Convex Conceptual Regions in Neural Representations

A Thesis
submitted to the Department of Computer Science
of the Faculty of Engineering, University of Freiburg
for the acquisition of the degree of
Bachelor of Science
in
Computer Science

Raphael Deuerlein
June 2026

Examiner: Prof. Joschka Boedecker
Supervisor: Hao Zhu

Convex Conceptual Regions in Neural Representations

Raphael Deuerlein

Department of Computer Science
University of Freiburg

universität freiburg

University of Freiburg
Freiburg im Breisgau, Baden-Württemberg, Germany

Copyright © 2026 Raphael Deuerlein
All rights reserved.

This publication is in copyright. Subject to statutory exception
no reproduction of any part in any form by print, photoprint,
microfilm, electronic or any other means may take place
without the written permission of the author.

Contents

Declaration

I hereby declare, that I am the sole author and composer of my thesis and that no other sources or learning aids, other than those listed, have been used. Furthermore, I declare that I have acknowledged the work of others by providing detailed references of said work. I hereby also declare, that my thesis has not been prepared for another examination or assignment, either wholly or excerpts thereof.

Place, Date

Signiture

Abstract

A geometric view of conceptual representation holds that concepts are represented as regions in an internal similarity space, and natural concepts as *convex* regions of that space. Building on this idea, recent work has shown that the decision regions of trained artificial neural networks become increasingly convex across processing layers. Whether biological neural populations exhibit comparable convex organization, however, has not yet been directly tested.

We adapt the graph-based convexity framework of Tětková *et al.* to neural spiking data and apply it to the IBL Brainwide Map dataset, in which mice perform a perceptual decision-making task. We measure convexity of population activity with respect to the animal's choice, using time within the decision interval as a substitute for the layer depth available in artificial networks.

We find that brainwide convexity of recorded population activity rises from near baseline at stimulus onset to substantially elevated values as the decision is expressed. It emerges across all regions examined but differs in timing and magnitude, and convex representation is positively associated with decision accuracy. These results indicate that convex organization of choice-related representations is a feature of biological neural populations, not only of artificial ones.

Chapter 1

Introduction

How does the mind represent the concepts it has learned? One influential answer is a geometric view, where concepts are regions in a representational space in which similar items lie close together [She87]. This turns categorization into a question about geometry, but it leaves a further one open. If concepts are regions, what shape do those regions take? Gärdenfors gives a specific answer: natural concepts correspond to *convex* regions, regions with no internal gaps, such that anything lying between two members of a category is itself a member [Gär90, Gär00]. Since this is a claim about relational structure rather than about any particular implementation, it can be asked of any representational system, biological or artificial.

If convex organization of internal representations is a genuine principle of neural systems, it should be observable across systems. While convex category structure has recently been demonstrated in artificial neural networks, a property found only in artificial systems leaves open whether it reflects something general about neural representations or merely an artifact of a particular architecture and training procedure. Observing it in biological neural systems would therefore provide additional support for the former.

Turning this from a conceptual claim into an empirical one requires a bridge from the abstract representational space to something measurable. The representational geometry framework provides it by treating the joint activity of a population of units, whether artificial activations or the firing rates of recorded neurons, as a point in a high-dimensional space, and treating the distances among these points as the representational relations the system has learned [KK13]. Under this framework the convexity prediction becomes concrete and testable: members of a category should occupy an approximately convex region of the population's activity space.

Tětková *et al.* recently operationalized exactly this for deep neural networks [TBD+25]. They tracked the convexity of class regions across the layers of trained networks using two measures: a graph-based measure, suited to the curved, low-dimensional structure on which representations tend to lie, and an interpolation-based Euclidean measure. Both revealed the consistent pattern across modalities and architectures, that convexity rises systematically with layer depth. They also showed that the convexity of pretrained representations predicts downstream few-shot performance after fine-tuning, indicating that this organization also has functional

relevance.

Whether biological neural populations exhibit the same emergence of convex category structure is, to date, an open question.

Transferring this framework to biological neural recordings is not straightforward. Four difficulties make it a methodological and empirical problem. First, recording coverage is small. We only ever observe a small subsample of the total neurons in a brain region. How far do findings from such a subsample extend to the full population? Second, the brain offers no discrete layers. Computation unfolds continuously through recurrent dynamics instead of a globally shared sequence of processing stages. We therefore miss an axis across which the emergence of structure can be compared. Third, population activity is shaped by many processes beyond the task itself. Neural states therefore represent more than just task-dependent variables. Fourth, there is no direct possibility of arbitrary querying of activation states to assign decision labels to unobserved states, which rules out the interpolation-based measurement of Euclidean convexity available in the artificial case.

With this thesis, we pursue two aims. The first is scientific: to test whether the convexity signature found in artificial networks also appears in biological neural representations, by asking whether the convexity of population activity with respect to an animal's choice increases over the course of a decision. The second is methodological: to develop a principled and reusable way of applying the graph-convexity framework to neural population recordings, so that the method can be applied to other datasets and analyses than those examined here.

The contributions of this thesis are the following:

- An adaptation of the graph-convexity framework of Tětková *et al.* from artificial networks to biological neural population recordings, in which time takes the place of layer depth, population activity is summarized as a discrete state vector at successive time points, and convexity is computed over these observed states alone.

- A brainwide analysis showing that convexity of neural representations with respect to choice is near baseline at stimulus onset and rising toward movement onset, following a non-linear, accelerating time course.

- A regional analysis showing that choice-related convexity emerges across the regions examined, while its temporal profile - timing, magnitude, and persistence - differs, together with a careful account of what these differences can and cannot support given the limited data per region.

- A behavioral analysis relating session-level convexity to decision accuracy, finding a positive, moderate correlation consistent with the functional expectation that convex organization supports decision making.

- `cvxnr`, a reusable Python implementation of the adapted pipeline for measuring graph-convexity in neural spiking data.

1.1 Thesis Outline

The remainder of this thesis is organized as follows. The *Background and Theory* chapter develops the theoretical and methodological foundations of the work, beginning with the geometric view of concepts and Gärdenfors' convexity hypothesis, connecting it to neural population geometry and graph-based convexity, reviewing the framework of Tětková *et al.* in artificial networks, and setting out in detail the challenges of transferring that framework to biological recordings. The *Methods* chapter describes how the adapted method is operationalized and applied to the IBL spiking data: the construction of population vectors and the temporal alignment of trials, the estimation of graph-convexity (computed as in Tětková *et al.*), and the brainwide, regional, and behavioral analyses. The *Results* chapter presents the empirical findings of each analysis. The *Discussion* chapter interprets these results, relates them to the predictions and to the findings of Tětková *et al.*, and examines the methodological limitations and the directions they open up for future work.

Chapter 2

Background and Theory

This chapter develops the conceptual and methodological foundations on which the rest of the thesis builds, moving from cognitive science theories of conceptual representation, to a measurable prediction about neural activity and the graph-based measure that tests it, and finally to the difficulties of applying that measure to biological recordings. Section 2.1 introduces the geometric view of concepts and Gärdenfors' proposal that natural categories form convex regions of a representational space. Section 2.2 connects this abstract space to neural activity through the representational geometry framework, which treats population firing rates as points in a high-dimensional space. Section 2.3 adapts the notion of convexity to the curved, low-dimensional structure on which neural and artificial representations tend to lie, motivating the graph-based convexity measure introduced by Tětková *et al.* and used throughout this work. Section 2.4 reviews how Tětková *et al.* apply this measure to deep neural networks and what they find. Section 2.5 considers the difficulties that arise in transferring the method to biological systems, motivating the adaptations introduced in the following chapter.

2.1 Theories of Conceptual Representation

2.1.1 Geometric View of Concepts

The question of how the mind represents concepts has a long history in cognitive science and philosophy. Classical theories of concepts proposed that categories are defined by necessary and sufficient features: something is a bird if and only if it has wings, feathers, and lays eggs. This account faces well-known empirical problems. Most people judge a robin to be a better example of a bird than a penguin, even though both satisfy the definitional criteria equally. That is why it has largely given way to more graded, similarity-based accounts.

A particularly influential alternative is the geometric view: concepts are not symbolic rules but regions in a representational space. This idea has empirical roots in work by Shepard, who showed that human similarity judgments are often well-approximated by distances in a geometric space [She87]. If stimuli are represented as

points in a space and similarity is proximity, then categories are regions (containing the set of points that belong to a given class). Categorization becomes a problem of geometry: given a new point, which region does it fall into?

This geometric view makes no claim about what the representational space is made of, it is only defined at the level of relational structure. It specifies relations such as distance, regions and boundaries rather than the mechanism by which these relations are physically implemented. This abstraction is what allows the framework to support a specific kind of cross-system comparison: not whether systems (e.g. brains and artificial networks) work the same way, but whether they organize representational space similarly. Making this comparison concrete requires linking the abstract representational space to something measurable in each system. For artificial networks, this is relatively direct, since internal activations can serve directly as coordinates. For biological systems, the bridge is less direct and will be addressed in Section 2.2.

2.1.2 Conceptual Spaces and Convexity

The geometric theory most directly relevant here is Gärdenfors' theory of conceptual space [Gär00, Gär90]. A conceptual space is a geometric space whose dimensions are *quality dimensions*. These are psychologically meaningful properties of the domain. In the domain of color, for example, relevant quality dimensions are hue, saturation, and brightness. Each stimulus is represented as a point whose coordinates specify its values along these dimensions, and similarity is represented as spatial proximity: stimuli close in the space are taken to be psychologically similar.

A central element of this framework is the proposal that natural concepts correspond to convex regions of conceptual space [Gär90, Gär00]. A region C in a Euclidean space is convex if, for any two points $x, y \in C$, the entire line segment connecting them also lies within C:

$$\forall x, y \in C,\ \forall \lambda \in [0,1] :\ \lambda x + (1-\lambda) y \in C.$$

Intuitively, a convex region has no internal gaps or concavities. It cannot exclude a point that lies between two points already contained within the region. A solid disc is convex, whereas a crescent or annulus is not. (This Euclidean formulation will later be extended to other geometric structures, where the role of the line segment is replaced by shortest paths or geodesics.)

Gärdenfors argues that convexity follows from organizing categories by similarity [Gär90, Gär00]. Consider a category that excluded some intermediate case between two of its members. For example, a shade of color that lies perceptually between two shades both classified as "red", but is itself classified as non-red. Such a category would assign the same label to two points while excluding an intermediate point that is more similar to both of them than they are to each other. Such a category would be structurally fragmented. Convexity can therefore be understood as a geometric expression of categorical coherence, where categories correspond to clusters of similar items, rather than to disjunctive or fragmented regions of the space.

A convergent formal motivation for the same prediction of convex conceptual regions comes from prototype theory [Ros73]. Prototype theory proposes that categories are organized around a central, most representative example, the *prototype*.

A robin may function as a prototypical bird, whereas ostriches and penguins are more atypical but still belong to the category. If categorization proceeds by assigning each stimulus to the category of its nearest prototype in conceptual space, the resulting partition of that space forms Voronoi cells, which are convex in Euclidean space [BV04].

Convexity also provides a geometric basis for interpolation-based generalization. If category regions are convex, then stimuli lying between known category members receive the same classification. Convex organization therefore supports smooth generalization across similar representations.

Conceptual space theory thus makes a concrete geometric prediction about category representations. Members of the same category should occupy coherent, approximately convex regions of representational space. The remainder of this chapter develops how this prediction can be tested in neural population data.

2.2 Neural Population Representation

The theories reviewed so far are formulated in terms of an abstract representational space, a geometrical space whose structure encodes relations like semantic similarity. Since these spaces are theoretical constructs, they are not directly observable, and testing whether such geometric structure is reflected in neural systems requires linking this abstract space to measurable neural activity.

This link is provided by the representational geometry framework [KK13], which treats the activity patterns of a population of neurons as points in a high-dimensional space. Geometric relations among these points, in particular distances, are then interpreted as reflecting the representational relations the system has learned. Under this assumption, the predictions of conceptual space theory can be reformulated as claims about neural population geometry, in which convex category structure corresponds to a testable property of representational organization.

2.2.1 Neural Population Vectors

Modern recording technologies such as Neuropixel probes make it possible to simultaneously record the spiking activity of hundreds to thousands of neurons in an awake, behaving animal [JSS+17]. The firing rate of each neuron summarizes its current response as a single scalar, the number of spikes it produces per unit time. At any moment t, the firing rates of all N simultaneously recorded neurons can be collected into a single vector, the neural population vector:

$$\mathbf{r}(t) = (r_1(t),\ r_2(t),\ \ldots,\ r_N(t)) \in \mathbb{R}^N.$$

Rather than asking what a single neuron encodes, the population approach asks what is encoded by the pattern of activity across the entire recorded ensemble [EH21]. Each distinct pattern, that is each point in $\mathbf{R}^N$, represents a different collective state of the population, and the geometric structure of those points is what the representational geometry framework analyzes. The predictions of conceptual space theory can therefore be restated in these terms: if category regions are convex in

the representational space, they should appear as convex regions in the geometry of neural population activity.

2.2.2 Neural Manifold Hypothesis

Although population vectors formally inhabit an N-dimensional space, neural activity is far from arbitrary. Because neural activity is constrained by correlations between neurons, not all combinations of firing rates are observed in practice [CK11]. Population vectors recorded during a given task tend to cluster near a much lower-dimensional structure embedded within the full recording space. This structure is called the *neural manifold*, and the hypothesis that neural activity is constrained to such a low-dimensional manifold is now well supported empirically [CY14, SPS+19].

To build intuition, consider the surface of the Earth. A traveler walking in a straight direction across the globe follows a curved trajectory when viewed externally from space. Shortest paths along such a surface are called geodesics, and they are not generally straight lines in the surrounding embedding space.

