Inferring Interpretable Neural Trajectories by Smoothness-Enhanced Embedding Learning

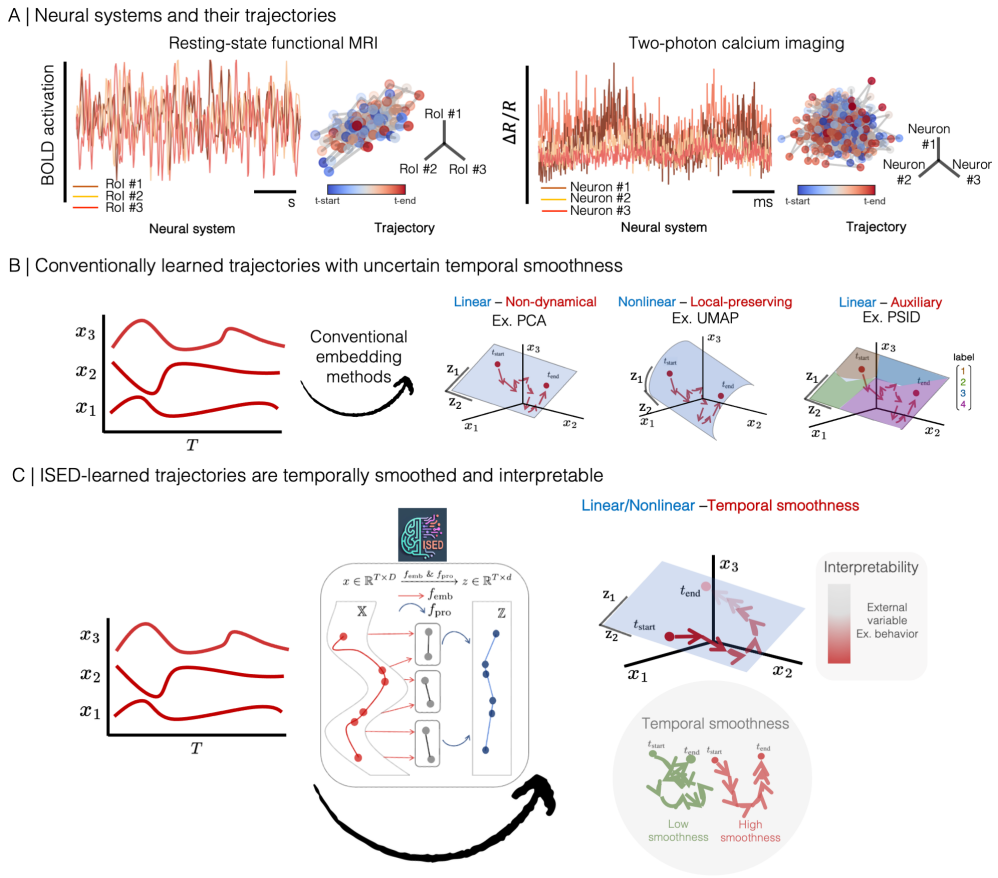
Abstract

Deriving interpretable low-dimensional trajectories from complex neural systems is a central challenge in computational neuroscience. Existing embedding methods often yield trajectories that are overly complex, non-intuitive, and temporally coherent, limiting their interpretability. To address these problems, we present smoothness-enhanced embedding learning, along with ISED, a novel subsequence-based, two-stage learning model that prioritizes temporal smoothness as a core principle for embedding neural dynamics. ISED learns smooth, interpretable neural trajectories that faithfully represent dynamics of high-dimensional neural systems. We validate the smoothness-enhanced trajectories derived by ISED across synthetic datasets and diverse neural systems, including local field potentials from behaving mice and resting-state fMRI from phenotypically distinct individuals, demonstrating its ability to achieve state-of-the-art inter-embedding consistency, superior behavioral interpretability, and robust phenotypical discriminability. By establishing temporal smoothness as a criterion for embedding neural systems, smoothness-enhanced embedding learning (with ISED) provides a powerful framework for interpreting neural dynamics across timescales and data modalities, setting a new standard for low-dimensional modeling of neural systems.

Keywords: Neural time series analysis; embedding learning; temporal smoothness; mutual information

1 Introduction

Neural systems are inherently dynamic, exhibiting continuous changes and evolving over time. Advanced recording techniques [1–3] have enabled researchers to record and analyze these systems as large-scale, dynamic entities across various temporal and spatial scales, capturing phenomena from single neuron-level human cortical activity[4] to population-level neuronal activations across lifespans of mice [5]. While these high-dimensional dynamic systems are foundational for neural analysis, they present significant computational challenges in deriving interpretable low-dimensional neural trajectories that provide an intuitive grasp of complex temporal dynamics [6] (Fig. 1(A)).



One approach to deriving these neural trajectories is to apply embedding methods to project high-dimensional neural data onto low-dimensional manifolds [7–11]. This leads to a central question in computational neuroscience. What is the embedding method for learning interpretable neural trajectories, or alternatively, what characteristic defines the interpretable geometry of these low-dimensional neural trajectories?

Classical embedding methods, such as PCA, suggest that interpretability may arise from their linearity; however, they lack dynamic components essential for modeling neural systems. In contrast, dynamic embedding approaches that preserve local structure, e.g., MDS [15], t-SNE [16], UMAP [17], often produce complex, non-intuitive trajectories. More recent auxiliary-aligned approaches, e.g., pi-VAE [18], PSID [19], Cebra [20], enable learned trajectories to align with external variables, such as behavioral data, thereby enhancing interpretability. However, these methods rely heavily on richness of auxiliary variables provided (Fig. 1(B)). This highlights the need for a more general, auxiliary-independent dynamic embedding method that can effectively derive interpretable trajectories for neural systems.

Fig. 1 Smoothness-enhanced trajectories of neural systems: background, limitations of conventional embedding methods, and the proposed approach.
(A) Background on neural system trajectories. Neural systems express themselves as complex, high-dimensional dynamics across various temporal resolutions, as seen in datasets such as resting-state functional MRI [12] and two-photon calcium imaging [13]. These trajectories, often visualized as start-to-end scatterplots, tend to be intricate and challenging to interpret due to their high dimensionality. To illustrate, three feature dimensions were randomly selected to display sample trajectories from each system, highlighting the complexity and limited intuitiveness of such visualizations.
(B) Limitations of conventional embedding methods. Conventional embedding methods project high-dimensional dynamics (x, leftmost figure) over time (T) onto a low-dimensional embedding manifold (z, right three figures), preserving the temporal length T. These methods differ by the linearity of the manifold and the properties of the dynamics [8, 14]. For instance, PCA is a linear, non-dynamic method, while UMAP, a nonlinear approach preserving only local structure, makes limited use of temporal information and disregards trajectory smoothness. Preferential subspace identification (PSID) identifies linear subspaces related to auxiliary variables, but depends on richness of these variables and does not ensure smooth trajectories. Supplementary Notes §A discusses additional examples. Overall, conventional methods often fail to incorporate temporal smoothness in their embeddings.
(C) Smoothness-enhanced embedding learning with ISED. The ISED model (center box) introduces smoothness-enhanced embedding learning to capture temporally smoothed neural trajectories. ISED operates in two stages: in the first stage, high-dimensional dynamics (x) are segmented into subsequences and embedded into low-dimensional representations through a learned embedding function (f_{emb}, indicated by red arrows). In the second stage, these low-dimensional subsequences are projected onto the embedding manifold using a learned projection function (f_{pro}, shown by blue arrows), producing smooth, interpretable trajectories of neural systems (z). These smoothness-enhanced trajectories exhibit high temporal smoothness and meaningful external correspondence.

In this study, we employed the principle of temporal smoothness to establish a general-purpose embedding paradigm – **temporal smoothness-enhanced embedding learning** to derive interpretable trajectories in complex neural systems (Fig. 1(C)). The concept of temporal smoothness [21–24] (Supplementary Notes §B), are well-supported by numerous neural recording studies [7, 25–30], suggesting that an underlying trajectory of a neural system exhibits smooth and predictable transitions over time to capture gradual changes along the temporal axis. Despite its recognized value, this smoothness principle has yet to be systematically integrated into embedding learning, leaving its full potential unexplored in the study of neural trajectories.

