quantize for Neural Activity Forecasting in Mouse Visual Cortex

**Anonymous authors**
Paper under double-blind review

## Abstract

Understanding complex animal behaviors hinges on deciphering the neural activity patterns within brain circuits, making the ability to forecast neural activity crucial for developing predictive models of brain dynamics. This capability holds immense value for neuroscience, particularly in applications such as real-time optogenetic interventions. While traditional encoding and decoding methods have been used to map external variables to neural activity and vice versa, they focus on interpreting past data. In contrast, neural forecasting aims to predict future neural activity, presenting a unique and challenging task due to the spatiotemporal sparsity and complex dependencies of neural signals. Existing transformer-based forecasting methods, while effective in many domains, struggle to capture the distinctiveness of neural signals characterized by spatiotemporal sparsity and intricate dependencies. To address this challenge, we here introduce *QuantFormer*, a transformer-based model specifically designed for forecasting neural activity from two-photon calcium imaging data. Unlike conventional regression-based approaches, *QuantFormer* reframes the forecasting task as a classification problem via dynamic signal quantization, enabling more effective learning of sparse neural activation patterns. Additionally, *QuantFormer* tackles the challenge of analyzing multivariate signals from an arbitrary number of neurons by incorporating neuron-specific tokens, allowing scalability across diverse neuronal populations.
Trained with unsupervised quantization on the Allen dataset, *QuantFormer* sets a new benchmark in forecasting mouse visual cortex activity. It demonstrates robust performance and generalization across various stimuli and individuals, paving the way for a foundational model in neural signal prediction.
Source code available at https://anonymous.4open.science/r/iclr2025_quantformer-E568.

## 1 Introduction

Complex animal behavior is believed to stem from the electrical activity of coordinated ensembles of neurons within specific brain circuits (Yuste, 2015; Yuste et al., 2024). For example, during sensory perception (Ohki et al., 2005; 2006) and motor coordination (Harpaz et al., 2014; Omlor et al., 2019; Santos et al., 2015), correlated patterns of electrical activity in groups of neurons are observed in the primary sensory and motor cortices (Chen et al., 2024; Inagaki et al., 2022; Panzeri et al., 2022). These activity patterns are structured both spatially and temporally, meaning different subsets of neurons are activated at distinct times. The patterns are further distinguished based on the sensory stimuli or motor outputs they represent (Kondapavulur et al., 2022; Miller et al., 2014; Rule & O'Leary, 2022). Importantly, the activity at any given moment is influenced by the recent history of the neuronal circuit (Boly et al., 2007; Leinweber et al., 2017; Luczak et al., 2022).

Neuronal activity patterns can be recorded in the intact brain using various methods, including electrophysiological recordings (Jun et al., 2017; Steinmetz et al., 2019) and optical techniques such as two-photon microscopy (Denk et al., 1990; Helmchen & Denk, 2005) combined with fluorescent activity reporters (Chen et al., 2013; Dana et al., 2019). These methods enable high-resolution, in vivo imaging of brain cell activity, allowing researchers to observe coordinated neuronal responses during sensory stimulation and motor execution. For instance, studies have shown specific neuronal ensembles encoding stimulus features and behavior in the sensory cortex (Buetfering et al., 2022; Carrillo-Reid et al., 2019), and in the motor cortex during motor programs (Serradj et al., 2023).

A key challenge in neuroscience is developing predictive models that can forecast neuronal activity in a given brain network based on past observations. This task holds significant scientific value, particularly for online closed-loop experiments, such as optogenetics, where real-time adjustments to experimental conditions can enhance intervention effectiveness. Our approach to **forecasting neural activity** differs fundamentally from traditional encoding and decoding methods. Decoding methods, such as those detailed in Azabou et al. (2023); Ye et al. (2023); Antoniades et al. (2023), focus on mapping internal neural variables (e.g., neural activity) to external variables, such as behavior occurring simultaneously with the neural response or the stimulus that elicited it. On the other hand, encoding methods Turishcheva et al. (2024a;b); Li et al. (2023); Xu et al. (2023a); Sinz et al., aim to map external variables to internal neural activity. In contrast, our goal is to model future neural activity without relying on synchronous data, emphasizing the unique challenge of forecasting rather than decoding past stimuli/behaviors or encoding past activity.

The motivation for predicting neuronal activity stems from its demonstrated effectiveness in investigating the sensorimotor cortex of humans and nonhuman primates (Truccolo et al., 2010). However, the application of neural activity forecasting to guide online optogenetic manipulation represents a novel and original advancement in this field. A key element in achieving this goal is leveraging data that is accessible in real-time scenarios. Traditional methods often employ spiking activity data (Schrimpf et al., 2018; Pei et al., 2021; Turishcheva et al., 2024a), which presents challenges due to limited accuracy of real-time spike inference. We thus shift the focus on raw fluorescence traces that provide a direct measure of neuronal activity, improving the precision of predictions and enabling effective manipulation in real-time experimental settings.

In this paper, we propose *QuantFormer*, a transformer-based model for two-photon calcium imaging forecasting using latent space vector quantization. Our approach reframes the forecasting problem as a classification task through vector quantization, enabling the learning of sparse activation spikes. Posing a regression problem as a classification task, even when the data is implicitly continuous, facilitates sparse coding (as already demonstrated in pixel (Van Den Oord et al., 2016b) and audio (Van Den Oord et al., 2016a) spaces), which is crucial given the relative rarity of neuronal activations.

QuantFormer first learns, in a masked auto-encoding fashion (Devlin et al., 2018; He et al., 2022), to compress input neural signals into a sequence of quantized codes, thus allowing self-supervised training by predicting masked items in the sequence. This strategy facilitates the pre-training of the model for downstream tasks by quantization learning while providing a natural way to approach forecasting as the prediction of masked future codes.

Scalability to an arbitrary number of neurons is achieved by learning and prepending a set of neuron-specific tokens to the input. These tokens empower the model to process data from all available neurons without the need to create one model for each neuron or to increase its complexity through multivariate analysis, thus facilitating the effective learning of neural dynamics.

*QuantFormer* was extensively evaluated on the publicly available Allen dataset (de Vries et al., 2020), the only existing benchmark - to our knowledge - providing raw fluorescence traces, and significantly surpassed other state-of-the-art forecasting methods in predicting both short- and long-term neural activity. Ablation studies also confirm the design choices underlying *QuantFormer*.

In summary, the key contributions of this work include:

- **Forecasting neural responses for optogenetic manipulation**: We propose a novel approach that leverages neural activity predictions to guide optogenetic interventions, a completely original concept in the literature.
- **Reframing forecasting as a classification problem**: By employing vector quantization, QuantFormer learns a discrete representation of neural signals, enabling the use of classification techniques to predict sparse neuronal activations effectively.
- **Handling arbitrary neural populations**: Our model uses neuron-specific tokens to facilitate the analysis of multivariate signals from any number of neurons, ensuring scalability and generalization across individuals and sessions.
- **Establishing a foundation model for mouse visual cortex**: Leveraging unsupervised learning on the Allen dataset, QuantFormer demonstrates robust forecasting across different stimuli, individuals, and experimental conditions, laying the groundwork for future research in neural signal prediction.

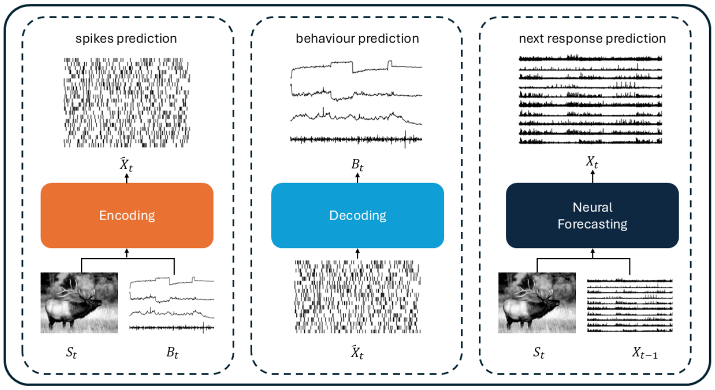

Figure 1: **Comparison of encoding, decoding, and forecasting tasks**. Encoding methods take a stimulus and behavioral variables at time $t$ to predict neural spikes at the same time point. In contrast, decoding methods work do the opposite, using spike responses at time $t$ to predict behavioral variables for that time step. Neural forecasting differs from both, as it uses the stimulus at time $t$ and raw fluorescence traces at time $t-1$ to predict neural responses at time $t$.

## 2 Related work

This paper introduces *QuantFormer*, a transformer-based method trained using self-supervision for neural forecasting on two-photon calcium imaging data.

In deep learning for two-photon calcium imaging, existing methods have predominantly focused on neuron segmentation (Soltanian-Zadeh et al., 2019; Sità et al., 2022; Bao et al., 2021; Xu et al., 2023b), as well as encoding and decoding tasks.
In particular, *decoding methods* map neural activity (internal variables) to external outcomes like behavior Azabou et al. (2023); Ye et al. (2023); Antoniades et al. (2023). These methods, which use neural activity as input, focus on decoding synchronous patterns, such as behaviors occurring alongside neural responses. However, their goal is not to predict future neural dynamics but to link current neural signals to external events.
*Encoding methods* do the opposite, mapping external variables (e.g., stimuli) to neural activity. Approaches such as Turishcheva et al. (2024a;b); Li et al. (2023); Xu et al. (2023a); Sinz et al. predict neural responses based on stimuli, but often rely on trial-averaged data and are not designed to forecast future neural activity on a single-trial basis without the use of synchronous behaviour variables, which are not accessible in online settings.
In contrast, the task we present in this work, *neural forecasting*, aims to predict future neural activity triggered by external inputs (e.g., visual stimuli) based on the neuron states, i.e., on its past neural data. This is essential for online, closed-loop experiments where forecasting future activity is required to manipulate neurons in real time. The difference between encoding, decoding and neural forecasting tasks is clarified in Fig. 1.

In terms of model architecture, *QuantFormer* is aligned with the recent trend in modeling univariate and multivariate single-dimensional time-series signals through transformers. LogTrans (Li et al., 2019) pioneered the use of transformers in univariate forecasting, utilizing causal convolutions to enhance attention locality. Informer (Zhou et al., 2021) improves efficiency in long-sequence forecasting with sparse attention. PatchTST (Nie et al., 2023) handles multichannel data by processing univariate signals in patches, limiting its ability to capture inter-variable correlations. Crossformer (Zhang & Yan, 2022) applies attention across both time and variable dimensions to exploit multivariate dependencies, with cross-window self-attention capturing long-range relationships. Pyraformer (Liu et al., 2021a) introduces pyramidal attention to represent multiresolution features. FEDformer (Zhou et al., 2022) replaces self-attention with Fourier decomposition and wavelet transforms for handling seasonal data patterns.

*QuantFormer* employs transformers where multivariate signals are handled through prepending tokens that encode specific neurons as well as tokens for stimulus encoding. It employs a pre-training procedure based on reconstructing masked input, in **an autoencoder configuration**, thus leveraging the extensive volumes of unlabeled neural signals from two-photon calcium imaging data in a self-supervised learning setting. This strategy has already demonstrated remarkable results in various research areas, including language modeling (Devlin et al., 2018), audio (Huang et al., 2022), and vision (He et al., 2022; Tong et al., 2022). Additionally, during pre-training, we also learn to quantize in a manner similar to image generation (Esser et al., 2021). However, this quantization approach has not been applied to pose forecasting as a code classification task in the neural signal data domain. Though not directly applied to two-photon calcium imaging, methodologies like BrainLM (Ortega Caro et al., 2023), based on the vanilla transformer, and SwiFT (Kim et al., 2023), which leverages Swin transformers (Liu et al., 2021b) trained on fMRI data, are more closely aligned with our approach, as they perform pre-training through self-learning. Both models are then tuned on downstream tasks, though only BrainLM includes brain state forecasting.