The same idea applies to neural data. Population vectors occupy a low-dimensional, curved structure embedded within $\mathbb{R}^N$, and the geometric relationships between them that are relevant to representation are relationships measured along that structure, not through the ambient space.

This has a direct consequence for how convexity should be measured. Convexity is a geometric property, and the answer it returns depends on which geometry is used. A category region that appears non-convex when assessed using straight lines through R^N may be perfectly convex when paths are required to follow the manifold instead. Section 2.3 develops this point and introduces a notion of convexity suited to manifold-structured data.

2.3 Graph-based Convexity

The appropriate generalization of convexity to a manifold replaces straight-line segments with geodesics, the shortest paths between points as measured along the manifold itself. In a flat Euclidean space, geodesics are straight lines; on a sphere, they are arcs of great circles; on a neural manifold, geodesics follow the curved surface of the underlying low-dimensional structure.

A region C is *geodesically convex* on a manifold $\mathcal{M}$ if, for any two points $x, y \in C$, there exists a geodesic path from x to y along $\mathcal{M}$ which lies entirely within C [TBD+25]. Geodesic convexity is the geometrically appropriate notion of convexity for manifold-structured data because it respects the intrinsic geometry of the representations rather than the geometry of the ambient embedding space.

However, geodesic convexity cannot be applied directly to neural data. The underlying manifold is not known explicitly and must instead be inferred from a finite sample of observed population states. As a result, exact geodesics cannot be computed. A discrete approximation is therefore required. The approach for sampled data used by Tětková *et al.* is to construct a graph over the observed data points, whose shortest paths serve as approximation to geodesics on the underlying manifold.

2.3.1 K-Nearest Neighbor Graph

Given a set of M population state vectors $\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_M$, a K-nearest neighbor graph (KNN) $G = (V, E)$ is constructed by connecting each data point $\mathbf{r}_i$ to its K nearest neighbors. Each $\mathbf{r}_i$ corresponds to a node in V, labeled with its respective class labels. Undirected edges are added between each node and its K nearest neighbors in Euclidean distance, weighted by that distance.

The KNN graph approximates the manifold's geometry because manifolds are locally Euclidean. In a sufficiently small neighborhood around any point, the manifold looks flat, and Euclidean distances approximate geodesic distances. Short edges in the KNN graph therefore correspond to short geodesic distances on the manifold, and longer geodesics are approximated by chains of such local connections, that is, shortest paths in the graph [BdSLT00]. As M increases and K is chosen appropriately, the graph approximation converges to the true manifold geometry.

The choice of K involves a tradeoff between connectivity and locality. A small K produces a sparse graph that may underconnect the manifold, potentially leading to disconnected components and poorly estimated geodesics. A large K produces a denser graph that better connects the manifold but may introduce edges between points that are not genuinely close on the manifold, creating shortcuts through the ambient space. In practice, K is treated as a tuneable parameter and evaluated through robustness analysis.

2.3.2 Graph Convexity

With the KNN graph defined, graph convexity replaces geodesics with shortest paths in the graph. A class $A \subseteq V$ is graph-convex if, for any two nodes $u, v \in A$, all nodes along every/a shortest path $\pi(u, v)$ in G also belong to A:

$$\forall u, v \in A,\ \forall w \in \pi(u, v) :\ w \in A.$$

By checking whether the shortest path between two same-class nodes passes only through same-class nodes, graph convexity provides a discrete approximation of geodesic convexity [TBD+25].

Perfect graph convexity is unlikely in empirical neural data. Neural recordings are noisy, sampling is finite, and representations of different classes may partially overlap. The condition that shortest paths between same-class nodes stay entirely within that class, is therefore rarely satisfied. In practice, convexity can instead be treated as a continuous property measuring the extent to which shortest paths between same-class samples remain within the class region.

For a class A and a set P of sampled same-class pairs, the convexity score is

$$\mathrm{Conv}(A) = \frac{1}{|P|} \sum_{(u,v)\in P} \frac{|\pi^*(u, v) \cap A|}{|\pi^*(u, v)|},$$

where $\pi^*(u, v) = \pi(u, v) \setminus \{u, v\}$ denotes the shortest path between u and v excluding the endpoints, $|\pi^*(u, v)|$ is the number of nodes on that path, and $|\pi^*(u, v) \cap A|$ is the number of those nodes belonging to class A. The score is the mean fraction of same-class nodes across sampled shortest paths, and takes values in $[0, 1]$, with 1 indicating

perfect graph convexity. Excluding the endpoints from $\pi(u, v)$ is necessary because, by construction, both endpoints belong to A. Including them would therefore inflate the score, particularly for short paths, where the endpoints make up a large fraction of $|\pi(u, v)|$.

Graph convexity has a second advantage beyond respecting manifold geometry. It is computed entirely from observed population states, without requiring access to unobserved or interpolated points. This matters because, unlike in deep networks, where arbitrary intermediate representations can be constructed and evaluated, biological neural recordings provide only the population states actually evoked by presented stimuli. Section 2.5.1 returns to this point in detail.

This graph-convexity measure was developed and validated by Tětková *et al.* in the context of deep neural networks [TBD+25], and is introduced in that setting in the following section.

2.4 Convexity in Deep Neural Networks

A deep neural network (DNN) transforms its input through a sequence of computational layers. At each layer, the activity of all units can be collected into an activation vector, formally analogous to the neural population vectors of Eq. (2.2.1) This vector defines a point in a high-dimensional representational space whose dimensions correspond to the units of that layer. Each layer therefore induces its own representational space, and the network as a whole can be understood as a sequence of such spaces, each progressively transforming the geometry inherited from the previous layer.

Like neural population activity, representations in DNNs are often concentrated near lower-dimensional manifold structure embedded within the ambient activation space [HS16, AHH18]. Consequently, geometric relationships among representations are not necessarily well captured by Euclidean structure alone, motivating the graph-based notion of convexity introduced in Section 2.3 [TBD+25].

A trained classifier induces a partition of each representational space according to predicted class label. The set of activation vectors assigned to a given class forms that class's decision region within that layer's representational space. Because internal activations are directly observable, geometric properties of these decision regions, including convexity, can be measured explicitly within the representational space induced by each layer. Deep networks are a particularly tractable setting for this. Unlike biological neural systems, their representations are fully observable at every stage, and arbitrary activation vectors, including interpolated ones, can be constructed and propagated through the network to obtain class assignments. Moreover, representations are organized into a discrete sequence of layers that is globally aligned across inputs, permitting systematic analysis of how representational geometry evolves with processing depth. Section 2.5 returns to this contrast and discusses why these properties do not hold for biological neural systems.

Representations in deep networks change from layer to layer. Early layers tend to encode relatively local or low-level features, whereas later layers encode increasingly abstract and task-relevant structure [ZF14]. As depth increases, representations of same-class inputs often become more clustered and increasingly separated from

representations of other classes. This progressive reorganization motivates examining how the convexity of decision regions evolves across layers. If category representations become increasingly structured during processing, this may be reflected in increasing convexity of the corresponding decision regions.

The following section describes how Tětková *et al.* [TBD+25] operationalize and measure the evolution of convex decision regions across network layers.

2.4.1 Pipeline of Tětková et al.

Tětková *et al.* [TBD+25] operationalize representational convexity by measuring the geometric organization of category representations at successive layers of a trained DNN. For each layer, the activation vectors produced by all inputs in the dataset are collected, forming a point cloud in the layer's representational space. Each point is labeled with the corresponding class by the network.

A KNN graph is then constructed over this point cloud using Euclidean distance as described in Section 2.3.1. The convexity score for each class is then computed (Eq. (2.3.2)) by sampling same-class pairs, finding shortest paths in the graph, and recording the fraction of intermediate nodes that belong to the same class. The layer-level convexity score is the mean of these fractions across all sampled pairs, ranging from 0 to 1, with higher values indicating that shortest paths remain largely within the class region. Repeating this procedure for every layer produces a convexity profile of the network, tracking the evolution of representational geometry across processing depth.

To complement the graph-based analysis, Tětková *et al.* also measure Euclidean convexity through interpolation. For randomly sampled same-class pairs, a set of equidistant points is placed along the straight-line segment connecting their activation vectors. These interpolated activation states are then propagated through the remaining layers of the network, and the resulting class assignments are recorded. The Euclidean convexity score is defined as the proportion of interpolated points that receive the same class label as the original pair.

This analysis relies on several properties specific to deep neural networks. First, internal activation vectors are directly and completely observable at every layer. Second, arbitrary activation vectors can be constructed and propagated through the remainder of the network to obtain class assignments. Third, all inputs pass through the same discrete sequence of processing stages, permitting systematic comparison of how representational geometry evolves with processing depth. These assumptions do not generally hold for biological neural systems and therefore pose significant challenges for extending the method to neural population data (see Section 2.5.3 for further discussion).

2.4.2 Finding and Interpretation

Applying this pipeline across artificial deep neural networks trained on multiple domains, Tětková *et al.* [TBD+25] report a consistent pattern. Both graph-based and Euclidean convexity increase systematically across processing depth. Convexity scores begin near the chance level expected under random labeling, approximately $\frac{1}{C}$ for C classes, and rise substantially towards later layers. This pattern is observed

across all tested modalities and architectures, suggesting that increasing convexity of category decision regions is a general property of trained DNNs rather than a domain-specific phenomenon.

These findings are consistent with the view that representations become progressively more task-relevant across layers. Early layers primarily encode local or low-level features that are shared across categories, whereas later layers increasingly separate category-specific structure. As representations associated with the same class become more coherent and distinct from those of other classes, decision regions correspondingly become more convex.

The functional relevance of this geometric organization is supported by a further result. Convexity measured in pretrained representations is positively associated with downstream few-shot learning performance after fine-tuning. Networks with more convex category structure tend to achieve higher accuracy when adapting to novel tasks from limited training examples. This relationship is consistent with the theoretical arguments from Section 2.1.2, that convex representations facilitate interpolation-based generalization.

2.5 Challenges in Applying the Pipeline to Neural Data

The graph-convexity framework of Tětková *et al.* was developed and validated in deep neural networks, where representational states are fully observable and can be manipulated directly. Biological neural recordings differ from this setting in several important respects. Internal representations are only partially observed, computation unfolds through ongoing recurrent dynamics rather than discrete processing stages, and neural activity reflects many processes beyond the task under investigation. Transferring the framework from artificial to biological systems requires a number of methodological assumptions and approximations, each of which is discussed in turn below.

2.5.1 No Arbitrary Representation Querying

In a DNN, activation states can be constructed and evaluated for arbitrary inputs, including points that interpolate between observed examples. This makes it possible to evaluate Euclidean convexity directly by passing interpolated activation vectors through the remaining layers and examining their class assignments, thereby probing the geometry of decision regions beyond the specific observations present in a dataset. No equivalent procedure exists for biological neural recordings. Only the population vectors actually evoked by presented stimuli are available, and arbitrary intermediate population states cannot be evaluated directly. Euclidean convexity therefore cannot be measured directly. The present work relies exclusively on graph-based convexity, which can be estimated from observed population states alone.

In principle, a classifier trained on neural data could approximate arbitrary querying, but given the limited number of trials typically available in decision-related recording sessions, this would introduce its own challenges and limitations.

2.5.2 Partial Observability

The mouse brain, from which most neural recordings of this kind are obtained, contains on the order of ∼70 million neurons[HHML06]. Neuropixel recordings of the kind analyzed here simultaneously record from hundreds to a few thousand units per session [JSS+17]. The recorded population is therefore a small sample of the underlying neural state space; a projection onto a subspace defined by whichever neurons happen to fall within electrode range.

Estimated convexity is not generally invariant under projection. A category representation that seems convex in the full population does not need to appear convex in any given recorded subpopulation, and the reverse also holds. Observed convexity estimates should therefore be interpreted as properties of the geometry visible in the recorded subpopulation rather than claims about the complete neural representation.

Nevertheless, consistent convexity patterns observed across sessions, animals, time points, or brain regions are unlikely to arise solely as artifacts of neuronal subsampling, and may therefore still provide meaningful insight into the organization of neural representations. Subsampling by random projection tends to disrupt rather than induce convex structure A.5, making it unlikely that high graph-convexity estimates would emerge consistently across analyses and regions from data that are not genuinely convex in the sampled space. This consideration motivates caution in interpreting the absolute magnitude of convexity estimates, and places greater emphasis on qualitative patterns. Whether convexity emerges, when it emerges, when it peaks, and when it decays.

2.5.3 Time as Substitute for Layer Depth

The original pipeline measures convexity at successive layers of a trained network, tracking how representational geometry evolves across the processing hierarchy. This is possible because feedforward networks provide a globally defined, discrete sequence of computation stages through which every input passes in the same order, as established in Section 2.4. Biological neural systems provide no comparable structure. Computation unfolds continuously in time, is distributed across multiple interacting regions, and involves extensive recurrent feedback [VGSS20]. There is therefore no single, globally shared axis of processing depth along which representations can be compared across inputs.

To obtain comparable discrete population states across trials, continuous spike trains are evaluated around discrete time points to yield scalar estimates of firing rate, constructing a sequence of neural population vectors (2.2.1). Within the adapted pipeline, time therefore serves as an operational substitute for layer depth. Successive time points provide an axis along which representations can be compared and the emergence of convex category structure tracked.

However, time points across trials do not define a globally consistent processing stage. The same behavioral outcome may be reached through partially different neural trajectories across trials [VGSS20, FHI+25], so a given time point does not need to correspond to the same internal state across repetitions. The use of time as a substitute for layer depth is therefore an approximation, and convexity profiles

across time should be interpreted with this in mind.