We propose a new embedding learning method called **information-based smoothness-enhanced embedding dynamics learning model** (**ISED**). ISED achieves smoothness by incorporating both within-subsequence temporal denoising and between-subsequence temporal prediction to learn low-dimensional trajectories with smooth and interpretable transitions. Extensive numerical simulations demonstrate that these ISED-learned smoothness-enhanced trajectories yield more accurate, continuous representations of the original system, significantly surpassing conventional embedding approaches in capturing true dynamics of neural systems.

In real neural systems, smoothness-enhanced trajectories are expected to have broad implications for neuroscientific discoveries that interpret behavior [31, 32]. To explore this, we applied smoothness-enhanced embedding learning to curated CA1 memory cell recordings from four exploring mice [33], producing consistent behavior-aligned, inter-mouse neural trajectories that provide insights into hippocampal dynamics. ISED-derived, temporally smoothed neural trajectories are also informative in reflecting phenotype distinctions in brain dynamics from resting-state human fMRI

data [12], enabling phenotype-specific exploration of brain dynamics through the lens of smoothness-enhanced trajectories. By employing temporal smoothness as a central criterion for embedding neural dynamics, the proposed smoothness-enhanced embedding learning offers a robust, versatile tool for representing complex neural systems in interpretable, low-dimensional form.

Results

Smoothness-enhanced trajectories capture the core of a synthetic system.

Overview of ISED in learning smoothness-enhanced trajectories.

In pursuit of learning smoothness-enhanced trajectories in modeled neural systems, we introduce ISED, a two-stage, subsequence-based unsupervised learning model to directly learn low-dimensional trajectories (Methods and Extended Data Fig. 1). For a modeled high-dimensional neural system represented as a D-dimensional multivariate time series of length T (input feature dynamics x), subsequences of this time series, defined by a pre-determined length k, can be automatically extracted and embedded onto a lower-dimensional manifold ($d \ll D$), yielding embedding subsequences $\tilde{z}$ via a learnable embedding function f_{emb}. These are then summarized into target embedding dynamics z through a learnable projection function f_{pro} (Methods and Algorithm 1). The specific forms of f_{emb} and f_{pro} are adaptable, allowing practitioners to tailor them as needed (Supplementary Notes §C).

Table 1 General steps to implement the ISED model.

Input:	High-dimensional neural dynamics and hyper-parameters
Output:	Trained embedding and projection functions.
Step	**Description**
(1)	Extract subsequences from high-dimensional dynamics of modeled systems.
(2)	Encode subsequences via the embedding function and compute local smoothness.
(3)	Project encoded subsequences with the projection function, calculating global smoothness.
(4)	Calculate mutual information between original and projected data.
(5)	Train embedding and projection functions by maximizing the mutual, local smoothness and global smoothness information.

General steps for implementing ISED on a neural system are outlined in Table 1. With defined subsequence length k, ISED maximizes mutual information shared between feature and embedding dynamics, $\mathrm{I}(x; z)$, ensuring that the learned embedding dynamics accurately represent the original data. To achieve targeted temporal smoothness, ISED applies two types of smoothness operations: temporal de-noising to enhance local within-subsequence smoothness, represented by Jacobian dynamics of embedding subsequences $J(\tilde{z})$, and temporal prediction to strengthen global between-subsequence smoothness, measured by temporal predictability of embedding dynamics $I_{pred}(z)$. Once trained, ISED uses optimized embedding and projection functions to transform new high-dimensional data into smoothness-enhanced trajectories.

Smoothness-enhanced trajectories in a synthetic system.

As proof of concept, we validated ISED on simulated data from a synthetic neural system. This simulation study assessed ISED's capability in learning smoothness-enhanced trajectories and evaluating the embedding performance of these trajectories. We generated 2D time series using a deterministic function with controlled randomness, providing ground-truth 2D latent variables. These variables were then mapped to high-dimensional (100D) feature space using RealNVP [34], yielding high-dimensional feature dynamics for the synthetic system (Methods and Fig. 2(A)). We applied various embedding methods to learn trajectories of the synthetic system, including a non-dynamical approach (PCA), a continuity-preserving method (UMAP), an auto-regressive embedding (AE+RNN), and our smoothness-enhanced approach (ISED). All methods were trained on identical samples, with evaluations on test data. The evaluation framework comprised three assessments: (1) temporal smoothness of the resulting trajectories, measured via first-order differences; (2) reconstruction accuracy of the resulting trajectories against the true 2D latent space; and (3) reconstruction accuracy between high-dimensional 100D feature dynamics and their decoded forms for methods with decoding or inverse transformation functions (Methods and Fig. 2(B)).

Smoothness-enhanced trajectories outperform conventionally learned trajectories.

We benchmarked our proposed ISED model against other embedding methods, including PCA, UMAP, and AE+RNN. ISED's smoothness-enhanced trajectories outperformed these methods by producing smooth, temporally predictable patterns with high fidelity in recovering true 2D latent structures in simulated systems (Fig. 2(C) and Extended Data Fig. 2(A)). ISED ranked highest across three evaluation metrics (Temporal smoothness − One-way ANOVA: $F_{(3,8)} = 60.74, p < 0.001$; R^2 on latent − One-way ANOVA: $F_{(3,8)} = 40.61, p < 0.001$; R^2 on feature − One-way ANOVA: $F_{(3,8)} = 24.93, p < 0.001$; Fig. 2(D) and Extended Data Fig. 2(B)). Full benchmarking results, including additional embedding methods, are documented in Supplementary Results §A.).

Smoothness underpins the success of an embedding method.

In ISED. Notably, the degree of smoothness in ISED-learned trajectories can be tuned, which determines overall embedding performance. As the subsequence length (k) increases, these trajectories evolve into slower-varying patterns with predictable trends (Fig. 2(E)), offering computational advantages and closely aligning with the 2D ground-truth latent space for accurate reconstruction of 100-dimensional feature dynamics (Temporal smoothness − One-way ANOVA: $F_{(3,8)} = 22.90, p < 0.001$; R^2 on latent − One-way ANOVA: $F_{(3,8)} = 4.37, p < 0.05$; R^2 on feature − One-way ANOVA: $F_{(3,8)} = 0.37, p = 0.77$; Pearson's correlation coefficient between smoothness and R^2 on latent space in ISED: $r = 0.57$, $p < 0.001$, linear fit $R^2 = 0.49$; Fig. 2(F)). An ablation study further examined respective roles of local and global smoothness, reinforcing the importance of temporal smoothness in ISED-derived trajectories (Extended Data Fig. 2(C)(D)).

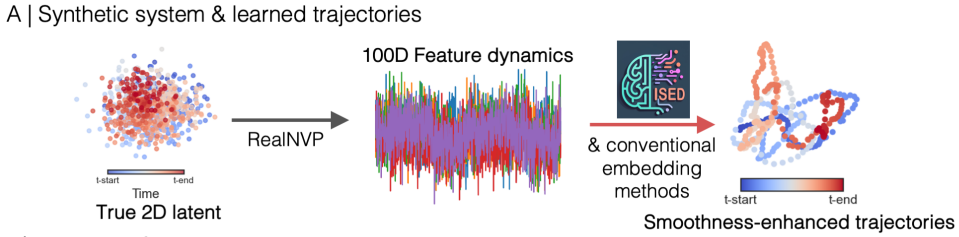

A | Synthetic system & learned trajectories
100D Feature dynamics
RealNVP
ISED
& conventional embedding methods
t-start
t-end
Time
True 2D latent
Smoothness-enhanced trajectories
B | Evaluation framework
Temporal smoothness
Reconstruction accuracy on latent[R^2]
Reconstruction accuracy on feature[R^2]
t_{start}
t_{end}
Less
More
Learned 2D latent
vs.
True 2D latent
Generated feature dynamics
True feature dynamics
C | Conventional vs. smoothness-enhanced trajectories
Linear – None (PCA)
Nonlinear – Local-preserving (UMAP)
Nonlinear – Autoregressive (AE+RNN)
Smoothness-enhanced (ISED)
D | Embedding performance
R^2 on latent
R^2 on feature
PCA
UMAP
AE+RNN
ISED
Smoothness
E | k-determined smoothness in ISED
Smoothness
z_2
z_1
k = 5
k = 50
k = 100
k = 125
ISED(k=5)
ISED(k=50)
ISED(k=100)
ISED(k =125)
F | Importance of smoothness in ISED
Correlation Coefficient: r = 0.57
Linear fit (R^2=0.32)
R^2 on latent
Smoothness (a.u.)
G | Relation between smoothness and reconstruction accuracy
$R^2 = 0.63$
R^2 on latent
Smoothness[a.u.]
PCA
UMAP
AE+RNN
ISED
H | Importance of smoothness in other embedding methods
PCA
PCA-smooth
AE+RNN
AE+RNN-sub
Smoothness (a.u.)
Latent R^2
AR order
k size

