Application of Bagged Copula-GP: Confirming Neural Dependencyn on Pupil Dilation

Anonymous authors
Paper under double-blind review

Abstract

Advancements in recording techniques has enabled the ability to record thousands of neurons simultaneously, shifting the needs within the field of computational neuroscience to powerful computational and statistical techniques. Copula-GP is a recently developed state-of-the-art parametric mutual information estimator found to outperform other novel non-parametric methods when utilized on highly dimensional data. Here, we utilized Copula-GP together with *Gaussian Process Factor Analysis* (GPFA) to investigate the information interaction between neuronal processes within the visual cortex of live mice and pupil dilation. We found usage of GPFA as a preprocessing step to Copula-GP was an effective means of investigating neuronal dependence, allowing flexibility in analysis and finding results in agreement with prior literature, and additionally extended Copula-GP with a bagging framework, allowing for the aggregation of model estimations and allowing for more accurate estimation accuracy and representation of dependency shape. We validated our bagging algorithm on simulated data sampled from known distributions, and utilized bagged Copula-GP with GPFA on said neuronal data to find results in agreement with baseline Copula-GP but with more stability.

1 Introduction

Not quite as many topics in the sciences have captured peoples' imaginations as the mysteries of human thought, behavior, and learning, a topic which has inspired fields ranging from psychology to philosophy Stewart (1822); Sternberg & Smith (1988). We now know that human thought occurs within the brain, and understanding the intrinsics of how and where such processes occur has applications in fields such as prosthetics and neurosurgery, not to mention how such findings often find a way to spill into related fields, such as machine learning Jangid et al. (2021). *Computational Neuroscience* aims to answer how cognition and behavior are encoded in the brain through the use of machine learning and statistical methods on large amounts of neuronal data Blundell et al. (2018); McDougal et al. (2017); Vu et al. (2018). Some of the primary questions on this frontier include: How is information encoded in populations of neurons, and how is this information related to behaviors observed Kudryashova et al. (2022)?

As advances in recording techniques have allowed for the recording of hundreds to thousands of neurons at once Jun et al. (2017); Dombeck et al. (2007), the bottlenecks keeping researchers from answering these central questions in neuroscience have shifted away from data-related issues to the purely computational and statistical Vu et al. (2018); Glaser et al. (2019); Quian Quiroga & Panzeri (2009). Answering such questions with machine learning comes down to analysis of the intricate highly-dimensional multivariate dependencies (i.e. dependencies and correlations between many random variables, many of which have different distributions) in recorded neuronal and behavioral data, much of which varies across spatial and temporal dimensions Kudryashova et al. (2022); Jun et al. (2017); Glaser et al. (2019); Quian Quiroga & Panzeri (2009). This becomes especially challenging once one considers how different behaviors can occur on a scale of hours or even days, whereas related neuronal activity can occur on a scale of milliseconds Pakan et al. (2018); Dombeck et al. (2007); Mathis et al. (2018), even more so as more variables are included in the data and the dependencies become even more computationally intense to analyze accurately (the "curse of dimensionality") Koppen (2000). Thus, in order to properly analyze such copious amounts of intricate

high-dimensional data, we need methods that are both resistant to high dimensionality as well as equipped with the ability to properly analyze dependencies between temporally-disparate variables in time series.

Parametric Copula-GP model for analyzing multidimensional neuronal and behavioral relationships Kudryashova et al. (2022), aims to fulfill this requirement via a novel framework for parametric estimation of multivariate distributions and mutual information through the use of *copulas*, a probabilistic structure adept at mutual information estimation. The core paper's contributions to the body of work includes the development of the Copula-GP python package, a package which allows for the training of copulas, model selection, and copula entropy extraction. Copula-GP was found to outperform other non-parametric mutual information estimators when predicting mutual information for a larger number of variables (≥ 10) Kudryashova et al. (2022). This combined with it's robustness to the "curse of dimensionality" makes it well suited for analysis of highly dimensional neuronal data.

Here, we validated usage of Copula-GP together with *Gaussian Process Factor Analysis* (GPFA; a dimensionality reduction method robust to qualities of neuronal data through which we are able to extract latent variables), doing so through confirming statistical dependence of neuronal data extracted from mice on pupil dilation. Within the literature, there is very little use of GPFA with mutual information estimators, much less conditional parametric copulas on neuronal data (or on any data). This project aimed to fill that gap, with the goal of improving the interpretability of the model by capturing processes in the brain across groups of neurons as latent trajectories extracted from spike-train data and estimating the mutual information between latent trajectories $X_1, X_2, \cdots, X_N$ conditioned on a behavioral variable Y, as well as the information interaction $I(X_1 : X_2 : \cdots : X_N \leftarrow Y)$ measuring statistical dependence in $X_1, X_2, \cdots, X_N$ captured by Y.

We also aimed to extend Copula-GP with a method of ensemble model selection through the use of *bagging*, wherein we aggregate copula estimations of individual Copula-GP estimators fit on separate samples of data, validating the ensemble method via entropy estimation of a known distribution and culminating in usage of bagged Copula-GP on aforementioned real life neuronal data. To summarize, the contributions of this paper include:

- Validation of combining novel Copula-GP with GPFA (the combination of which we dub *Copula-GPFA* as shorthand), with the expected benefits of robustness to dimensionality and efficient, detailed analysis of neuronal dependence.
- Justification of possible improvement to the Copula-GP algorithm through the addition of bagging, confirmed via validation tests on synthetic data sampled from known copula distributions.
- Confirming dependencies between neuronal data from the visual cortex on pupil dilation through the use of bagged Copula-GPFA.

The code repository for work shown is omitted for anonymity.

2 Neuroscience Preliminaries

In general, if a system of neurons encodes the information that goes into behaviors or cognition then we expect uncertainty in neuron spikes to be captured in behaviors during recorded spike-trains, or in other terms a significant quantity of mutual information and/or information interaction between spike signals and behaviors Gerstner et al. (2014). By extension, if we are able to accurately estimate mutual information then we should thusly be able to map systems of neurons to behaviors and cognition.

As stated in the introduction, computational neuroscience aims to answer questions surrounding the encoding of cognition and behavior in parts of the brain. Hurdles in answering such questions in the past stemmed from the difficulty of recording groups of neurons' behavior with certainty Kudryashova et al. (2022), however we now have the technology needed for accurate single-neuron resolution spike train trial recordings in the form of *Neuropixel probes* Steinmetz et al. (2021). As such, the hurdle of neuron-behavior-stimulus mutual information analysis is no longer centered on lack of data, moving instead to the need for accurate and efficient computational and statistical mutual information estimation techniques, as well as required robustness to

dimensionality when a large number of individual neurons and behavioral variables are analysed Kudryashova et al. (2022); Onken & Panzeri (2016); Vu et al. (2018).

3 Related Work

There are many choice methods in the realm of neuronal mutual information estimation. Popular novel ones include Bias-Improved Kraskov-Stögbauer-Grassberger (BI-KSG) Gao et al. (2018) and the Mutual Information Neuronal Estimator (MINE) Belghazi et al. (2021), both of which are effective non-parametric methods. The Copula-GP method implemented in the core paper by Kudryashova et al. (2022) is a parametric estimator found to generally outperform MINE and BI-KSG in mutual estimation accuracy on highly-dimensional data ($\geq$ 10 dimensions), and as such is our chosen method of neuronal mutual information analysis. It is a method based on *copulas*, a multivariate distribution with uniform marginals representative of cumulative distributions and a history of applications ranging from scientific analysis of dependency (as they are generally used in computational neuroscience Jenison & Reale (2004)) to the more purely analytical and predictive (i.e. predicting the outcome of an insurance claim Hu & O'Hagan (2021)). We delve more into what copulas are and how they're useful in section 4.

Copulas have been used in mutual-information estimation and dependency analysis in a variety of *in vivo* and *in silicon* neuronal data, including spike data, 2-photon calcium imaging, and multi-modal neuronal datasets Onken & Panzeri (2016); Shahbaba et al. (2014); Berkes et al. (2008); Jenison & Reale (2004), and have been successfully used for both mutual information estimation and dependency shape analysis in neuronal data Kudryashova et al. (2022); Shahbaba et al. (2014); Berkes et al. (2008). However, recent advances in GPU accelerated processing power (through the use of popular libraries such as *PyTorch* Paszke et al. (2019) and *GPyTorch* Gardner et al. (2018)) have allowed for further refinement of copula implementations and expanded the number of practical use cases, allowing for the implementation of such a method for high-dimensional neuronal data, like that of Copula-GP Kudryashova et al. (2022).

4 Copulas: An Introduction

When looking at the relationships between random variables, we often seek to examine the "shape" of variables' dependencies, even when such variables do not share a distribution Calsaverini & Vicente (2009). One of the choice methods when it comes to dependency and mutual information analysis are *copulas*, which are multivariate cumulative distributions with uniform marginal distributions Safaai et al. (2018); Kudryashova et al. (2022); Mitskopoulos et al. (2022). These marginal uniform distributions are "attached" to other distributions via a *Cumulative Distribution Function (CDF)* Van Vliet (2023). There are two things special about a copula distribution: firstly, by *Sklar's Theorem*, **every** multivariate distribution $F_{\mathbf{X}}$ where $\mathbf{X} \sim F_{\mathbf{X}}$ may be described by a copula $C(\mathbf{X})$ by means of attachment of CDFs to a copula's marginals Nelsen (2006); Safaai et al. (2018); Van Vliet (2023). This allows copulas to be an effective means of describing convoluted multivariate distributions, assuming one can sufficiently estimate marginals' CDFs. Secondly, **the negative entropy of a copula distribution $-H_C(\mathbf{X})$ is equivalent to the mutual information between marginals** $I(\mathbf{X}) = I(X_1, X_2, \cdots)$ Ma & Sun (2011); Jenison & Reale (2004). As such, should one be able to estimate a conditional copula $C(\mathbf{X}|Y)$ (i.e. through parametric means), one can gather several important mutual information and information interaction quantities. In the context of this paper, we are able to leverage this to experimentally estimate the amount of uncertainty in neuronal data captured in a behavioral variable, and we estimate parametric copulas via Copula-GP (explored in section 5). Copulas can be scaled in construction via a *vine construction*, where a highly dimensional copula C is described by a hierarchy of trees of bivariate copulas, with increased conditioning on marginals further up the hierarchy Onken & Panzeri (2016); Czado & Nagler (2022). This division of a multivariate distribution into bivariate building blocks allows copula vines to be more robust to dimensionality, given high performance in the bivariate case Kudryashova et al. (2022); Onken & Panzeri (2016). Formal copula theory is further explored in section C.

5 Copula-GP

To parameterize a copula, one can use *Gaussian Process (GP)* as a means of describing conditioning. GP is in essence a regression attempting to estimate a relationship between variables by means of Bayesian inference. GP can be used to estimate a distribution $\mathbf{f}$ to sample a function $f \sim \mathbf{f}$ describing parameterization over a domain $[0, 1]$, which can then be used in conjunction with GPLink functions to parameterize copula variants (see table 2 in the appendix for precise definitions of GPLinks) Kudryashova et al. (2022); Hernández-Lobato et al. (2013). While copulas have been utilized in modelling multivariate neuronal dependence and mutual-information Kudryashova et al. (2022); Onken & Panzeri (2016); Mitskopoulos et al. (2022), GP-treated vine copulas had only been extensively used on financial time series prior to the core paper Kudryashova et al. (2022); Hernández-Lobato et al. (2013). The Copula-GP package was implemented in Python and deployed in Kudryashova *et. al's* paper Kudryashova et al. (2022) (which was published in 2022), and is one of the few (if not only) practical implementations of a GP-treated copula vine model designed with GPU based acceleration of computation via *Pytorch* Paszke et al. (2019). It is also the first of such models designed with use on neuronal data in mind, and as such literature on the uses of GP vine copulas on modelling neuronal data is heavily limited, with the main source for such claims being the core paper. In simplest terms, the package serves as a framework for GP copula model parameter estimation, copula model selection, and model deployment as a vine, with additional built in visualization and entropy extraction utility. The core paper also utilizes comparison to other popular methods of neuronal mutual information extraction, yielding superior results in highly-dimensional data. **Any subsequent claim surrounding the Copula-GP package without a citation attached in this section comes courtesy of Kudryashova et al. (2022)**.

6 Entropy and Mutual Information Estimation

As Copula-GP models a copula distribution, possible states of variables can be sampled from it. As such, the joint mutual information $I(X_1 : X_2 : \cdots : Y) = H_C(\mathbf{X}|Y) - H_C(\mathbf{X})$ between parameter Y and multivariate distribution $\mathbf{X} = \{X_1, X_2, \dots\}$ can be derived from $C(\mathbf{X}|Y)$ (the C-vine copula Copula-GP estimates). Computing this directly via integration over the estimated copula density is computationally intensive due to nested integrals however, and as such a "estimated approach" is explored. We choose to find the MC estimate Robert & Casella (1999) of $H_C(X)$ (which is also utilized in the core paper), which involves sampling from the distribution $C(X|\hat{y}_i)$ with parameterization in N random values $\hat{y}_i \sim \mathcal{U}(0, 1)$ and estimating the conditional entropy as the mean

$$H_C(X|Y) = \int_{\mathrm{dom}(y)} H_C(X|y_i)dy_i \approx \frac{1}{N}\sum_{i=1}^{N} H_C(X|\hat{y}_i). \tag{1}$$

Similarly, we estimate the unconditional entropy as the mean entropy of the system found when fed true values of the parameterizing variable $y_i \in [0, 1]$, i.e

$$H_C(X) = \mathbb{E}_y(H_C(X|y)) \approx \frac{1}{N}\sum_{n=1}^{N} H_C(X|y_n). \tag{2}$$

Due to the law of large numbers, as N goes to infinity both approximations become more accurate. To find the copula entropy values $H_C(X|y_i)$, we utilize the Copula-GP's vine implementation's `entropy()` function to estimate conditional entropy for a fixed conditioning y via integrating over the estimated probability density of the copula.