2.5.4 Task-Irrelevant Variability in Population Activity

In the feedforward network considered by Tětková *et al.* [TBD+25], internal representations are deterministic functions of the current input and the learned model parameters. Repeated presentation of the same input produces the same activation pattern, and the representational space exclusively encodes entirely task-relevant structure.

Biological neural populations differ fundamentally in this respect. Neural activity is continuous, and the state of the population at any given moment reflects contributions from multiple concurrent processes, including sensory processing, motor preparation, arousal, attention, prior experience, and spontaneous fluctuations [VGSS20]. Population vectors recorded during a task therefore do not represent the task in isolation and reflect task-relevant processing as only one component among many. Repeated presentations of the same stimulus therefore do not generally produce identical population states.

This has direct consequences for the interpretation of convexity. Convexity estimates derived from neural recordings characterize the geometry of the observed population state space, not that of a hypothetical isolated task representation. In behavioral paradigms where decisions are expressed through actions, decision formation and movement preparation become intertwined, and emerging convexity may reflect motor planning, category representation, or both. Convexity should therefore be interpreted as a property of the neural population states available during behavior, not as a pure measure of categorical organization.

2.5.5 Regional Organization

Unlike artificial neural networks, which are typically analyzed as unified systems, biological recordings are obtained from specific brain regions that contribute differentially to cognition. Different regions may encode distinct task-relevant or task-irrelevant variables [SZHCH19, IAB+25]. There is therefore no reason to expect convexity to emerge uniformly throughout the brain.

In practice, however, the available data limit which regions can be meaningfully analyzed. The number of sessions per region is small, and sources of variability such as neuronal subsampling vary across sessions and regions in ways that are difficult to control for. Regional estimates are therefore strongly shaped by the particular properties of their individual recordings. The focus of regional analysis in this study is accordingly on intra-regional temporal dynamics rather than cross-regional comparison of absolute values.

Taken together, these considerations shape how the graph-convexity framework of Tětková *et al.* [TBD+25] can be applied to biological neural recordings, though they do so in different ways. The absence of arbitrary representation querying is not a problem that requires methodological adaptation, it is precisely why the present work relies exclusively on graph-based convexity, which operates only on observed population states and never requires interpolated ones. Partial observability similarly does not call for a methodological fix, but it constrains interpretation.

Convexity estimates should be read as properties of the geometry visible in the recorded subpopulation, not as claims about the complete neural representation. The most substantive adaptation concerns the absence of a shared processing hierarchy. Because neural computation unfolds continuously in time rather than through a discrete sequence of stages, time serves as a functional substitute for layer depth. The following chapter develops a methodology that operationalizes this substitution and applies the graph-convexity framework to biological neural recordings within these constraints.

Chapter 3

Methods

This chapter develops the methodology through which the graph-convexity framework from Tetkova *et al.* [TBD+25] is adapted and applied to biological neural recordings. Because the original framework was formulated for artificial neural networks, several design choices are required to translate it to electrophysiological data, including the representation of population activity and the handling of variable-duration trials across sessions with different recorded neurons.

The chapter is organized as follows. Section 3.1 describes the IBL Brainwide Map dataset and the behavioral task from which the recordings are drawn. Section 3.2 presents the main analysis pipeline, which proceeds in six stages: trial filtering, temporal alignment, firing-rate estimation, KNN graph construction, convexity approximation, and aggregation across sessions. Section 3.3 describes two further analyses, which extend or alter the main pipeline: A region-specific variant of the pipeline examining convexity dynamics relative to movement onset, and a behavioral correlation analysis linking convexity to task performance. Section 3.4 introduces the software implementation.

3.1 Dataset

All analyses are based on electrophysiological recordings and behavioral data from the International Brain Laboratory (IBL) Brainwide Map Project [Int23], described in detail by [IAB+25]. The dataset was collected across multiple laboratories using a standardized experimental protocol designed to measure brainwide neural activity during a controlled perceptual decision-making task. Recordings were obtained using up to two simultaneous Neuropixels probes [JSS+17] per session, allowing simultaneous recording of spiking activity from hundreds of neurons across multiple brain regions

The behavioral task requires head-fixed mice to detect the location of a visual grating stimulus and report their decision by rotating a wheel with their forepaws. On each trial, after a quiescence period during which no wheel movement is permitted, a stimulus of variable contrast appears on either the left or right side of a screen. The subject must then rotate the wheel and thereby move the stimulus towards

the center. Correct responses are rewarded with sugar water, whereas incorrect responses result in a timeout period. The task is therefore binary. Each trial is associated with either a leftward or rightward choice.

Recordings in this dataset were spike-sorted using a custom version of Kilosort [PSK+16] developed by the IBL [IAB+25], yielding spike trains for each unit. Resulting units were assigned quality labels based on a set of quality metrics defined by the IBL pipeline [IAB+25], where a quality label of 1.0 indicates well-isolated single neurons. Only sessions with a correct choice rate above 75% are included in the IBL release [IAB+25].

Recording sessions vary in the number of recorded neurons, the identity of those neurons, and covered brain regions, because the precise set of sampled neurons depends on probe placement, which is not identical across animals and sessions.

3.2 Main Analysis Pipeline

Because neural population vectors are defined over different sets of recorded neurons in different sessions and subjects, direct geometric comparison of population states across sessions is not meaningful. All geometric analyses are performed independently within each session, and convexity measures are subsequently aggregated across sessions.

The analysis pipeline proceeds in six stages, the first five of which operate independently within each recording session. First, trials are filtered by decision time to reduce temporal variability across trials. Second, the decision interval of each retained trial is mapped onto a shared set of normalized evaluation points, enabling comparison across trials of different durations. Third, population firing-rate vectors are estimated at each evaluation point. Fourth, a KNN graph is constructed over the resulting population vectors at each evaluation index. Fifth, a convexity score is computed for each choice class and averaged to yield a single session-level convexity profile. Last, session-level convexity profiles are aggregated at each evaluation point to obtain a single mean convexity profile as result.

3.2.1 Session Selection

Sessions were selected from the full IBL Brainwide Map dataset according to following criteria. Sessions were required to contain at least 100 neurons with a quality label of 1.0 as assigned by the IBL quality metrics pipeline (from here on named good-quality neurons). From the remaining 297 sessions, we randomly selected 100 for analysis.

3.2.2 Preprocessing

To improve comparability of neural population states across trials, trials are filtered according to decision time prior to all analyses. We defined the start of the decision interval as stimulus onset, the time at which the visual stimulus appeared on screen, and the end of the interval as movement onset, the time of the first recorded movement. Movement onset is therefore treated as the time a decision is expressed.

For each trial k, let $T_{k,start}$ be the time of stimulus onset and $T_{k,movement}$ the time of movement onset. The decision time D_k is then defined as elapsed time between the stimulus appearance and the first recorded wheel movement:

$$D_k = T_{k,\text{movement}} - T_{k,\text{start}}.$$

Trials were retained if their decision time fell within the range $[50, 400]$ ms. The rationale is that trials with similar decision times are more likely to exhibit comparable temporal structure in their underlying neural dynamics, making population vectors evaluated at the same normalized time point more comparable across trials. This filtering step partially addresses the challenge of comparing population vectors across trials of variable duration and differing neural trajectories Section 2.5.3)

After filtering, a median of 61.4% of trials per session were retained (IQR [52.6, 73.1]%; range [36.4, 86.3]%). The full distribution of decision times, with the filtering bounds indicated, is shown in Appendix A.2, and the sensitivity of results to the choice of bounds is evaluated in Appendix A.3.2.

Trials in which no wheel movement was recorded (for which no decision time is defined) were excluded from all analyses. All analyses were restricted to good-quality neurons as defined in Section 3.2.1.

3.2.3 Temporal Alignment

For each trial k, let D_k denote the decision time and $T_{k,start}$ the starting time as defined in Section 3.2.2. To enable temporal alignment across trials of different durations, a set of 30 evenly spaced normalized evaluation points

$$\alpha_j = \frac{j-1}{29}, j = 1, \ldots, 30$$

is defined on [0, 1] and shared across all trials and session. The corresponding absolute evaluation time for trial k at evaluation index j is

$$t_{k,j} = T_{k,\text{start}} + \alpha_j D_k$$

Population activity evaluated for a given α_j therefore corresponds to the same relative position within the decision interval regardless of the absolute duration of the trial. These normalized evaluation points serve as a functional analogue of network layers, addressing the absence of discrete processing stages in biological neural recordings (Section 2.5.3).

3.2.4 Firing Rate Estimation

Neural activity is represented as population firing-rate vectors evaluated at the decision-time normalized evaluation time points defined in Section 3.3. For each absolute evaluation time point $t_{k,j}$ and neuron i, the firing rate $r_i(t_{k,j})$ is estimated by counting spikes (of neuron i) within a 50 ms window centered on $t_{k,j}$ and dividing by the window duration. Collecting the firing rates of all simultaneously recorded neurons yields the population vector

$$\mathbf{r}_k(j) = (r_1(t_{k,j}), r_2(t_{k,j}), \ldots, r_N(t_{k,j})) \in \mathbb{R}^N,$$

where N denotes the number of recorded neurons in the corresponding session. The dimensionality N varies across sessions because the number of recorded good units is session dependent.

For a fixed evaluation point j, the collection

$$\{\mathbf{r}_1(j), \mathbf{r}_2(j), \ldots, \mathbf{r}_M(j)\}$$

defines the set of neural population states observed across all M trials of the session. These vectors form the basis for all subsequent graph-based analyses.

Firing rates are used without further normalization. Robustness with respect to window width and the choice of firing-rate estimator is evaluated in Appendix A.3.1, including an alternative Gaussian-smoothed estimate.

3.2.5 KNN Graph Construction

For each recording session s and evaluation index j, a weighted undirected KNN graph $G(j) = (V, E)$ is constructed (Section 2.3.1) over the corresponding set of population activity vectors $\{\mathbf{r}_1(j), \mathbf{r}_2(j), \ldots, \mathbf{r}_M(j)\}$. Each node $v_i \in V$ corresponds to a population vector $\mathbf{r}_i(j) \in \mathbb{R}^N$ and is labeled with the binary choice of that trial. Each node is connected to its K nearest neighbors under Euclidean distance in $\mathbb{R}^N$, with edge weights given by the corresponding distances. Following Tětková *et al.* [TBD+25], we set $K = 10$, and the robustness of all results with respect to the choice of K is evaluated in Appendix A.3.3. No dimensionality reduction is applied prior to graph construction, preserving the full geometry of the recorded population activity.

3.2.6 Convexity Approximation

Given a graph $G(j) = (V, E)$, convexity is evaluated separately for each behavioral choice class by examining whether shortest paths between same-class nodes remain within that class.

Let $A \subseteq V$ denote the set of nodes belonging to one choice class. A set P of 500 same-class node pairs is sampled uniformly without replacement. For each pair $(u, v) \in P$, a shortest path $\pi(u, v)$ is computed using Dijkstra's algorithm [Dij59].

To not inflate convexity rates, only intermediate nodes are considered. Let $\pi^*(u, v)$ denote the set of intermediate nodes on the shortest path between u and v:

$$\pi^*(u, v) = \pi(u, v) \setminus \{u, v\}.$$

Convexity for an individual node pair is quantified as the proportion of nodes along the path that belong to the same class:

$$C(u, v) = \begin{cases} 1 & \text{if } u,\ v \text{ directly connected} \\ 0 & \text{if no path exists between } u,\ v\ . \\ \frac{|\pi^*(u,v) \cap A|}{|\pi^*(u,v)|} & \text{otherwise} \end{cases}$$

If a direct connection between u and v exists, a score of 1 is assigned; if no path exists, a score of 0.

The convexity score for class A is then defined as the average over all sampled pairs:

$$\mathrm{Conv}_A(j) = \frac{1}{|P|} \sum_{(u,v)\in P} C(u,v).$$

This measure lies in the interval [0,1], where a value of 1 indicates that shortest paths between same-class nodes remain entirely within the class. Convexity is computed independently for both choice classes and averaged to obtain a single session-level convexity estimate

$$\mathrm{Conv}(j) = \frac{\mathrm{Conv}_A(j) + \mathrm{Conv}_B(j)}{2}.$$

3.2.7 Aggregation across Sessions

For each session s, the procedure described above yields a convexity profile

$$\{\mathrm{Conv}^{(s)}(1), \ldots, \mathrm{Conv}^{(s)}(30)\},$$

where $\mathrm{Conv}^{(s)}(j)$ denotes the convexity estimate at evaluation index j. To obtain aggregated/group-level estimates, we averaged convexity values uniformly across all S included sessions at each evaluation index j:

$$\mathrm{Score}(j) = \frac{1}{S} \sum_{s=1}^{S} \mathrm{Conv}^{(s)}(j).$$

The resulting profile across evaluation indices constitutes the primary outcome of the brainwide analysis.

3.3 Further Analyses

The analysis pipeline described above constitutes the primary brainwide analysis. For the following supplementary analyses, specific components of the pipeline were modified, while all remaining processing steps were performed identically.

3.3.1 Regional Analysis

To investigate to what extent different brain regions exhibit distinct convexity dynamics surrounding the moment a decision is formed and expressed, we performed a region-specific variant of the analysis pipeline, in which the temporal alignment and composition of the neural population vectors were modified. Evaluation points were defined relative to movement onset rather than the normalized decision interval, spanning a symmetric window before and after this event.

Brain regions were defined according to level 5 of the Allen Common Coordinate Framework [WDL+20], chosen as a compromise between anatomical granularity and recording coverage across sessions. A region was included in the analysis if it was covered by at least 25 sessions containing at least 40 good-quality neurons (Section 3.2.1) from that region. For each included region, the 25 sessions with the highest

number of good-quality neurons from that region were selected. The full pipeline was then applied independently to each region and its corresponding sessions.