Fig. 2 Smoothness-enhanced trajectories of synthetic systems.
(A) Synthetic system setup. We generated synthetic 2D time series as ground-truth latent dynamics, projecting them onto a 100D feature manifold using a randomly initialized RealNVP, which served as input time series. Multiple embedding methods (PCA, UMAP, AE+RNN, and multi-timescale ISED) were applied to learn 2D trajectories from these 100D features.
(B) Evaluation framework. We evaluated resulting 2D trajectories from different embedding methods for temporal smoothness, fidelity to true 2D latents, and ability to reconstruct 100D feature dynamics.
(C) Comparison of learned trajectories. We visually compared learned trajectories across methods (PCA, UMAP, AE+RNN, and ISED), displaying aligned scatterplots for each. Here, results are shown for the $\alpha = 0.5$ synthetic system, with full results in Extended Data Fig. 2(A). Scatterplot points are color-coded by time index, transitioning from blue to red to indicate temporal progression.
(D) Quantitative evaluation of embedding methods. We quantitatively assessed the temporal smoothness and reconstruction accuracy of 2D trajectories and 100D feature dynamics for each embedding method (PCA, UMAP, AE+RNN-sub, and ISED). The results were mapped onto a triangular spider plot, with each vertex representing a normalized value for a different metric. Full results can be found in Extended Data Fig. 2(B), Supplementary Results §A and Supplementary Fig. 1.
(E) Effect of subsequence length k on smoothness in ISED. Scatterplots depict ISED-learned trajectories with various subsequence lengths ($k = 5/50/100/125$), showing a general increase in smoothness as k grows. Points are color-coded similarly to (D).
(F) Importance of smoothness in ISED-learned trajectories. Left: We assessed smoothness, 2D latent reconstruction accuracy, and 100D feature reconstruction accuracy for each k in a triangular spider plot with normalized metrics. Right: We examined correlations between temporal smoothness and other metrics across k values, noting that performance peaked at $k = 125$ (approximately one-quarter of the original dynamics' dominant period), with performance declining beyond this threshold.
(G) Correlations across embedding methods. We analyzed the relationship between smoothness and embedding fidelity (2D latent reconstruction) across various methods.
(H) Incorporating smoothness in other methods. Beyond ISED, we applied temporal smoothness enhancements to standard methods, producing variants like PCA-smooth (based on PCA; the left figure) and AE+RNN-sub (based on AE+RNN; the right figure). Additional examples are in Supplementary Results §A.

Across embedding methods. Across embedding methods, temporal smoothness of resulting trajectories strongly correlates with reconstruction accuracy (Pearson's correlation between smoothness and R^2 on latent: $r = 0.70$, $p < 0.001, R^2 = 0.63$; Fig. 2(G)). To further investigate this relationship, we tested the effect of added temporal smoothing on standard methods. Beyond ISED, we created smoothed versions of PCA (PCA-smooth) and AE+RNN (AE+RNN-sub) with temporal smoothing operations applied during embedding learning (Methods). These modifications increased temporal smoothness (Temporal smoothness: PCA: 0.02 vs. PCA-smooth: 0.16, AE+RNN: 0.01 vs. AE+RNN-sub($k = 125$): 0.61; Fig. 2(H)) and improved reconstruction fidelity to the 2D latent ground truth(Reconstruction accuracy (2D latent): PCA: 0.16 vs. PCA-smooth: 0.26, AE+RNN: 0.2 vs. AE+RNN-sub($k = 125$): 0.49; Fig. 2(H)). Similar correlations between smoothness and embedding performance were also observed in these cases (Pearson's correlation between smoothness and R^2 on latent: PCA-smooth: $r = 0.83, p < 0.001, R^2 = 0.70$, AE+RNN-sub: $r = 0.95, p < 0.001, R^2 = 0.91$). Additional examples are provided in Supplementary Results §A and Supplementary Fig. 2.

Smoothness-enhanced neural trajectories of the hippocampal system in exploring mice.

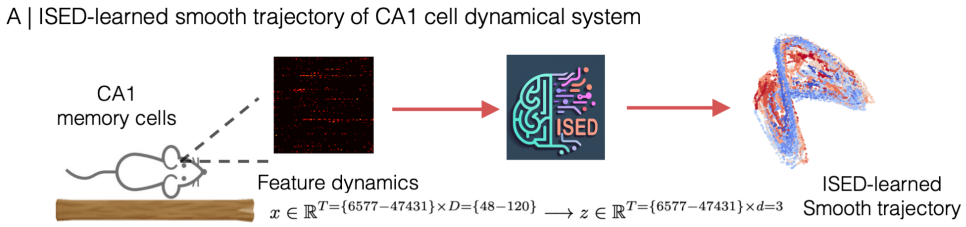
Building on the success of smoothness-enhanced trajectories in previously derived synthetic systems, we applied this approach to real neural systems. These ISED-learned trajectories are particularly valuable for high-dimensional neuroscientific data requiring smoothed temporal representations. Our initial focus was on learning smoothness-enhanced trajectories from the micro-level CA1 hippocampal region of four exploring mice [33]. These smoothness-enhanced trajectories were subsequently analyzed to assess their relevance in understanding hippocampal dynamics (Fig. 3(A)).

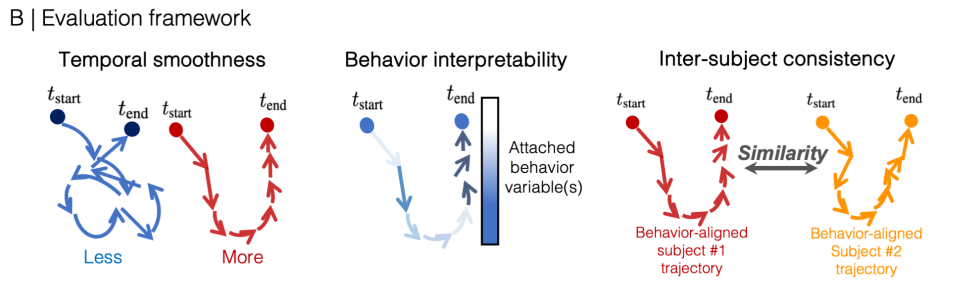
Smoothness-enhanced neural trajectories enable interpretation of behavior.