7 Extension of Copula-GP: Bagging

Bagging is a bootstrapping technique wherein multiple "weak" models are trained and their outcomes aggregated in some way to create a "stronger" model Breiman (1996), typically reducing variance and increasing model bias. We propose a formal justification of copula bagging, taking inspiration from the "Random Forest" algorithm for bagged regression Breiman (2001); we assume the existence of some true copula function

C describing a multivariate distribution X. We take N samples $\{S_1, S_2, \ldots, S_N\}$ and get a copula $C^{(n)}$ representing the best copula fit possible on an individual sample S_n. We then come to a final estimate via a mean aggregation of our estimates. Suppose $C^{(n)}$ is a C-vine estimate, and let $C^{(n)}_{i,j}$ represent the i-th copula in the j-th layer of the n-th estimate. We can create a mean mixture copula $\hat{C}_{i,j}$ representing the final estimate for the corresponding true copula $C_{i,j}$ as $\hat{C}_{i,j} = \frac{1}{N}\sum_{n=1}^{N} C^{(n)}_{i,j}$. As $N \to \infty$, by the law of large numbers (and the fact that copulas are distributions) we find $\hat{C}_{i,j} \to C_{i,j}$, and so $\hat{C} \to C$. In other words, by fitting copulas on individual spike-train trials and taking the aggregate, we estimate a copula that more closely aligns with the true probabilistic relationship than those aggregated.

7.1 Weighted Aggregation Methods

A simple unweighted mean is appropriate in the infinite case. In practice however, we often find that a naive average is insufficient when examining a finite number of samples; suppose we have finite N uniform length samples S_n and thus have a set of observations $S = \{S_1, S_2, \ldots, S_N\}$, fitting corresponding copula estimates $\hat{C}_n$. If the n-th sample is an outlier sample and so $\hat{C}_n$ does not resemble the true copula C_n, we naturally should account for that. As such, in addition to a naive average, we may choose to weigh models based off various Bayesian criterion. We utilized a Bayesian model aggregation approach utilized by S. Hu *et. al* Hu & O'Hagan (2021), which aggregated copulas via *Bayesian Information Criterion* (BIC). The BIC value for the n-th estimated copula is given as

$$\mathrm{BIC_n} = -2\log\mathcal{L}(\hat{C}_n|X,\Theta) + p_n \log|X|, \tag{3}$$

where $\log\mathcal{L}(\hat{C}_n|X,\Theta)$ is the log-likelihood of copula $\hat{C}_n$ under observations X and estimated model parameters Θ, and p_n is the total number of estimated parameters of $\hat{C}_n$ (in the case of Copula-GP, each bivariate copula estimated is a mixture copula and so possesses mixing and dependence parameters for each copula mixed; if the mixture is a singleton mixture, we only count dependence parameter. Note independence copulas have no dependence parameter). BIC essentially rewards model log-likelihood with penalty in number of parameters scaling with sample-size Schwarz (1978); Hu & O'Hagan (2021), with small BIC values are being preferred (as negative as possible). We arrive at weights for the n-th model as

$$W_{n,\mathrm{BIC}} = \frac{\exp(-\frac{1}{2}BIC_n)}{\sum_m \exp(-\frac{1}{2}BIC_m)}. \tag{4}$$

Hu & O'Hagan (2021) We also utilize the *Akaike Information Criterion* (AIC) for comparison, which is given for the n-th copula estimation as $\mathrm{AIC_n} = -2\log\mathcal{L}(X|\hat{C}_n,\Theta) + 2p_k$ and extract weights $W_{n,\mathrm{AIC}}$ accordingly via equation 4, replacing BIC_n with AIC_n. The AIC is similar to the BIC in that it rewards log-likelihood but has a more relaxed penalty in number of parameters. If a more complex copula matches the true distribution, this can lead to possible improvements in aggregation accuracy, but can also lead to over-fitting via over-parameterization Zhou (2021) (as we are already utilising mixtures of copulas with a high ceiling in number of parameters, we expect the latter to be true). We additionally include dynamic weighting of estimates via calculating the above information criteria and their respective weights point-wise as opposed to setting weights to be constant over the observations X; we call point-wise aggregation *dynamic bagging* and constant weighting *static bagging*. We validate our bagging methods with different aggregation methods on data samples from randomly generated bivariate copulas in section 10.

To aggregate C-vines, we can bag one layer at a time; we first bag copulas in the current layer normally, and gather cumulative conditional probabilities as pseudo-observations via the bagged copulas' ccdfs. We then utilize them for BIC, AIC, and R^2 calculations in the next layers' bagging process, and in doing so effectively propagate the previous layers' weightings to the next layer. As subsequent bagged copulas will have worse BIC, AIC, and R^2 calculations if their corresponding C-vines' previous layers were found to be bad fits, we in effect prioritize relationships with weaker conditioning using this method.

7.2 Aggregation Implementation

Copula-GP estimates a parametric C-vine along a given 1-D parameterization input X, modeling each individual copula as a mixture and getting estimated dependence parameterizations $\{\theta_{i,j}(X)\}$ and estimated

mixing parameterizations $\{\phi_{i,j}(Y)\}$ for each of i mixture copulas (with each mixture copula being a mix of $j^{(i)}$ copula variants). As such, our actual implemented bagging procedure is to aggregate along these estimated parameters for each unique copula variant in each mixture (we count each rotation variant of a Clayton or Gumbel copula as unique; this algorithm is outlined more formally in algorithm 1). In addition, we implemented model selection for each C-vine we aggregate, allowing for the bagged vine to contain copula mixtures possibly unexamined in the heuristic based model selection process. The C-vine aggregation process requires aggregation of copulas by layer and storing the cumulative distributions of each copula via the bagged copula ccdfs for weight extraction in the next layer.

Algorithm 1 Implemented bivariate mixture copula bagging algorithm. Here, mixture copulas consist of mixed copula variants *variants*, their corresponding dependency parameter values Θ (for each of the m inputs), and their corresponding mixing parameters *mix*. We assume weights have been defined over all m input points.

```
Input
    mixture: A list of mixture copulas to aggregate.
    weights: Corresponding mixture weights.
Output
    mixture(Variant List, Mix Parameters, Θ): A new mixture copula of variants Variant List with
    corresponding mixture parameters Mix Parameters and dependency parameters Θ.
procedure BagCopulas(mixtures, weights)
    N ← 0                                                    ▷ Unique variant counter.
    indexes ← dict()                                         ▷ Indexing dictionary.
    Variants Seen ← set()                                    ▷ Set of Copula Variants
    Variant Counts ← dict()                                  ▷ Counting dictionary.
    Total Variant Weight ← dict()                            ▷ Weighting dictionary.
    for i, mixture in enumerated(mixtures) do
        for n, variant in enumerated(mixture.variants) do
            if variant not in Variants Seen then
                Variants Seen.add(variant)
                indexes[variant] ← N
                Variant Counts[N] ← 0.0
                Total Variant Weight[N] ← 0.0
                N ← N + 1
            idx ← indexes[variant]
            Variant Counts[idx] ← self + 1
            indexes[(i, n)] ← idx
            Total Variant Weight[idx] ← self + weights[i]
    Variant List ← list(N)                                   ▷ Final list of variants.
    Mix Parameters ← 0_(N×m)                                 ▷ Mixture parameters.
    Θ ← 0_(N×m)                                              ▷ Dependency parameters.
    for i, mixture in enumerated(copulas) do
        for n, variant in enumerated(mixture.variants) do
            idx ← indexes[(i, n)]
            Variant List[idx] ← variant
            Mix Parameters[idx] ← self + mixture.mix[n] * weights[i]
            Θ[idx] ← self + mixture.Θ[n] * weights[i] / Total Variant Weight[idx]
    return new mixture(Variant List, Mix Parameters, Θ)
```

8 Gaussian-Process Factor Analysis

A very common technique in the realm of dimension reduction for highly dimensional neuronal data is *Gaussian-Process Factor Analysis* (GPFA), which factors a matrix of observations X into a product of a

loading matrix Ψ and a scoring matrix Θ as $X = \Psi\Theta$ (the *Factor Analysis* step) producing a low-dimensional group of trajectories, and then smooths said data by means of fitting a GP (the *Gaussian Process* step) Pirš & Štrumbelj (2022). This operates under the assumption of a linear relationship between observations X and latent trajectories Θ (utilizing Ψ as a transformation matrix) Yu et al. (2009), and models the GP with bias d as

$$X = \Psi\Theta + d + \epsilon, \;\; \epsilon \sim \mathcal{N}(0, \sigma_\epsilon^2) \tag{5}$$

with parameters $(d, \Psi, \sigma_\epsilon^2)$.

8.1 Utilization with Copula-GP and Proposed Benefits

The Copula-GPFA process consists of GPFA applied to neuronal data to extract latent trajectory estimations $\hat{X}_1, \hat{X}_2, \ldots,$ followed by fitting a Copula-GP C-vine estimator on neuronal trajectories. Doing so, we are able to efficiently extract mutual information estimates describing not just the information interaction between the examined part of the brain and some parameterizing variable, but the mutual information between the brain processes driving neurons instead of individual neurons themselves Yu et al. (2009); Ma & Sun (2011); Kudryashova et al. (2022), enhancing the interpretability of individual copulas estimated. In addition, by utilizing GPFA to reduce a large number of neurons to a more manageable amount, we reduce the computational intensity of mutual information analysis; if we extract 13 trajectories from $\sim$150 neurons then from the naive estimate for the complexity of training Copula-GP estimators (see core paper Kudryashova et al. (2022)) we can come to an upper-bound of $150^2 \div 13^2 \approx 133$ times faster Copula-GP training.

9 Data Providence: *Neuropixels*

The main data set we wished to explore is the *Visual Coding: Neuropixels* dataset, a publicly available dataset containing spike-signals recorded from the visual cortex of live mice at single neuron spatial resolution utilizing novel Neuropixel probes, a high-fidelity and -resolution brain probe developed in 2021 Steinmetz et al. (2021). Due to the volume of *in vivo* data recorded, as well as the quality metrics included in the data, it is an invaluable tool for analysis of underlying processes within the visual cortex. All technical claims surrounding the data in question is sourced from the technical white paper (see D). For more information surrounding data production and specificities not covered in the body of this paper, i.e specifics on how probes were used and how visual stimuli were produced, one should see the technical white paper. The spike data was collected via insertion of *Neuropixel* probes into the visual cortex of live mice. Probes record spike signals from points in space at single neuron resolution. These points are called "units," and are where neurons *might* be. Units have quality metrics attached, which can be used to filter for units of better quality (i.e. units with less noise, units which more certainly record spikes from only one neuron, etc). See section D for specifics on data retrieval and cleaning.

Pupil dilation has been found to have close ties to processes within the visual cortex Franke et al. (2022); Bombeke et al. (2016); Larsen et al. (2018). As this data is directly from the visual cortex, we expected a significant statistical dependence between pupil dilation and visual cortex trajectories. As such, the motivation behind utilizing this dataset is as an experimental method of model validation; Copula-GPFA should be able to extract a significant amount of information interaction between trajectories extracted from the visual cortex and pupil dilation.

10 Bagged Copula-GP Validation Tests

To validate our bagging methodology, we tested various Copula-GP aggregation methods alongside baseline unbagged Copula-GP on generated bivariate copulas (the "true" copula). C-vines are made up of bivariate copula building blocks, with C-vine model selection essentially being made up of consecutive bivariate copula selections. As such, validation alongside baseline Copula-GP on bivariate copulas alone is indicative of improvements in C-vine entropy estimation via bagging methods described.

For all validation tests in this section, the task was to accurately replicate the true copula's entropy and dependency shape. For validation, the weighted aggregation methods utilized were 1. Copulas weighted

point-wise dynamically by BIC / AIC on input, 2. Copulas weighted statically by BIC / AIC on input, and 3. naive average of copulas. In addition, we also examine copula entropy point-wise root mean squared error (RMSE) of baseline and bagging methods.

To generate the data, we drew 10000 samples from the true copula, splitting samples and their corresponding parameterizations into train and test sets (80:20 split). Validation tests 1 and 2 utilized parameterization on a normally distributed variable $x \sim \mathcal{N}(0.5, 0.2)$, restricted to the interval $[0, 1]$. Validation test 3 utilized a parameterization in a time-based variable t scaling linearly from 0 to 1. Copula training and model selection utilized train set samples, with accuracy validation utilizing metrics described on test set samples. All validations utilized the heuristic algorithm for individual estimator model selection, and for bagged estimations 4 estimators underwent individual training and model selection routines. We include random seed, module version, and hardware information for test reproducability purposes in the appendix (see G).

Table 1: True and predicted mean copula entropies $\overline{H}_C$ with point-wise RMSE of validation tests. Closest to actual / best scores in bold. We find that BIC Dynamically bagged Copula-GP consistently outperforms alternative methods in accuracy.