Finally, regarding the data, all the encoding and decoding methods discussed above rely on spiking data (as illustrated in Fig. 1). While spiking data reflects a more processed stage of neural signals, its practical application in online settings is limited due to the challenges of real-time spike inference, which often misses a significant portion of spikes (Huang et al., 2021). Given these limitations, we opted for raw fluorescence traces, which can be captured in real-time and circumvent the pitfalls of spike inference, making them more suitable for online neural forecasting. This decision inherently guided us towards the Allen Visual Coding dataset (de Vries et al., 2020), which provides a comprehensive large-scale benchmark for the mouse visual cortex, encompassing both raw fluorescence traces (unlike other existing benchmarks such as BrainScore (Schrimpf et al., 2018), Neural Latents'21 (Pei et al., 2021), and SENSORIUM (Wang et al., 2023)) and spiking activity.

# 3 Method

## 3.1 Overview

Our approach consists of two distinct training phases: *pre-training* through masked auto-encoding and *downstream training* addressing neural activity classification and forecasting. In the pre-training phase, we train a vector-quantized auto-encoder to reconstruct a sequence of non-overlapping neuronal signal patches - following similar procedures from computer vision (Dosovitskiy et al., 2020; He et al., 2022) - a fraction of which is replaced with a `[MASK]` token. The objective of this task is twofold: first, it encourages the model to learn an expressive and reusable feature representation of neuronal signal for downstream tasks; second, it lays the foundation for its use as a forecasting tool, by using `[MASK]` tokens as placeholders for future signal.

In the downstream phase, we employ the pre-trained encoder to predict neuron activations in response to visual stimuli. As mentioned above, this prediction task can be framed as a time series forecasting task, with the objective of predicting the temporal development of a neuronal response. Alternatively, it can be seen as a classification problem, where an *active* (i.e., neuron activation) or *inactive* (i.e., normal neuron activity) label is associated to the neural signal recorded after stimulus visualization.

## 3.2 Problem formulation

Let $\mathcal{S} = \{s_1, \dots, s_S\}$ be the set of stimuli to which subjects can be exposed, and let $\mathcal{N} = \{n_1, \dots, n_N\}$ be the set of neurons under analysis. We define an observation $\mathbf{o} = (\mathbf{x}_b, \mathbf{x}_f, n, s, a)$ to be the set of signals associated to neuron $n \in \mathcal{N}$ when presenting stimulus $s \in \mathcal{S}$: $\mathbf{x}_b \in \mathbb{R}^{L_b}$ is the *baseline activity*, i.e., the neuronal activity *before* the stimulus onset, while $\mathbf{x}_f \in \mathbb{R}^{L_f}$ is the *response activity*, i.e., the neuronal activity *after* the stimulus onset; $a \in \{0, 1\}$ denotes whether neuron $n$ is *active* after the presentation of stimulus $s$, and $L_b$ and $L_f$ denote the temporal length of the different portions in the recorded signals, for a given sample rate $r$. According to Chen et al. (2013), a neuron is marked as *active* ($a = 1$) if the response window has an average gain of 10% over the average baseline luminescence.

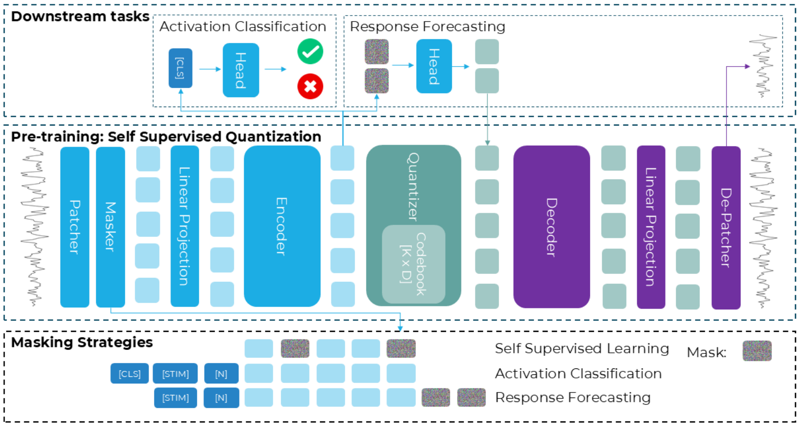

Figure 2: ***QuantFormer* architecture**. During **pre-training** we employ a **self-supervision quantization strategy** that learns to reconstruct the randomly-masked patches along a quantization scheme. For **response forecasting**, `[NEURON]` and `[STIM]` tokens are prepended to the input, and neuronal response patches are masked; the model predicts for the masked patches quantized codes that are converted, through the quantization decoder learned during self-training, to a continuous signal. For **activation classification**, an additional `[CLS]` token is included in the sequence, and its output embedding is fed to the activation classifier.

The ultimate goal of the proposed approach is to predict neuronal activity in response to a stimulus, by modeling either $p(\mathbf{x}_f|\mathbf{x}_b, n, s)$ (when posing the task as time series forecasting) or $p(a|\mathbf{x}_b, n, s)$ (when posing the task as a classification problem).

### 3.3 Pre-training stage

We pose our self-supervised pre-training as a masked auto-encoding task, with the objective of learning a general representation that models neuronal activity patterns with a view towards response forecasting. In order to make the representation as general as possible (as downstream training will be responsible for specialization), in this stage we aim to reconstruct *the entire signal* for an observation $\mathbf{o}$, i.e., the concatenation of $\mathbf{x}_b$ and $\mathbf{x}_f$, while *ignoring both the neuron identity and the presented stimulus*. Formally, let $p(\mathbf{x})$ be the distribution of concatenated baseline and response neuronal signals, with $\mathbf{x} \in \mathbb{R}^{L_b+L_f}$, and let $p(\mathbf{m}|\mathbf{x})$ be a masking function that removes a random portion from $\mathbf{x}$. We want to learn a latent representation $\mathbf{z}$, from which an estimate of the unmasked signal $\mathbf{x}$ can be reconstructed as $p(\mathbf{x}|\mathbf{z})$. Following common practices, we model $p(\mathbf{z}|\mathbf{m})$ and $p(\mathbf{x}|\mathbf{z})$ as an encoder-decoder network sharing the latent representation. Additionally, we introduce a quantization layer (Huh et al., 2023) on the latent representation, in order to enforce that the latent representations are mapped to a predefined set of embeddings. While not strictly necessary for pre-training, quantization yields a twofold usefulness for our purposes. First, it enables a categorical representation of neuronal signal components, allowing downstream tasks to pose forecasting as a classification problem rather than a regression one, which has been shown to be easier to optimize (Van Den Oord et al., 2016a). Second, quantization addresses the sparsity of neuronal activations as it forces the model to focus on a limited number of prototypes, encouraging reuse of codes corresponding to common patterns and potentially reducing the impact of over-represented components. We thus define a set of embeddings $E = \{\mathbf{e}_1, \ldots, \mathbf{e}_K\}$, with $\mathbf{e}_i \in \mathbb{R}^d$ and $K$ being the codebook dimension. In this setting, we distinguish between the continuous distribution $p(\mathbf{z}_e|\mathbf{m})$ produced by the encoder network and the categorical distribution $p(\mathbf{z}_q|\mathbf{m})$ obtained after quantization, defined as:

$$p(\mathbf{z}_q = k|\mathbf{m}) = \begin{cases} 1 & \text{with } k = \arg\min_i \|\mathbf{z}_e(\mathbf{m}) - \mathbf{e}_i\|_2 \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

The decoder is then supposed to learn the distribution of $p(\mathbf{x}|\mathbf{z}_q)$. In order to train the entire model, due to impossibility of backpropagating through the quantization operator, a straight-through estimator (Yin et al., 2019) is employed, directly copying the gradient of the reconstruction loss $\mathcal{L}_{\text{rec}}$ with respect to the quantized representation, $\nabla_{z_q}\mathcal{L}_{\text{rec}}$, to the output of the encoder. We implement $\mathcal{L}_{\text{rec}}$ as the weighted mean squared error between the original unmasked signal $\mathbf{x}$ and the decoder's output, separately taking into account the masked portion $\mathbf{x}_m$ and the unmasked portion $\mathbf{x}_u$:

$$\mathcal{L}_{\text{rec}}\left(\mathbf{x}, \hat{\mathbf{x}}, a\right) = (1 + a\beta)\left[\alpha\mathcal{L}_{\text{MSE}}(\mathbf{x}_m, \hat{\mathbf{x}}_m) + \mathcal{L}_{\text{MSE}}(\mathbf{x}_u, \hat{\mathbf{x}}_u)\right], \tag{2}$$

where $\alpha = 2$ emphasizes the importance of predicting masked elements, and $\beta$ is chosen to compensate for the sparsity of neuron activations, by setting its value depending on the ratio between inactive and active neuron observations. Note that this kind of compensation is possible because the model receives an input sequence for a single neuron at a time: multivariate approaches (e.g., Crossformer (Zhang & Yan, 2022)) are unable to balance active and inactive neurons, since a single input packs multiple neurons together. We complement the reconstruction loss with quantization and commitment losses from (Huh et al., 2023), in order to simultaneously train the encoder and learn the codebook.

From an implementation perspective, we employ transformer architectures to model both the encoder and the decoder. Similarly to common approaches in computer vision, we segment the input signal $\mathbf{x}$ into a set of *patches* $\{\mathbf{x}_1, \ldots, \mathbf{x}_P\}$, with $\mathbf{x}_i \in \mathbb{R}^{(L_b+L_f)/P}$ (padding can be applied to make the dimensionality an integer value). A linear projection transforms each patch $\mathbf{x}_i$ into a *token* $\mathbf{t}_i \in \mathbb{R}^d$, which includes positional encoding. The masking function $\mathbf{m}$ replaces a fraction $P_m$ of tokens with a learnable `[MASK]` token with the same dimensionality as each $\mathbf{t}_i$, producing a masked sequence $\mathbf{m} = \{\mathbf{m}_1, \ldots, \mathbf{m}_P\}$. Following the above formulation:

$$p(\mathbf{m}|\mathbf{x}) = p(\mathbf{m}_1, \ldots, \mathbf{m}_P|\mathbf{t}_1, \ldots, \mathbf{t}_P) = \prod b_i, \tag{3}$$

where each $b_i$ is a Bernoulli random variable with probability $P_m$, such that $\mathbf{m}_i =$ `[MASK]` if $b_i = 1$, and $\mathbf{m}_i = \mathbf{t}_i$ otherwise. A masked input $\mathbf{m}$ is then fed to the transformer encoder $\mathbf{z}_e$ and quantized into $\mathbf{z}_q$, keeping the same dimensionality as the masked input, i.e., $\mathbf{z}_q \in \mathbb{R}^{P\times d}$. The transformer decoder models $p(\mathbf{x}|\mathbf{z}_q)$ and includes a final projection layer that restores the patch dimensionality from the token representation; merging the resulting patches yields the reconstructed neuronal signal.

### 3.4 Downstream tasks

After pre-training the encoder-decoder network, we employ it for adaptation to specific downstream tasks, namely, *neuron activation prediction* and *response forecasting*.