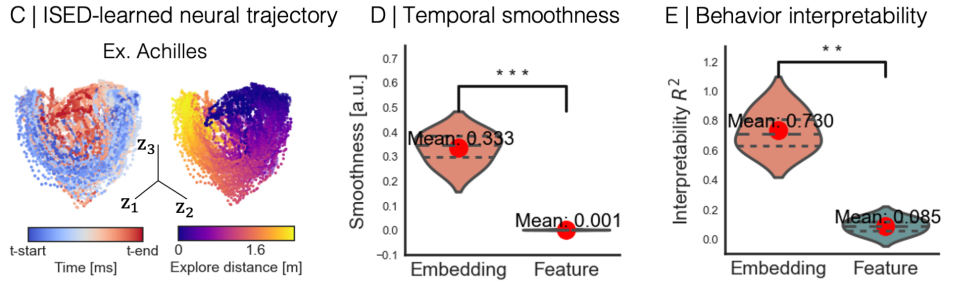
Focusing on smoothness-enhanced neural trajectories, we examine their temporal smoothness, intra-mouse behavioral interpretability, and inter-mouse behavioral consistency (Fig. 3(B)). These smoothness-enhanced neural trajectories exhibit continuous temporal transition patterns from start to finish. Notably, they align with behavior-related changes in exploration patterns (Fig. 3(C)). Quantitatively, compared to the original feature dynamics, these smoothness-enhanced neural trajectories (embedding dynamics) are not only smoother (Temporal smoothness − Paired t-test between embedding and feature dynamics: $t_3 = 10.39, p < 0.001, Cohen's\ d = 5.19$; Fig. 3(D)), but also show superior behavioral interpretability (Behavior interpretation analysis − Paired t-test between embedding and feature dynamics: $t_3 = 7.64, p < 0.01, Cohen's\ d = 3.82$; Fig. 3(E)). Interestingly, even without incorporating behavioral information during trajectory learning, these trajectories exhibit strong behavioral relevance, showing high similarity to behavior-identified embeddings (produced using PSID [19]) among the four mice (Pearson's correlation coefficient (averaged over four mice): $\bar{r}_4 = 0.84$; Extended Data Fig. 3(A)).

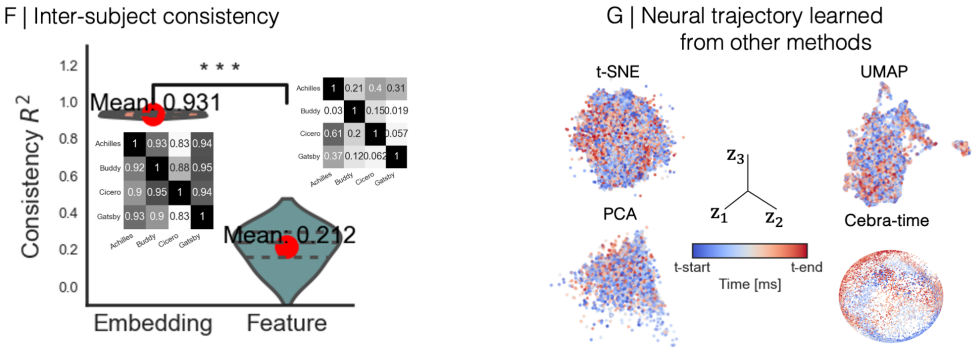
Smoothness-enhanced neural trajectories are idiosyncratic but consistent across behaviorally aligned mice.

As each mouse behaved differently in the exploratory task (Supplementary Results §B and Supplementary Fig. 4(A)), ISED captures the idiosyncrasy in learned mouse-wise smoothness-enhanced neural trajectories, reflected by their differing rolling variation patterns (Supplementary Fig. 4(B)). Remarkably, these idiosyncratic trajectories converge to uniform patterns across mice after behavior alignment, indicating highly consistent inter-subject trajectories for behaviorally aligned mice (Behavior-aligned inter-mouse consistency − Paired t-test between embedding and feature dynamics: $t_3 = 13.36, p < 0.001, Cohen's\ d = 0.99$; Fig. 3(F)). Notably, this neuron-behavior congruence across subjects is only evident in ISED-learned smoothness-enhanced trajectories, not in the original dynamics (Inter-subject representational similarity analysis [35] − Mantel's test between inter-subject behavior and neural similarity matrices: for feature dynamics: $r = -0.61$, for embedding dynamics: $r = 0.58$; Extended Data Fig. 3(B)).


A | ISED-learned smooth trajectory of CA1 cell dynamical system
CA1
memory cells
ISED
Feature dynamics
$x \in \mathbb{R}^{T=\{6577-47431\}\times D=\{48-120\}} \longrightarrow z \in \mathbb{R}^{T=\{6577-47431\}\times d=3}$
ISED-learned
Smooth trajectory


B | Evaluation framework
Temporal smoothness
Behavior interpretability
Inter-subject consistency
t_{start}
t_{end}
Less
More
Attached
behavior
variable(s)
Similarity
Behavior-aligned
subject #1
trajectory
Behavior-aligned
Subject #2
trajectory


C | ISED-learned neural trajectory
Ex. Achilles
z_1
z_2
z_3
t-start
t-end
Time [ms]
0
1.6
Explore distance [m]
D | Temporal smoothness
Smoothness [a.u.]
Mean: 0.333
Mean: 0.001
Embedding
Feature
E | Behavior interpretability
Interpretability R^2
Mean: 0.730
Mean: 0.085
Embedding
Feature


F | Inter-subject consistency
Consistency R^2
Mean: 0.931
Mean: 0.212
Embedding
Feature
Achilles
Buddy
Cicero
Gatsby
G | Neural trajectory learned
from other methods
t-SNE
UMAP
PCA
Cebra-time
z_1
z_2
z_3
t-start
t-end
Time [ms]

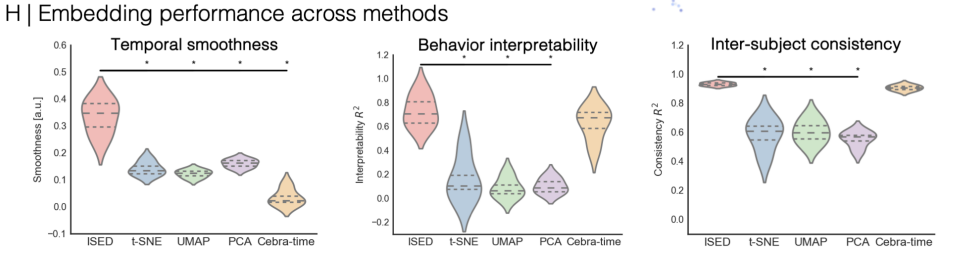

H | Embedding performance across methods
Temporal smoothness
Behavior interpretability
Inter-subject consistency
Smoothness [a.u.]
Interpretability R^2
Consistency R^2
ISED
t-SNE
UMAP
PCA
Cebra-time

Fig. 3 Smoothness-enhanced neural trajectories of the hippocampal system in mice.
(A) CA1 hippocampal system and ISED-learned smoothness-enhanced neural trajectories. Neural data were recorded from CA1 memory cells of four exploring mice [33]. We applied ISED to mouse-specific high-dimensional neural data to derive smoothness-enhanced trajectories.
(B) Evaluation framework for mouse-wise trajectories. Each trajectory was evaluated for temporal smoothness, behavior interpretability, i.e., predictability of the trajectory with respect to exploratory behavior, and inter-subject consistency aligned to behavior across mice.
(C) Smoothness-enhanced neural trajectory for mouse `Achilles`. The ISED-learned smooth neural trajectory for mouse `Achilles` is displayed as two scatter plots, color-coded by time index (left) and behavior distance (right), respectively.
(D) Comparison of temporal smoothness: smoothness-enhanced trajectories vs. original feature dynamics. Temporal smoothness of ISED-learned trajectories was compared to original feature dynamics for all four mice, controlling for dimensionality.
(E) Behavior interpretability of trajectories. We assessed the linear predictability of both ISED-learned smoothness-enhanced trajectories and original feature dynamics in relation to each mouse's exploratory behavior.
(F) Behavior-aligned inter-mouse consistency of smoothness-enhanced trajectories vs. original feature dynamics. We evaluated inter-mouse trajectory similarity after behavior alignment, with consistency results shown in two correlation matrices for both embedding and original feature dynamics.
(G) Comparison with other embedding methods. For mouse `Achilles`, we compared ISED with other methods (t-SNE, UMAP, PCA, Cebra-time) by showing learned trajectories, color-coded by time index. Full trajectories for all four mice are available in Supplementary Results §B and Supplementary Fig. 3.
(H) Benchmark ISED against other embedding methods. Each embedding method's performance was evaluated based on temporal smoothness, behavior interpretability, and inter-subject consistency.

Smoothness-enhanced neural trajectories outperform conventional neural trajectories.