For these sessions, we distributed evaluation time points uniformly between 150 ms before and 150 ms after movement onset for each trial. The offset of evaluation index $j = 1, \ldots, 30$ relative to movement onset is

$$\delta_j = -150 + \frac{300}{29}(j-1) \quad \text{ms},$$

and the corresponding absolute evaluation time for trial k is

$$t_{k,j} = T_{k,\text{movement}} + \delta_j,$$

where $T_{k,\text{movement}}$ denotes the time of movement onset in trial k (Section 3.1). Unlike the decision-time-normalized alignment in Section 3.3, the temporal offsets δ_j are fixed across trials and do not depend on k. All trials share the same absolute time window around movement onset, so population states are compared relative to a common reference point.

For a given region and session, we defined the set of good-quality neurons assigned to that region as a subset $\mathcal{M} \subseteq \{1, \ldots, N\}$, where N is the number of recorded good-quality neurons in the session. We then restricted all neural population vectors to the neurons in $\mathcal{M}$. Let $M = |\mathcal{M}|$ denote the number of region-specific neurons in that session, with $M \leq N$. The resulting region-restricted population vector for trial k at evaluation index j is

$$\mathbf{r}_k(j) \in \mathbb{R}^M,$$

constructed analogously to Section 3.4 using only the firing rates of these M neurons. The remaining pipeline, with KNN graph construction (Section 3.5) and convexity approximation (Section 3.6), was applied identically using these region-restricted population vectors, and session-level convexity profiles were aggregated (Eq. (3.2.7)) across the 25 sessions selected for that region.

This gives a movement-aligned convexity profile $\{\text{Conv}(1), \ldots, \text{Conv}(30)\}$ for each analyzed region, allowing direct comparison of convexity dynamics across brain regions around the time of movement initiation. Corresponding analyses at atlas levels 4 and 6 are reported in Appendix A.6.

3.3.2 Behavioral Correlation

To assess whether convex decision regions in neural population states are related to task performance, session-level convexity estimates obtained from the brainwide analysis were compared with behavioral accuracy.

For each session s, the convexity profile as computed in the main brainwide analysis (Section 3.2)

$$\{\text{Conv}^s(1), \ldots, \text{Conv}^s(30)\}$$

was reduced to a single scalar by averaging the final ten evaluation points, corresponding to the last third of the normalized decision interval:

$$\hat{C}^{(s)} = \frac{1}{10} \sum_{j=21}^{30} \text{Conv}^{(s)}(j).$$

This interval was chosen because decision-related representations are expected to be most fully developed near the end of the decision period. Averaging across multiple evaluation points additionally reduces variability arising from individual time points.

We quantified behavioral performance as decision accuracy. Unlike the convexity analysis, which uses the decision-time–filtered trials (Section 3.2.2), accuracy was computed over all trials in a session, since it is a behavioral property of the whole session and filtering by decision time would bias the estimate toward specific trials. For each session s, decision accuracy was defined as the fraction of all trials resulting in a correct choice:

$$\hat{A}^{(s)} = \frac{\text{number of correct trials in } s}{\text{total number of trials in } s}.$$

The linear association between late-interval convexity and decision accuracy was quantified using the Pearson correlation coefficient

$$r = \frac{1}{S-1} \sum_{s=1}^{S} \left(\frac{\hat{C}^{(s)} - \bar{C}}{\sigma_C} \right) \left(\frac{\hat{A}^{(s)} - \bar{A}}{\sigma_A} \right).$$

where $\bar{C}, \sigma_C$ and $\bar{A}, \sigma_A$ denote the sample mean and standard deviation of the convexity scores and accuracy values, respectively, computed across all S sessions.

Spearman's rank correlation coefficient was additionally computed as a nonparametric robustness check, relaxing the assumption of a linear relationship between the two variables.

3.4 Software Implementation

The analyses presented in this thesis were conducted using cvxnr, an open-source Python package for convexity analysis of neural spiking data developed as part of this work. The package implements the complete pipeline described in this chapter: session selection and trial filtering, temporal alignment, firing-rate estimation, KNN graph construction, graph-convexity approximation, and aggregation. It also produced every result reported in the following chapters. It is written as a reusable library with a documented API: although access to the IBL Brainwide Map dataset is integrated directly, the package operates identically on any local spiking dataset provided in the supported format, so the method can be applied beyond the recordings analyzed here. The package architecture, dependencies, and usage are described in Appendix B, and the code is publicly available at https://github.com/nrgrp/cvxnr.git.

Chapter 4

Results

This chapter reports the results of the three analyses described in Chapter 3.

4.1 Brainwide Analysis

Neural population representation during the perceptual decision-making task exhibits clear convex structure with respect to the animal's choice (Figure 4.1).

At stimulus onset ($\alpha = 0$)), the mean convexity score is approximately 0.53, only slightly exceeding the baseline value obtained from shuffled labels (Appendix A.4). During the initial portion of the decision interval, convexity scores remain relatively stable and close to baseline.

As the trial progresses, convexity increases gradually, with the rate of increase becoming more pronounced during the second half of the decision interval. The mean convexity score reaches its maximum value of approximately 0.74 near movement onset ($\alpha = 1$), corresponding to an increase of approximately 0.21 relative to stimulus onset.

4.2 Regional Analysis

Across all analyzed regions, convexity of neural population representations followed a similar movement-aligned temporal profile (Figure 4.2), consisting of an initial baseline period, a pronounced increase beginning before movement onset, sustained elevated values after movement onset, and a gradual decline toward the end of the analyzed time window. However, substantial regional differences were observed in the magnitude and temporal evolution of these dynamics.

The earliest increase in convexity was observed in Isocortex, which began to rise well before movement onset and reached comparatively high values already during the pre-movement period. MY-mot, in contrast, remained close to baseline for longer but then exhibited the steepest rise of all regions, overtaking Isocortex shortly before movement onset and reaching the highest peak convexity overall (~ 0.84) shortly after movement onset. STRd displayed a distinct temporal profile characterized by a prolonged increase in convexity. Rather than plateauing shortly after movement

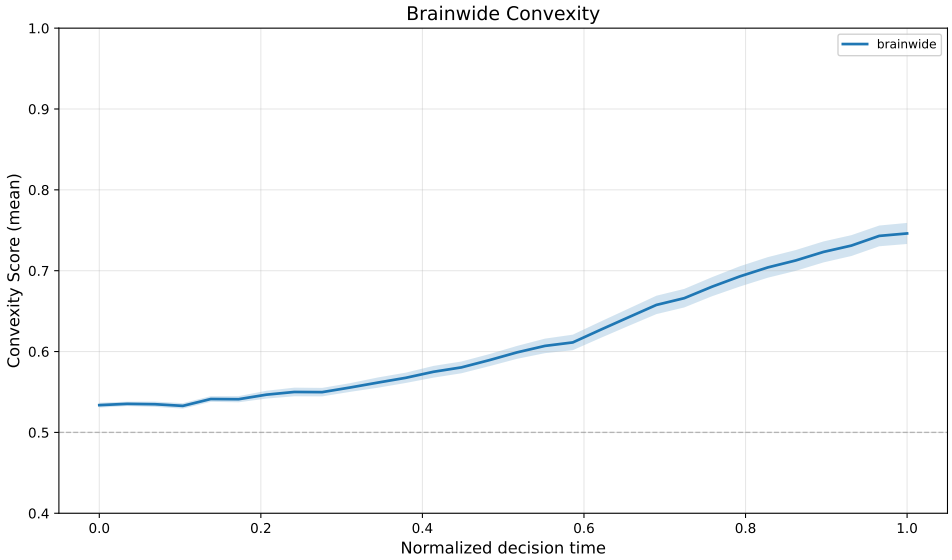

Figure 4.1 Time course of mean convexity scores across the normalized decision window from stimulus onset ($\alpha = 0$) to movement onset ($\alpha = 1$). Values represent the mean across 100 sessions, shaded areas indicate SEM.

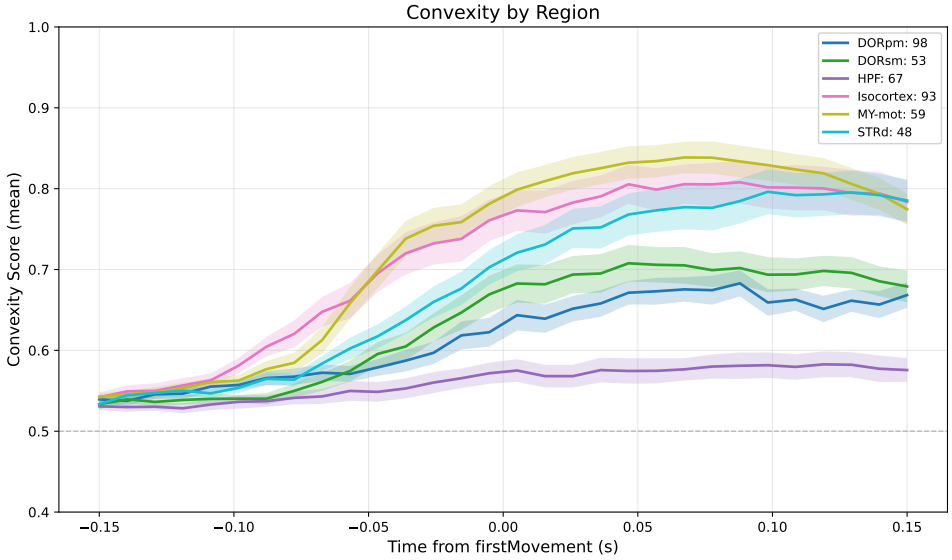

Figure 4.2 Mean convexity time course across sessions for different regions, computed over the 150 ms before and after the first recorded wheel movement. For each region, the mean is computed across 25 sessions in which the region had at least the indicated number of neurons (shown in the legend).

onset, its convexity continued rising throughout the early post-movement period and reached its maximum later in the trial, resulting in comparatively high values at the end of the analyzed time window.

DORsm and DORpm displayed intermediate dynamics, following the general movement-aligned pattern but with more moderate increases, plateauing at lower values (∼0.68–0.71) than MY-mot, Isocortex, and STRd, and without the early onset or prolonged post-movement rise seen in those regions.

HPF showed the weakest modulation overall, with convexity remaining close to baseline throughout and increasing only modestly (from ~ 0.53 to ~ 0.58) around movement onset, the smallest change of any region analyzed.

Overall, all analyzed regions exhibited a movement-aligned increase in convexity, but differed substantially in the timing, magnitude, and persistence of this increase. Isocortex showed the earliest emergence of elevated convexity, MY-mot exhibited the steepest rise and highest peak values, and STRd showed the most prolonged post-movement increase, while DORsm, DORpm, and HPF showed comparatively modest changes, with HPF remaining nearly at baseline throughout.

4.3 Behavioral Correlation

We found that sessions with higher convexity later in the decision interval tended to exhibit higher decision accuracy. Both variables were positively correlated, with a Pearson correlation coefficient of $r = 0.35$ ($p < 0.001$, 95% CI $[0.16, 0.51]$) across 100 sessions (4.3). A Spearman correlation ($r = 0.33$, $p < 0.001$) resulted in a comparable estimate, indicating that the relationship is not driven by outliers or by a small number of extreme sessions.

This relationship is moderate in size, accounting for roughly 12% of the variance in task accuracy across sessions, and considerable scatter remains around the regression line. Sessions with similar convexity scores can differ substantially in accuracy.

Together, these results demonstrate that convex structure in neural population representations emerges progressively over the course of the decision interval. Regional analyses further reveal that while this pattern is mostly consistent in its general shape, regions differ in the timing, magnitude, and persistence of their convexity dynamics. Finally, late-interval convexity is positively associated with behavioral accuracy across sessions, suggesting a link between the geometry of neural population representations and task performance.

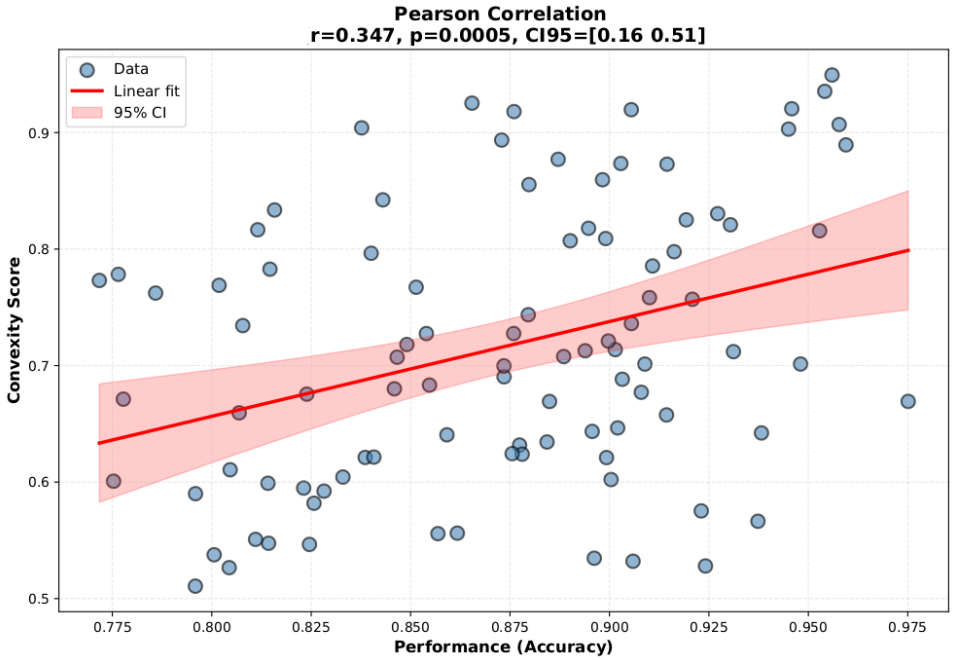

Figure 4.3 Session-level correlation between behavioral performance and late-window neural convexity. Scatter plot of task accuracy against mean convexity score during the late decision window ($\alpha > \frac{2}{3}$) across 100 sessions. Each point represents one session. The red line shows the linear regression fit with 95% confidence interval (shaded). Pearson correlation: $r = 0.347$, $p = 0.0005$; Spearman robustness check: $r = 0.333$, $p = 0.0009$.