Model	Test 1 $\overline{H}_C$	RMSE	Test 2 $\overline{H}_C$	RMSE	Test 3 $\overline{H}_C$	RMSE
True Copula	-0.1440	-	-0.6864	-	0.6864	-
Baseline	-0.0000	0.1442	-0.6680	0.0694	-0.4556	0.2587
BIC Dynamic	**-0.0802**	**0.0647**	**-0.6764**	0.0719	**-0.5628**	**0.1388**
BIC Static	-0.0791	0.0656	-0.6725	0.0713	-0.5589	0.1418
AIC Dynamic	-0.0796	0.0652	-0.6664	0.0748	-0.5382	0.1615
AIC Static	-0.0790	0.0658	-0.6662	0.0658	-0.5389	0.1596
naive average	-0.0798	0.0651	-0.6738	**0.0651**	-0.5580	0.1420

10.1 Validation 1: Low-Entropy 4-Copula Mixture Copula

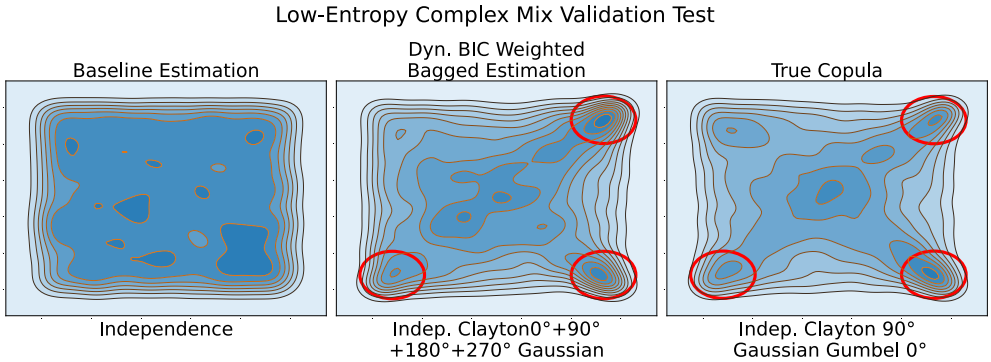


Figure 1: Average predicted copulas of BIC dynamically bagged and baseline Copula-GP on low-entropy copula validation test set, with true copula for comparison. Contour lines of copulas are visible. Note how the distribution "tricks" the baseline estimator into picking the independence copula, but the bagged estimate is able to capture major tail dependencies in the true distribution (circled in red).

Our first validation test was on a low entropy copula with a resemblance to the independence copula we expected our bagging algorithm to perform well on. Baseline Copula-GP when facing such a copula can often "give up" early into heuristic model selection and select independence if it finds non-independence copulas possess too low WAIC. We expect that bagging is able to make up for this flaw in baseline heuristic model selection via it's bias-variance trade-off, and thus pick up tail dependencies in the model.

We found that our expectations were held. In table 1 we observe that bagging methods all performed better than baseline (which selected independence), with BIC dynamic bagging possessing the best model accuracy and RMSE. Finally, in figure 1 we find that the BIC dynamically bagged estimated copula captures major tail distributions in the true copula, whereas the baseline estimation of independence does not.

10.2 Validation 2: High-Entropy 3-Copula Mixture Copula

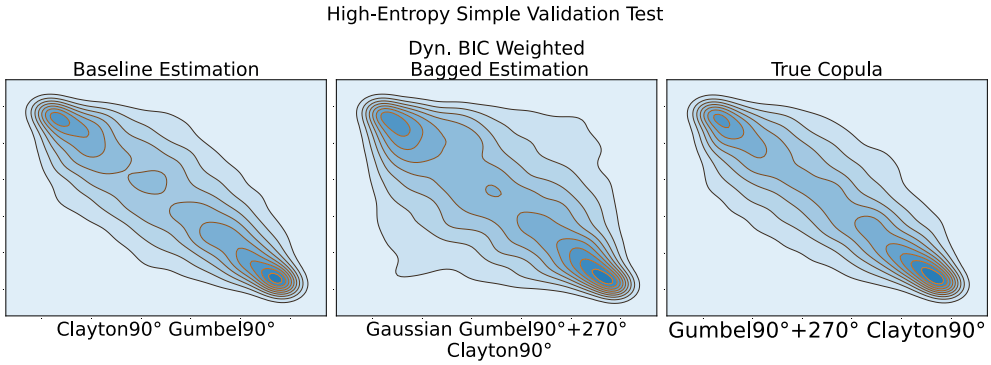


Figure 2: Average predicted copulas of BIC dynamically bagged and baseline Copula-GP on high-entropy copula validation test set, with true copula for comparison. contour lines of copulas are visible. We find that bagging methods can result in incorrect selection of copula variant leading to possible misrepresentation of dependency shape, however the bagged estimation outperforms baseline in mean copula entropy estimated.

For this validation test, we expected baseline to perform well and are interested in if the bias induced by bagging can result in worse performance on distributions the baseline is well suited for. The true copula here is a high-entropy copula with simple shape, something that in the core paper Kudryashova et al. (2022) was found to be well suited to baseline Copula-GP as a method.

We find from table 1 that all tested methods had similar performance. The BIC dynamically weighted estimate had the best estimated mean copula entropy, with the naive average estimate possessing superior RMSE. In figure 2 we find that the BIC dynamically weighted estimation included a gaussian copula and as such makes incorrect assumption surrounding the dependency shape of the true copula. Despite this, the majority of bagged copula estimates still result in similar if not slightly better performance in copula entropy prediction.

10.3 Validation 3: High-Entropy 3-Copula Mixture Copula, Parameterized Linearly in Time

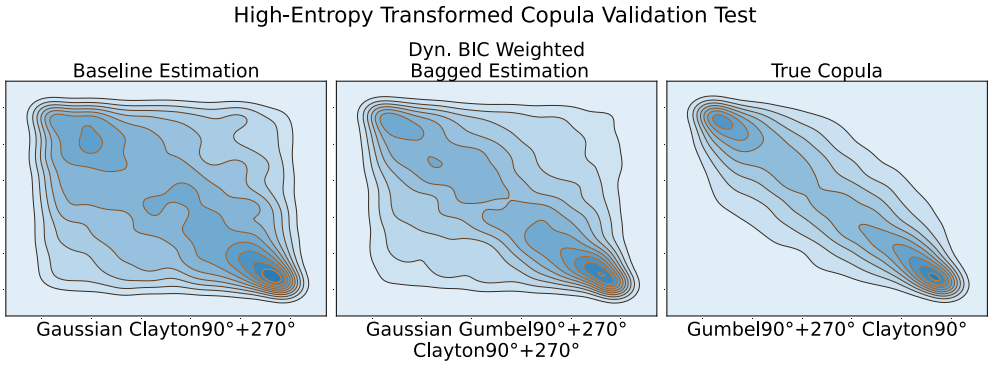


Figure 3: Average predicted copulas of BIC dynamically bagged and baseline Copula-GP on transformed high-entropy copula validation test set, with true copula for comparison. contour lines of copulas are visible. We find that the baseline estimation possesses less of a resemblance to the true copula than the bagged estimation.

Our final validation was on a copula parameterized in time. In the core paper, it was found that "transformed" cases with dependency shape changing as a function in time can be more challenging for Copula-GP to predict the entropy of accurately. As such, we utilize the copula in validation test 2 and change it's parameterization to be in time t scaling linearly from 0 to 1. We maintain continuity when splitting into train and test set, and

as such the train set consists of parameterization in $t = 0.0$ to 0.8, and the test set consists of parameterization in $t = 0.8$ to 1.0. We hope that bagging distribution estimations made by estimators trained on continuous subsets of training data can induce some robustness to time-based transformation in true dependency shape.

Our hopes were confirmed, as we find in table 1 that bagging methods all find better accuracy than baseline. BIC dynamically weighted bagging results in the best mean copula entropy estimate and RMSE. From figure 3 we see that the baseline estimated copula possesses much less of a resemblance to the true copula than the bagged estimate.

10.4 Validation Discussion

In general, we found that bagged copula estimations utilizing 4 estimators possesses similar or better accuracy in copula entropy estimation than baseline. The best bagging method appears to be BIC dynamically aggregated copulas, possessing the best accuracy for all validation tests conducted. In addition, AIC weighted aggregation methods find worse performance than the BIC counterparts. Considering the difference between AIC from BIC is a more lenient penalty in number of parameters, this can be attributed to overfitting due to over-parameterization. From validation test 1, we see that bagged estimates can catch dependencies in distributions that the baseline perceives as independence. In C-vine model selection, as copulas gain more conditioning marginal variables' relationships tend to more closely resemble conditional independence Kudryashova et al. (2022). As such, we may conclude that a bagged estimator might be more likely to pick up tail dependencies in conditioned relationships where the baseline estimator might assume independence. From validation tests 2 and 3 we find that the bagged estimator is more robust to transformations in dependency shape over time, and as such may yield better accuracy when predicting marginal variables' dependency shape for time-based data, such as spike train data. Finally, we note that the BIC dynamically bagged estimator does not over-estimate copula entropy in validation, much like was found of baseline Copula-GP in the core paper.

11 Exploration of Experimental Neuronal Dataset

For this section, the objective was to utilize Copula-GPFA to confirm high statistical dependence between the visual cortex and pupil dilation. We hypothesised significant information interaction, thereby effectively confirming that the visual cortex of the brain is linked to pupil dilation and agreeing with prior results in the literature Bombeke et al. (2016); Ganea et al. (2020). GPFA was fit on 100 consecutive spike trains during Drifting Gratings stimulus presentations, each of which have a uniform temporal length of 2s followed by a 1s interval during which no stimulus is presented. We fit the model for the entirety of the dataset with the dimensionality of the elbow of the log-likelihood plot $n = 13$ (figure 9). We removed drift and concatenated trials as per the process described in section F.1 to produce semi-continuous data, mapped onto the domain $(0, 1)$ via additional interim steps described in section F.1, and moved on to Copula-GP application.

11.1 Copula-GP Application

We utilized Copula-GP to estimate C-vines over the concatenation of GPFA-treated trials, possessing 12 layers (78 copulas total) with parameterization in normalized pupil dilation, testing both baseline unbagged and bagged Copula-GP. For model bagging, we utilized the BIC dynamically weighted bagged Copula-GP over 4 subsets of semi-continuous data representing 20 trials each, and estimated negative copula entropy $-H_C(X)$ over time for 15 continuous trials. We expected either similar or lesser (more negative) copula entropy estimates of bagged Copula-GP as baseline, as was observed in validation.

We can see the outcome of $-H_C(X)$ and interaction information $I(X \leftarrow Y)$ calculations alongside normalized pupil dilation in figure 4. Bagged Copula-GP estimated a copula with mean entropy $-H_C(X) = 13.4027$ bits (95% CI of 0.7069) and mean $-H_C(X|Y) = 7.3323$ bits (95% CI of 5.9693). Copula entropy appears to be highly correlated with pupil dilation, with decreases in copula entropy occurring with decreases in pupil dilation. In addition, the information interaction stays firmly *negative*, implying the multivariate distribution of neuronal trajectories possesses statistical dependence on pupil dilation (thereby confirming our hypothesis).

From figure 5, we can see that like the bagged estimate the baseline negative copula entropy appears to be dependent on pupil dilation, with large dips occurring with pupil dilation. We found a mean negative copula entropy $-H_C(X)$ estimation of 13.4745 bits (95% CI of 3.0720) and a mean negative conditional copula entropy $-H_C(X|Y)$ empirical estimation of 7.2221 bits (95% CI of 6.1190), with X being the neuronal trajectories and Y being pupil dilation. However, we find the bagged estimate is less sensitive to small changes in pupil dilation than the baseline estimate. In addition, the bagged entropy estimate appears to steadily hover around the mean negative entropy found for most observed time-buckets where as the unbagged estimate appears almost linear in pupil dilation (see figure 5). These differences aside, the bagged and baseline mean copula entropy and information interaction estimates were close, implying that the bagged estimator does not inherently out- or under-perform baseline unbagged *Copula-GPFA* in the case of the *in vivo* data.

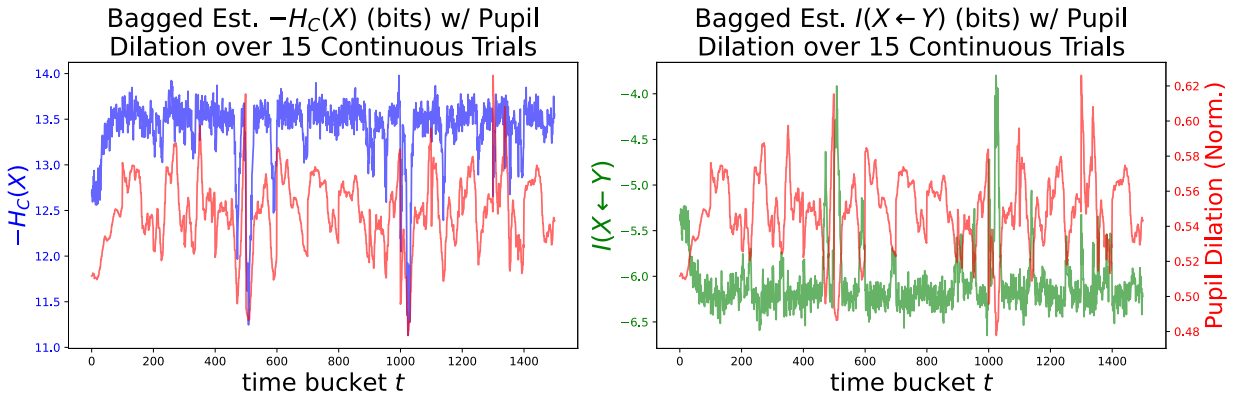


Figure 4: Pupil dilation (in red), copula entropy of neuronal trajectories parametric in pupil dilation utilizing bagged estimation (in blue), and corresponding information interaction (in green) through time utilizing estimated copula entropy. Copula entropy was estimated via estimating a distribution with BIC dynamically bagged Copula-GP.