Compared to neural trajectories learned using conventional methods such as t-SNE, UMAP, PCA, and Cebra-time, ISED-learned smoothness-enhanced trajectories are the only ones that clearly indicate temporal progression from start to finish (Fig. 3(G)). These smooth transitions also feature highly predictable patterns, as reflected by the highest determinism rates observed in recurrence analysis (Supplementary Fig. 5.). Quantitatively, smoothness-enhanced trajectories show significant advantages over the other methods, e.g., t-SNE, UMAP, PCA and Cebra-time; Fig. 3(H) on smoothness (Temporal smoothness − Mann-Whitney U test between ISED and other methods: $U = 16/16/16/16$, $p < 0.05/p < 0.05/p < 0.05/p < 0.05$), behavior interpretability (Intra-mouse behavior interpretability − Mann-Whitney U test between ISED and other methods: $U = 16/16/16/10, p < 0.05/p < 0.05/p < 0.05/p < 0.05$), and inter-subject consistency (Behavior-aligned inter-mouse consistency − Mann-Whitney U test between ISED and other methods: $U = 16/16/16/14, p < 0.05/p < 0.05/p < 0.05/p < 0.05$).

Smoothness-enhanced brain trajectories of resting-state cortical systems in healthy and ASD individuals.

In addition to modeling the micro-level hippocampal system of exploring mice, we also applied smoothness-enhanced embedding learning to a macro-level cortical system with phenotypic conditions, demonstrating its ability to derive smoothness-enhanced brain trajectories for phenotyping. The macro-level cortical system was derived from

curated, preprocessed resting-state functional MRI time series from the ABIDE-I database [12], which includes 117 individuals, either healthy controls (HC) or diagnosed with autism spectrum disorder (ASD). ISED, with an optimally defined wide subsequence length, was used to learn brain trajectories from these individual-level fMRI time series (Methods and Fig. 4(A)). Resulting brain trajectories were evaluated for smoothness, inter-subject consistency, and phenotype decodability (Fig. 4(B)).

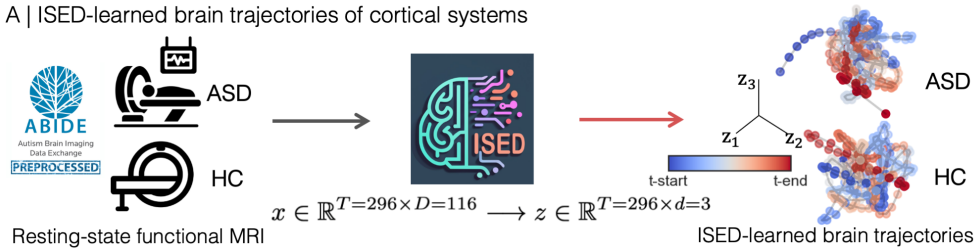
Smoothness-enhanced brain trajectories are consistent among individuals.

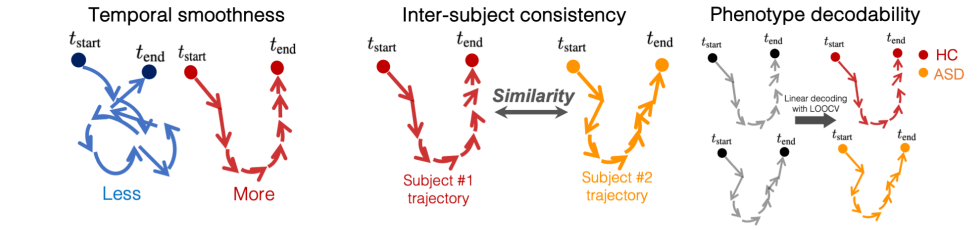
In contrast to the complex and non-intuitive original (feature) dynamics (Supplementary Results §C and Supplementary Fig. 7), the learned smoothness-enhanced brain trajectories for both HC and ASD individuals can be well characterized by their temporal smoothness, showing clear transitions from start to end (Temporal smoothness – Paired t-test between embedding and feature dynamics: $t_{116} = 41.90, p < 0.001, Cohen's\ d = 3.84$; Fig. 4(C)). Additionally, these trajectories show remarkable inter-subject consistency, irrespective of phenotypic condition (Inter-subject consistency – Paired t-test between embedding and feature dynamics: $t_{116} = 50.12, p < 0.001, Cohen's\ d = 4.34$; Fig. 4(D)).

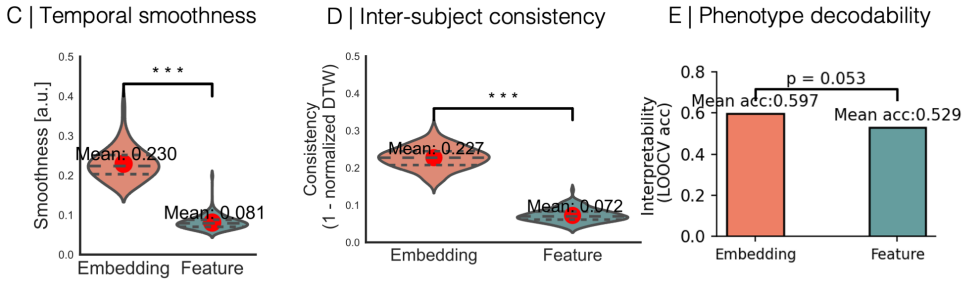
Smoothness-enhanced brain trajectories differentiate between ASD and HC individuals.

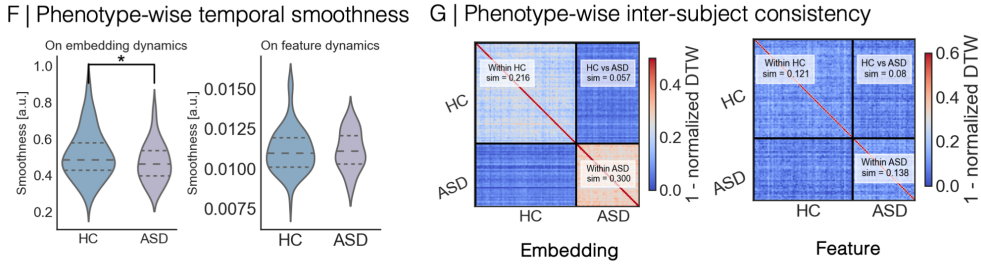
As decoding features, smoothness-enhanced brain trajectories offer better phenotypic differentiation between ASD and HC individuals than the original dynamics (Phenotype decoding – Paired McNemar's test: $\chi^2 = 19.0, p = 0.053$; Fig. 4(E)). Only these smoothness-enhanced trajectories show significant differences in temporal smoothness between ASD and HC groups (Temporal smoothness analysis – Independent t-test: $t_{115} = 3.87, p < 0.05$; Fig. 4(F)). Importantly, when examining inter-subject consistency, these smoothness-enhanced brain trajectories also capture between-subject similarities that align with phenotypic groupings (Inter-subject consistency (averaged across individuals in groups): of HC individuals: $\bar{r} = 0.22$, within ASD individuals: $\bar{r} = 0.30$, between HC and ASD individuals: $\bar{r} = 0.057$; Fig. 4(G)).

Additionally, these trajectories remain competitive with conventionally learned trajectories from PCA, t-SNE, UMAP, and Cebra-time embedding methods across temporal smoothness (Temporal smoothness – Mann-Whitney U test between ISED and other methods: $U = 14103/14153/13733/14161$, $p < 0.05/p < 0.05/p < 0.05/p < 0.05$; Fig. 4(H)), inter-subject consistent (Inter-subject consistency similarity – Mann-Whitney U test between ISED and other methods: $U = 14161/14160/14089/14161$, $p < 0.001/p < 0.001/p < 0.001/p < 0.001$; Fig. 4(H)), and phenotypically congruent brain trajectories at rest (Phenotype decoding – Paired McNemar's test between ISED and other methods: $\chi^2 = 21.10/5.45/1.02/39.10$, $p < 0.05/p = 0.894/p = 0.590/p < 0.01$; Fig. 4(H)).