Chapter 5

Discussion

5.1 Findings

5.1.1 Brainwide Analysis

The brainwide analysis shows that convexity of neural representations with respect to choice is near baseline at stimulus onset and rises toward movement onset. This rise is not linear. Convexity remains close to baseline through the early portion of the defined decision interval and accelerates during its second half.

This pattern mirrors the layerwise increase reported by Tětková *et al.* [TBD+25] in DNNs. This parallel should not be overstated. The correspondence is one of overall shape, not a literal mapping between time points and layers. As developed in Section 2.5.3, biological recordings offer no natural segmentation of processing stages, and neural trajectories leading to a decision can differ substantially, in part because the same decision in the IBL task can be reached through different strategies, such as stimulus-driven or prior-driven, depending largely on stimulus contrast [FHI+25]. Convexity arising only later into the trial is therefore expected, and aligns well with the findings of the paper.

Because choice is reported through wheel movement, the late acceleration of convexity coincides with the period in which movement is prepared and initiated. The same geometric separation could reflect an emerging decision representation, a developing motor plan, or an entangled decision–action state. The brainwide analysis alone cannot separate these, and whether they can be separated at all in a task of this kind may be an open question in the field rather than something a different analysis design could resolve.

The brainwide analysis pools over sessions whose recordings cover neural activity from partially distinctive regions, obscuring region-specific differences in convex structure.

5.1.2 Regional Analysis

The regional analysis shows that the movement-aligned convexity profile is broadly similar across regions in its general shape, with a baseline period, an increase

beginning before movement onset, elevated values after onset, and a gradual decline. The persistence of this general shape under the movement-aligned analysis used here, in addition to the normalized decision time analysis of the previous section, indicates that the overall convexity dynamics are robust to the choice of alignment. Regions differ, however, in the timing, magnitude, and persistence of this increase. Convexity rises earliest and is sustained longest in Isocortex; it rises latest but most steeply in MY-mot, which reaches the highest peak; STRd shows the most prolonged post-movement increase; and HPF shows the weakest modulation, remaining close to baseline throughout.

While a functional interpretation of these differences may be tempting, dynamics are also subject to the subsampling and session-selection problem, which is especially acute in regional analysis, where data is more limited. Since different regions draw on different sessions and animals, cross-regional differences may partly reflect differences between sessions and subjects, rather than between regions. This is particularly important given the small session counts per region.

The differences should therefore be treated with care.

In addition, the analyzed regions are still too coarse to align them with specific cognitive functions, and analysis of each region pools over sessions that may cover functionally distinct subregions, so the region-level convexity profiles are averages over potentially heterogeneous structure rather than a property of a functionally unitary area.

A last point to consider regarding the temporal profiles concerns the large bin sizes used in firing-rate estimation. These smooth the population activity, which can blur sharp temporal dynamics. Apparent differences in the timing and steepness of regional increases should therefore be read with this smoothing in mind.

Taken together, the regional analysis supports the conclusion that movement-aligned convexity of neural representation is a general feature across the regions examined, at least for the sampled populations. The differences in detailed dynamics are real and hint at genuine regional structure, but they cannot, on the present data, be cleanly separated from the artifacts introduced by subsampling and session selection. They are therefore best read as an indication that regional differences in convex structure are a promising target for future work rather than as established functional distinctions.

5.1.3 Behavioral Correlation

Sessions with more convex neural representations late in the decision interval tend to have higher decision accuracy. The correlation is positive and moderate (Pearson $r = 0.35$, Spearman $r = 0.33$), and the agreement between the two coefficients indicates it is not driven by outliers.

This is consistent with the theoretical expectation from Section 2.1.2 that convex organization supports smooth generalization, and should therefore be functionally beneficial. It is also consistent with Tětková *et al.*'s [TBD+25] findings, that convexity in pretrained representations predicts downstream few-shot performance. The parallel is partial, however, and worth stating precisely. Their result links convexity to later learning capacity, measuring performance after fine-tuning, whereas the present result links convexity to concurrent task accuracy within the same sessions.

Both tie convex structure to a functional benefit, but the relationship is predictive in their case and concurrent here.

The measured strength of the relationship should not be overstated. It accounts for roughly 12% of the variance in accuracy across sessions, and considerable scatter remains: sessions with similar convexity scores can differ substantially in accuracy. Some of this scatter is expected. The brainwide convexity scores pool over neurons that may come from distinct regions with differing convex structure, and the subsampling of neurons induces additional variance from session to session. Convexity therefore explains only part of the variation in session-level performance.

A further limitation constrains interpretation. The analysis establishes no causal direction. More convex geometry may support better decisions, better decision processes may produce more convex geometry, or both may follow from a common factor. The cross-session design is especially exposed to this last possibility, since sessions differ in ways like stimulus statistics and response bias, that could influence convexity and accuracy together.

5.2 Methodological Considerations

5.2.1 Bin sizes and temporal alignment

The firing-rate estimates used here rely on a larger bin size than the 20 ms common in much of the related literature, and larger bins yielded higher convexity scores in practice (Appendix A.3.1). Larger bins dampen the effect of small temporal misalignments at points where trials would otherwise be alignable. The cost is that this same smoothing can hide sharper temporal dynamics. The interpretable anchor should therefore be the late portion of the interval, where a decision representation is expected to have formed, resting on the assumption that this late state is comparatively independent of the trajectory leading to it. Still, no alignment strategy can fully synchronize heterogeneous trials. Both the decision-time-normalized alignment of the brainwide analysis and the flat event-aligned method of the regional analysis remain imperfect for trials of different length and different underlying neural trajectory. This imperfection is likely one reason why larger bins are useful, and it is better addressed at the source than through smoothing: future work that divides trials by internal state or isolates the basis of each decision should yield more comparable trajectories.

5.2.2 Partial observability and statistical power

The most fundamental limitation, established in Section 2.5.2, is that each recording is a small and somewhat arbitrary sample of the underlying neural state space, and that estimated convexity is not invariant under projection onto such a sample. Observed convexity therefore cannot be read as the convexity actually present in the full population: a small sample may fail to capture structure that is there, and in theory could also suggest structure that is not. But as shown empirically in Appendix A.5, random subsampling tends to disrupt rather than create convex structure. So the bias from partial observability runs against finding convexity, not

toward it. Consistently high convexity across sessions, animals, and regions can therefore be treated as a meaningful signal rather than an artifact. But the focus should remain on where and when convexity emerges rather than absolute values.

A further limitation is the size of the dataset. Convexity is averaged over sessions, but the sessions are few and not interchangeable. Each records a different animal and captures only a subsample of the neural population. The expected variance of scores between sessions therefore is high. When session count for an analysis is low, results are strongly shaped by the particular sessions used. The regional analysis is the most exposed, where session count per region is especially low.

5.3 Further Directions

Euclidean convexity could not be measured here, because biological recordings do not permit querying of arbitrary population states (Section 2.5.1). Training a classifier on the neural population would allow arbitrary points in activation space, including interpolated states, to be assigned a choice label, making it possible to apply the interpolation-based portion of the Tětková *et al.* [TBD+25] pipeline to neural data and to compare graph-based and Euclidean convexity directly, as was done for DNNs. The main difficulty is the limited number of trials in decision-related sessions, which limits how well such a classifier can be trained.

A second direction is to separate decision strategies. If trials were split by strategy, for example, more stimulus-driven versus more prior-driven [FHI+25] , the neural trajectories within each group should be better aligned, and it would be informative to see whether convexity then emerges earlier. An alternative version of this idea would be to model internal states explicitly, applying the pipeline separately to each inferred state. The state model could be fit with the strategy-inference approach of Mohammadi *et al.* [MAIP25], or with a general latent-factor framework such as the multi-convex programming approach of Zhu *et al.* [ZYHB25].

Finally, as recording technology scales toward larger and more complete population coverage, repeating these analyses on substantially larger samples would directly address the subsampling limitation above, and would allow absolute convexity magnitudes and genuine cross-regional comparisons to be interpreted with greater confidence. However, this should be qualified. More neurons are not always better. Meaningfully eliminating the partial-observability problem would require access to something close to the full population state space, which raises both experimental and computational costs. The appropriate target is adequate coverage of the relevant subpopulations rather than maximal neuron count for its own sake. With more data, the regional analysis could also be pushed to finer subregions, avoiding the pooling over functionally distinct areas, which also limits the present results.

Chapter 6

Conclusion

This thesis asked whether the increasing convexity of internal representations found in artificial neural networks also appears in biological neural representations, and set out to adapt the method developed for these networks so that it could be applied to biological recordings.

Within the limits of the data, the scientific answer is affirmative. Convexity with respect to choice is near baseline at stimulus onset and rises toward movement onset along a non-linear, accelerating course. The pattern is robust to the choice of trial alignment and main analysis parameters, appears across the regions examined, and is associated with higher decision accuracy across sessions.

This is the same direction of effect that Tětková *et al.*report across the layers of deep networks, now obtained in a system that shares none of their architecture or training, which suggests that increasingly convex representations are not specific to artificial classifiers, but may reflect something more general about how neural systems organize choice-related representations. The parallel is one of shape rather than a literal mapping of time onto layers, and in a task where choice is reported through movement the emerging geometry cannot be cleanly separated from the motor preparation it is entangled with. It may reflect the decision, the developing action, or an entangled state of both. These are limits on interpretation, rather than on the effect itself.

The methodological answer is the pipeline itself: an adaptation of graph-based convexity to neural population recordings that substitutes time for layer depth, summarizes population activity as discrete state vectors, and measures convexity over observed states alone, released as `cvxnr` for use on other datasets and research questions.

The strongest limitation is also the most encouraging note to end on. Because random subsampling tends to destroy convex structure rather than create it, the convexity measured from these small, partial recordings is most likely a conservative reflection of what is present in the full population. The signature is therefore visible despite the conditions working against it. Fuller coverage, better-aligned trials, and a direct Euclidean comparison stand to sharpen the picture rather than overturn it.

Appendix A

Dataset and Preprocessing

A.1 Session Selection and Inclusion Counts

This appendix documents how sessions and neurons were selected from the IBL Brainwide Map release for each analysis (Sections 3.1 and 3.2.1).

Brainwide analysis

Starting from the full Brainwide Map release of 459 sessions, sessions were required to contain at least 100 neurons with a quality label of 1.0 (good-quality neurons). 297 sessions met this criterion, of which 100 were selected at random for the brainwide analysis. The selection funnel is summarized in Table A.1.

Criterion	Sessions
Full Brainwide Map release	459
$\geq$ 100 good-quality neurons	297
Randomly selected for analysis	100

Table A.1 Session inclusion funnel for the brainwide analysis. Good-quality neurons are units with an IBL quality label of 1.0.

Across the 100 analyzed sessions, the number of good-quality neurons per session had a median of 184 (IQR $[140, 238]$; range $[105, 469]$). This count determines the dimensionality N of the population vectors in each session (Section 3.2.4).

Regional analysis

For the regional analysis, brain regions were defined at level 5 of the Allen Common Coordinate Framework [WDL+20]. A region was included if it was covered by at least 25 sessions containing at least 40 good-quality neurons from that region. For each included region, the 25 sessions with the highest such neuron counts were used (Section 3.3.1). The included regions and their per-region neuron counts are listed in Table A.2.

Region	Median	Range
	Good-quality neurons / session	
Isocortex	144	[93, 330]
MY-mot	97	[59, 170]
STRd	62	[48, 136]
DORsm	70	[53, 173]
DORpm	132	[98, 309]
HPF	105	[67, 258]

Table A.2 Regions included in the regional analysis (Allen CCF level 5), with the number of sessions and the per-session count of region-specific good-quality neurons. All regions used the 25 sessions with the highest neuron counts meeting the inclusion threshold.

A.2 Distribution of Decision Times

This section reports the distribution of decision times across the 100 sessions included in the brainwide analysis and the effect of the decision-time filter applied during preprocessing (Section 3.2.2).

Decision time D_k was defined as the elapsed time between stimulus onset and the first recorded wheel movement (Section 3.2.2). Across all trials prior to filtering, decision times were right-skewed, with a median of 0.15 ms (IQR $[0.11, 0.22]$ ms).

Trials were retained for analysis if their decision time fell within $[50, 400]$ ms Section 3.2.2). After filtering, a median of 64.1% of trials per session were retained (IQR $[52.6, 73.1]$%; range $[36.4, 86.3]$%). This corresponded to a median of 431 retained trials per session (IQR [340, 522]; range [247, 664])

The filtering bounds and the resulting distribution are shown in Figure A.1.

A.3 Parameter Robustness

Unless otherwise noted, all robustness analyses use the default configuration of the brainwide pipeline: a 50 ms firing-rate window, a decision-time filtering range of $[50, 400]$ ms, and $K = 10$ nearest neighbors. Each subsection varies one parameter while holding the others fixed, and all analyses aggregate across the 100 sessions used in the brainwide analysis.