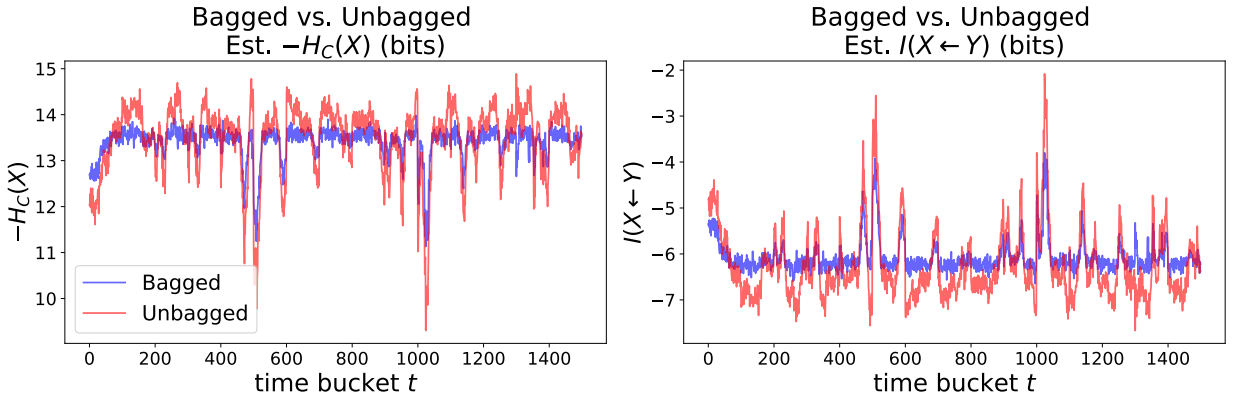


Figure 5: BIC dynamically bagged vs unbagged *Copula-GP* estimated entropies and information interactions. The BIC dynamically bagged entropies curve seems to hover steadily around a mean, and does not appear to be as volatile as unbagged entropies.

12 Discussion

We validated Copula-GPFA as a mutual information and information interaction extraction method on *in vivo* data extracted from mice, confirming significantly negative information interaction between neuronal

trajectories and pupil dilation and agreeing with prior findings linking the visual cortex with pupil dilation Bombeke et al. (2016); Ganea et al. (2020); Zylberberg et al. (2012). In addition, Copula-GPFA also allows for easy interpretability of findings, allowing for visualization of both neuronal trajectories and information quantities over time. We also justify a significant reduction in computational complexity via GPFA utilized with Copula-GP, using complexity estimates made in the core paper. In other words, **Copula-GPFA is able to efficiently provide accurate and meaningful dependency and information quantification analysis and results.** We also introduced copula bagging as a means to possibly improve copula estimation, complete with both formalistic and practical justifications. We implemented bagging as an extension to the Copula-GP code-base (locally), and **identified the superiority of bagged estimations over baseline in tail-dependency identification and estimated copula entropy accuracy in the bivariate case.** In addition, we utilized the bagged extension on the experimental *in vivo* data, finding results in agreement with baseline estimation of mean copula entropy with more stability over time, however whether this stability is indicative of the true distribution or not goes unconfirmed. As Copula-GPFA is not inherently a method restricted to the area of neuroscience, it is entirely plausible Copula-GPFA is able to be meaningfully applied to other fields where dependency analysis is useful (i.e. bio-informatics or quantitative finance).

While we did not test robustness of the bagged extension to higher dimensionality, as C-vine selection within the Copula-GP algorithm consists of individual selections of bivariate building blocks we hypothesize that benefits found in the bivariate case propagate into cases of higher dimensions. Our implementation of bagging also follows in the footsteps of the formalistic approach of the original Copula-GP implementation, and as such maintains flexibility in use cases outside the realm of spike train data and neuroscience. We also acknowledge that the ability to bag copula estimations allows for the aggregation of Copula-GP estimators to be fit on individual trials of data, thereby resolving possible violations of local smoothness and continuity assumptions caused by concatenation of GPFA treated spike train trials (due to time-to-fit concerns and limitations in compute-power, this goes unutilized as of now). Finally, we note our bagging implementation can be used *without* heuristic model selection, and one could utilize our implementation to bag singleton mixtures (mixture copulas of only a single variant) as an alternative efficient means of model selection. Finally, we note that while we did not thoroughly investigate individual bivariate copulas within the C-vines estimated, *each bivariate copula itself represents a dependency relationship.* As such, there is room for more analysis.

12.1 Further Validation, Optimization, and Use of Bagging Methods

Our validations find that weighted copula estimate aggregation can yield effective improvements in copula tail dependency identification and copula entropy accuracy. The obvious extension to validations made would of course be to confirm if such improvements firmly carry over to higher dimensions, however because Copula-GP implementation and the structure of the C-vine we believe that benefits found in the bivariate case will propagate into higher dimensions. In addition, further refinement of aggregation methods may be considered, i.e. clustering methods such as k-means as a way to select copula estimates for datasets. Use of parallel processing in training bagged estimators concurrently might also be a natural optimization for bagged Copula-GP; one could even bag bivariate copulas immediately after they've completed model selection on their subsets of data during vine training.

That being said, benefits found in the bivariate case are still entirely applicable. Bivariate copulas are still effectively used in fields from computational finance Cherubini et al. (2011) to bio-informatics Ray et al. (2020). As such, the improvements made to Copula-GP in bivariate copula estimation via the addition of dynamically weighted aggregation methods may be used to robustify bivariate copula selection effectiveness in such use cases.

References

Mário Antunes, Diogo Gomes, and Rui L. Aguiar. Knee/Elbow Estimation Based on First Derivative Threshold. In *2018 IEEE Fourth International Conference on Big Data Computing Service and Applications (BigDataService)*, pp. 237–240, March 2018. doi: 10.1109/BigDataService.2018.00042. URL `https://ieeexplore.ieee.org/abstract/document/8405717`.

Bence Bagi, Michael Brecht, and Juan Ignacio Sanguinetti-Scheck. Unsupervised discovery of behaviorally relevant brain states in rats playing hide-and-seek. *Current Biology*, 32(12):2640–2653.e4, June 2022. ISSN 09609822. doi: 10.1016/j.cub.2022.04.068. URL https://linkinghub.elsevier.com/retrieve/pii/S0960982222007047.

Mohamed Ishmael Belghazi, Aristide Baratin, Sai Rajeswar, Sherjil Ozair, Yoshua Bengio, Aaron Courville, and R. Devon Hjelm. MINE: Mutual Information Neural Estimation, August 2021. URL http://arxiv.org/abs/1801.04062. arXiv:1801.04062 [cs, stat].

P. Berkes, F. Wood, and J. Pillow. Characterizing neural dependencies with copula models. December 2008. URL https://www.semanticscholar.org/paper/Characterizing-neural-dependencies-with-copula-Berkes-Wood/41d56cacdecbf781ee7c219780f45ee948f71bf2.

Gérard Biau and Erwan Scornet. A random forest guided tour. *TEST*, 25(2):197–227, June 2016. ISSN 1863-8260. doi: 10.1007/s11749-016-0481-7. URL https://doi.org/10.1007/s11749-016-0481-7.

Inga Blundell, Romain Brette, Thomas A. Cleland, Thomas G. Close, Daniel Coca, Andrew P. Davison, Sandra Diaz-Pier, Carlos Fernandez Musoles, Padraig Gleeson, Dan F. M. Goodman, Michael Hines, Michael W. Hopkins, Pramod Kumbhar, David R. Lester, Bóris Marin, Abigail Morrison, Eric Müller, Thomas Nowotny, Alexander Peyser, Dimitri Plotnikov, Paul Richmond, Andrew Rowley, Bernhard Rumpe, Marcel Stimberg, Alan B. Stokes, Adam Tomkins, Guido Trensch, Marmaduke Woodman, and Jochen Martin Eppler. Code Generation in Computational Neuroscience: A Review of Tools and Techniques. *Frontiers in Neuroinformatics*, 12, November 2018. ISSN 1662-5196. doi: 10.3389/fninf.2018.00068. URL https://www.frontiersin.org/journals/neuroinformatics/articles/10.3389/fninf.2018.00068/full. Publisher: Frontiers.

Klaas Bombeke, Wout Duthoo, Sven C. Mueller, Jens-Max Hopf, and C. Nico Boehler. Pupil size directly modulates the feedforward response in human primary visual cortex independently of attention. *NeuroImage*, 127:67–73, February 2016. ISSN 1053-8119. doi: 10.1016/j.neuroimage.2015.11.072. URL https://www.sciencedirect.com/science/article/pii/S105381191501109X.

Leo Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, August 1996. ISSN 1573-0565. doi: 10.1007/BF00058655. URL https://doi.org/10.1007/BF00058655.

Leo Breiman. Random Forests. *Machine Learning*, 45(1):5–32, October 2001. ISSN 1573-0565. doi: 10.1023/A:1010933404324. URL https://doi.org/10.1023/A:1010933404324.

R. S. Calsaverini and R. Vicente. An information-theoretic approach to statistical dependence: Copula information. *Europhysics Letters*, 88(6):68003, December 2009. ISSN 0295-5075. doi: 10.1209/0295-5075/88/68003. URL https://dx.doi.org/10.1209/0295-5075/88/68003.

Umberto Cherubini, Sabrina Mulinacci, Fabio Gobbi, and Silvia Romagnoli. *Dynamic Copula Methods in Finance*. John Wiley & Sons, November 2011. ISBN 978-0-470-68307-1. Google-Books-ID: wZaADwAAQBAJ.

Claudia Czado and Thomas Nagler. Vine Copula Based Modeling. *Annual Review of Statistics and Its Application*, 9(1):453–477, 2022. doi: 10.1146/annurev-statistics-040220-101153. URL https://doi.org/10.1146/annurev-statistics-040220-101153. _eprint: https://doi.org/10.1146/annurev-statistics-040220-101153.

Gabriela Czanner, Sridevi V. Sarma, Demba Ba, Uri T. Eden, Wei Wu, Emad Eskandar, Hubert H. Lim, Simona Temereanca, Wendy A. Suzuki, and Emery N. Brown. Measuring the signal-to-noise ratio of a neuron. *Proceedings of the National Academy of Sciences*, 112(23):7141–7146, June 2015. doi: 10.1073/pnas.1505545112. URL https://www.pnas.org/doi/abs/10.1073/pnas.1505545112. Publisher: Proceedings of the National Academy of Sciences.

Michael Denker and Moritz Kern. Elephant 1.0.0, November 2023. URL https://zenodo.org/records/10107318.

Daniel A. Dombeck, Anton N. Khabbaz, Forrest Collman, Thomas L. Adelman, and David W. Tank. Imaging large-scale neural activity with cellular resolution in awake, mobile mice. *Neuron*, 56(1):43–57, October 2007. ISSN 0896-6273. doi: 10.1016/j.neuron.2007.08.003.

R. M. Dudley. Sample Functions of the Gaussian Process. In Evarist Giné, Vladimir Koltchinskii, and Rimas Norvaisa (eds.), *Selected Works of R.M. Dudley*, Selected Works in Probability and Statistics, pp. 187–224. Springer, New York, NY, 2010. ISBN 978-1-4419-5821-1. doi: 10.1007/978-1-4419-5821-1_13. URL https://doi.org/10.1007/978-1-4419-5821-1_13.

Katrin Franke, Konstantin F. Willeke, Kayla Ponder, Mario Galdamez, Na Zhou, Taliah Muhammad, Saumil Patel, Emmanouil Froudarakis, Jacob Reimer, Fabian H. Sinz, and Andreas S. Tolias. State-dependent pupil dilation rapidly shifts visual feature selectivity. *Nature*, 610(7930):128–134, October 2022. ISSN 1476-4687. doi: 10.1038/s41586-022-05270-3. URL https://www.nature.com/articles/s41586-022-05270-3. Publisher: Nature Publishing Group.

Dan Alin Ganea, Alexander Bexter, Mathias Günther, Pierre-Marie Gardères, Björn M. Kampa, and Florent Haiss. Pupillary Dilations of Mice Performing a Vibrotactile Discrimination Task Reflect Task Engagement and Response Confidence. *Frontiers in Behavioral Neuroscience*, 14, September 2020. ISSN 1662-5153. doi: 10.3389/fnbeh.2020.00159. URL https://www.frontiersin.org/articles/10.3389/fnbeh.2020.00159. Publisher: Frontiers.

Weihao Gao, Sewoong Oh, and Pramod Viswanath. Demystifying Fixed k -Nearest Neighbor Information Estimators. *IEEE Transactions on Information Theory*, 64(8):5629–5661, August 2018. ISSN 1557-9654. doi: 10.1109/TIT.2018.2807481. URL https://ieeexplore.ieee.org/abstract/document/8294268. Conference Name: IEEE Transactions on Information Theory.

Jacob Gardner, Geoff Pleiss, Kilian Q Weinberger, David Bindel, and Andrew G Wilson. GPyTorch: Black-box Matrix-Matrix Gaussian Process Inference with GPU Acceleration. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper/2018/hash/27e8e17134dd7083b050476733207ea1-Abstract.html.