A | ISED-learned brain trajectories of cortical systems
ABIDE
Autism Brain Imaging Data Exchange
PREPROCESSED
ASD
HC
ISED
$x \in \mathbb{R}^{T=296 \times D=116} \longrightarrow z \in \mathbb{R}^{T=296 \times d=3}$
z_3
z_1
z_2
t-start
t-end
ASD
HC
Resting-state functional MRI
ISED-learned brain trajectories


B | Evaluation framework
Temporal smoothness
Inter-subject consistency
Phenotype decodability
t_{start}
t_{end}
Less
More
Similarity
Subject #1 trajectory
Subject #2 trajectory
Linear decoding with LOOCV
HC
ASD


C | Temporal smoothness
D | Inter-subject consistency
E | Phenotype decodability
Smoothness [a.u.]
Mean: 0.230
Mean: 0.081
Embedding
Feature
Consistency (1 - normalized DTW)
Mean: 0.227
Mean: 0.072
Interpretability (LOOCV acc)
p = 0.053
Mean acc:0.597
Mean acc:0.529


F | Phenotype-wise temporal smoothness
G | Phenotype-wise inter-subject consistency
On embedding dynamics
On feature dynamics
Smoothness [a.u.]
HC
ASD
Within HC sim = 0.216
HC vs ASD sim = 0.057
Within ASD sim = 0.300
Within HC sim = 0.121
HC vs ASD sim = 0.08
Within ASD sim = 0.138
1 - normalized DTW
Embedding
Feature

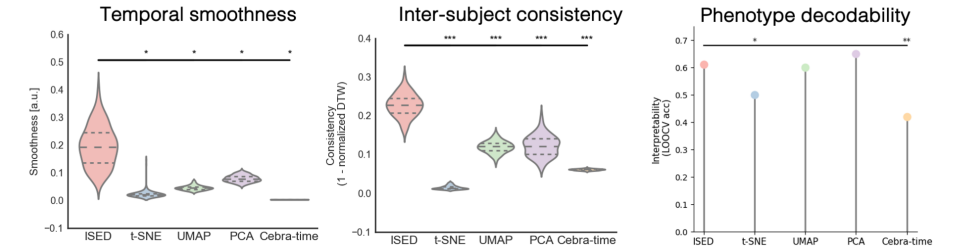

H | Embedding performance of other methods
Temporal smoothness
Inter-subject consistency
Phenotype decodability
Smoothness [a.u.]
Consistency (1 - normalized DTW)
Interpretability (LOOCV acc)
ISED
t-SNE
UMAP
PCA
Cebra-time

Fig. 4 Smoothness-enhanced brain trajectories of cortical systems in healthy and ASD individuals.
(A) Human cortical system and ISED-learned smoothness-enhanced brain trajectories. Resting-state fMRI data from the ABIDE-I database [12], including healthy controls (HC) and individuals with autism spectrum disorder (ASD), were used to generate temporally smoothed brain trajectories with ISED. Shown here are trajectories from a randomly selected ASD individual (upper) and a randomly selected HC individual (lower).
(B) Evaluation framework for brain trajectories. Each trajectory was assessed based on three metrics: temporal smoothness, inter-subject consistency, and phenotype decodability (with the added linear decoder).
(C) Temporal smoothness: ISED-learned smoothness-enhanced trajectories vs. original feature dynamics. Temporal smoothness was evaluated for ISED-learned trajectories and original feature dynamics across 117 individuals, ensuring controlled dimensionality.
(D) Inter-subject consistency: ISED-learned smoothness-enhanced trajectories vs. original feature dynamics. We assessed the similarity between all individuals using the $1 -$ normalized DTW index, comparing both the embedding (trajectories) and original feature dynamics.
(E) Phenotype decodability: ISED-learned smoothness-enhanced trajectories vs. original feature dynamics. We measured the decoding performance for phenotype classification based on both the ISED-learned trajectories and the original feature dynamics. The performance was measured using a leave-one-out cross-validation (LOOCV) scheme in phenotype classification.
(F) Phenotypic differences in temporal smoothness. Temporal smoothness was compared between HC and ASD individuals for both ISED-learned smoothness-enhanced trajectories and original dynamics.
(G) Phenotype-induced differences in inter-subject consistency. Using phenotype information, inter-subject consistency was assessed within HC, within ASD, and between HC and ASD groups, with correlation matrices representing both smoothness-enhanced trajectories and original dynamics.
(H) Benchmarking ISED against other embedding methods. ISED performance was compared with other embedding methods (t-SNE, UMAP, PCA, and Cebra-time) in terms of temporal smoothness, inter-subject consistency, and phenotype decodability. Additional trajectory visualizations from other methods are available in Supplementary Results §C and Supplementary Fig. 6.

Discussion

Low-dimensional trajectories of neural systems provide dynamic portraits that capture time-related changes, yet conventional embedding methods often yield unintuitive and uninterpretable trajectories that hinder analysis and interpretation. By upholding the temporal smoothness principle, we developed smoothness-enhanced embedding learning, embodied in the ISED learning model, which enables data-driven learning of interpretable, smooth trajectories. Extensive numerical simulations demonstrate that ISED outperforms conventional methods, not only in embedding performance, but also in enhancing temporal smoothness. When applied to real neural systems, from the micro-scale hippocampal system in behaving mice to the macro-scale cortical system in resting-state individuals with phenotypic conditions, ISED produces low-dimensional, temporally smoothed neural and brain trajectories. These smoothness-enhanced trajectories are behaviorally interpretable, consistent among subjects, and discriminative between phenotypic conditions, making them valuable for a wide range of neuroscientific investigations.

The enhanced temporal smoothness of our approach raises important questions about the role of smoothness in deriving and interpreting neural trajectories. Our smoothness-enhanced embedding learning method differs from prior smoothness-driven approaches in two key aspects. First, unlike non-learnable methods like

Savitzky-Golay filtering [36] or simple moving averages applied in a post-hoc manner [37, 38], ISED learns smoothed trajectories directly from modeled neural systems. Second, it differs from approaches that enforce adherence to predefined smoothing functions, such as Gaussian processes in GPFA [39] or auto-regressive functions in sequential autoencoders [40]. Temporal smoothing, while often an implicit part of existing embedding methods, has not previously been systematically embedded in model learning processes, as in ISED (Supplementary Notes §A and Supplementary Table 1).

What role do these non-smooth but interpretable trajectories play in understanding neural dynamics? Cebra-derived trajectories of hippocampal systems among four mice achieved competitive inter-subject consistency and behavioral interpretability compared to our smooth, ISED-derived trajectories, despite their non-smooth presentation (Extended Data Fig. 4(A)). Remarkably, these non-smooth yet interpretable trajectories showed both geometric and topological similarities to our smooth and interpretable trajectories (Extended Data Fig. 4(B)(C)), suggesting that smoothness may not be the sole factor in achieving interpretable representations. Instead, these findings indicate that trajectories can retain interpretability through alignment with behavioral data or by preserving underlying structural features, even without smooth temporal transitions.

The similarity between non-smooth, interpretable trajectories from Cebra and the smoothed trajectories produced by ISED highlights multiple viable pathways to meaningful, low-dimensional representations of complex neural systems. While ISED represents a significant advance in producing intuitive, interpretable neural trajectories, our findings indicate that temporal smoothness may not be a strict requirement for interpretability. Moreover, the effectiveness of ISED in learning smooth trajectories depends on recording duration, with shorter recordings potentially leading to suboptimal results. Future research should address this limitation and extend experimental validation across diverse neural systems to clarify whether temporal smoothness is an essential criterion for interpretable neural trajectories or a computational preference for analytic convenience.

Methods

Information-based smoothness-enhanced embedding dynamics learning model (ISED)

We propose a two-stage subsequence-based embedding learning model, **information-based smoothness-enhanced embedding dynamics learning model** (**ISED**), designed to learn smoothness-enhanced low-dimensional trajectories (embedding dynamics) to represent various high-dimensional neural systems. From this point onward, the terms "embedding dynamics" (primarily used in machine learning) and "trajectories" (primarily used in neuroscience) will be used interchangeably.