#### 3.4.1 Neuronal activation prediction

Given an observation $\mathbf{o} = (\mathbf{x}_b, \mathbf{x}_s, n, s, a)$, our goal is to predict whether the target neuron responds to the stimulus or not, by modeling $p(a|\mathbf{x}_b, n, s)$. We adapt the pre-trained encoder to this task, with some modifications designed to inject neuron-specific and stimulus-specific knowledge, which was ignored during the self-supervised training. We first introduce a learnable `[CLS]` token (Devlin et al., 2018; Dosovitskiy et al., 2020), whose representation at the output of the encoder is fed to a linear binary classifier, marking the neuron as *active* or *inactive*. We then define a set of stimulus-specific learnable tokens $\{\texttt{[STIM]}_1, \ldots, \texttt{[STIM]}_S\}$, one for each possible stimulus, and a set of neuron-specific learnable tokens $\{\texttt{[NEURON]}_1, \ldots, \texttt{[NEURON]}_N\}$, one for each neuron under analysis; all $\texttt{[STIM]}_i$ and $\texttt{[NEURON]}_j$ tokens are $d$-dimensional vectors, i.e., with the same dimension as the encoder input tokens. Given the observation $\mathbf{o} = (\mathbf{x}_b, \mathbf{x}_s, n, s, a)$, we segment and project the baseline signal $\mathbf{x}_b$ into tokens $\{\mathbf{t}_1, \ldots, \mathbf{t}_P\}$, and then feed the encoder network with $\{\texttt{[CLS]}, \texttt{[NEURON]}_n, \texttt{[STIM]}_s, \mathbf{t}_1, \ldots, \mathbf{t}_P\}$.

Feeding neuron and stimulus identifiers to the encoder is a key aspect of the approach: since our backbone does not inherently handle multivariate data, we compensate for this lack by providing learnable neuron identifiers, making the model able to learn distinct activation patterns for each neuron; similarly, stimuli identifiers provide a means for the model to discover the specific stimuli a neuron responds to. Also, we only feed the baseline signal $\mathbf{x}_b$ to the encoder, since the response $\mathbf{x}_f$ likely contains neuron activation information, which would defeat the purpose of the classifier.

The transformer-based architecture also allows us to tackle this task in two different training settings: *prompt-tuning* and *fine-tuning*. With prompt-tuning, only soft prompts (i.e., `[CLS]`, all $\texttt{[STIM]}_i$ and all $\texttt{[NEURON]}_j$) can be optimized, while the encoder remains frozen. With fine-tuning, all encoder parameters can be updates. In this task, neither the quantizer nor the decoder are used.

### 3.4.2 Neuronal response forecasting

The objective of this task is to model $p(\mathbf{x}_f|\mathbf{x}_b, n, s)$, in order to predict the response time series $\mathbf{x}_f$ from the baseline signal $\mathbf{x}_b$, preceding the stimulus onset. A possible approach to this problem consists in using the pre-trained encoder-decoder network, by masking all input tokens related to the portion of signals to be predicted, and read the forecast signal as the encoder output. However, as mentioned above, the pre-trained model lacks neuron/stimulus specialization, which is needed to handle different neuronal activity dynamics. Moreover, while the pre-trained encoder is trained to capture the underlying patterns of the input data for filling in missing information, this does not necessarily imply the capability to directly predict future values of a time series. To address these issues, we act in two ways: similarly to the previous task, $\texttt{[STIM]}_i$ and $\texttt{[NEURON]}_j$ tokens are added to the beginning of the sequence, to provide the model with specific information; second, rather than fine-tuning the entire model, we append a classification network after the encoder for predicting the codebook indices corresponding to masked tokens only. This approach, besides simplifying the architecture, can be specifically tailored to understand the dynamics of neuronal activity post-stimulus.

Given an input observation $\mathbf{o} = (\mathbf{x}_b, \mathbf{x}_s, n, s, a)$, we convert the baseline signal $\mathbf{x}_b$ into tokens $\{\mathbf{t}_1, \ldots, \mathbf{t}_P\}$, and construct the encoder input as $\{\texttt{[NEURON]}_n, \texttt{[STIM]}_s, \mathbf{t}_1, \ldots, \mathbf{t}_P, \texttt{[MASK]}, \ldots, \texttt{[MASK]}\}$: the number of $\texttt{[MASK]}$ tokens, $M$, depends on the length $L_f$ of the response signal. $\texttt{[STIM]}_i$ and $\texttt{[NEURON]}_j$ tokens are learned separately from their counterparts in the activation classification downstream task. We denote the output of the encoder corresponding to masked tokens as $\{\mathbf{h}_1, \ldots, \mathbf{h}_M\}$, and feed it to a classification network $\phi$, implemented as a multi-layer perceptron. We compute the set of targets $\{y_1, \ldots, y_M\}$, with $y_i \in \{1, \ldots, K\}$, by feeding the full signal, i.e., the concatenation of $\mathbf{x}_b$ and $\mathbf{x}_f$ to the original pre-trained encoder, reading out the quantization indeces into which the response portion is encoded. We then train the classifier and learn the soft prompts by optimizing the cumulative cross-entropy loss over masked tokens:

$$\mathcal{L}_{\text{rf}} = -\sum_{i=1}^{M} \log \phi\left(\mathbf{h}_i\right)_{y_i} \tag{4}$$

with $\phi\left(\mathbf{h}_i\right)_c$ being the $c$-th component of the predicted class distribution for the $i$-th masked token. At inference time, the predicted codebook indeces replace the masked tokens and the entire sequence is fed to the pre-trained decoder for reconstructing the forecast response. Note that both the codebook and the decoder are frozen at this stage, while the encoder can be frozen too (thus learning soft prompts only during training) or optionally fine-tuned.

## 4 Experimental results

### 4.1 Dataset

The Allen Brain Observatory Dataset comprises over 1,300 two-photon calcium imaging experiments, organized into more than 400 containers. Each container, representing all the experimental data from a single mouse, consists of three 90-minute sessions foreseeing the administration of multiple stimuli. We selected 11 containers (i.e., mice) previously used in Sità et al. (2022). Each container has at least three complete sessions available. The original dataset includes various types of stimuli: drifting gratings, static gratings, natural scenes, natural movies, locally sparse noise and spontaneous activity (Observatory, 2017). However, we excluded natural movies, as isolating individual neuron responses is challenging, and spontaneous activity, as it is not stimulus-related. Within each session, every stimulus type is presented across three distinct sub-sessions. Each stimulus may be shown once or multiple times. The presentation of a single stimulus, along with its corresponding neural response, is referred to as a trial. In total, we used 236,808 multivariate signals representing neuron responses from the selected mice (additional details in Table A-1).

Moreover, in our classification downstream task, we examine whether there is a response to a given stimulus within a defined response window. Across all mice and their neurons, we identify a total of 2,287,735 positive (*active*) responses, while normal activity (non-responsive, *inactive*) samples amount to approximately 40 million.

To mitigate temporal correlations and prevent overlap between training and test sets, we partition the data on a per-subsession basis. Specifically, we allocate two subsessions for training and one for testing, with each subsession separated by 10-15 minutes. Furthermore, we ensure that training and testing data are distinctly separated by exposing the mouse to other stimuli during the interim period, thus eliminating potential temporal correlations between signals.

### 4.2 Training procedure and metrics

*QuantFormer* is pre-trained in self-supervision as a masked auto-encoding task through quantization, using data from all subjects. The full model consists of 6 layers and 8 attention heads for both the encoder and the decoder, with a hidden size $d$ of 128 and a mask ratio $P_m$ of 0.2. We train with Adam (Kingma & Ba, 2014) for 50 epochs, a learning rate of $10^{-4}$ and a batch size of 32. In both downstream tasks, we fine-tune the encoder for 100 epochs with a learning rate of $10^{-3}$. The length of baseline and response signals in each observation is respectively 3 seconds and 2 seconds at sample rate $r = 30$, resulting in padded sequence lengths $L_b = 96$ and $L_f = 64$. The number of quantized codes $K$ is set to 32. As we diverge from these values we note that performance decreases in both tasks (see Tables A-2 and A-3 in the *Appendix*). This confirms the sparsity of crucial information in brain signals, which can be encoded with as few as 32 indices (the performance decrease was less sensitive to the dimensionality $d$).

Downstream tasks were conducted separately for each subject and stimulus category, with results reported as mean and standard deviation across all runs. We also evaluate generalization using the *leave-one-category-out* strategy, excluding specific stimulus categories or mice from pre-training and using the excluded data for downstream training. As metrics, we use balanced accuracy, precision, recall, and $F_1$ for classification, and MSE, SMAPE, Pearson correlation and structural similarity index (SSIM) for forecasting.

The selected competitors for our approach, based on code availability and adaptability to the tasks, are Autoformer (Wu et al., 2021), Informer (Zhou et al., 2021), Cross-Former (Zhang & Yan, 2022), and BrainLM (Ortega Caro et al., 2023). We use BrainLM pre-trained on large fMRI data (due to observed similarities between mice and humans (Eppig et al., 2015)), fine-tuned on our data (BrainLM$_{ft}$), and trained from scratch. Additionally, we include a simple LSTM-based baseline, that we empirically found to mostly predict the signal's mean. All experiments are conducted on a workstation with an 8-core CPU, 64GB RAM, and an NVIDIA A6000 GPU (48GB VRAM).

### 4.3 Results

We initially focus on assessing model performance in stimuli response classification; results are shown in Table 1.

Table 1: **Performance on stimuli response classification**. All metrics marked with * have $p \ll 0.01$, while metrics with ** have $p < 0.05$ using one-sided Wilcoxon test.

| **Method** | **Acc** (↑) | $F_1$ (↑) | **Prec** (↑) | **Rec** (↑) |
|---|---|---|---|---|
| LSTM | 61.53 ± 12.75* | 31.00 ± 27.71* | 40.07 ± 39.42* | 35.08 ± 22.91* |
| Autoformer | 58.50 ± 06.11* | 26.21 ± 17.06* | 65.67 ± 27.18* | 17.36 ± 12.47* |
| Informer | 60.77 ± 07.13* | 28.69 ± 15.53* | 55.59 ± 24.82* | 23.11 ± 15.69* |
| BrainLM | 59.66 ± 13.97* | 22.71 ± 32.79* | 27.25 ± 39.39* | 19.56 ± 02.83* |
| BrainLM$_{ft}$ | 62.31 ± 14.67* | 29.33 ± 03.48* | 36.58 ± 42.11* | 24.91 ± 29.69* |
| Cross-former | 75.51 ± 04.45** | 63.89 ± 07.59** | 85.71 ± 03.73 | 51.49 ± 09.03** |
| **QuantFormer** | **77.39** ± 03.88 | **66.94** ± 06.51 | **85.89** ± 00.04 | **55.27** ± 07.82 |

*QuantFormer* and Cross-former showcase superior performance compared to other methods, with ours yielding slightly better performance. However, as we discussed earlier, the primary advantage of the quantization strategy lies in its ability to frame a regression task as a classification task for better modeling outliers such as neuron activation. This is demonstrated in Table 2, where we report forecasting metrics, computed on a gradient-based normalization process that scales each signal by dividing it by its accumulated gradient, ensuring that the signals are on a comparable scale based on their overall rate of change (see Sect. III in the *Appendix* for more details). Figure 3 presents qualitative examples of forecast activation predicted by *QuantFormer* and its competitors.