Wulfram Gerstner, Werner M. Kistler, Richard Naud, and Liam Paninski. *Neuronal Dynamics: From Single Neurons to Networks and Models of Cognition*. Cambridge University Press, July 2014. ISBN 978-1-107-06083-8. Google-Books-ID: D4j2AwAAQBAJ.

Joshua I. Glaser, Ari S. Benjamin, Roozbeh Farhoodi, and Konrad P. Kording. The roles of supervised machine learning in systems neuroscience. *Progress in Neurobiology*, 175:126–137, April 2019. ISSN 0301-0082. doi: 10.1016/j.pneurobio.2019.01.008. URL https://www.sciencedirect.com/science/article/pii/S0301008218300856.

José Miguel Hernández-Lobato, James R Lloyd, and Daniel Hernández-Lobato. Gaussian Process Conditional Copulas with Applications to Financial Time Series. July 2013.

Sen Hu and Adrian O'Hagan. Copula Averaging for Tail Dependence in Insurance Claims Data, March 2021. URL http://arxiv.org/abs/2103.10912. arXiv:2103.10912 [stat].

Aisha Jangid, Laxmi Chaudhary, and Komal Sharma. Computational Neuroscience and Its Applications: A Review. In Mohammad Shorif Uddin, Avdhesh Sharma, Kusum Lata Agarwal, and Mukesh Saraswat (eds.), *Intelligent Energy Management Technologies*, pp. 159–169, Singapore, 2021. Springer. ISBN 9789811588204. doi: 10.1007/978-981-15-8820-4_16.

Rick L. Jenison and Richard A. Reale. The Shape of Neural Dependence. *Neural Computation*, 16(4):665–672, April 2004. ISSN 0899-7667. doi: 10.1162/089976604322860659. URL https://doi.org/10.1162/089976604322860659.

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, Mladen Barbic, Timothy J. Blanche, Vincent Bonin, João Couto, Barundeb Dutta, Sergey L. Gratiy, Diego A. Gutnisky,

Michael Häusser, Bill Karsh, Peter Ledochowitsch, Carolina Mora Lopez, Catalin Mitelut, Silke Musa, Michael Okun, Marius Pachitariu, Jan Putzeys, P. Dylan Rich, Cyrille Rossant, Wei-lung Sun, Karel Svoboda, Matteo Carandini, Kenneth D. Harris, Christof Koch, John O'Keefe, and Timothy D. Harris. Fully integrated silicon probes for high-density recording of neural activity. *Nature*, 551(7679): 232–236, November 2017. ISSN 0028-0836, 1476-4687. doi: 10.1038/nature24636. URL https://www.nature.com/articles/nature24636.

Mario Koppen. The curse of dimensionality. 2000.

Nina Kudryashova, Theoklitos Amvrosiadis, Nathalie Dupuy, Nathalie Rochefort, and Arno Onken. Parametric Copula-GP model for analyzing multidimensional neuronal and behavioral relationships. *PLOS Computational Biology*, 18(1):e1009799, January 2022. ISSN 1553-7358. doi: 10.1371/journal.pcbi.1009799. URL https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1009799. Publisher: Public Library of Science.

Rylan S. Larsen, Emily Turschak, Tanya Daigle, Hongkui Zeng, Jun Zhuang, and Jack Waters. Activation of neuromodulatory axon projections in primary visual cortex during periods of locomotion and pupil dilation, December 2018. URL https://www.biorxiv.org/content/10.1101/502013v1. Pages: 502013 Section: New Results.

Jian Ma and Zengqi Sun. Mutual Information Is Copula Entropy. *Tsinghua Science & Technology*, 16 (1):51–54, February 2011. ISSN 1007-0214. doi: 10.1016/S1007-0214(11)70008-6. URL https://www.sciencedirect.com/science/article/pii/S1007021411700086.

Alexander Mathis, Pranav Mamidanna, Kevin M. Cury, Taiga Abe, Venkatesh N. Murthy, Mackenzie Weygandt Mathis, and Matthias Bethge. DeepLabCut: markerless pose estimation of user-defined body parts with deep learning. *Nature Neuroscience*, 21(9):1281–1289, September 2018. ISSN 1546-1726. doi: 10.1038/s41593-018-0209-y. URL https://www.nature.com/articles/s41593-018-0209-y. Number: 9 Publisher: Nature Publishing Group.

Robert A. McDougal, Thomas M. Morse, Ted Carnevale, Luis Marenco, Rixin Wang, Michele Migliore, Perry L. Miller, Gordon M. Shepherd, and Michael L. Hines. Twenty years of ModelDB and beyond: building essential modeling tools for the future of neuroscience. *Journal of Computational Neuroscience*, 42(1):1–10, February 2017. ISSN 1573-6873. doi: 10.1007/s10827-016-0623-7. URL https://doi.org/10.1007/s10827-016-0623-7.

Lazaros Mitskopoulos, Theoklitos Amvrosiadis, and Arno Onken. Mixed vine copula flows for flexible modeling of neural dependencies. *Frontiers in Neuroscience*, 16:910122, September 2022. ISSN 1662-453X. doi: 10.3389/fnins.2022.910122. URL https://www.frontiersin.org/articles/10.3389/fnins.2022.910122/full.

Thomas Nagler and Claudia Czado. Evading the curse of dimensionality in nonparametric density estimation with simplified vine copulas. *Journal of Multivariate Analysis*, 151:69–89, October 2016. ISSN 0047-259X. doi: 10.1016/j.jmva.2016.07.003. URL https://www.sciencedirect.com/science/article/pii/S0047259X16300471.

Roger B. Nelsen. *An introduction to copulas*. Springer series in statistics. Springer, New York Berlin Heidelberg, 2. ed edition, 2006. ISBN 978-0-387-28659-4 978-1-4419-2109-3.

A. Onken and S. Panzeri. Mixed vine copulas as joint models of spike counts and local field potentials. December 2016. URL https://www.semanticscholar.org/paper/Mixed-vine-copulas-as-joint-models-of-spike-counts-Onken-Panzeri/07bb8182920fce1ec11522a1dbc82d546658efb1.

Janelle MP Pakan, Valerio Francioni, and Nathalie L Rochefort. Action and learning shape the activity of neuronal circuits in the visual cortex. *Current Opinion in Neurobiology*, 52:88–97, October 2018. ISSN 0959-4388. doi: 10.1016/j.conb.2018.04.020. URL https://www.sciencedirect.com/science/article/pii/S0959438818300400.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/hash/bdbca288fee7f92f2bfa9f7012727740-Abstract.html.

Felix Pei, Joel Ye, David 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, January 2022. URL http://arxiv.org/abs/2109.04463. arXiv:2109.04463 [cs, q-bio].

Gregor Pirš and Erik Štrumbelj. Bayesian Gaussian process factor analysis with copula for count data. *Expert Systems with Applications*, 197:116645, July 2022. ISSN 0957-4174. doi: 10.1016/j.eswa.2022.116645. URL https://www.sciencedirect.com/science/article/pii/S0957417422001336.

Rodrigo Quian Quiroga and Stefano Panzeri. Extracting information from neuronal populations: information theory and decoding approaches. *Nature Reviews Neuroscience*, 10(3):173–185, March 2009. ISSN 1471-0048. doi: 10.1038/nrn2578. URL https://www.nature.com/articles/nrn2578. Number: 3 Publisher: Nature Publishing Group.

Sumanta Ray, Snehalika Lall, and Sanghamitra Bandyopadhyay. CODC: a Copula-based model to identify differential coexpression. *npj Systems Biology and Applications*, 6(1):1–13, June 2020. ISSN 2056-7189. doi: 10.1038/s41540-020-0137-9. URL https://www.nature.com/articles/s41540-020-0137-9. Publisher: Nature Publishing Group.

Christian P. Robert and George Casella. Monte Carlo Integration. In Christian P. Robert and George Casella (eds.), *Monte Carlo Statistical Methods*, pp. 71–138. Springer, New York, NY, 1999. ISBN 978-1-4757-3071-5. doi: 10.1007/978-1-4757-3071-5_3. URL https://doi.org/10.1007/978-1-4757-3071-5_3.

Houman Safaai, Arno Onken, Christopher D. Harvey, and Stefano Panzeri. Information estimation using nonparametric copulas. *Physical Review E*, 98(5):053302, November 2018. ISSN 2470-0045, 2470-0053. doi: 10.1103/PhysRevE.98.053302. URL http://arxiv.org/abs/1807.08018. arXiv:1807.08018 [cs, math, q-bio, stat].

Eric Schulz, Maarten Speekenbrink, and Andreas Krause. A tutorial on Gaussian process regression: Modelling, exploring, and exploiting functions. *Journal of Mathematical Psychology*, 85:1–16, August 2018. ISSN 0022-2496. doi: 10.1016/j.jmp.2018.03.001. URL https://www.sciencedirect.com/science/article/pii/S0022249617302158.

Gideon Schwarz. Estimating the Dimension of a Model. *The Annals of Statistics*, 6(2):461–464, 1978. ISSN 0090-5364. URL https://www.jstor.org/stable/2958889. Publisher: Institute of Mathematical Statistics.

Babak Shahbaba, Bo Zhou, Shiwei Lan, Hernando Ombao, David Moorman, and Sam Behseta. A Semiparametric Bayesian Model for Detecting Synchrony Among Multiple Neurons. *Neural Computation*, 26(9):2025–2051, September 2014. ISSN 0899-7667, 1530-888X. doi: 10.1162/NECO_a_00631. URL https://direct.mit.edu/neco/article/26/9/2025-2051/8006.

Claude Elwood Shannon and Warren Weaver. *The mathematical theory of communication*. University of Illinois Press, Urbana, 1998. ISBN 978-0-252-72546-3. URL https://www.loc.gov/catdir/enhancements/fy1616/98230924-b.html. OCLC: 40716662.

Nicholas A. Steinmetz, Cagatay Aydin, Anna Lebedeva, Michael Okun, Marius Pachitariu, Marius Bauza, Maxime Beau, Jai Bhagat, Claudia Böhm, Martijn Broux, Susu Chen, Jennifer Colonell, Richard J. Gardner, Bill Karsh, Fabian Kloosterman, Dimitar Kostadinov, Carolina Mora-Lopez, John O'Callaghan,

Junchol Park, Jan Putzeys, Britton Sauerbrei, Rik J. J. van Daal, Abraham Z. Vollan, Shiwei Wang, Marleen Welkenhuysen, Zhiwen Ye, Joshua T. Dudman, Barundeb Dutta, Adam W. Hantman, Kenneth D. Harris, Albert K. Lee, Edvard I. Moser, John O'Keefe, Alfonso Renart, Karel Svoboda, Michael Häusser, Sebastian Haesler, Matteo Carandini, and Timothy D. Harris. Neuropixels 2.0: A miniaturized high-density probe for stable, long-term brain recordings. *Science*, 372(6539):eabf4588, April 2021. doi: 10.1126/science.abf4588. URL https://www.science.org/doi/full/10.1126/science.abf4588. Publisher: American Association for the Advancement of Science.

Robert J. Sternberg and Edward E. Smith. *The Psychology of Human Thought*. CUP Archive, February 1988. ISBN 978-0-521-31115-1. Google-Books-ID: DZ44AAAAIAAJ.

Dugald Stewart. *Elements of the Philosophy of the Human Mind*. E. and E. Hosford, 1822. Google-Books-ID: h98AAAAAYAAJ.

Ben Van Vliet. Abe Sklar's, March 2023. URL https://papers.ssrn.com/abstract=4198458.

Mai-Anh T. Vu, Tülay Adalı, Demba Ba, György Buzsáki, David Carlson, Katherine Heller, Conor Liston, Cynthia Rudin, Vikaas S. Sohal, Alik S. Widge, Helen S. Mayberg, Guillermo Sapiro, and Kafui Dzirasa. A Shared Vision for Machine Learning in Neuroscience. *Journal of Neuroscience*, 38(7): 1601–1607, February 2018. ISSN 0270-6474, 1529-2401. doi: 10.1523/JNEUROSCI.0508-17.2018. URL https://www.jneurosci.org/content/38/7/1601. Publisher: Society for Neuroscience Section: Tech-Sights.

Sumio Watanabe. [PDF] A widely applicable Bayesian information criterion | Semantic Scholar, August 2012. URL https://www.semanticscholar.org/paper/A-widely-applicable-Bayesian-information-criterion-Watanabe/d9c9b74c8cbfa5475c8e91312efac956bf0518ab.

A. Wilson and H. Nickisch. Kernel Interpolation for Scalable Structured Gaussian Processes (KISS-GP). March 2015. URL https://www.semanticscholar.org/paper/Kernel-Interpolation-for-Scalable-Structured-Wilson-Nickisch/767d0625c4767c6b8afa0b1b30deafed7e0e8f08.

Byron M. Yu, John P. Cunningham, Gopal Santhanam, Stephen I. Ryu, Krishna V. Shenoy, and Maneesh Sahani. Gaussian-Process Factor Analysis for Low-Dimensional Single-Trial Analysis of Neural Population Activity. *Journal of Neurophysiology*, 102(1):614–635, July 2009. ISSN 0022-3077, 1522-1598. doi: 10.1152/jn.90941.2008. URL https://www.physiology.org/doi/10.1152/jn.90941.2008.