Formally, ISED learns an embedding function (f_{emb}) and a projection function (f_{pro}) to map the original D-dimensional feature dynamics x onto d-dimensional embedding dynamics (trajectories) z via a subsequence-based two-stage mapping: $x \in \mathbb{R}^{T\times D} \xrightarrow{\text{subsequence}} \tilde{x} \in \mathbb{R}^{T\times k\times D} \xrightarrow{f_{\text{emb}}} \tilde{z} \in \mathbb{R}^{T\times k\times d} \xrightarrow{f_{\text{pro}}} z \in \mathbb{R}^{T\times d}$. Here, we denote $\tilde{x}$ and $\tilde{z}$ as the feature and (to-be-learned) embedding subsequences. T and i mark the total time length and specific time index of the dynamics. k represents the uniformed subsequence length and $D \gg d$.

Since the subsequence length k directly controls the temporal smoothness of the resulting embedding dynamics, it can either be set as a user-defined hyperparameter or determined heuristically prior to implementation. For varying k values, it enables ISED to learn a range of smoothness-enhanced embedding dynamics with different degrees of temporal smoothness. Depending on the structure of the input data, feature subsequences $\tilde{x}$ can be generated from the feature dynamics x through various operations. The parameter d, which defines the dimensionality of the embedding dynamics, can be predefined using either a hypothesis-driven or data-driven approach.

Defining k in ISED implementation

In ISED implementation, adjusting the k values allows us to learn a variety of trajectories with different degrees of smoothness. A larger k tends to produce smoother trajectories but at the cost of preserving fine details of the original dynamics, whereas a smaller k retains more detailed dynamics but results in less smooth trajectories. Drawing inspiration from the commonly used heuristic for setting the time-offset τ in time-delayed embedding learning [41, 42], we set the range of k to (one quarter of) the dominant period (λ) of the original dynamics: $k \in [\frac{1}{4}(\lambda_{\text{min}}), \frac{1}{4}(\lambda_{\text{max}})]$. In practice, for a given system, we compute this range of k and apply linear extrapolation to derive multiple k values, enabling the learning of trajectories corresponding to multiple timescales.

With opted k, we offer two major ways to obtain feature subsequences ($\tilde{x} \in \mathbb{R}^{T\times k\times D}$) from the provided feature dynamics ($\tilde{x} \in \mathbb{R}^{T\times D}$) that preserve the entire time length T.

1. Sliding window approach with padding method
   With a sliding window approach, we can create T subsequences by moving the window one step at a time: $\tilde{x} = [\tilde{x}_i, \tilde{x}_{i+1}, ..., \tilde{x}_{i+k-1}]$, where i ranges from 1 to

T. This approach will result in T subsequences of length k. For the last $k-1$ subsequence, we circumvent the edge issue by padding this subsequence with zeros.

2. Circular buffer approach
   An alternative method to obtain T subsequences without requiring padding for the last subsequence is through the use of a circular buffer approach. These subsequences $\tilde{x}$ can be expressed as: $\tilde{x}_i = \left[\tilde{x}_{(i) \mod T}, \tilde{x}_{(i+1) \mod T}, \ldots, \tilde{x}_{(i+k-1) \mod T}\right]$, for $i = 1, 2, \ldots, T$, where i ranges from 1 to T. This approach ensures that all subsequences are of length k, with the subsequence indices wrapping around once T is reached. As a result, a total of T subsequences are crafted, each containing k-length timesteps, without the need for padding.

For high-dimensional time series data with many time steps, the sliding window with padding approach is preferable. However, for data with lower temporal resolution, the circular buffer approach is more suitable.

Determining d in ISED implementation

In the context of embedding learning, determining the optimal embedding dimensionality (the value of d in our ISED implementation) remains an open research question in computational neuroscience [14, 43, 44]. Some approaches [7, 45] have adopted for a cut-off value method, in which the optimal dimensionality is selected through a greedy search that captures the majority of variance in the data. Others [46, 47] have pre-set the value of d to meet specific needs, such as for 3D visualization. In ISED, the optimal d can be determined through either a data-driven approach, such as local intrinsic dimensionality estimation [48], or a hypothesis-driven approach, in which d is chosen based on requirements of downstream tasks.

ISED implementation in detail

With defined subsequence length, each ISED implementation involves two computational steps in learning smoothness-enhanced embedding dynamics (Extended Data Fig. 1):

1. Maximizing mutual information between the feature and embedding dynamics to ensure that the learned embedding dynamics retain the essential structure of the original dynamics.
2. Applying two temporal smoothing operations to encourage smooth progression of the embedding dynamics along the temporal axis.

Mutual information between feature and embedding dynamics $\mathbf{I}(\boldsymbol{x}; \boldsymbol{z})$.

Fundamentally, the primary objective of any embedding method, including ISED, is to preserve information contained in the original dynamics. From the perspective of information theory, this objective is achieved by maximizing the mutual information (I(;)) between the original dynamics x and the final, to-be-learned embedding dynamics z:

$$\max \mathrm{I}(x; z)$$
$$\text{where } z = f_{\text{pro}}\big(f_{\text{emb}}(\tilde{x})\big).$$

Here, f_{emb} denotes a learnable embedding function, i.e., $f_{\text{emb}} : \mathbb{R}^{T\times k\times D} \rightarrow \mathbb{R}^{T\times k\times d}$, which reduces the dimensionality of x. f_{pro} is a learnable projection function that projects the learned embedding subsequences $\tilde{z}$ onto the final embedding dynamics z. The choice of f_{emb} can be flexibly selected from existing off-the-shelf feature extractors, such as deep recurrent neural networks with various architectures. Similarly, the choice of f_{pro} is left to the practitioner, as long as it meets the requirement of achieving global smoothness – ensuring that the modeled embedding subsequences are predictive of one another along the temporal axis (see the later section on global smoothness information).

Local and global smoothness

Maximizing the mutual information between the feature and embedding dynamics ensures that learned embeddings closely approximate the feature dynamics, but it does not inherently control temporal smoothness. To achieve smoothness-enhanced embedding dynamics, we incorporate two temporal smoothness operations during embedding learning: local within-subsequence smoothness and global between-subsequence smoothness modules.

Local (within-subsequence) smoothness information

The first source of temporal smoothness arises from fluctuations within each learned embedding subsequence. To address this, we apply a temporal de-noising approach that reduces random fluctuations in subsequences. Given a subsequence length of k, one way to achieve this is by minimizing the Jacobian of the dynamics in embedded subsequences, $J_k(\tilde{z})$. The magnitude of this Jacobian reflects the stability of the output, i.e., the embedding subsequence, in response to changes in its internal dynamics, i.e., fluctuations in the subsequence. A large magnitude in $J_k(\tilde{z})$ suggests instability in the dynamics of $\tilde{z}$, potentially indicating chaotic behavior. To ensure smooth transitions in the dynamics and prevent abrupt changes or amplifications, a smaller magnitude is preferred. Therefore, the objective is to minimize this Jacobian term:

$$\min J_k(\tilde{z}).$$

This minimization elevates the temporal smoothness within learned embedding subsequences, ultimately leading to the production of smooth embedding dynamics.

Global (between-subsequence) smoothness information

Another factor contributing to temporal smoothness is the predictability between subsequences [49]. To address this, we incorporate a second smoothing operation, i.e., temporal prediction, to further enhance the smoothness of the embedding dynamics. In this approach, the choice of f_{pro} is refined to serve not only as a function

that condenses embedding subsequences $\tilde{z}$ into embedding dynamics z, but also as one that enforces the first-order (or higher-order) Markov property in the modeled embedding dynamics [50, 51]. This global smoothness is captured through the temporal predictive information term $I_{\text{pred}}(z)$ [52], which measures predictability of the current time-step in the embedding dynamics $(f_{\text{pro}}(z_{i'}))$ based on previous time-steps in the learned embedding dynamics $(f_{\text{pro}}(z_i))$. To further enhance smoothness in the learned embedding dynamics, we maximized this global smoothness information:

$$\begin{aligned} &\max I_{\text{pred}}\big(z\big) \\ &\text{where } z = f_{\text{pro}}(\tilde{z}), \end{aligned}$$

thereby improving the temporal smoothness in the resulting embedding dynamics.