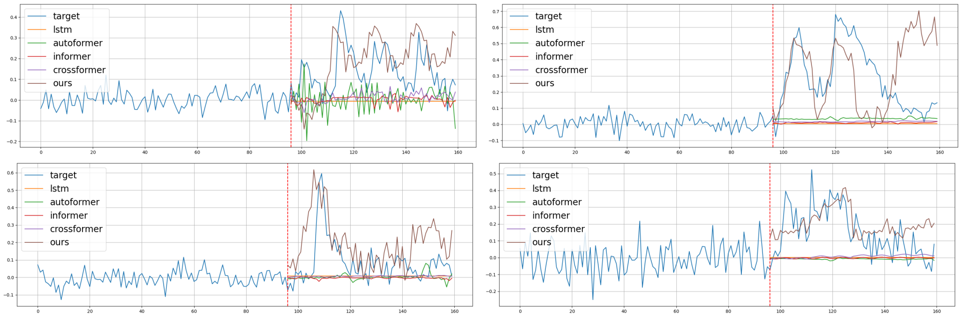

Figure 3: **Qualitative analysis of stimuli response forecasting performance by *QuantFormer* and its competitors**: forecasting examples for each type of stimuli: drifting gratings (top-left), static gratings (top-right), natural scenes (bottom-left) and locally sparse noise (bottom-right). More examples can be found in Section IV of the *Appendix*.

Both quantitative and qualitative results highlight that *QuantFormer* models sparse nature of neural responses better than competitors that predominantly model signals' mean.

Table 2: **Performance on stimuli response forecasting** of *QuantFormer* compared to existing forecasting methods. All metrics marked with * have $p \ll 0.01$, while metrics with ** have $p < 0.05$ using one-sided Wilcoxon test.

| **Method** | **MSE** (↓) | **SMAPE** (↓) | **Corr** (↑) | **SSIM** (↑) |
|---|---|---|---|---|
| LSTM | 60349.339 ± -* | 0.943 ± 0.037* | 0.273 ± 0.129* | 0.003 ± 0.004* |
| AutoFormer | 19.828 ± 11.728* | 0.800 ± 0.033* | 0.312 ± 0.085* | 0.011 ± 0.005* |
| Informer | 0.285 ± 0.376** | 0.707 ± 0.051* | 0.302 ± 0.033* | 0.022 ± 0.017* |
| BrainLM | 0.605 ± 2.852 | 0.701 ± 0.158* | 0.253 ± 0.125* | 0.008 ± 0.034* |
| BrainLM$_{ft}$ | 0.457 ± 0.825 | 0.697 ± 0.183* | 0.337 ± 0.112** | 0.001 ± 0.035* |
| Cross-former | 2.011 ± 2.749* | 0.723 ± 0.062* | 0.292 ± 0.087* | 0.036 ± 0.020* |
| **QuantFormer** | **0.247** ± 0.078 | **0.656** ± 0.137 | **0.338** ± 0.075 | **0.069** ± 0.062 |

Cross-referencing classification (Table 1) and forecasting performance (Table 2), it becomes apparent that *QuantFormer* excels in both tasks, unlike other methods such as Informer (Zhou et al., 2021) and Cross-former (Zhang & Yan, 2022), which specialize in only one. For instance, while Informer exhibits good forecasting metrics, its classification metrics, especially recall, fall short. This may stem from Informer generating responses with activations surpassing the mean signal, but not reaching the threshold for positive classification. Conversely, Cross-former achieves good classification accuracy but struggles with forecasting, likely due to its tendency to predict constant responses that lead to positive classifications while diverging from actual response patterns.
To substantiate the design choices behind *QuantFormer*, we conduct an ablation study to analyze the importance of different components in the model architectures for classification and forecasting tasks, focusing only on "drifting gratings" stimuli for simplicity. We start by evaluating the performance of our encoder backbone when trained from scratch, using cross-entropy for classification and MSE for forecasting (referred to as $Baseline$ in Table 3). The model is provided with pre-stimulus neuronal activity together with the [STIM] token. We then extend this by prepending the sequence with the [NEURON] token (indicated as $Learnable\ tokens$ in Table 3). Additionally, we evaluate the effects of quantization pre-training on model performance compared to pre-training using a standard auto-encoder scheme without quantization (indicated as $AE$ in Table 3[1]). The results demonstrate that forecasting mostly benefits from embedding quantization.

We also explore pre-training benefits for Informer (Zhou et al., 2021), Autoformer (Wu et al., 2021), and Cross-former (Zhang & Yan, 2022) using quantization and standard auto-encoding. However,

[1] Due to space limits, we report only two metrics for classification and two for forecasting.

Table 3: **Ablation study for learnable tokens and quantization on "drifting gratings" stimuli**

| | **Classification** | | **Forecasting** | |
|---|---|---|---|---|
| **Method** | **Acc** (↑) | $F_1$ (↑) | **MSE** (↓) | **Corr** (↑) |
| Baseline | 75.88 ± 4.08 | 64.32 ± 6.62 | 0.021 ± 0.015 | 0.147 ± 0.077 |
| ↪Learnable tokens | 77.53 ± 3.89 | 67.29 ± 6.39 | 0.023 ± 0.018 | 0.147 ± 0.055 |
| ↪AE | 77.22 ± 4.25 | 66.10 ± 7.41 | 0.019 ± 0.014 | 0.207 ± 0.082 |
| ↪Quantization | **77.66** ± 3.78 | **67.42** ± 6.35 | **0.016** ± 0.009 | **0.252** ± 0.095 |

their architectures face two challenges (details in Sect. V of the *Appendix*): 1) combining channel and time information in embeddings creates an information bottleneck, making quantization impractical for temporal patterns; 2) the imbalance between sparse activations and normal signals requires a training strategy targeting individual neurons. Unlike methods that process all neurons simultaneously, *QuantFormer* uses `[NEURON]` tokens to capture individual neuron dynamics.

We then assess the generalization performance of *QuantFormer* on different subjects and stimuli with a leave-one-out strategy. Table 4 shows that *QuantFormer* generalizes effectively across various scenarios, with performance metrics similar to those in Table 2, underscoring its potential as a foundational model for large-scale studies of the mouse visual cortex.

Table 4: **Generalization performance of *QuantFormer* across subjects and stimuli.**

| | **Classification** | | **Forecasting** | |
|---|---|---|---|---|
| | **Acc** (↑) | $F_1$ (↑) | **MSE** (↓) | **Corr** (↑) |
| Subjects | 77.32 ± 4.04 | 67.18 ± 6.58 | 0.367 ± 0.558 | 0.344 ± 0.154 |
| Stimuli | 76.78 ± 3.88 | 67.45 ± 6.26 | 0.411 ± 0.578 | 0.392 ± 0.142 |

In an additional analysis (detailed in the appendix), we examined attention score maps and the latent space of discrete codes and neuron embeddings to understand activation predictions and model interpretability. Attention rollout (Fig. A-8) showed neuron activation predictions are mainly driven by `[NEURON]` token activity, with pre-stimulus patches and the stimulus token adapting to the specific stimuli. 2D t-SNE on neuron embeddings (Fig. A-10) revealed that the `[NEURON]` token encodes neuron-specific statistics like activation probability, while 2D t-SNE on the codebook (Fig. A-9) showed discrete codes capture distinct patterns, but their reconstruction is context-dependent, highlighting sequence context in predictions.

## 5 Conclusion

We presented *QuantFormer*, a transformer-based model using latent space vector quantization to capture sparse neural activity patterns in two-photon calcium imaging. By framing the regression problem as classification and leveraging unsupervised vector quantization, *QuantFormer* outperforms state-of-the-art methods in response classification and forecasting. Trained and tested on a subset of the Allen dataset, it excels in learning sparse activation spikes and capturing long-term dependencies, making it a versatile and robust tool for understanding neural dynamics.

A possible limitation of *QuantFormer* includes the lack of an inhibition mechanism may lead to sequences of high activation responses, contrary to the typical single activation observed in biological neurons. As future work, *QuantFormer* will be trained on the entire Allen dataset, as well as adapted to spiking neural data (in order to use other existing benchmarks), to enhance generalization capability for creating a foundation model for the mouse visual cortex.

## REFERENCES

Antonis Antoniades, Yiyi Yu, Joseph Canzano, William Wang, and Spencer LaVere Smith. Neuroformer: Multimodal and multitask generative pretraining for brain data, 2023. 2, 3

Mehdi Azabou, Vinam Arora, Venkataramana Ganesh, Ximeng Mao, Santosh Nachimuthu, Michael Mendelson, Blake Richards, Matthew Perich, Guillaume Lajoie, and Eva L. Dyer. A unified, scalable framework for neural population decoding. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. 2, 3

Yijun Bao, Somayyeh Soltanian-Zadeh, Sina Farsiu, and Yiyang Gong. Segmentation of neurons from fluorescence calcium recordings beyond real time. *Nature machine intelligence*, 3(7):590–600, 2021. 3

Mélanie Boly, Evelyne Balteau, Caroline Schnakers, Christian Degueldre, Gustave Moonen, André Luxen, Christophe Phillips, Philippe Peigneux, Pierre Maquet, and Steven Laureys. Baseline brain activity fluctuations predict somatosensory perception in humans. *Proceedings of the National Academy of Sciences*, 104(29):12187–12192, 2007. 1

Christina Buetfering, Zihui Zhang, Margarita Pitsiani, John Smallridge, Ellen Boven, Sacha McElligott, and Michael Häusser. Behaviorally relevant decision coding in primary somatosensory cortex neurons. *Nature neuroscience*, 25(9):1225–1236, 2022. 1

Luis Carrillo-Reid, Shuting Han, Weijian Yang, Alejandro Akrouh, and Rafael Yuste. Controlling visually guided behavior by holographic recalling of cortical ensembles. *Cell*, 178(2):447–457, 2019. 1

Susu Chen, Yi Liu, Ziyue Aiden Wang, Jennifer Colonell, Liu D Liu, Han Hou, Nai-Wen Tien, Tim Wang, Timothy Harris, Shaul Druckmann, et al. Brain-wide neural activity underlying memory-guided movement. *Cell*, 187(3):676–691, 2024. 1

Tsai-Wen Chen, Trevor J Wardill, Yi Sun, Stefan R Pulver, Sabine L Renninger, Amy Baohan, Eric R Schreiter, Rex A Kerr, Michael B Orger, Vivek Jayaraman, et al. Ultrasensitive fluorescent proteins for imaging neuronal activity. *Nature*, 499(7458):295–300, 2013. 1, 4

Hod Dana, Yi Sun, Boaz Mohar, Brad K Hulse, Aaron M Kerlin, Jeremy P Hasseman, Getahun Tsegaye, Arthur Tsang, Allan Wong, Ronak Patel, et al. High-performance calcium sensors for imaging activity in neuronal populations and microcompartments. *Nature methods*, 16(7):649–657, 2019. 1

Saskia EJ de Vries, Jerome A Lecoq, Michael A Buice, Peter A Groblewski, Gabriel K Ocker, Michael Oliver, David Feng, Nicholas Cain, Peter Ledochowitsch, Daniel Millman, et al. A large-scale standardized physiological survey reveals functional organization of the mouse visual cortex. *Nature neuroscience*, 23(1):138–151, 2020. 2, 4

Winfried Denk, James H Strickler, and Watt W Webb. Two-photon laser scanning fluorescence microscopy. *Science*, 248(4951):73–76, 1990. 1

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 2, 4, 6

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 4, 6

Janan T. Eppig, Judith A. Blake, Carol J. Bult, James A. Kadin, Joel E. Richardson, and the Mouse Genome Database Group. The mouse genome database (mgd): facilitating mouse as a model for human biology. *Nucleic Acids Research*, 43(D1):D726–D736, 2015. doi: 10.1093/nar/gku967. 8

Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 12873–12883, 2021. 4

Naama Kadmon Harpaz, Tamar Flash, and Ilan Dinstein. Scale-invariant movement encoding in the human motor system. *Neuron*, 81(2):452–462, 2014. 1

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 16000–16009, 2022. 2, 4

Fritjof Helmchen and Winfried Denk. Deep tissue two-photon microscopy. *Nature methods*, 2(12): 932–940, 2005. 1

Lawrence Huang, Peter Ledochowitsch, Ulf Knoblich, Jérôme Lecoq, Gabe J Murphy, R Clay Reid, Saskia EJ de Vries, Christof Koch, Hongkui Zeng, Michael A Buice, Jack Waters, and Lu Li. Relationship between simultaneously recorded spiking activity and fluorescence signal in gcamp6 transgenic mice. *eLife*, 10:e51675, mar 2021. ISSN 2050-084X. doi: 10.7554/eLife.51675. URL https://doi.org/10.7554/eLife.51675. 4

Po-Yao Huang, Hu Xu, Juncheng Li, Alexei Baevski, Michael Auli, Wojciech Galuba, Florian Metze, and Christoph Feichtenhofer. Masked autoencoders that listen. *Advances in Neural Information Processing Systems*, 35:28708–28720, 2022. 4

Minyoung Huh, Brian Cheung, Pulkit Agrawal, and Phillip Isola. Straightening out the straight-through estimator: Overcoming optimization challenges in vector quantized networks. In *International Conference on Machine Learning*. PMLR, 2023. 5, 6

Hidehiko K Inagaki, Susu Chen, Kayvon Daie, Arseny Finkelstein, Lorenzo Fontolan, Sandro Romani, and Karel Svoboda. Neural algorithms and circuits for motor planning. *Annual review of neuroscience*, 45:249–271, 2022. 1

James J Jun, Nicholas A Steinmetz, Joshua H Siegle, Daniel J Denman, Marius Bauza, Brian Barbarits, Albert K Lee, Costas A Anastassiou, Alexandru Andrei, Çağatay Aydın, et al. Fully integrated silicon probes for high-density recording of neural activity. *Nature*, 551(7679):232–236, 2017. 1

Peter Yongho Kim, Junbeom Kwon, Sunghwan Joo, Sangyoon Bae, Donggyu Lee, Yoonho Jung, Shinjae Yoo, Jiook Cha, and Taesup Moon. Swift: Swin 4d fmri transformer. *arXiv preprint arXiv:2307.05916*, 2023. 4

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 8

Sravani Kondapavulur, Stefan M Lemke, David Darevsky, Ling Guo, Preeya Khanna, and Karunesh Ganguly. Transition from predictable to variable motor cortex and striatal ensemble patterning during behavioral exploration. *Nature communications*, 13(1):2450, 2022. 1

Marcus Leinweber, Daniel R Ward, Jan M Sobczak, Alexander Attinger, and Georg B Keller. A sensorimotor circuit in mouse cortex for visual flow predictions. *Neuron*, 95(6):1420–1432, 2017. 1

Bryan M. Li, Isabel Maria Cornacchia, Nathalie Rochefort, and Arno Onken. V1t: large-scale mouse v1 response prediction using a vision transformer. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. URL https://openreview.net/forum?id=qHZs2p4ZD4. 2, 3

Shiyang Li, Xiaoyong Jin, Yao Xuan, Xiyou Zhou, Wenhu Chen, Yu-Xiang Wang, and Xifeng Yan. Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting. *Advances in neural information processing systems*, 32, 2019. 3

Shizhan Liu, Hang Yu, Cong Liao, Jianguo Li, Weiyao Lin, Alex X Liu, and Schahram Dustdar. Pyraformer: Low-complexity pyramidal attention for long-range time series modeling and forecasting. In *International conference on learning representations*, 2021a. 3

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 10012–10022, 2021b. 4

Artur Luczak, Bruce L McNaughton, and Yoshimasa Kubo. Neurons learn by predicting future activity. *Nature machine intelligence*, 4(1):62–72, 2022. 1

Jae-eun Kang Miller, Inbal Ayzenshtat, Luis Carrillo-Reid, and Rafael Yuste. Visual stimuli recruit intrinsically generated cortical ensembles. *Proceedings of the National Academy of Sciences*, 111 (38):E4053–E4061, 2014. 1

Yuqi Nie, Nam H. Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. In *International Conference on Learning Representations*, 2023. 3

Allen Brain Observatory. Technical whitepaper: Stimulus set and response analysis. 2017. URL https://community.brain-map.org/uploads/short-url/uOe7nlLdLLIIivh5PeL8a0g7gV7.pdf. 7

Kenichi Ohki, Sooyoung Chung, Yeang H Ch'ng, Prakash Kara, and R Clay Reid. Functional imaging with cellular resolution reveals precise micro-architecture in visual cortex. *Nature*, 433(7026): 597–603, 2005. 1

Kenichi Ohki, Sooyoung Chung, Prakash Kara, Mark Hübener, Tobias Bonhoeffer, and R Clay Reid. Highly ordered arrangement of single neurons in orientation pinwheels. *Nature*, 442(7105): 925–928, 2006. 1

Wolfgang Omlor, Anna-Sophia Wahl, Pia Sipilä, Henry Lütcke, Balazs Laurenczy, I-Wen Chen, Lazar T Sumanovski, Marcel van't Hoff, Philipp Bethge, Fabian F Voigt, et al. Context-dependent limb movement encoding in neuronal populations of motor cortex. *Nature communications*, 10(1): 4812, 2019. 1

Josue Ortega Caro, Antonio Henrique Oliveira Fonseca, Christopher Averill, Syed A Rizvi, Matteo Rosati, James L Cross, Prateek Mittal, Emanuele Zappala, Daniel Levine, Rahul M Dhodapkar, et al. Brainlm: A foundation model for brain activity recordings. *bioRxiv*, pp. 2023–09, 2023. 4, 8

Stefano Panzeri, Monica Moroni, Houman Safaai, and Christopher D Harvey. The structures and functions of correlations in neural population codes. *Nature Reviews Neuroscience*, 23(9):551–567, 2022. 1

Felix Pei, Joel Ye, David M. Zoltowski, Anqi Wu, Raeed H. Chowdhury, Hansem Sohn, Joseph E. O'Doherty, Krishna V. Shenoy, Matthew T. Kaufman, Mark Churchland, Mehrdad Jazayeri, Lee E. Miller, Jonathan Pillow, Il Memming Park, Eva L. Dyer, and Chethan Pandarinath. Neural latents benchmark '21: Evaluating latent variable models of neural population activity. In *Advances in Neural Information Processing Systems (NeurIPS), Track on Datasets and Benchmarks*, 2021. URL https://arxiv.org/abs/2109.04463. 2, 4

Michael E Rule and Timothy O'Leary. Self-healing codes: How stable neural populations can track continually reconfiguring neural representations. *Proceedings of the National Academy of Sciences*, 119(7):e2106692119, 2022. 1

Fernando J Santos, Rodrigo F Oliveira, Xin Jin, and Rui M Costa. Corticostriatal dynamics encode the refinement of specific behavioral variability during skill learning. *Elife*, 4:e09423, 2015. 1

Martin Schrimpf, Jonas Kubilius, Ha Hong, Najib J. Majaj, Rishi Rajalingham, Elias B. Issa, Kohitij Kar, Pouya Bashivan, Jonathan Prescott-Roy, Franziska Geiger, Kailyn Schmidt, Daniel L. K. Yamins, and James J. DiCarlo. Brain-score: Which artificial neural network for object recognition is most brain-like? *bioRxiv preprint*, 2018. URL https://www.biorxiv.org/content/10.1101/407007v2. 2, 4

Najet Serradj, Francesca Marino, Yunuen Moreno-López, Amanda Bernstein, Sydney Agger, Marwa Soliman, Andrew Sloan, and Edmund Hollis. Task-specific modulation of corticospinal neuron activity during motor learning in mice. *Nature Communications*, 14(1):2708, 2023. 1

Fabian Sinz, Alexander S Ecker, Paul Fahey, Edgar Walker, Erick Cobos, Emmanouil Froudarakis, Dimitri Yatsenko, Zachary Pitkow, Jacob Reimer, and Andreas Tolias. Stimulus domain transfer in recurrent models for large scale cortical population prediction on video. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc. URL https://proceedings.neurips.cc/paper_files/paper/2018/hash/9d684c589d67031a627ad33d59db65e5-Abstract.html. 2, 3

Luca Sità, Marco Brondi, Pedro Lagomarsino de Leon Roig, Sebastiano Curreli, Mariangela Panniello, Dania Vecchia, and Tommaso Fellin. A deep-learning approach for online cell identification and trace extraction in functional two-photon calcium imaging. *Nature Communications*, 13(1):1529, 2022. 3, 7

Somayyeh Soltanian-Zadeh, Kaan Sahingur, Sarah Blau, Yiyang Gong, and Sina Farsiu. Fast and robust active neuron segmentation in two-photon calcium imaging using spatiotemporal deep learning. *Proceedings of the National Academy of Sciences*, 116(17):8554–8563, 2019. 3

Nicholas A Steinmetz, Peter Zatka-Haas, Matteo Carandini, and Kenneth D Harris. Distributed coding of choice, action and engagement across the mouse brain. *Nature*, 576(7786):266–273, 2019. 1

Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training. *Advances in neural information processing systems*, 35:10078–10093, 2022. 4

W. Truccolo, L. Hochberg, and J. Donoghue. Collective dynamics in human and monkey sensorimotor cortex: predicting single neuron spikes. *Nature Neuroscience*, 13(1):105–111, 2010. doi: 10.1038/nn.2455. URL https://doi.org/10.1038/nn.2455. 2

Polina Turishcheva, Paul G. Fahey, Laura Hansel, Rachel Froebe, Kayla Ponder, Michaela Vystrčilová, Konstantin F. Willeke, Mohammad Bashiri, Eric Wang, Zhiwei Ding, Andreas S. Tolias, Fabian H. Sinz, and Alexander S. Ecker. The dynamic sensorium competition for predicting large-scale mouse visual cortex activity from videos, 2024a. URL https://arxiv.org/abs/2305.19654. 2, 3

Polina Turishcheva, Paul G. Fahey, Michaela Vystrčilová, Laura Hansel, Rachel Froebe, Kayla Ponder, Yongrong Qiu, Konstantin F. Willeke, Mohammad Bashiri, Ruslan Baikulov, Yu Zhu, Lei Ma, Shan Yu, Tiejun Huang, Bryan M. Li, Wolf De Wulf, Nina Kudryashova, Matthias H. Hennig, Nathalie L. Rochefort, Arno Onken, Eric Wang, Zhiwei Ding, Andreas S. Tolias, Fabian H. Sinz, and Alexander S Ecker. Retrospective for the dynamic sensorium competition for predicting large-scale mouse primary visual cortex activity from videos, 2024b. URL https://arxiv.org/abs/2407.09100. 2, 3

Aaron Van Den Oord, Sander Dieleman, Nils Zeghidour, Francesco Tacchino, Shariq Ganaie, Gholamreza Haffari, and Andrew Senior. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016a. URL https://arxiv.org/abs/1609.03499. 2, 5

Aäron Van Den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2016b. URL http://proceedings.mlr.press/v48/oord16.html. 2

Eric Y Wang, Paul G Fahey, Kayla Ponder, Zhuokun Ding, Andersen Chang, Taliah Muhammad, Saumil Patel, Zhiwei Ding, Dat Tran, Jiakun Fu, et al. Towards a foundation model of the mouse visual cortex. *bioRxiv*, 2023. 4

Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *CoRR*, abs/2106.13008, 2021. URL https://arxiv.org/abs/2106.13008. 8, 9, 20

Aiwen Xu, Yuchen Hou, Cristopher M Niell, and Michael Beyeler. Multimodal deep learning model unveils behavioral dynamics of V1 activity in freely moving mice. *bioRxivorg*, May 2023a. 2, 3

Zhehao Xu, Yukun Wu, Jiangheng Guan, Shanshan Liang, Junxia Pan, Meng Wang, Qianshuo Hu, Hongbo Jia, Xiaowei Chen, and Xiang Liao. Neuroseg-ii: A deep learning approach for generalized neuron segmentation in two-photon ca2+ imaging. *Frontiers in Cellular Neuroscience*, 17:1127847, 2023b. 3

Joel Ye, Jennifer Collinger, Leila Wehbe, and Robert Gaunt. Neural data transformer 2: Multi-context pretraining for neural spiking activity. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 80352–80374. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/fe51de4e7baf52e743b679e3bdba7905-Paper-Conference.pdf. 2, 3

Penghang Yin, Jiancheng Lyu, Shuai Zhang, Stanley Osher, Yingyong Qi, and Jack Xin. Understanding straight-through estimator in training activation quantized neural nets. *arXiv preprint arXiv:1903.05662*, 2019. 6

Rafael Yuste. From the neuron doctrine to neural networks. *Nature Reviews Neuroscience*, 16(8): 487–497, 2015. 1

Rafael Yuste, Rosa Cossart, and Emre Yaksi. Neuronal ensembles: Building blocks of neural circuits. *Neuron*, 2024. 1

Yunhao Zhang and Junchi Yan. Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting. In *The eleventh international conference on learning representations*, 2022. 3, 6, 8, 9, 20

Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pp. 11106–11115, 2021. 3, 8, 9, 20

Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *International conference on machine learning*, pp. 27268–27286. PMLR, 2022. 3

# A Appendix

## I Dataset detailed information

The Allen Brain Data Observatory is a resource from the Allen Institute for Brain Science that provides a comprehensive collection of data on the mouse visual cortex. This resource is designed to facilitate research and understanding of brain function, particularly in the context of how sensory information is processed. It contains various types of data regarding the mouse visual cortex ranging from cell connectivity to spontaneous neuronal activity and to stimulus-response data.

For the experiments conducted in this work, as explained in the dataset description subsection, we used the responses to the following four types of stimuli:

- **Drifting gratings:** A full field drifting sinusoidal grating at a spatial frequency of 0.04 cycles/degree was presented at 8 different directions (from 0° to 315°, separated by 45°), at 5 temporal frequencies (1, 2, 4, 8, 15 Hz). Each pattern was shown for 2 seconds, followed by 1 second of a uniform gray background before the next pattern appeared. Also blank sweeps (shown every 20 gratings) are included in this type of stimulus. Each condition (combination of temporal frequency and direction) was presented 15 times across session A. The response time was evaluated on a window of 2 seconds after the stimulus onset.
- **Static gratings:** A full field static sinusoidal grating was presented at 6 different orientations (separated by 30°), 5 spatial frequencies (0.02, 0.04, 0.08, 0.16, 0.32 cycles/degree), and 4 phases (0, 0.25, 0.5, 0.75). Each stimulus was presented for 0.25 seconds, without intergray period. Also, blank sweeps were shown every 25 gratings are included in this type of stimulus. Each condition (combination of spatial frequency, orientation and phase) was presented 50 times across session B. The response time was evaluated on a window of 0.5 seconds after the stimulus onset.
- **Locally Sparse Noise:** This type of stimulus consisted of a 16 x 28 array of pixels, each 4.65 degrees on a side. In each medium gray frame of the stimulus (presented for 0.25 seconds) a small number (11) of pixels were randomly changed to be white or black. 9000 different frames was presented once across session C. The response time was evaluated on a window of 0.5 seconds after the stimulus onset.
- **Natural Scenes:** 118 natural images selected from Berkeley Segmentation Dataset (Martin et al., 2001), van Hateren Natural Image Dataset (van Hateren and van der Schaaf, 1998), McGill Calibrated Colour Image Database (Olmos and Kingdom, 2004) were presented in grayscale for 0.25 seconds each, with no inter-image gray period. Each image was presented 50 times, in random order, and the response period was evaluated in 0.5 seconds after the stimulus onset.

The experimental settings is depicted in Fig. A-1.

Table A-1: **Stimuli administration protocol, dataset information and experiment durations.** Window refers to the length of data (seconds after the administration of the corresponding stimulus) considered for response forecasting. Duration is the time in minutes needed for executing a downstream training epoch for the corresponding stimulus type. The average number of neurons per mouse is 241, with a standard deviation of 63.

| [redacted] | **Acquisition protocol** | | | **Dataset information** | | **Duration** |
|---|---|---|---|---|---|---|
| **Stimuli type** | **# instances** | **# trials** | **window (s)** | **# mice** | **# signals** | **# time (m)** |
| Drifting gratings | 41 | 628 | 2 | 11 | 6.908 | 0.43 |
| Static gratings | 120 | 6000 | 0.5 | 11 | 66.000 | 2.4 |
| Locally sparse noise | 9000 | 9000 | 0.5 | 11 | 99.000 | 2.04 |
| Natural scenes | 118 | 5900 | 0.5 | 11 | 64.900 | 1.2 |
| Total | 9.279 | 21.528 | – | 11 | 236.808 | 6.07 |

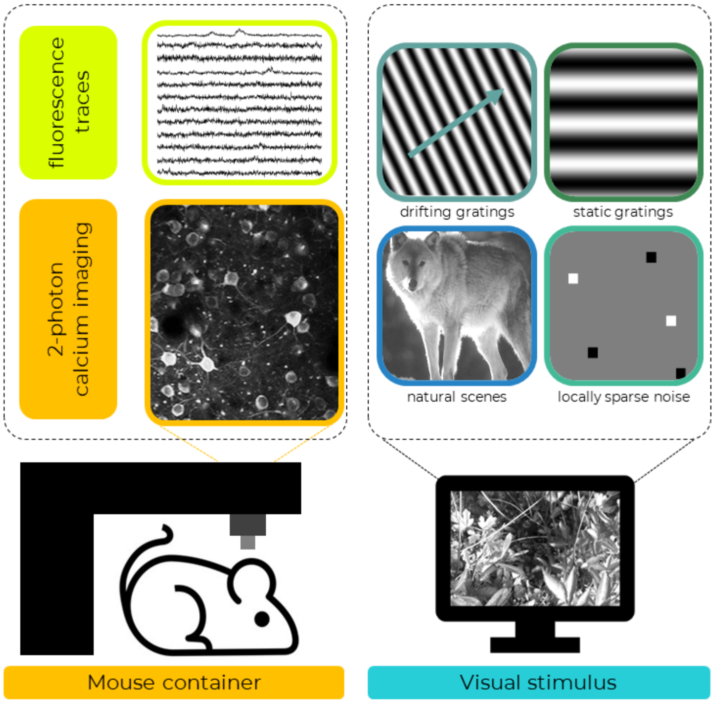

Figure A-1: The Allen dataset. Fluorescence time series are extracted from the two-photon calcium images (*Left*). Examples of the stimuli used (*Right*).

The 11 container ids used for the experiments in this work are: 511507650, 511510667, 511510675, 511510699, 511510718, 511510779, 511510855, 511510989, 526481129, 536323956 and 543677425.

## II Hyperparameter search for quantization and embedding dimensionality

In order to determine the optimal values for the number of quantization indices $(K)$ and embedding dimensionality $(d)$, shared by both the quantized codes and the transformer models, we conduct an exploratory hyperparameter tuning on the responses to "drifting gratings" stimuli only. Such choice was made because this stimuli category needs less time for complete training sessions. First, we fix the value of $d$ to 128 and perform classification and forecasting experiments varying the value of $K$. Our model achieved the best correlation score for a value of $K$ equal to 32. Afterwards, we repeated the same experiments using that number of quantized vectors and we varied the value of parameter $d$ instead.

Table A-2: **$K$ and $d$ parameter value search for forecasting.**

| | | | **Forecasting** | | |
|---|---|---|---|---|---|
| **Value** | **MSE** (↓) | **MAE** (↓) | **SMAPE** (↓) | **Corr** (↑) | **SSIM** (↑) |
| | | | **Quantization indexes $K$ with $d$ = 128** | | |
| $K = 4$ | 0.028 ± 0.014 | 0.132 ± 0.032 | 0.744 ± 0.049 | 0.161 ± 0.050 | 0.010 ± 0.028 |
| $K = 8$ | 0.033 ± 0.006 | 0.161 ± 0.019 | 0.673 ± 0.174 | 0.201 ± 0.065 | 0.060 ± 0.0730 |
| $K = 16$ | 0.015 ± 0.008 | 0.081 ± 0.024 | 0.647 ± 0.063 | 0.219 ± 0.072 | 0.091 ± 0.058 |
| $K = 32$ | 0.026 ± 0.005 | 0.128 ± 0.015 | 0.641 ± 0.154 | 0.257 ± 0.077 | 0.090 ± 0.086 |
| $K = 64$ | 0.032 ± 0.071 | 0.136 ± 0.035 | 0.637 ± 0.105 | 0.218 ± 0.081 | 0.077 ± 0.073 |
| $K = 128$ | 0.040 ± 0.040 | 0.141 ± 0.076 | 0.631 ± 0.108 | 0.221 ± 0.074 | 0.076 ± 0.072 |
| $K = 256$ | 0.028 ± 0.015 | 0.118 ± 0.041 | 0.712 ± 0.103 | 0.149 ± 0.043 | 0.035 ± 0.051 |
| $K = 512$ | 0.027 ± 0.016 | 0.115 ± 0.043 | 0.769 ± 0.105 | 0.168 ± 0.077 | 0.014 ± 0.053 |
| | | | **Embedding dimensionality $d$ with $K = 32$** | | |
| $d = 64$ | 0.031 ± 0.01 | 0.152 ± 0.028 | 0.662 ± 0.18 | 0.157 ± 0.01 | 0.061 ± 0.06 |
| $d = 128$ | 0.026 ± 0.005 | 0.128 ± 0.015 | 0.641 ± 0.154 | 0.257 ± 0.077 | 0.090 ± 0.086 |
| $d = 256$ | 0.051 ± 0.008 | 0.179 ± 0.016 | 0.764 ± 0.062 | 0.234 ± 0.080 | 0.016 ± 0.029 |
| $d = 512$ | 0.027 ± 0.028 | 0.113 ± 0.064 | 0.751 ± 0.111 | 0.134 ± 0.020 | 0.015 ± 0.052 |

Table A-3: **Classification performance for varying values of $K$ and $d$.**

| | **Classification** | |
|---|---|---|
| **Value** | **Acc** (↑) | $F_1$ (↑) |
| | **Quantization indexes $K$ with $d$ = 128** | |
| $K = 4$ | 76.76 ± 4.83 | 66.54 ± 8.80 |
| $K = 8$ | 76.80 ± 4.34 | 66.57 ± 8.04 |
| $K = 16$ | 77.24 ± 4.72 | 67.04 ± 8.37 |
| $K = 32$ | 77.96 ± 4.33 | 66.06 ± 8.32 |
| $K = 64$ | 77.45 ± 4.62 | 65.70 ± 7.32 |
| $K = 128$ | 77.17 ± 4.92 | 66.74 ± 8.31 |
| $K = 256$ | 77.04 ± 4.76 | 66.80 ± 7.67 |
| $K = 512$ | 76.90 ± 4.99 | 66.63 ± 8.08 |
| | **Embedding dimensionality $d$ with $K$ = 32** | |
| $d = 64$ | 76.86 ± 4.40 | 66.63 ± 7.91 |
| $d = 128$ | 77.96 ± 4.33 | 66.06 ± 8.32 |
| $d = 256$ | 77.19 ± 4.66 | 66.67 ± 7.99 |
| $d = 512$ | 64.70 ± 14.06 | 37.02 ± 34.74 |

The optimal values for $K$ and $d$ were decided by the highest value of Pearson correlation obtained in the downstream task of forecasting (Table A-2, best correlation obtained for values $K$ = 32 and $d$ = 128). Table A-3, instead, shows the performance obtained in the classification downstream task for varying values of $K$ and $d$.