A.3.1 Firing-Rate Estimation

The overall shape of the convexity time course is consistent across all tested window widths (Figure A.2), demonstrating that the results are robust to the exact choice of window. Larger windows produce an earlier rise in convexity. Activity from later in the decision period contributes to estimates at earlier evaluation points, when spikes get integrated over a wider interval. Smaller windows preserve finer temporal resolution but yield lower peak convexity values. A width of 50 ms was chosen

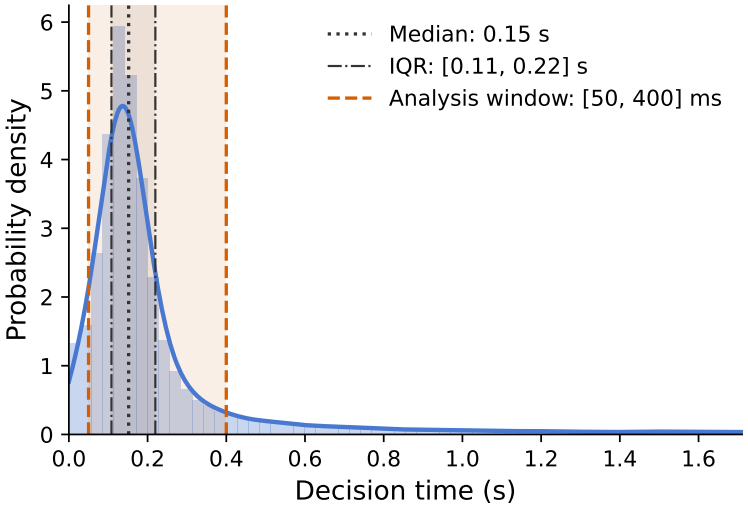

Figure A.1 Distribution of decision times across all trials in the 100 analyzed sessions, prior to filtering, pooled together. Decision time is the interval between stimulus onset and the first recorded wheel movement. Red dashed vertical lines mark the lower (50 ms) and upper (400 ms) filtering bounds, the shaded region indicates retained trials. Trials with no recorded wheel movement, for which no decision time is defined, are excluded.

to preserve temporal resolution without sacrificing the stability of the convexity estimates.

To test whether the results depend on the specific estimator rather than the window width, we additionally compared the default box-window spike count against a Gaussian-smoothed firing-rate estimate, in which spikes are convolved with a Gaussian kernel rather than counted within a hard-edged window. The two estimators yield convexity time courses of the same shape and comparable magnitude (Figure A.3), confirming that the results are robust to the choice of firing-rate estimator and not specific to the binning approach used.

A.3.2 Decision-Time Filtering Bounds

Stricter filtering, which retains a smaller, more temporally homogeneous subset of trials, leads to higher convexity scores (Figure A.4). This is consistent with the expectation that trials with more similar decision times exhibit more comparable temporal structure (Section 3.2.2). However, stricter filtering also reduces the number of trials available per session, introducing a tradeoff between temporal homogeneity and the statistical stability of the per-session convexity estimates. The bounds of $[50, 400]$ ms where chosen as a compromise between these competing considerations, retaining trials with more similar decision times while preserving enough trials per session for stable convexity estimation.

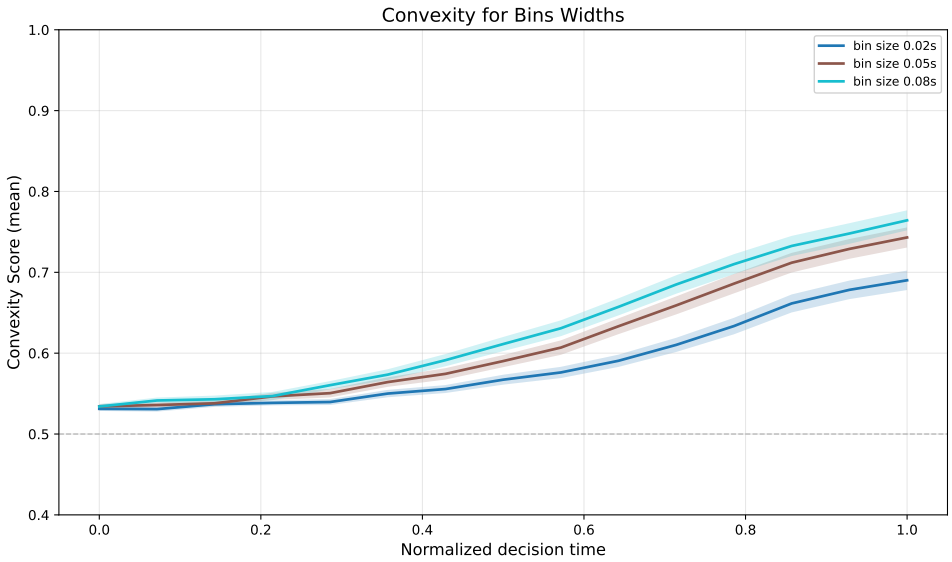

Figure A.2 Mean convexity time course across sessions for different firing-rate window widths.

A.3.3 Number of Nearest Neighbors K

Higher values of K lead to higher convexity scores throughout the time course (Figure A.5), including the early period before convexity rises. Part of this offset reflects graph construction rather than a property of the representation, as directly connected same-class pairs score 1 by construction, and the fraction of such pairs grows with K relative to the number of nodes M. This is consistent with the shuffled-label baseline (Appendix A.4), which carries no temporal structure, yet rises with K. Conversely, smaller values of K increase the chance of disconnected graphs, in which unreachable pairs score 0 and lower the overall score (2.3.2).

The chosen value of $K = 10$ keeps connectivity low enough to limit this inflation while still yielding a connected graph in most sessions. Going from $K = 10$ to $K = 20$ appears to raise the baseline more than it raises the emerged peak convexity. As the relevant quantity is the captured convexity, the height of the converged score above baseline, $K = 20$ offers no clear benefit. The extra connectivity in this case likely adds mostly construction-driven inflation rather than captured structure. Across all tested values of K, the qualitative shape of the time course, an initial flat period followed by a progressive rise, is preserved. The parameter K shifts mostly the absolute convexity values rather than the temporal profile.

A.4 Shuffled-Label Baseline

A chance-level baseline was obtained by randomly shuffling trial choice labels 100 times at each evaluation point and averaging the resulting convexity scores across

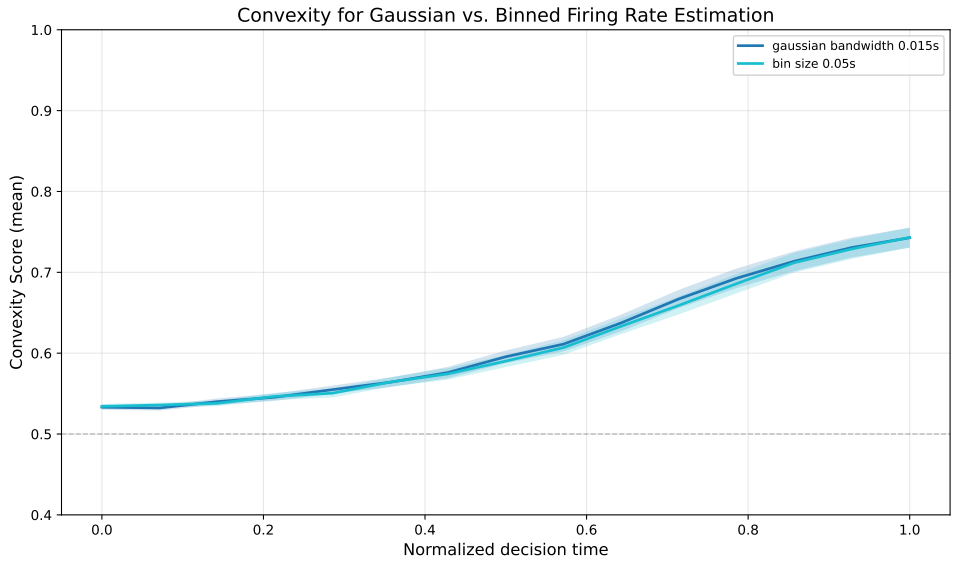

Figure A.3 Mean convexity time course across sessions for two firing-rate estimators: the default box-window spike count and a Gaussian-smoothed estimate (kernel $\sigma = x$ ms).

permutations. The baseline is approximately flat across the decision period at each value of K, indicating that the permuted labels carry no temporal structure. Its level rises with K, from ~ 0.51 at $K = 5$ to ~ 0.55 at $K = 20$ (Figure A.6), consistent with the effect of direct connections in the graph construction (Appendix A.3.3). At the default $K = 10$ the baseline sits at approximately 0.52, slightly above the chance level of $1/C = 0.5$ expected for two choice classes. This provides the reference against which the convexity increase reported in Section 4.1 and Section 4.2 is assessed.

A.5 Neuron Subsampling

Randomly subsampling neurons removes dimensions of the neural population activity vector and is therefore a projection onto a lower-dimensional subspace. This analysis empirically tests the claim that such a projection breaks the convex organization rather than inducing it.

We randomly selected 30 sessions with at least 200 good-quality neurons. For each subsample size $n \in \{20, 35, 50, 100, 200\}$ we randomly subsampled n neurons and computed convexity scores, repeating this 30 times per session as a bootstrap estimate. As for the behavioral correlation, each convexity time course was reduced to a scalar by averaging the scores over the last third of evaluation points (3.3.2)

Scalar convexity decreases as fewer neurons are retained, falling from approximately 0.70 at $n = 200$ to approximately 0.60 at $n = 20$ (Figure A.7). The projection onto fewer dimensions therefore reduces convexity. Before this scalar reduction, the full convexity time courses (Figure A.8) show that the effect is not a constant

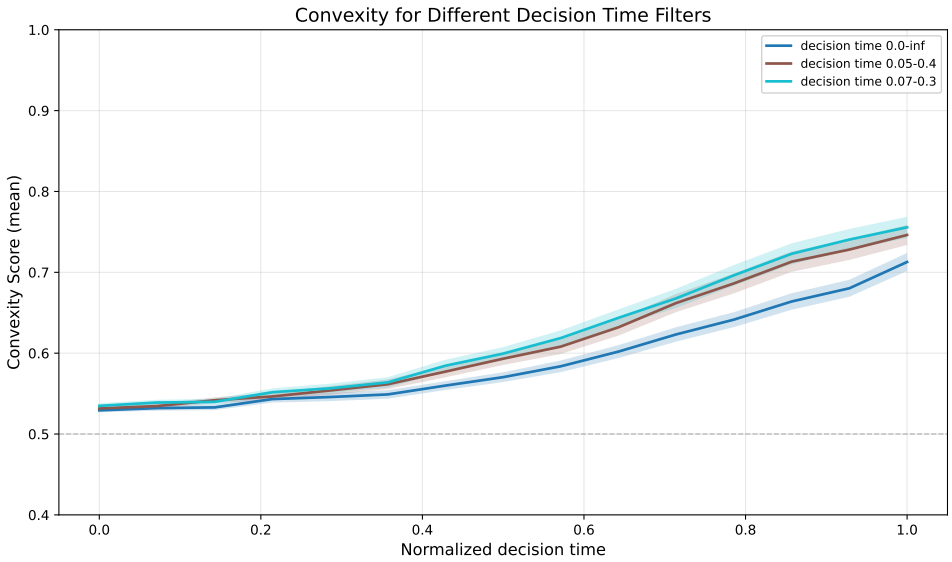

Figure A.4 Mean convexity time course across sessions for different decision-time filtering bounds.

shift across the decision period, and can therefore not be attributed to merely a baseline increase. At trial start the subsample sizes differ by only about 0.01 in convexity between the smallest and largest, all lying close to the shuffled-label baseline (Appendix A.4). This spread grows over the decision period, so that the separation between subsample sizes emerges almost entirely in the rise of convexity rather than at trial start, indicating that it does capture meaningful structure.

Subsampling in this analysis therefore reduces convexity, consistent with the claim that projection via random subsampling breaks present convex organization rather than inducing it.

A.6 Regional Analysis at Other Atlas Levels

The main-text regional analysis defines regions at level 5 of the Allen Common Coordinate Framework [WDL+20]. To check that the results do not depend on this choice of granularity, we repeated the movement-aligned analysis at the adjacent coarser (level 4) and finer (level 6) level. These analyses mostly re-group the same recordings under different anatomical boundaries: the underlying sessions and neurons largely overlap with the level-5 analysis, so agreement across levels reflects robustness to where the regional boundaries are drawn rather than independent replication.

At both levels the same organization seen at level 5 is preserved (Figures A.9 and A.10). All regions rise from a common baseline of approximately 0.53 around movement onset, motor-associated regions (MBmot and MY at level 4; MO and CP at level 6) show the strongest and earliest increases, thalamic and cortical regions

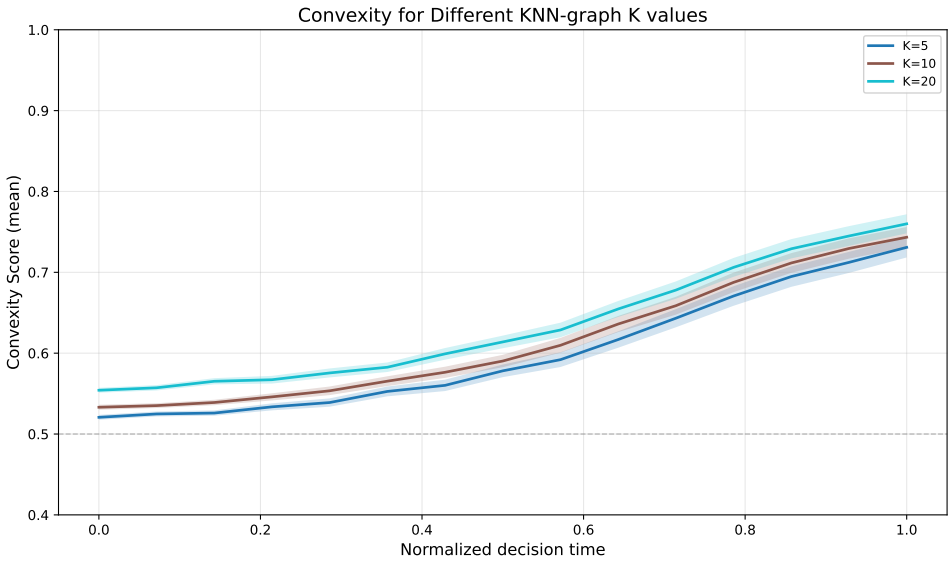

Figure A.5 Mean convexity time course across sessions for different values of K in the KNN graph construction.

are intermediate, and hippocampal regions remain weakest, barely departing from baseline. The ordering and the weak hippocampal modulation therefore persist whether the brain is divided more coarsely or more finely, indicating that the reported dynamics are not an artifact of the level-5 boundaries.