Zhi-Hua Zhou. Why over-parameterization of deep neural networks does not overfit? *Science China Information Sciences*, 64(1):116101, January 2021. ISSN 1674-733X, 1869-1919. doi: 10.1007/s11432-020-2885-6. URL https://link.springer.com/10.1007/s11432-020-2885-6.

Ariel Zylberberg, Manuel Oliva, and Mariano Sigman. Pupil Dilation: A Fingerprint of Temporal Selection During the "Attentional Blink". *Frontiers in Psychology*, 3, August 2012. ISSN 1664-1078. doi: 10.3389/fpsyg.2012.00316. URL https://www.frontiersin.org/journals/psychology/articles/10.3389/fpsyg.2012.00316/full. Publisher: Frontiers.

A Information Theory Essentials

Information theory, in broad terms, is the study of formal information quantification and communication Shannon & Weaver (1998). The key measure of information in this field is *entropy*, defined as $H(X) = -\mathbb{E}_X(\log p(x)),\ x \sim X$. Essentially, entropy measures the number of *bits of information* needed to encode the total uncertainty of event X Shannon & Weaver (1998), and as such acts as a quantification of uncertainty. We also define the conditional entropy of X given the outcome y of event Y as

$$H(X|Y) = -\mathbb{E}_{X,Y}(\log \frac{p(x,y)}{p(y)}),\ \ x \sim X,\ y \sim Y, \tag{6}$$

and the *mutual information* between X and Y as $I(X:Y) = H(X) - H(X|Y)$. Mutual information thus represents the *reduction in bits of uncertainty* surrounding X given the outcome of Y. In addition, this quantity is *symmetric* Shannon & Weaver (1998), and so $I(X:Y) = H(Y) - H(Y|X)$ as well. In other words, **mutual information describes statistical co-dependence between events X and Y.** In addition, we note that mutual information is positive semi-definite (≥ 0), and so only ever represents a reduction in uncertainty (this is intuitive; the unconditioned outcome of an event naturally encompasses all conditions). We extend this to the *conditional mutual information* of X and Y given the outcome of event Z, $I(X:Y|Z) = H(X|Z) - H(X,Y|Z)$, where $H(X,Y|Z)$ is the *conditional joint entropy* of X and Y given the outcome of Z, or the bits of uncertainty in the joint system of X and Y under condition z Shannon & Weaver (1998). Finally, mutual information is easily extendable to a definition of *joint mutual information* via replacing singular entropy with it's joint definition (as traditional entropy itself is not the primary concern of this paper, we will skip rigorously defining these terms for brevity) Shannon & Weaver (1998). Joint mutual information and it's conditional counterpart are the primary quantities of interest we wish to estimate in this paper as we wish to extract the *interaction information* between $\mathbf{X}$ and Y $I(\mathbf{X} \leftarrow Y) = I(\mathbf{X}|Y) - I(\mathbf{X})$, representing the change in the mutual information between marginals of $\mathbf{X}$ when Y is learned, or the amount of uncertainty surrounding $\mathbf{X}$ captured in Y.

B Python Version and Hardware Utilized

Computations were made on the Edinburgh University compute server utilizing 128 GB of system memory, an Intel Xeon Gold 6142 processor, an NVidia 2080, and an NVidia 2080Ti. Plots were made on a laptop with 8 GB of system memory and an Apple M2 processor. The python version utilized for all computations and plots made was 3.10.6; *Elephant* 1.0.0 and Copula-GP 0.0.5 were utilized for computations, and *Matplotlib* 3.6.1 and *Seaborn* 0.10.0 were utilized for plot production.

C Copulas and their Entropy

In this section, we expand further upon formal copula theory.

C.1 Definition and Basic Properties

Copulas are multivariate distributions with uniform marginal distributions, which themselves usually represent marginal variables' cumulative distributions Kudryashova et al. (2022); Safaai et al. (2018) A d-dimensional copula function $C(u_1, u_2, \ldots, u_d) : [0,1]^d \to [0,1]$ is defined as a *cumulative distribution function (CDF)* of a vector on the unit hyper-cube $[0,1]^d$ with uniform marginals $\mathcal{U}_{[0,1]}$:

$$C(u_1, u_2, \ldots, u_d) = \Pr(U_1 \leq u_1, U_2 \leq u_2, \ldots, U_d \leq u_d), \tag{7}$$

where $U_n \sim \mathcal{U}_{[0,1]}$ Kudryashova et al. (2022); Safaai et al. (2018). Variables of non-uniform distribution are "attached" to the marginals through the use of CDFs (which by definition map to the interval $[0,1]$ Van Vliet (2023)) as functions of non-uniformly distributed variables, creating a copula as a joint CDF Nelsen (2006). *Sklar's theorem* states that for a d-dimensional random vector $\mathbf{X} = \{X_1, X_2, \cdots, X_d\}$ and it's CDF $F_{\mathbf{X}}$ with marginals $F_1, F_2, \cdots, F_d$, there exists a copula C such that $\forall \mathbf{x} \in \mathbb{R}^d$, $\mathbf{x} \sim \mathbf{X}$,

$$F_{\mathbf{X}}(x_1, x_2, \ldots, x_d) = C(F_1(x_1), F_2(x_2), \ldots, F_d(x_d)), \;\; x_i \in \mathbb{R}. \tag{8}$$

Safaai et al. (2018); Van Vliet (2023) What is most valuable about this construction is the allowance for the random variables $X_1, X_2, \cdots, X_d$ to take *any* distribution, allowing for meaningful dependency and mutual information analysis between variables of different probabilistic distributions Safaai et al. (2018). We can also condition the copula on some continuous variable y, allowing for the parametrization of the copula for a variable like time, phase, other marginals, etc Kudryashova et al. (2022):

$$F_{\mathbf{X}}(x_1, x_2, \ldots, x_d|y) = C(F_1(x_1|y), F_2(x_2|y), \ldots, F_d(x_d|y)|y), \;\; x_i, y \in \mathbb{R}. \tag{9}$$

Depending on how we define a copula's CDF, we can dynamically change the shape of the copula to represent different dependency relationships, with the intention of better fitting the marginal variables' dynamic dependency shape (i.e. dynamic tail dependencies, transitions into different copula families over time, dynamic

increases or decreases in marginal distributions' correlation) Safaai et al. (2018); Kudryashova et al. (2022). Doing so imposes strong assumptions on data however, and can introduce biases in analysis when the shape of the copula does not fit the true relationship of the individual variables Safaai et al. (2018). Various copula families used in the core paper of this project Kudryashova et al. (2022) are depicted in figure 6.

C.2 Copula Entropy

As before, we consider a random vector $\mathbf{X} = \{X_1, X_2, \ldots, X_d\}$, where F_i describes a marginal distribution of X_i. let $\mathbf{u} = [u_1, u_2, \ldots, u_d]$, with $u_i = F_i(x_i)$. By equation 8 we know $F_{\mathbf{X}}$ is described by a copula C. We define their copula entropy as

$$H_C(\mathbf{X}) = -\int_C c(\mathbf{u}) \log c(\mathbf{u}) d\mathbf{u}, \tag{10}$$

where $c(\mathbf{u})$ represents the probability density of the copula at $\mathbf{u}$

$$c(\mathbf{u}) = \frac{\partial^d C}{\partial u_1 \partial u_2 \ldots \partial u_d}. \tag{11}$$

Ma & Sun (2011); Jenison & Reale (2004) Let $\mathbf{x}$ be a random sample $\mathbf{x} \sim \mathbf{X}$.

$$\begin{aligned} I(\mathbf{X}) &= \int_{\mathbf{x}} F_X(\mathbf{x}) \log \frac{F_X(\mathbf{x})}{\prod_i F_i(x_i)} d\mathbf{x} \\ &= \int_{\mathbf{x}} c(\mathbf{u_x}) \prod_i F_i(x_i) \log c(\mathbf{u_x}) d\mathbf{x} \\ &= \int_{\mathbf{x}} F_X(\mathbf{x}) c(\mathbf{u_x}) \log(c(\mathbf{u_x})) d\mathbf{x} \\ &= \int_C c(\mathbf{u}) \log c(\mathbf{u}) d\mathbf{u} \\ &= -H_C(\mathbf{X}). \end{aligned} \tag{12}$$

Ma & Sun (2011); Jenison & Reale (2004) Ma & Sun (2011); Jenison & Reale (2004) In other words, **mutual information is negative copula entropy.** This also implies that copula entropy is negative semi-definite as opposed to traditional entropy, which is positive semi-definite.

Now, suppose we have some other random variable $y \sim Y$. We can extend the definition of copula entropy in equation 10 to be parameterized and thus conditioned in a outside variable y as

$$H_C(\mathbf{X}|y) = -\int_C c(\mathbf{u}|y) \log c(\mathbf{u}|y) d\mathbf{u}, \;\; c(\mathbf{u}|y) = \frac{\partial^d C}{\partial F_{1|y} \partial F_{2|y} \ldots \partial F_{d|y}} \tag{13}$$

(where each $F_{i|y}$ is the respective conditional marginal probability of X_i) Ma & Sun (2011) and can extend further to the definition of *conditional copula entropy* as $H_C(\mathbf{X}|Y) = \int_y H_C(\mathbf{X}|y_i) = -I(\mathbf{X}|Y)$. Information interaction between $\mathbf{X}$ and Y can thus be defined as $I(\mathbf{X} \leftarrow Y) = I(\mathbf{X}|Y) - I(\mathbf{X}) = H_C(\mathbf{X}) - H_C(\mathbf{X}|Y)$, and so if one obtains parametric copula estimates for $\mathbf{X}$ conditioned on Y one can estimate $I(\mathbf{X} \leftarrow Y)$. In the setting of this paper, we hope for the information interaction between the two to be firmly *negative*, implying information surrounding $\mathbf{X}$ is captured in Y and a statistical dependence between the two being posited. Working under the assumption that the brain is a source of behavioral and motor functions such as pupil dilation, then if a part of the brain is tied to such a function we expect the information interaction to be significantly negative.

C.3 Vine Copulas

To scale against higher dimensions, copulas can take a *vine copula construction*, which factorize multivariate distributions into conditional distributions modelled as singular bivariate copulas Onken & Panzeri (2016); Mitskopoulos et al. (2022). A single "vine" in this construction may be represented as a hierarchy of trees,

where each node represents a single CDF (with increasing conditioning for each tree) and each edge is a pair copula. The cumulative distributions of each tree's bivariate building blocks is then used as the nodes of the next tree in the hierarchy, and so extends the set of conditioning variables into the next tree Czado & Nagler (2022). A multivariate distribution $f_{\mathbf{X}}$ with univariate marginal variable distributions $X_1, X_2, \cdots, X_d$ can be factorized in this way as

$$f_{\mathbf{X}}(x_1, \cdots, x_d) = \prod_{k=1}^{d} f(x_k) \prod_{j=1}^{d-1} \prod_{i=1}^{d-j} C_{j,i|1,\cdots,j-1} \left(F(x_j|x_1, \cdots, x_{j-1}), F(x_{i+j}|x_1, \cdots, x_{j-1}) \right). \tag{14}$$

(This specific factorization is called a *canonical vine*, or C-vine Onken & Panzeri (2016).) What is special about vine copulas is both assumed independence between particular marginals in a tree (due to conditional independence) and the decomposition of a high dimensional distribution into bivariate building blocks. As such sampling from and analysis of a high dimensional vine copula is not as affected by the curse of dimensionality given robustness in the bivariate case Czado & Nagler (2022); Nagler & Czado (2016). This, combined with the ability for parameterization in time and/or space, make vine copulas well suited for analysis of complex neuronal recordings Onken & Panzeri (2016); Kudryashova et al. (2022).

C.4 Gaussian-Process for Dependence Parameterization

We further expand upon vine copulas with the addition of *Gaussian Process (GP)* as a method of estimating parameterization of the relationship between a copula's marginals and dependency set Kudryashova et al. (2022); Hernández-Lobato et al. (2013). A GP estimator is a regression attempting to estimate the true relationship between dependent and independent variables by defining a distribution over functions such that observed points might follow a Gaussian distribution (or, in short, by means of Bayesian inference) Schulz et al. (2018). In more rigid terms, we assume output y of a function f given a set of observations $\mathbf{x}$ can be given as

$$y = f(\mathbf{x}) + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \sigma_\epsilon^2), \quad \mathbf{f} \sim \mathcal{GP}\left(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x'})\right) \tag{15}$$

where σ_ϵ^2 takes the usual definition as the variance of ϵ, m is the expected value of $\mathbf{f}$ given $\mathbf{x}$, and k is the covariance between the outputs of two separate sets of observations $\mathbf{x}$ and $\mathbf{x'}$ (this specific parameter is often called the *kernel* of the estimated distribution $\mathcal{GP}$) Schulz et al. (2018). Different methods of estimating k, m, and σ_ϵ^2 exist Schulz et al. (2018); Dudley (2010); Hernández-Lobato et al. (2013), and so a method should be selected based on use case. Incorporating GP into our copula vine, we are able to extend our multivariate distribution to have dynamic dependency relationships estimated by means of GP. To express this mathematically, we redefine individual copulas to take the approximate form

$$F_{\mathbf{X}}(x_1, \cdots, x_d|y') = C(F_1(x_1|\theta_1(y')), F_2(x_2|\theta_2(y')), \cdots, F_d(x_d|\theta_d(y'))|\theta'(y')), \tag{16}$$

where $\theta_1, \cdots, \theta_d, \theta' : \mathbb{R} \to \text{dom(y)}$ are functions mapping some parameter y' onto the domain of the conditioning variable y representing the relationships between $x_1, \cdots, x_d$ and y', often called *GPLink* functions Kudryashova et al. (2022); Schulz et al. (2018). These GPlink functions are what we predict via GP, and how GPLinks are estimated varies depending on the specific use case and implementation of GP Schulz et al. (2018); Kudryashova et al. (2022); Hernández-Lobato et al. (2013). The main innovation extending copulas in this way is both uncertainty in how the relationship between parameters and marginals are defined, as well as parameterization of marginals' dependence. The latter is particularly useful when the actual dependence relationships in the data are not static and may instead be dynamic over time Kudryashova et al. (2022).