## III Forecasting metrics for un-normalized signals

Table A-4 presents forecasting metrics without normalization, where a basic mean signal baseline yields among the highest performance. However, regression metrics on un-normalized signals, given their sparse nature, does not accurately reflect the true forecasting capabilities of tested models. This motivates our normalization method, which normalizes signals dividing them by the sum of their absolute derivatives, emphasizing the rate of change. This approach highlights true forecasting capabilities and ensures that mean-baseline performance sets the lowest boundary (e.g., $\inf$ for MSE, MAE), penalizing models that predict around the average.

Table A-4: **Regression metrics on stimuli response forecasting using un-normalized responses.**

| **Method** | **MSE (↓)** | **MAE (↓)** | **SMAPE (↓)** | **Corr (↑)** | **SSIM (↑)** |
|---|---|---|---|---|---|
| Baseline | 0.095 ± 0.341 | 0.058 ± 0.008 | 0.829 ± 0.009 | 0.335 ± 0.002 | 0.122 ± 0.031 |
| LSTM | 0.093 ± 0.395 | 0.505 ± 0.007 | 0.883 ± 0.062 | 0.252 ± 0.322 | 0.246 ± 0.026 |
| Autoformer | 0.098 ± 0.123 | 0.074 ± 0.022 | 0.062 ± 0.025 | 0.118 ± 0.011 | 0.077 ± 0.037 |
| Informer | 0.097 ± 0.379 | 0.062 ± 0.010 | 0.857 ± 0.049 | 0.118 ± 0.011 | 0.098 ± 0.032 |
| BrainLM | 0.103 ± 0.388 | 0.057 ± 0.008 | 0.902 ± 0.049 | 0.106 ± 0.006 | 0.111 ± 0.031 |
| BrainLM$_{ft}$ | 0.132 ± 0.451 | 0.057 ± 0.008 | 0.858 ± 0.057 | 0.107 ± 0.073 | 0.098 ± 0.042 |
| Cross-former | 0.301 ± 1.210 | 0.060 ± 0.009 | 0.771 ± 0.039 | 0.138 ± 0.032 | 0.096 ± 0.032 |
| **QuantFormer** | 0.445 ± 1.230 | 0.236 ± 0.106 | 1.55 ± 0.082 | 0.138 ± 0.017 | 0.015 ± 0.022 |

## IV Response forecasting examples

Due to space limitations in the main paper here we report more examples of response forecasts of the tested models to all four categories of stimuli (*Natural scenes* in Figure A-2, *Drifting gratings* in Figure A-3, *Static gratings* in Figure A-4 and *Locally sparse noise* in Figure A-5). All examples showcase the superior capability of *QuantFormer* to model neuron activation w.r.t. competitors.

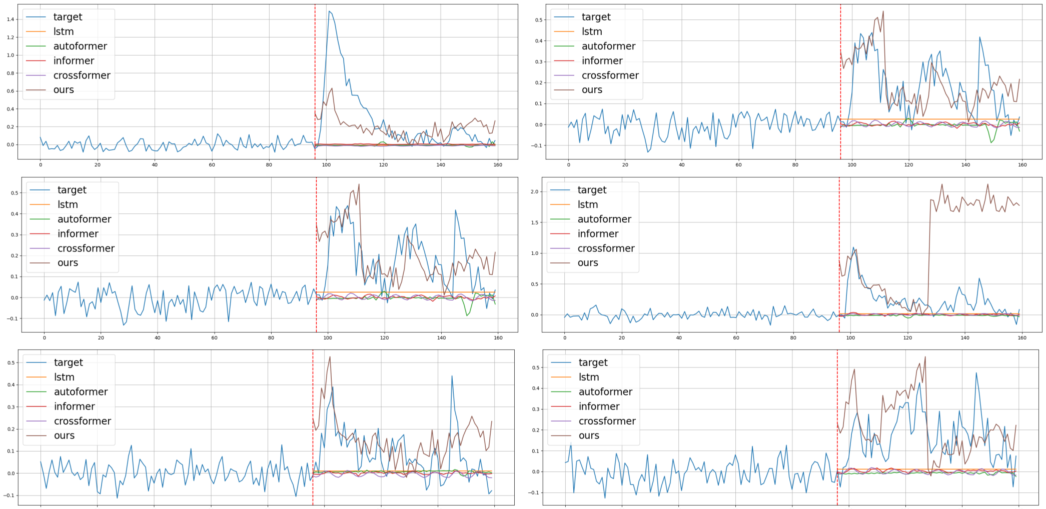

Figure A-2: **Examples of response forecasting by *QuantFormer* and its competitors on natural scenes**.

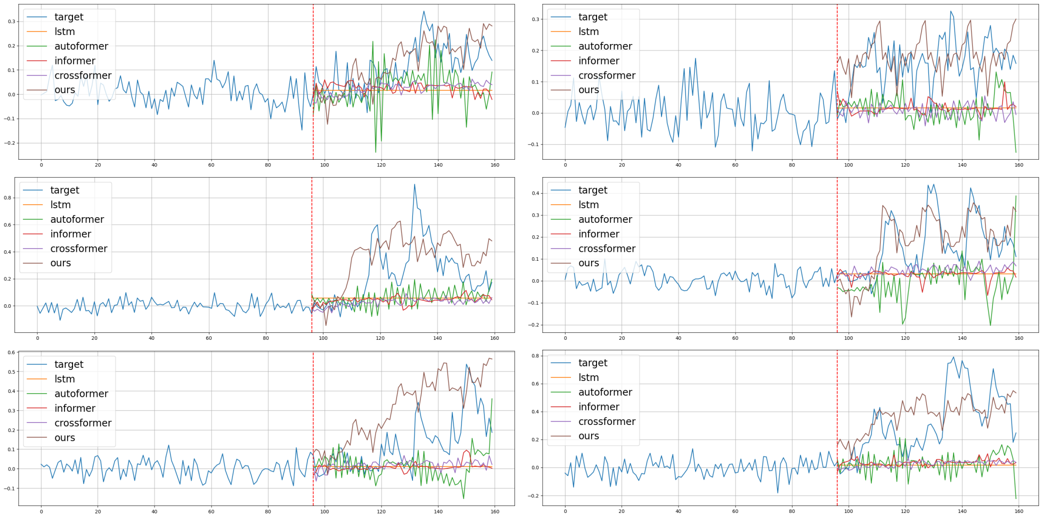

Figure A-3: **Examples of response forecasting by *QuantFormer* and its competitors on drifting gratings**.

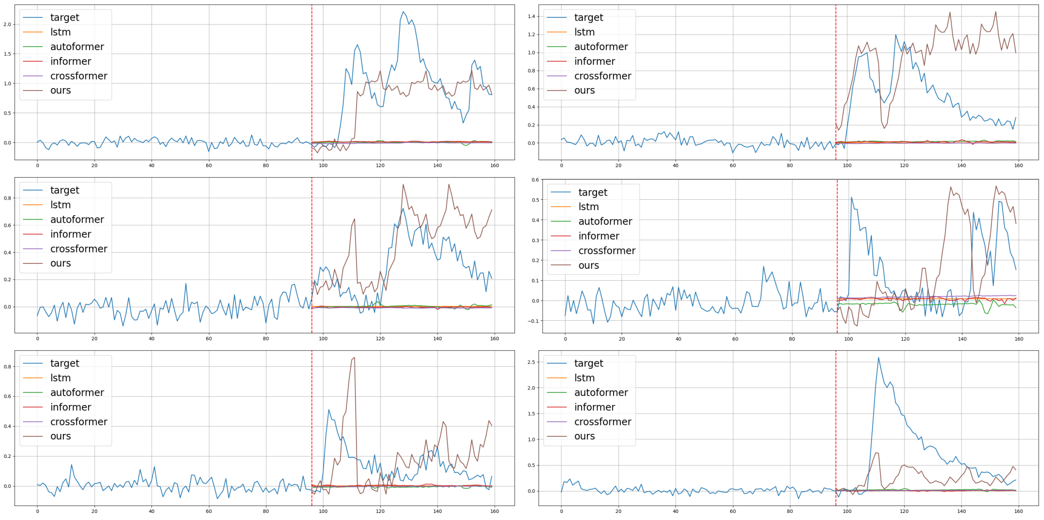

Figure A-4: **Examples of response forecasting by *QuantFormer* and its competitors on static gratings**.

## V Application of self-supervised quantization on competitors

One might question why our pre-training and quantization strategy was not applied to other methods, especially those based on transformer architectures. The primary reason lies in the substantial modifications required to integrate auto-encoding pre-training and quantization into these approaches.

Firstly, quantization is infeasible for Informer (Zhou et al., 2021) and Autoformer (Wu et al., 2021), due to their reliance on embedding layers along the channel dimension, whereas our method embeds temporally patched data. The goal of quantization is to derive robust temporal representations and patterns. Encoding channel combinations with single codes would create an information bottleneck, emphasizing channel patterns over temporal ones.

Secondly, quantization cannot be directly applied to Crossformer (Zhang & Yan, 2022). Although Crossformer performs patching and embedding both channel-wise and temporally, it introduces a two-stage attention mechanism across time and channels. Theoretically, quantization could be

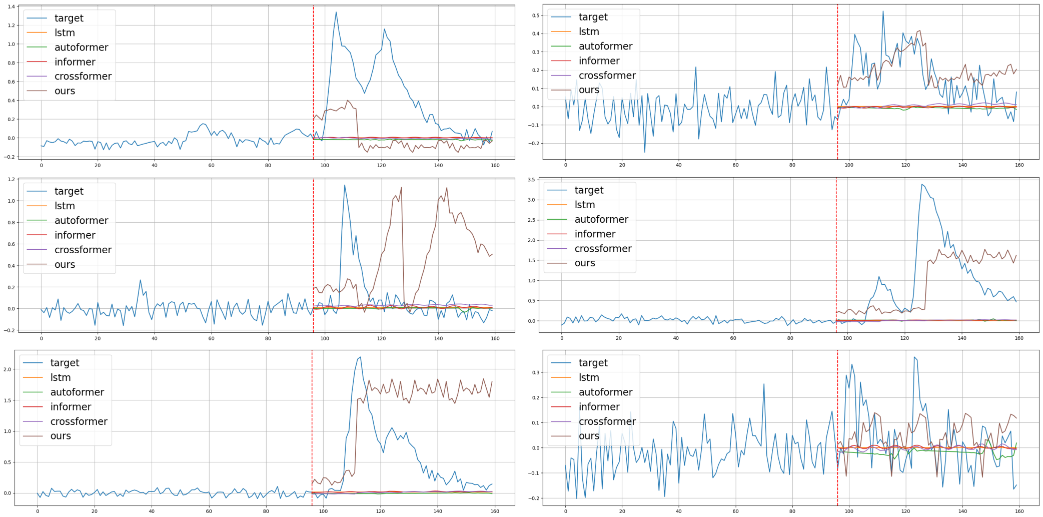

Figure A-5: **Examples of response forecasting by *QuantFormer* and its competitors on locally sparse noise**.

implemented; however, pre-training constraints prevent shuffling, altering, or discarding channels. With only 10% of neurons active per trial, this causes an imbalance during pre-training, leading the quantizer to optimize losses using a limited number of samples for the actual activation. This results in limited number of quantization codes (3) that mostly describe normal activity signals (the majority in the training data), thus leading to a high quantization error, as shown in Fig. A-6. Our approach mitigates this by allowing the exclusion of non-active neurons to maintain data balance during pre-training, thus obtaining a much lower quantization error, as shown in Fig. A-7.