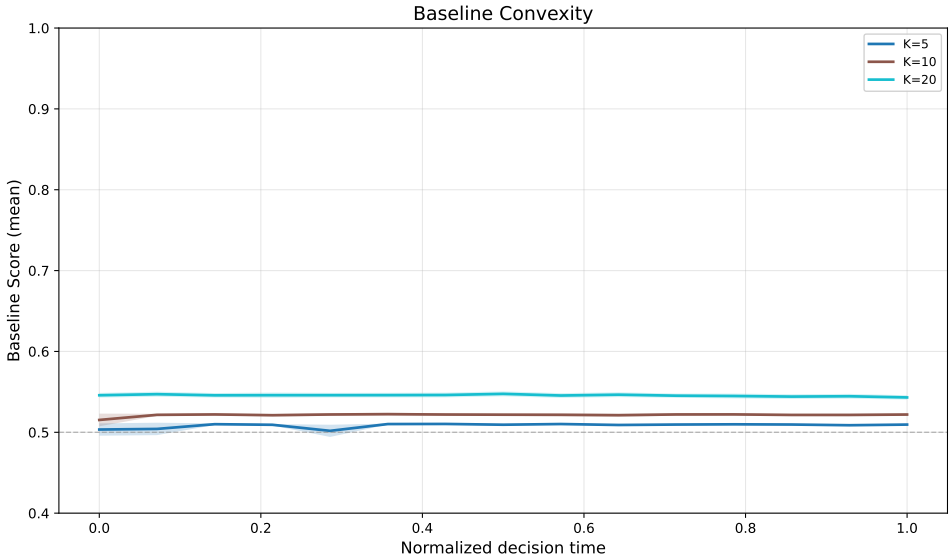

Figure A.6 Shuffled-label baseline convexity across the decision period for three values of K. Labels are randomly permuted 100 times at each evaluation point and convexity averaged across permutations, aggregated over the 100 sessions of the brainwide analysis. Shaded areas indicate SEM.

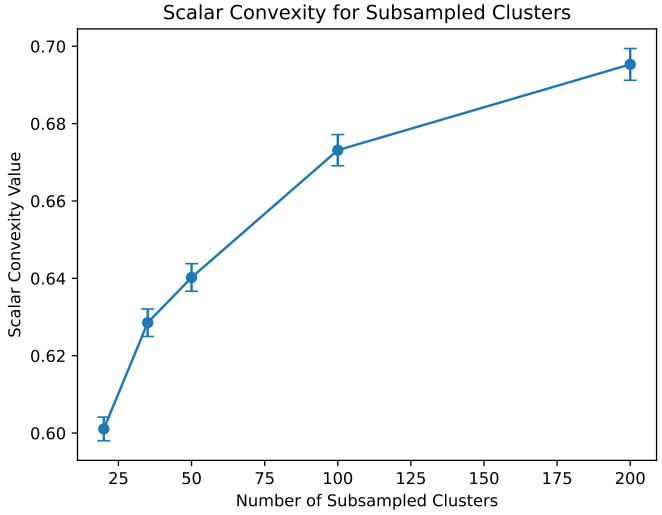

Figure A.7 Scalar convexity as a function of the number of subsampled neurons, aggregated over 30 sessions with at least 200 good-quality neurons and 30 bootstrap repetitions per session and subsampled size. Each time course was reduced to its mean over the last third of evaluation points before averaging. Error bars indicate SEM across sessions.

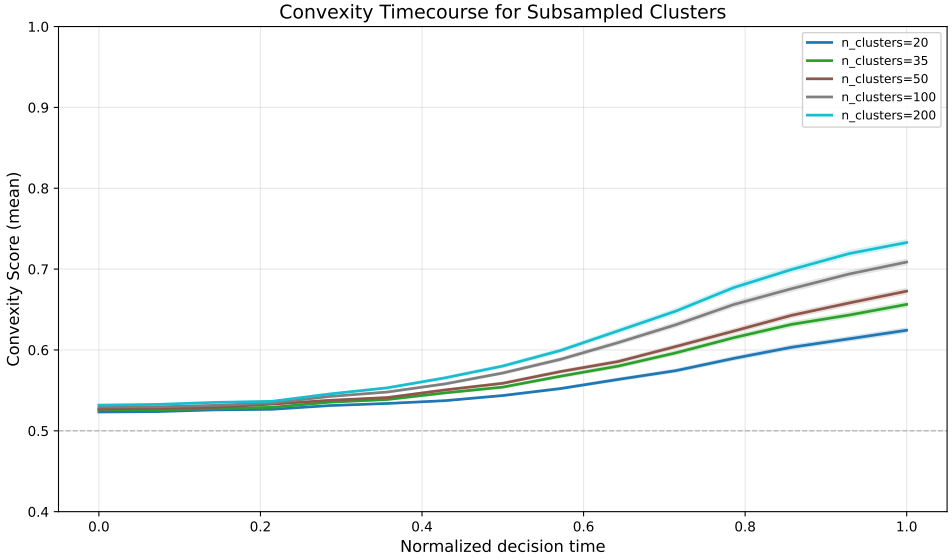

Figure A.8 Convexity time course for different numbers of subsampled neurons, averaged over 30 sessions with at least 200 good-quality neurons and 30 bootstrap repetitions per session and subsampled size. Shaded areas indicate SEM across sessions.

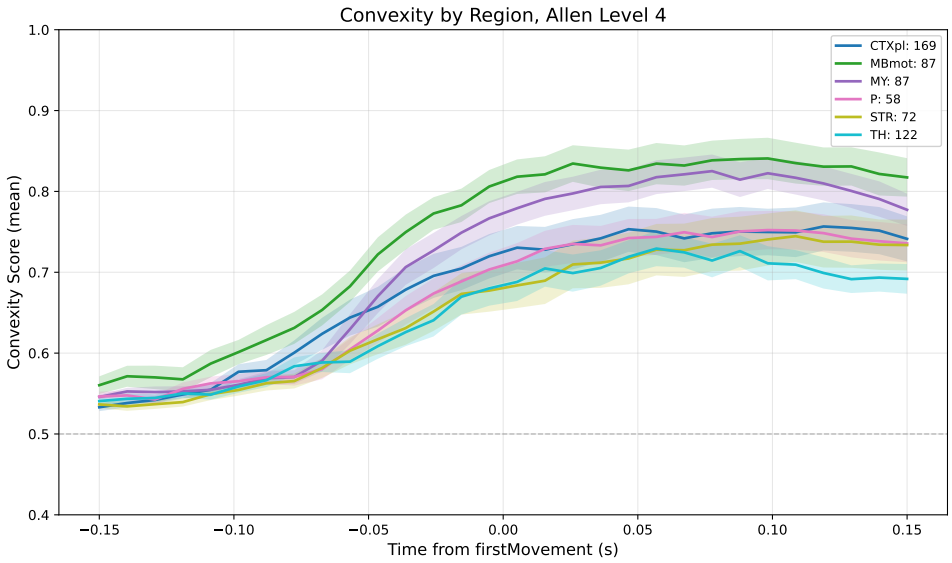

Figure A.9 Movement-aligned convexity time course at Allen CCF level 4. Each curve is the mean across the 25 sessions with the highest region-specific neuron counts for that region. The selected sessions differ between regions. Shaded areas indicate SEM. Legend values give the minimum region-specific good-quality neuron count across the 25 selected sessions. CTXpl, cortical plate; MBmot, midbrain motor-related; MY, medulla; P, pons; STR, striatum; TH, thalamus.

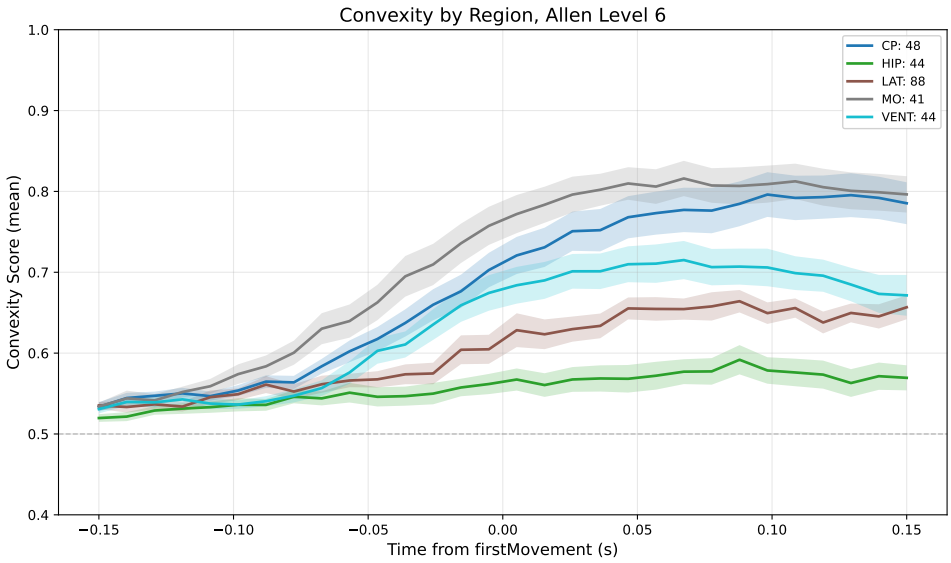

Figure A.10 Movement-aligned convexity time course at Allen CCF level 6. Each curve is the mean across the 25 sessions with the highest region-specific neuron counts for that region. The selected sessions differ between regions. Shaded areas indicate SEM. Legend values give the minimum region-specific good-quality neuron count across the 25 selected sessions. CP, caudoputamen; HIP, hippocampal region; LAT, lateral thalamic group; MO, somatomotor areas; VENT, ventral thalamic group.

Appendix B

Software Implementation

`cvxnr` is an open-source Python package for convexity analysis of neural spiking data, developed as part of this thesis. It implements the analysis pipeline described in Chapter 3 as a documented, reusable library and produced all results reported in this work. Access to the IBL Brainwide Map dataset is integrated directly, and the package runs identically on any local spiking dataset in the supported format (Section B.6).

This appendix is a practical reference. It describes how an analysis run is structured, gives a complete worked example, and then documents the data format and the main configuration options.

B.1 Analysis Run Overview

Conceptually, a full pipeline run proceeds through the six stages of Chapter 3.

1. **Session selection:** Select one or more recording sessions, which fullfill selection criteria (enough neurons in given region or total neurons)
2. **Filtering:** Filter trials by decision time to reduce temporal heterogeneity across trials.
3. **Trial alignment:** Align trials by computing evaluation time points for each trial relative to behavioural events.
4. **Firing rate estimation:** Estimate a population firing-rate vector at each evaluation point.
5. **Score computation:** Compute convexity score for each evaluation point by building a KNN graph over those vectors and measuring how well shortest paths stay in given class.
6. **Aggregation:** Aggregate the resulting timecourses across sessions.

Three objects configure a run and recur throughout the API. An `AlignmentWindow` specifies the temporal alignment; an `AnalysisConfig` bundles the analysis parameters - firing-rate method and width, decision-time filter, neighbourhood size K - and

a data source (`IBLDataSource` or `LocalDataSource`) supplies the recordings. The high-level entry point `analyze_multiple_sessions` takes these and runs the full pipeline after session selection in one call. Functions for each individual stage are also importable and can be called directly, as shown in Section B.5.

Two implementation details are worth noting. First, each session is analyzed in an isolated subprocess, which bounds memory growth across long batch runs. Second, multiple parameter configurations can be evaluated in a single call, so that each session is loaded only once across an entire parameter grid.