Table 2: Different GPLink functions used in Copula-GP for different copula types. GPLinks are used to parameterize dependence of the particular copula in a continuous variable. Courtesy of Kudryashova et al. (2022).

Copula	Domain	GPLink(f): $\mathbb{R} \to \text{dom}(c_j)$
Independence	-	-
Gaussian	$[-1, 1]$	$\text{Erf}(f/1.4)$
Frank	$(-\infty, \infty)$	$0.3 \cdot f + \text{sign}(f) \cdot (0.3 \cdot f)^2$
Clayton	$[0, \infty)$	$\text{Exp}(0.3 \cdot f)$
Gumbel	$[1, \infty)$	$1 + \text{Exp}(0.3 \cdot f)$

C.5 Benefits of Parametric Vs. Non-parametric Copulas

One possible criticism surrounding copulas is that they impose assumptions surrounding the marginal distributions' dependency shape; imposing a certain copula can introduce bias in the data if said copula's assumptions are not met. While this bias could effect our findings in the non-parametric case, utilization of copula parameterization and usage of mixed copula construction (see eq. equation 19) circumvents this through dynamic tailoring of copula variant to the parameterizing value Kudryashova et al. (2022); Onken & Panzeri (2016).

D Data Retrieval and Cleaning

Visual Coding - Neuropixels is split into recording sessions, where during each session a mouse test subject was exposed to varying visual stimuli, and is publicly available via the `allensdk` python package (visit `https://portal.brain-map.org/` and navigate to "circuits and behavior", then "Neuropixels"). Stimuli were presented in "blocks" of similar kinds of stimuli of varying duration. For this project, we examine specifically the first 100 stimuli presentations of the first "drifting gratings" block (block 2), as this way we mitigate variation in neuronal response due to differing stimulus length. In addition, within this block individual presentations are of uniform length (2 seconds of presentation, with a inter-presentation break of 1 second), and so the block is easily divisible into separate spike train trials (the GPFA implementation used is best suited for a trial-by-trial format).

The session data contains the full spike recordings of all recorded units within the subject's visual cortex. As stated prior, these units possess quality metrics that correspond to how accurate and noisy unit recordings are. Of these, we utilize Signal to Noise Ratio (SNR) and Inter Spike Interval (ISI) Violation rate. SNR corresponds to the ratio of the maximum unit waveform amplitude to one standard deviation of the waveform, and acts as a metric of how noisy the unit recording is Biau & Scornet (2016). ISI violation rate is the percentage of unit spikes that occur during what should be the corresponding neuron's refractory period, and serves as a metric to determine whether multiple neurons and/or electronic interference are being picked up in a single unit's recordings Czanner et al. (2015).

Session data used here was pulled from a single session (session ID 756029989). By setting a lower bound for SNR and an upper bound for ISI violation rate, we are able to filter overly noisy and unreliable units' spike recordings out of the data set. Utilizing a SNR lower bound of 3 and an ISI violation rate upper bound of 0.05 (5%), we found that in this session 26.0% of units meet these quality thresholds (178 out of the original total of 684). Example spike data for a single stimulus presentation of 2 seconds is shown in figure 11.

We also utilized pupil area recordings (cm^2) to parameterize the copulas in pupil dilation. Measurements were recorded every 33ms and possess a relatively large range, with sporadic large spikes in pupil dilation (see pupil dilation curves in figure 10a). As such, we applied a rolling mean with a window of 10 entries for smoothing followed by a robust normalization procedure Robust(X):

$$\text{Robust}(X) = \frac{X - Q_1(X)}{Q_3(X) - Q_1(X)}, \tag{17}$$

where $Q_1(X), Q_3(X)$ are the 1st and 3rd quartile values of X. Doing so, we reduce the impact of outliers in the data. As the input for parameterizing values for fitting a C-vine via Copula-GP must be the interval

$[0, 1]$, we map onto via an additional min-max normalization

$$\text{MinMax}(X) = \frac{X - \min(X)}{\max(X) - \min(X)}. \tag{18}$$

We can observe the difference between the raw and processed data in figure 10. See figure 10c for the distribution this preprocessing regime created, which appears roughly normal with few outliers.

The technical white-paper for the dataset is also available at:
`https://brainmapportal-live-4cc80a57cd6e400d854-f7fdcae.divio-media.net/filer_public/80/75/8075a100-ca64-429a-b39a-569121b612b2/neuropixels_visual_coding_-_white_paper_v10.pdf`.

E Copula-GP Implementation

The vine construction implemented in Copula-GP used copula building blocks from five distinct families: independence, Gumbel, Gaussian, Frank, and Clayton copulas, with the independence copula preferred for independent variables (see figure 6). The factorization of the vine is that of the C-vine described by equation 14, with accommodations made for GP-parameterization. To fully capture the tail dependencies and negative correlations in relationships between marginal distributions, *mixed copulas* were utilized. In the core paper, these are defined as

$$C_{mixed}(\mathbf{X}|Y) = \sum_{j=1}^{K} \phi_j(Y) C_j(\mathbf{X}|\theta_j(Y)), \tag{19}$$

where K is the number of elements, ϕ_j is the *concentration* of the jth copula (c_j), and θ_j is the jth copula's parameter GPLink. The GPLink for each copula is determined by it's copula variant, as shown in figure 2, with θ being defined by GPLink(f), where f is sampled from $\theta \sim \mathcal{N}(\mu, K_\lambda(X, X))$ (the choice of GPLink depends on the kind of copula; see 2). GP is also utilized to parameterize the concentrations ϕ_j, which are defined as

$$\phi_j = (1 - t_j) \prod_{m=1}^{j-1} t_m, \quad t_m = \Phi\left(\tilde{f}_m + \Phi^{-1}\left(\frac{M - m - 1}{M - m}\right)\right), \quad t_M = 0, \tag{20}$$

where Φ is the CDF of a standard normal distribution and $\tilde{f}_m$ is sampled from $\tilde{\mathbf{f}}_\mathbf{m} \sim \mathcal{N}(\tilde{\mu}_m, \quad \tilde{K}_{\tilde{\lambda}_m}(Y, Y))$. This gives us $2M - 1$ sets of hyper parameters to estimate, $\{\lambda\}_M$ kernel hyperparameters for each GPLink θ and $\{\tilde{\lambda}\}_{M-1}$ kernel hyperparameters for each concentration function ϕ, estimated via the methods described in section E.1.

E.1 Copula-GP Model Selection and Parameter Estimation

As stated in the body of this paper, the parameters for the distributions used in GP must be estimated. In the Copula-GP framework, this is accomplished through the use of *stochastic variational inference (SVI)*, with SVI being scaled to high dimensions by means of *Kernel Interpolation for Scalable Structured Gaussian Process (KISS-GP)* Wilson & Nickisch (2015). These methods were specifically used for efficient implementation in aforementioned python libraries *PyTorch* and *GPyTorch* Paszke et al. (2019); Gardner et al. (2018). For model hyper-parameter selection, *Watanabe-–Akaike information criterion (WAIC)* was used, a metric which aims to maximise the Akaike information criterion (AIC) by means of a Bayesian-approach (we more rigorously define AIC in section 7). Given f_1, f_2 and the true distribution g, WAIC examines the difference in log-likelihood i.e. $I(g : f_1) - I(g : f_2) = -\mathbb{E}(\log f_1(X) - \log f_2(X))$, and selects the model with the greater mutual information Watanabe (2012). The space of models is too large to find the optimal model when considering the number of combinations of copulas shown in 6, and as such the core paper implements a greedy algorithm of minimising WAIC which can be used with all copula types and a heuristic algorithm specifically tuned for certain combinations of copula types. For this project, we utilize the heuristic approach of copula selection.

E.2 Dimensionality Selection

Before we can utilize GPFA, we first must isolate which number of dimensions n to reduce down to. For the *in vivo* dataset we chose n via investigating the log likelihood of GPFA fits extracted from a 3-fold cross validation, from target dimensionality $n = 1$ to 50. We then plotted the log likelihoods and saw both where the elbow, or the point of maximum concave slope curvature representing a "good enough" dimension to reduce down to Antunes et al. (2018) of the log likelihood curve was. While there are computational methods for elbow selection Antunes et al. (2018), we investigated few enough points to allow for the elbow to be selected ourselves visually.

F GPFA Specifics

F.1 Additional Post-GPFA Interim Processing Required

Copula-GP's C-vine framework is fit on single-trial continuous, however *Elephant*'s GPFA implementation produces trajectory data that is split into trials. A solution to this would be to concatenate trials trajectory wise, however per-trial drift in trajectory means can result in weak Copula-GP fit performance if these lead to large jump discontinuities and thus loss of smoothness in the data; the kernel for the GP-link functions used in parameterization will require more restrictions as it encodes the smoothness of the data (among other things) Schulz et al. (2018); Yu et al. (2009). For the *in vivo* data, we found in figure 7 that drift occurs in the 1 second inter-stimulus break in the average trial. As such, we crop this period out of each trial (50 points), and concatenate trials together trajectory-wise. While this is by far not a perfect solution, it allows the data to remain roughly smooth at the cost of continuity in time and residual (small) jump discontinuities, as well as isolating the data to only when stimulus presentations are occurring. See figure 8 for a example of trial-to-trial discontinuities created by this interim step.

F.2 Python Implementation of GPFA Used

The implementation of GPFA utilized is sourced from the *Elephant* (Electrophysiology Analysis Toolkit) python library, with the package's 1.0.0 release (used for the contents of this paper) being published November 10, 2023 Denker & Kern (2023). The *Elephant* package was specifically designed for use on neuronal data, motivated by a push to release a standardized python package for use in computation neuroscience. The GPFA module has specifically seen use in recent papers Pei et al. (2022); Bagi et al. (2022), and since release has become quite popular. The GPFA module receives some dimension n to reduce down to and time-bucket size m (we utilize 10ms and 20ms buckets) to instantiate a GPFA python object. This object can then be fit on a number of spike train recordings of uniform temporal length given the start and end time of each recording, summing the number of spikes for each time bucket and utilizing the bucketed spike counts as the observation matrix in equation equation 5. As each projection corresponds to the expectation of the latent trajectories $\mathbb{E}(\Theta|\Psi)$, these parameters are estimated in the *Elephant* implementation via expectation maximization Yu et al. (2009).

G Validation Test Reproducability

All validation tests were done using variations of arguments for the `validate_bagging.py` python file.

Test 1 For test 1, we utilized:

- Random Seed: 859448723
- Num. Estimators: 4
- Maximum Copula Elements: 5
- Dimensions: 2

- Shuffling of Data: Yes
- Input Type: Random

Test 2 For test 2, we utilized:

- Random Seed: 859443
- Num. Estimators: 4
- Maximum Copula Elements: 3
- Dimensions: 2
- Shuffling of Data: Yes
- Input Type: Random

Test 3 For test 3, we utilized:

- Random Seed: 859443
- Num. Estimators: 4
- Maximum Copula Elements: 3
- Dimensions: 2
- Shuffling of Data: No
- Input Type: Linear

H Additional Figures

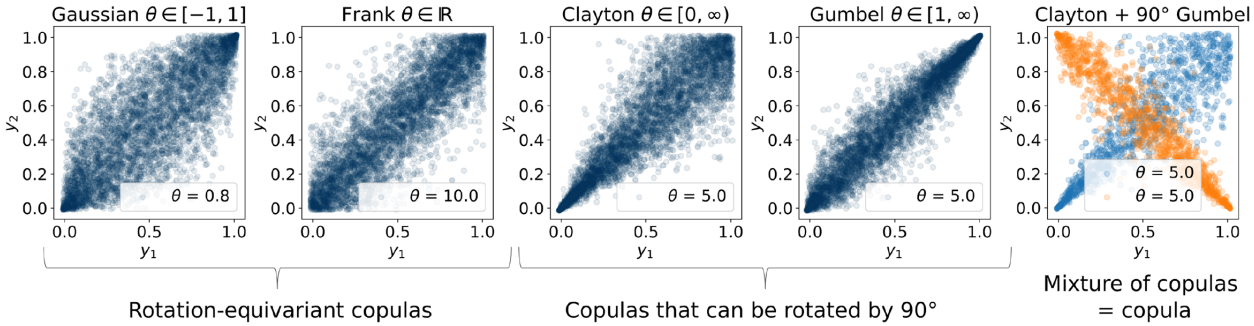
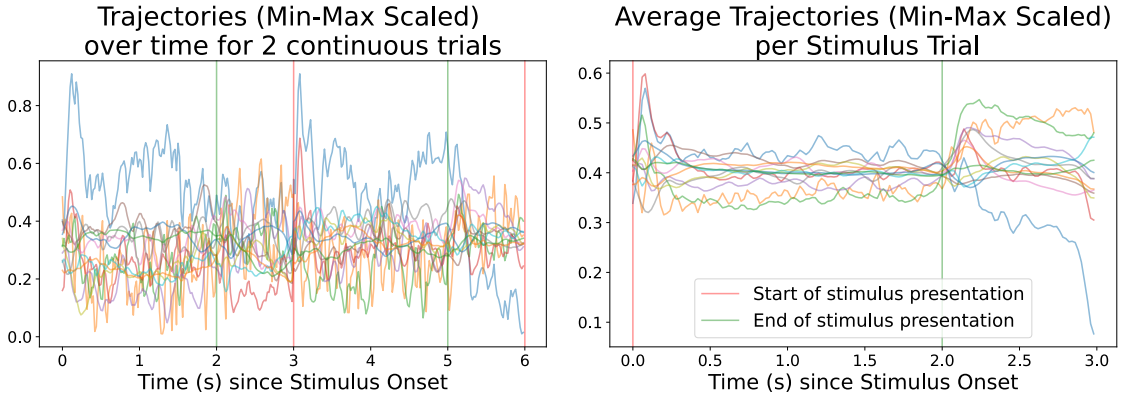


Figure 6: Various kinds of copulas utilized in Copula-GP. Note the difference in tail distribution representation, as well as how combining copula variants make a new *mixed* copula. Original source Kudryashova et al. (2022), with permission.