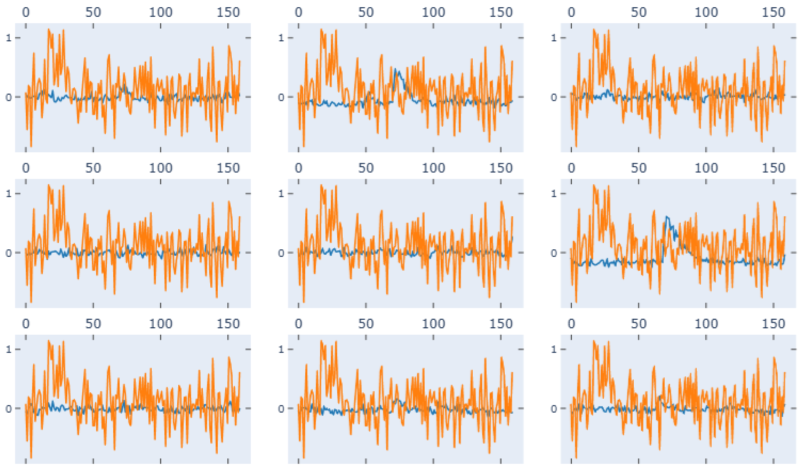

Figure A-6: **Cross-former quantization failure.** In blue, the target signals, while in orange the predicted responses when using quantization.

Furthermore, pretraining itself is problematic for similar reasons. Different containers possess unique channels, necessitating significant alterations to existing methods for effective pretraining. For Autoformer and Informer, each container and experiment would require a dedicated embedding layer to map input channel dimensions into a unified latent space. For Crossformer, introducing a pad token and padding mask might make pretraining feasible, but there would be no consistency in channel order across different containers and experiments. This inconsistency would result in channel attention learning non-generalizable dependencies. Even within a single container, such as a mouse, the number of channels and their order vary across experiments.

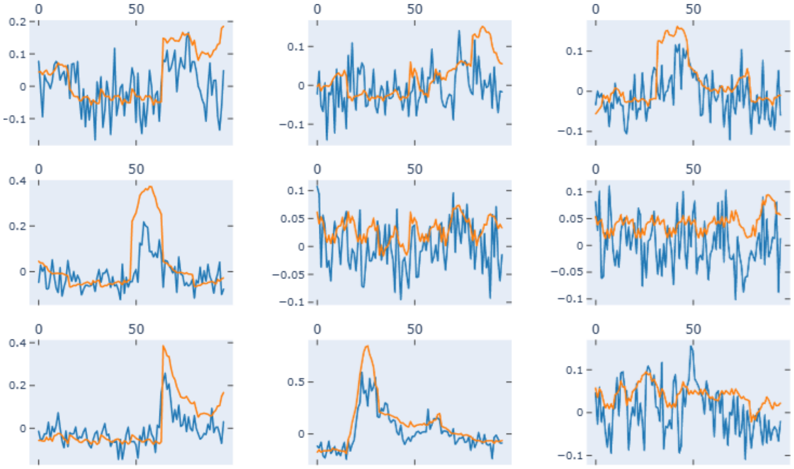

Figure A-7: ***QuantFormer* quantization performance.** In blue, the target signals, while in orange the predicted responses when using quantization.

Thus, our strategy is more appropriate for pretraining, given the inherent challenges and limitations of adapting other methods for this purpose.

## VI Attention maps

We here present in Fig. A-8 attention score maps computed through attention-rollout on *QuantFormer* for neuron activation prediction for all the four types of stimuli: drifting gratings, static gratings, natural scenes, and locally-sparse noise. These maps reveal that [NEURON] token activity predominantly influences predictions, followed by pre-stimulus patches and stimulus token, with the model adapting pre-stimulus information based on the specific stimuli delivered. These attention maps show distinct activation patterns requiring further investigation by neuroscientists.

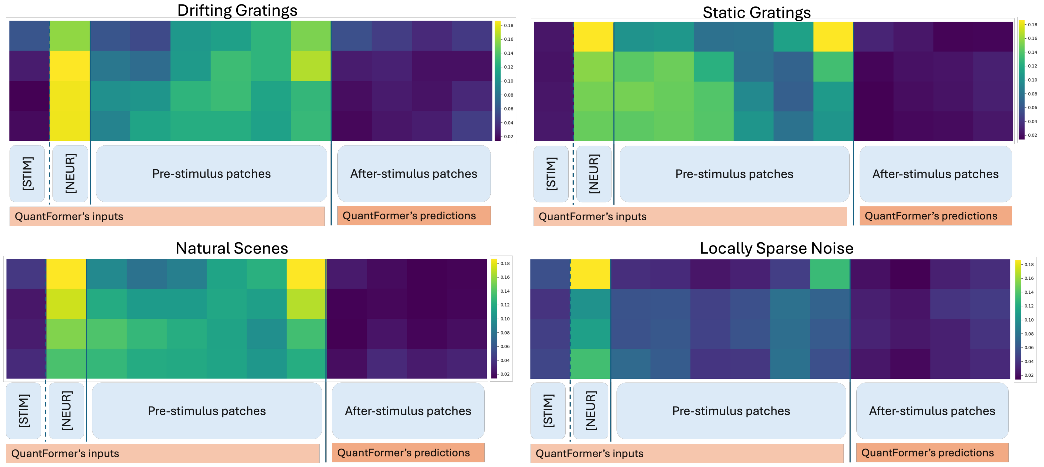

Figure A-8: Attention maps for all stimuli type.

## VII Interpretability of learned codes

Fig. A-9 shows the latent space structure of discrete codes learned by vector quantization. We performed 2D t-SNE on a learned codebook to observe sequence patterns. Subfigure (a) shows that amplitude increases along the x-axis when plotting codes on the same scale. Subfigure (b) reveals pattern variability after normalizing the scale. Interestingly, despite having a relatively small number of codes, the reconstructed representation heavily depends on the sequence, as shown in Subfigure (c): we generated sequences with bursts of the same code, except for one typically representing a peak (e.g., code 19), highlighted between red dashed lines. The replaced code's amplitude and shape vary based on context, indicating that while codes represent patterns, the reconstruction depends on the whole sequence.

## VIII Interpretability of neuron embeddings

To undestrand what is encoded into neuron embeddings, we visualized through t-SNE neuron embeddings from a downstream task. We find that neuron embeddings encode information such as activation frequency and response statistics. Colors in Fig. A-10 denote whether the measured quantity is above or below a threshold.

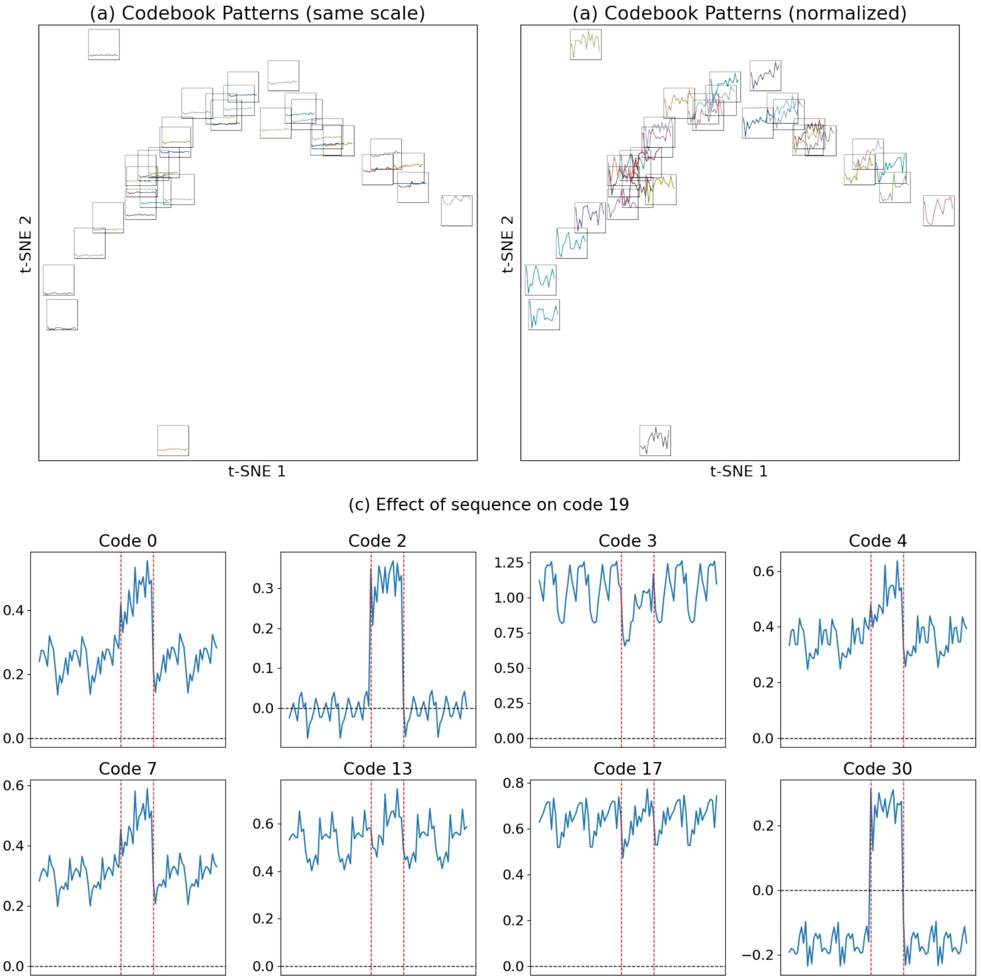

Figure A-9: Interpretability of codes. (a) t-SNE of a codebook, with patterns representation in the same scale. We can appreciate along the first axis the amplitude variation. (b) Same as before, but with normalization to appreciate differences in patterns. (c) Effect of sequence.

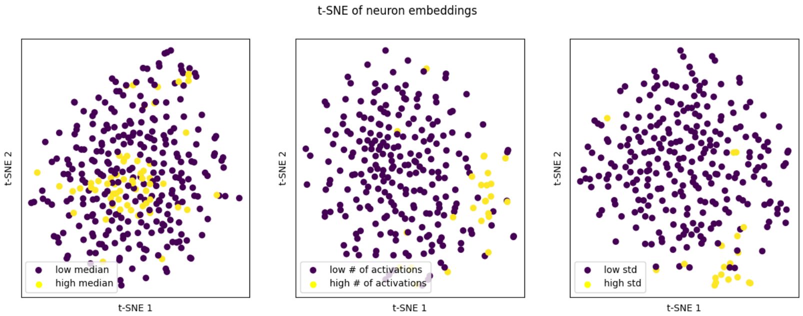

Figure A-10: Interpretability of neuron embeddings. We show t-SNE examples of neuron embeddings. We found that similar neurons in the latent space have also similar statistics like the median, the number of activations or the standard deviation.