B.2 Installation

`cvxnr` requires Python $\geq$ 3.10 and depends on `numpy`, `scipy`, `scikit-learn`, `python-igraph` (for the convexity graph), `matplotlib`, and the IBL data stack (`one-api`, `ibllib`, `brainbox`, `iblatlas`) for IBL access. It is installed from source:

```
git clone https://github.com/nrgrp/cvxnr.git
cd cvxnr
make install
```

B.3 Code Examples

The following reproduces a brainwide-style run: select 100 sessions, align each trial by normalizing across the stimulus onset to movement onset interval, compute convexity scores, and aggregate across sessions.

```
from cvxnr import (IBLDataSource, AnalysisConfig,
    AlignmentWindow,
                   analyze_multiple_sessions)
from cvxnr.multi_session_analysis import save_results

# 1. Select sessions
ibl_source = IBLDataSource() # initializa data source
sessions = ibl_source.get_available_sessions(
    min_clusters=100, n_sessions=100)

# 2. Configure alignment and analysis parameters
alignment = AlignmentWindow(events=['stimOn_times', '
    firstMovement_times'])
config    = AnalysisConfig(rate_method='binned', width=0.05,
                           dt_filter=(0.05, 0.4), convex_K=10)

# 3. Run the full pipeline
results = analyze_multiple_sessions(
    sessions=sessions, n_eval=30,
    configurations=config, alignment=alignment)
save_results(results, 'results.pkl')

# Optional: use build-in plotting API
```

```
plot_timecourse(results, 'figures/brainwide_timecourse.png')
```

Variations. A few common adjustments to the example above:

Flat time single event-alignment. Passing a single event with fixed padding spaces evaluation points uniformly in real time around that event. For example, ±150 ms around movement onset:

```
alignment = AlignmentWindow(events=['firstMovement_times'],
                            front=0.15, tail=0.15)
```

Parameter grid. Passing lists instead of single values runs a grid search. Each session is loaded once and analyzed under every configuration, and each configuration carries its own results:

```
from cvxnr import build_param_grid
configs = build_param_grid(
    binned_bin_sizes=[0.02, 0.05],      # binned configurations
    gaussian_bandwidths=[0.01, 0.02],   # and kernel
        configurations
    decisionTime_filters=[(0.05, 0.4), 0.8],
    convex_K=[10, 20])
    # results in 2 x 2 x 2 x 2 = 16 configurations
```

Region-specific analysis. A `regions` argument threads through session selection and analysis, restricting the population to neurons in the given Allen-atlas region(s). This is the mechanism underlying the regional analyses of Section 3.3.1

```
sessions = IBLDataSource().get_available_sessions(
    regions=['Isocortex'], min_clusters=40, n_sessions=25)
results  = analyze_multiple_sessions(
    sessions=sessions, n_eval=30, configurations=config,
    alignment=alignment, regions=['Isocortex'])
```

Additional scores. By default only the convexity score is computed. The shuffled baseline and the hubness diagnostics (Section B.7) are requested by name through the `scores` argument of the analysis functions:

```
results = analyze_multiple_sessions(
    sessions=sessions, n_eval=30, configurations=config,
    alignment=alignment, scores=['convexity', 'baseline', '
        robinhood', 'k_skewness'])
```

Single session. `analyze_session` runs one session directly, which is convenient for inspecting intermediate results or when doing changes to the pipeline:

```
result = analyze_session(
    session_id=session_id,              # a single session
        identifier
    n_eval=30, configuration=config,
    alignment=alignment, regions=['VISp'])
```

B.4 Command-Line Interface

The package utility can also be used as a command-line tool, with a thin wrapper around the Python API. It provides three commands: `analyze` runs the pipeline and pickles the results, `plot` renders convexity timecourses from one or more result files, and `list-sessions` reports the sessions matching a selection criterion. Session selection is controlled by flags (`regions`, `n-sessions`, `min-clusters`, `sessions`, `data`); the analysis parameters themselves - rate method, width, neighbourhood size K, decision-time filter, alignment, and `n_eval` - are supplied through a built-in preset or a YAML configuration file. Precedence runs flags over config file over preset.

The brainwide run as the example above can be reproduced with a configuration file, which carries the analysis parameters, alignment, and `n_eval`:

```
cvxnr analyze results.pkl --config analysis.yaml \
      --n-sessions 100 --min-clusters 100
cvxnr plot results.pkl --output figures/brainwide_timecourse.
    png
```

Or an example region-restricted run with no configuration file and a built-in preset instead:

```
cvxnr list-sessions --regions VISp --min-clusters 40   # check
    availability
cvxnr analyze visp.pkl --regions VISp --n-sessions 25 \
      --min-clusters 40 --preset standard
```

Because the CLI only wraps the API, the finer-grained controls of the preceding sections, like additional parameters or scores, and the manual pipeline, remain available only through the Python interface. The full flag reference and an annotated configuration template are maintained in the repository.

B.5 Manual Pipeline

Each stage of the pipeline is available as stand-alone functions that can be imported and called directly. The sequence below illustrates the basic flow for analyzing a single session.

```
1 trials  = ibl_source.load_trials(session_id)
2 trials  = decision_time_filter(trials, (0.05, 0.4))   # keep
      50-400 ms trials
3 spikes, clusters = ibl_source.load_spiking_data(session_id)
4
5 # abs. evaluation time points, shape (n_trails, n_eval)
6 t_eval = evaluation_time_points(trials, n_eval=30, alignment=
      alignment)
7
8 # firing rate estimations, shape (n_eval, n_trials, n_clusters)
9 rates  = binned_firing_rates(spikes, t_eval, config.width)
```

```
10
11 # convexity at the first evaluation point
12 score = compute_convexity(rates[0], trials['choice'],
13                            n_neighbors=config.convex_K)
```

This manual path is intended for inspecting intermediate results or doing modifications to the pipeline. Spiking data are large, and looping over many sessions by hand leads to memory buildup which can easily crash the analysis. For batch analysis, prefer `analyze_session` or `analyze_multiple_sessions`, which run each session in an isolated subprocess for clean resource management.

B.6 Input Data Format

The IBL data source loads data directly through the ONE API, so no manual file layout is needed. Local datasets are instead supplied as a flat directory, with one subdirectory per session and one metadata.yaml.

```
data_root/
|- metadata.yaml           # brainmap type, event-name config
|- session_001/
|   |- spikes_times.npy    # spike times,       sh. (n_spikes,)
|   |- spikes_clusters.npy # cluster per spike, sh. (n_spikes,)
|   |- clusters.csv        # columns: cluster_id, acronym
|   |- trials.csv          # columns: choice, <event>_times,
|- session_002/
|- ...
```

`metadata.yaml` declares whether cluster acronyms follow the Allen Atlas hierarchy (`brainmap: AllenAtlas`), in which case a region query resolves through the hierarchy and also matches sub-regions, as for IBL data; or should be string-matched exactly in a flat hierarchy (`brainmap: other`). Both data sources implement the same interface, so the rest of the pipeline is identical regardless of data origin. A local dataset is used by substituting the data source for the IBL one:

```
1 source   = LocalDataSource('path/to/data_root')
2 sessions = source.get_available_sessions(min_clusters=200,
      n_sessions=30)
```

B.7 Additional Options

Additional score Types. To provide a reference for what counts as above chance convexity values for a given graph structure, the package can also be used to compute a label-shuffled baseline: the same score with the choice labels randomly permuted, averaged over several permutations. This gives the chance level for that specific session and configuration, rather than assuming a fixed value.

The package additionally reports two label-independent diagnostics over the same graph that flag when a score may be unreliable. Both measure *hubness*: the

tendency, in high-dimensional embeddings, for a few points to dominate others' nearest-neighbour lists and thereby distort shortest-path structure. Strong hubness at a given configuration is a signal to treat the corresponding convexity estimate with caution.

The baseline and the two diagnostics are not computed by default. They are requested by name through the `scores` argument of the analysis functions, as shown in the example above.

Additional parameters. Beyond the core parameters, `AnalysisConfig` accepts a `params` dictionary for less common options used in the robustness analyses of this thesis:

`correct_trials_only` restrict the analysis to rewarded trials.

`n_clusters_sample` randomly subsample this many clusters before rate estimation, for subsampling control analyses.

`baseline_permutations` number of label permutations averaged for the shuffled baseline, if baseline as score is requested (default 50).

For example:

```
config = AnalysisConfig(
    rate_method='binned', width=0.05, dt_filter=0.8, convex_K=10,
    params={'correct_trials_only': True, 'baseline_permutations': 100})
```

B.8 Reproducibility and Availability

This appendix gives only a basic overview of the package's functionality Most functions accept further options, and only the core API and commands are described here. The complete API and command-line reference is documented in the GitHub repository. The package is released under the Apache License 2.0 and is available at https://github.com/nrgrp/cvxnr. Scripts to regenerate the figures in Chapter 4 are provided in https://github.com/nrgrp/cvxnr/tree/main/scripts.

References

[AHH18] Georgios Arvanitidis, Lars Kai Hansen, and Søren Hauberg. Latent space oddity: On the curvature of deep generative models. In *International Conference on Learning Representations (ICLR)*. OpenReview.net, 2018.

[BdSLT00] Mira Bernstein, Vin de Silva, John C. Langford, and Joshua B. Tenenbaum. Graph approximations to geodesics on embedded manifolds. Technical report, Stanford University, Department of Psychology, 2000.

[BV04] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[CK11] Marlene R. Cohen and Adam Kohn. Measuring and interpreting neuronal correlations. *Nature Neuroscience*, 14(7):811–819, 2011.

[CY14] John P. Cunningham and Byron M. Yu. Dimensionality reduction for large-scale neural recordings. *Nature Neuroscience*, 17(11):1500–1509, 2014.

[Dij59] Edsger W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1:269–271, 1959.

[EH21] R. Becket Ebitz and Benjamin Y. Hayden. The population doctrine in cognitive neuroscience. *Neuron*, 109(19):3055–3068, 2021.

[FHI+25] Charles Findling, Félix Hubert, International Brain Laboratory, Luigi Acerbi, Brandon Benson, Julius Benson, Daniel Birman, Niccolò Bonacchi, E. Kelly Buchanan, Sebastian Bruijns, Matteo Carandini, Joana A. Catarino, Gaelle A. Chapuis, Anne K. Churchland, Yang Dan, Felicia Davatolhagh, Eric E. J. DeWitt, Tatiana A. Engel, Michele Fabbri, Mayo A. Faulkner, Ila Rani Fiete, Laura Freitas-Silva, Berk Gerçek, Kenneth D. Harris, et al. Brain-wide representations of prior information in mouse decision-making. *Nature*, 645(8079):192–200, 2025.

[Gär90] Peter Gärdenfors. Induction, conceptual spaces and ai. *Philosophy of Science*, 57(1):78–95, 1990.

[Gär00] Peter Gärdenfors. *Conceptual Spaces: The Geometry of Thought*. MIT Press, Cambridge, MA, 2000.

[HHML06] Suzana Herculano-Houzel, Bruno Mota, and Roberto Lent. Cellular scaling rules for rodent brains. *Proceedings of the National Academy of Sciences*, 103(32):12138–12143, 2006.

[HS16] Olivier J. Hénaff and Eero P. Simoncelli. Geodesics of learned representations. In *International Conference on Learning Representations (ICLR)*, 2016.

[IAB+25] International Brain Laboratory, Dora Angelaki, Brandon Benson, Julius Benson, Daniel Birman, Niccolò Bonacchi, et al. A brain-wide map of neural activity during complex behaviour. *Nature*, 645(8079):177–191, 2025.

[Int23] International Brain Laboratory. Data release—brainwide map—q4 2022, 2023.

[JSS+17] James J. Jun, Nicholas A. Steinmetz, Joshua H. Siegle, Daniel J. Denman, Marius Bauza, Brian Barbarits, Albert K. Lee, Costas A. Anastassiou, Alexandru Andrei, Cagatay Aydin, Mladen Barbic, Timothy J. Blanche, Vincent Bonin, Joao Couto, Barundeb Dutta, Sergey L. Gratiy, Diego A. Gutnisky, Michael Hausser, 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, 2017.

[KK13] Nikolaus Kriegeskorte and Rogier A. Kievit. Representational geometry: Integrating cognition, computation, and the brain. *Trends in Cognitive Sciences*, 17(8):401–412, 2013.

[MAIP25] Zeinab Mohammadi, Zoe C. Ashwood, International Brain Laboratory, and Jonathan W. Pillow. Identifying the factors governing internal state switches during nonstationary sensory decision-making. *Nature Communications*, 16:11684, 2025.

[PSK+16] Marius Pachitariu, Nicholas Steinmetz, Shabnam Kadir, Matteo Carandini, and Kenneth D. Harris. Fast and accurate spike sorting of high-channel count probes with kilosort. *bioRxiv*, 2016.

[Ros73] Eleanor H. Rosch. Natural categories. *Cognitive Psychology*, 4(3):328–350, 1973.

[She87] Roger N. Shepard. Toward a universal law of generalization for psychological science. *Science*, 237(4820):1317–1323, 1987.

[SPS+19] Carsen Stringer, Marius Pachitariu, Nicholas Steinmetz, Matteo Carandini, and Kenneth D. Harris. High-dimensional geometry of population responses in visual cortex. *Nature*, 571:361–365, 2019.

[SZHCH19] Nicholas A. Steinmetz, Peter Zatka-Haas, Matteo Carandini, and Kenneth D. Harris. Distributed coding of choice, action and engagement across the mouse brain. *Nature*, 576(7786):266–273, 2019.

[TBD+25] Lenka Tětková, Thea Brüsch, Teresa Dorszewski, Fabian Martin Mager, Rasmus Ørtoft Aagaard, Jonathan Foldager, Tommy Sonne Alstrøm, and Lars Kai Hansen. On convex decision regions in deep network representations. *Nature Communications*, 16:5419, 2025.

[VGSS20] Saurabh Vyas, Matthew D. Golub, David Sussillo, and Krishna V. Shenoy. Computation through neural population dynamics. *Annual Review of Neuroscience*, 43:249–275, 2020.

[WDL+20] Quanxin Wang, Song-Lin Ding, Yang Li, Josh Royall, David Feng, Phil Lesnar, Nile Graddis, Maitham Naeemi, Benjamin Facer, Anh Ho, Tim Dolbeare, Brandon Blanchard, Nick Dee, Wayne Wakeman, Karla E. Hirokawa, Aaron Szafer, Susan M. Sunkin, Seung Wook Oh, Amy Bernard, John W. Phillips, et al. The allen mouse brain common coordinate framework: A 3d reference atlas. *Cell*, 181(4):936–953.e20, 2020.

[ZF14] Matthew D. Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European Conference on Computer Vision (ECCV)*, pages 818–833. Springer, 2014.

[ZYHB25] Hao Zhu, Shengchao Yan, Jasper Hoffmann, and Joschka Boedecker. Multi-convex programming for discrete latent factor models prototyping. *arXiv preprint arXiv:2504.01431*, 2025.