(a) (b)

Figure 7: Trajectories extracted via GPFA (dimension $n = 13$) representing dynamics driving recorded neuronal activity. Figure 7a are two trials' trajectory data as functions in time that have been concatenated, giving the appearance of continuity. Figure 7b is the mean trial. Note the clear vertical drift in mean and single trial trajectories post-stimulus presentation stop. Data is scaled to the range [0.01,0.99] to match Copula-GP's input range of (0,1). Lines are transluscent for visibility purposes.

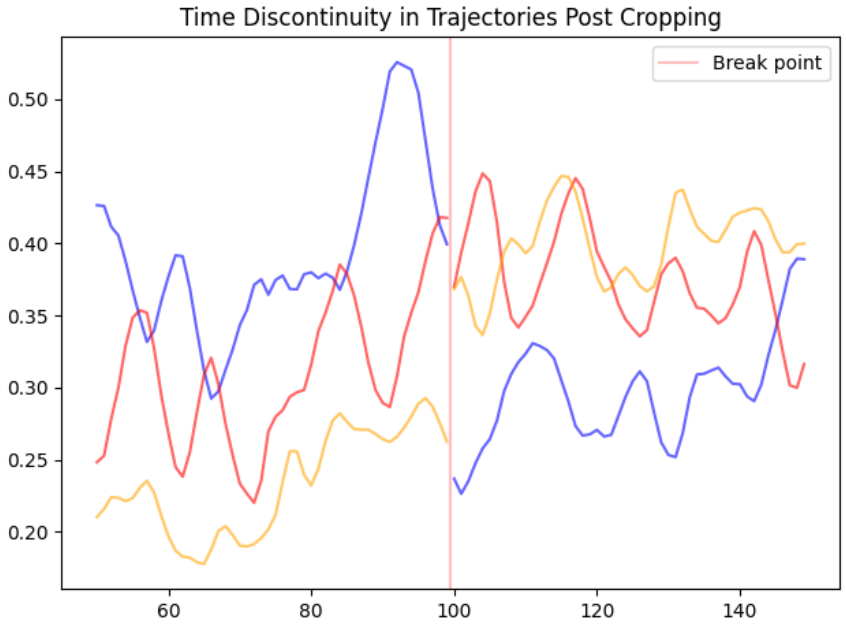


Figure 8: Discontinuities present in trajectory data post-cropping and -concatenation (3 of 13 trajectories shown). Such discontinuities can negatively effect GP performance when estimating GP-link functions during the Copula-GP fit process.

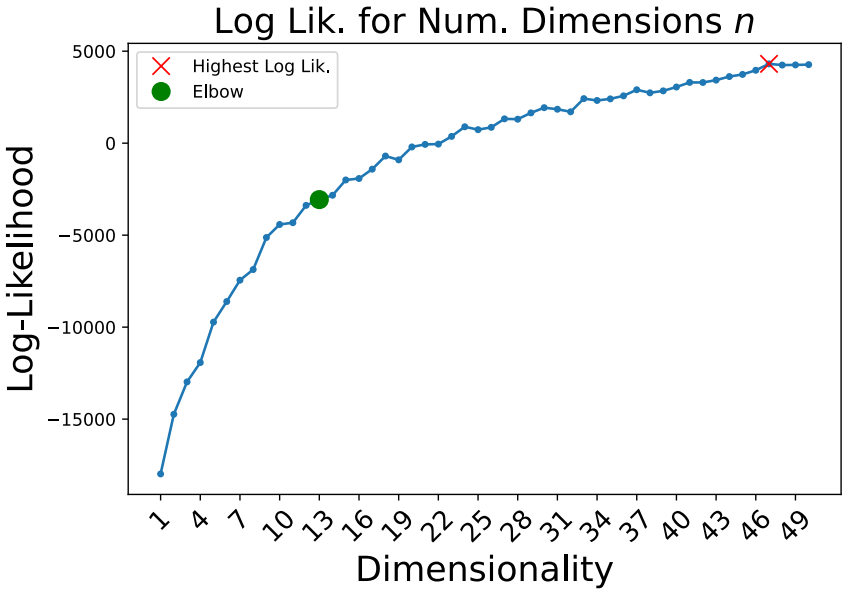
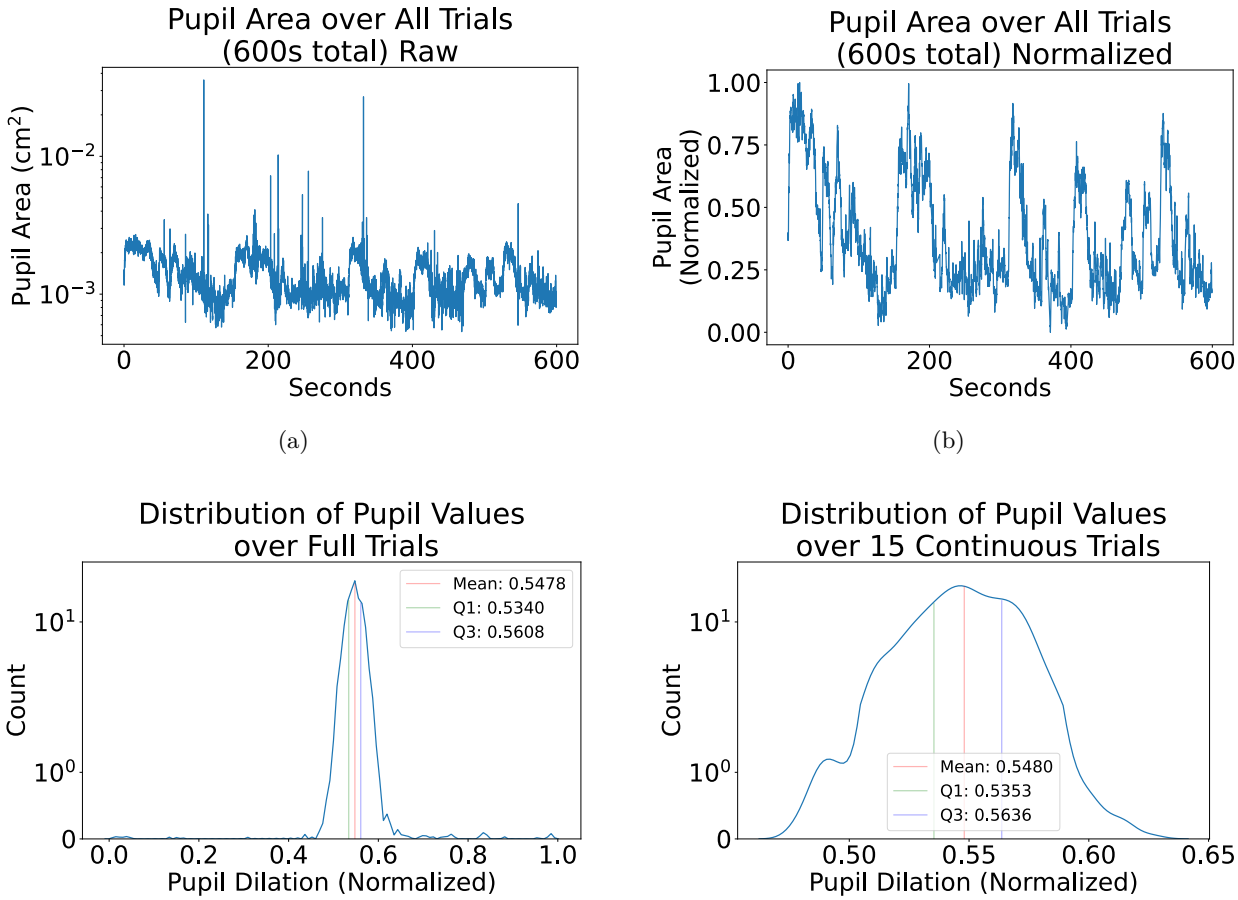


Figure 9: Log-likelihood curve for GPFA fit on trials as function of dimension n. Elbow and max log-likelihood found marked. Note that only dimension up to and including $n = 50$ were tested, and it is entirely possible log-likelihood to increase further in dimensions.



(a) (b) (c) (d)

Figure 10: Figures 10a and 10b represent pupil dilation as function of time over all trials (duration 600s), raw (10a, pupil area in cm^2 scaled logistically) and processed (10b, pupil area smoothed and normalized). Processing consisted of rolling mean (window of 10 entries), a robust scaling, and a min-max scaling. Note the removal of strong outliers through processing, as well as the large range necessitating a logistic scale in values present in the raw recordings. Figure 10c represents the distribution over all pupil dilation data utilized (100000 continuous samples / 100 trials). Figure 10d represents the distribution over 1500 continuous samples (15 trials). Note the use of logistic scale in figures 10a, 10c, and 10d.

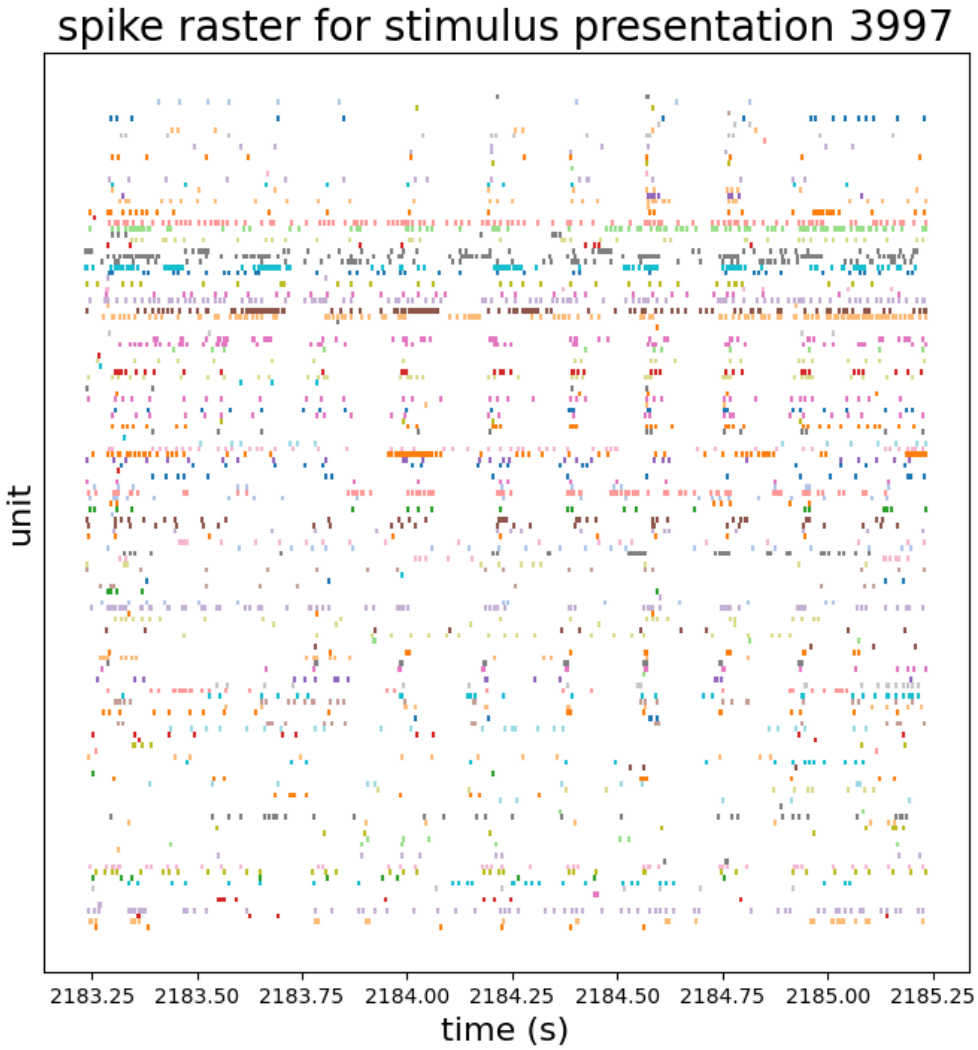


Figure 11: Spike raster of a single drifting gratings stimulus presentation (duration of 2s), from the session utilized in model validation (session ID 756029989). Only those spikes with SNR lower bound of 3 and ISI violation rate upper bound of 0.05 are shown. Note correlation in spike events present in the raster plot, as well as heightened neuronal spike-rate at the beginning of stimulus presentation.