Optimization and loss functions

Armed with three optimization objectives: $\max \mathrm{I}\big(x; z\big)$, $\min J_k\big(\tilde{z}\big)$ and $\max I_{\text{pred}}(z)$ in ISED, we unfold corresponding computable loss functions in the following sections.

$\mathrm{I}\big(x; z\big)$ *estimation*

Due to the difficulty in direct computing the mutual information $\mathrm{I}\big(x; z\big)$, which can be expressed into $\int\int p(z,x) \log \frac{p(x|z)}{p(x)} dz dx$, we approximate this mutual information term through estimating the density ratio $\frac{p(x|z)}{p(x)}$. As $p(x|z)$ can be parameterized by an added generative model (or a decoder) g, such density ratio can be estimated through by the simple mean square error between generated and given feature subsequences $||g\big(f_{\text{pro}}(f_{\text{emb}}(\tilde{x}))\big) - x||^2$. This mean square error is commonly viewed as the reconstruction error in widely used recurrent neural networks [53], generative models [54], and auto-encoder based models in time series analysis [55].

We then employ the local variational approximation technique [56] [51] to develop a loss function to train the embedding function f_{emb}, the projection function f_{pro}, and an added decoder g jointly. Given a set of T pairs of time-aligned feature dynamics and subsequences $\{\mathbb{X}, \tilde{\mathbb{X}}\} = \big\{\{x_1, \tilde{x}_1\}, ..., \{x_t, \tilde{x_t}\}\big\}$, we aim to optimize the loss:

$$\mathrm{I}\big(x; z\big) \approx -\mathbb{E}_{(x_i, \tilde{x}_i) \in (\mathbb{X}, \tilde{\mathbb{X}})} \log \Big[\sum_{i=1}^{T} ||g\big(f_{\text{pro}}(f_{\text{emb}}(\tilde{x}_i))\big) - x_i||^2 \Big]$$

Optimizing this approximated mutual information loss ensures the learnable f_{emb} and f_{pro} to capture the essence of modeled feature subsequences.

Additionally, aside from its aid in optimization, the added learnable generative model g also enables ISED to obtain the decoding capacity, allowing learned embedding dynamics to be reconstructed to recover the original high-dimensional feature dynamics for fidelity check.

$J_k(\tilde{z})$ ***computation***

By assuming the learnable embedding function f_{emb} as a recurrent neural network, we can compute the Jacobian of dynamics within the embedding subsequences $J_k(\tilde{z})$, which accounts for the volume of change in the probability space due to the learning of the embedding function.

Within k subsequence length, the Jacobian of subsequence dynamics can be further derived,

$$\begin{aligned} J_k(\tilde{z}) &= \frac{\partial \tilde{z}_k}{\partial \tilde{z}_{k-1}} \\ &= \big(\phi'(W\tilde{x})\big)W, \end{aligned}$$

where W is the trainable weight matrix in our embedding function f_{emb} (in a standard recurrent neural network that ignores the bias) and ϕ' is the derivative of the applied differentiable nonlinear transformation in f_{emb}.

As this Jacobian is computed in each embedding subsequence, it yields a $k \times k$ Jacobian matrix. Then, for a given time-aligned set of T feature subsequences $\tilde{x}_i \in \tilde{\mathbb{X}}; \tilde{\mathbb{X}} = \{\tilde{x}_1, ..., \tilde{x}_t\}$, we summarize the time-averaged (across the subsequence length k) log transformation of the determinate of the Jacobian matrix for each subsequence. This leads to the to-be-minimized final loss function for $J_k(\tilde{z})$ as:

$$\mathbb{E}_{\tilde{x}_i \in \tilde{\mathbb{X}}}\left[\sum_{i=1}^{T}\Big(\frac{1}{k}\log|\det\big[\big(\phi'(W\cdot\tilde{x}_i)\big)W\big]|\Big)\right]$$

$I_{\textit{pred}}(z)$ ***estimation***

The global smoothness information term $I_{\text{pred}}(z)$ captures the temporal predictability of learned embedding dynamics throughout the time, i.e., the capability of past embedding (with 1 time-step lag) to predict the current embedding. This information term can then be reduced to $\log\big[\frac{p(z_i|z_{i-1})}{p(z_i)}\big]$ [52], where z_i and z_{i-1} denote the current and past embeddings, respectively. As with the previous mutual information estimation, we can also use various types of positive scoring functions (i.e., a simple dot-product function) to approximate $\big[\frac{p(z_i|z_{i-1})}{p(z_i)}\big]$ [57]. We then develop the time-variant contrastive learning-based loss function as follows.

$$\begin{aligned} I_{\text{pred}}(z) &= \log\big[\frac{p(z_i|z_{i'})}{p(z_i)}\big] \\ &\approx -\mathbb{E}_{(\tilde{x}_i,\tilde{x}_{i-1},\tilde{x}_{j;j\notin\{i,i-1,\ldots,1\}})\in\tilde{\mathbb{X}}}\log\Bigg[\\ &\frac{f_{\text{pro}}(f_{\text{emb}}(\tilde{x}_i))\cdot f_{\text{pro}}(f_{\text{emb}}(\tilde{x}_{i-1}))}{\sum_{x_j\in\tilde{\mathbb{X}}} f_{\text{pro}}(f_{\text{emb}}(\tilde{x}_i))\cdot f_{\text{pro}}(f_{\text{emb}}(\tilde{x}_{j;j\notin\{i,i-1,\ldots,1\}}))}\Bigg], \end{aligned}$$

where $\tilde{\mathbb{X}}$ denotes a set of time-aligned T numbers of feature subsequences $\tilde{x}_i \in \tilde{\mathbb{X}}; \tilde{\mathbb{X}} = \{\tilde{x}_1, ..., \tilde{x}_t\}$, in which $\tilde{x}_i$, $\tilde{x}_{i-1}$, and $\tilde{x}_{j;j\notin\{i,i-1,...,1\}}$ represent the current, past, and a random picked feature subsequence that is neither past nor current, respectively.

Maximizing this loss term encourages the current embedding subsequences to be predictable from only the past embedding subsequences along the temporal axis.

Overall optimization function

Armed with above-mentioned three component-wise loss functions for ISED, given time-aligned T numbers of feature dynamics $x_i \in \mathbb{X}$ and subsequences $\tilde{x}_i \in \tilde{\mathbb{X}}$, the overall to-be-minimized loss function for our ISED learner $\mathcal{L}_{\text{ISED}}$ can be derived as,

$$\begin{aligned}\mathcal{L}_{\text{ISED}} = \mathbb{E}_{(x_i,\tilde{x}_i)\in(\mathbb{X},\tilde{\mathbb{X}})} \log \left[\sum_{i=1}^{T} \left\| g\left(f_{\text{pro}}\left(f_{\text{emb}}(\tilde{x}_i)\right)\right) - x_i \right\|^2\right] \\ + \mathbb{E}_{\tilde{x}_i\in\tilde{\mathbb{X}}} \left[\sum_{i=1}^{T} \left(\frac{1}{k} \log \left|\det\left[\left(\phi'(W\cdot\tilde{x}_i)\right) W\right]\right|\right)\right] \\ + \mathbb{E}_{(\tilde{x}_i,\tilde{x}_{i-1},\tilde{x}_{j;j\notin\{i,i-1,...,1\}})\in\tilde{\mathbb{X}}} \log \left[\frac{f_{\text{pro}}\left(f_{\text{emb}}(\tilde{x}_i)\right)\cdot f_{\text{pro}}\left(f_{\text{emb}}(\tilde{x}_{i-1})\right)}{\sum_{x_j\in\tilde{\mathbb{X}}} f_{\text{pro}}\left(f_{\text{emb}}(\tilde{x}_i)\right)\cdot f_{\text{pro}}\left(f_{\text{emb}}(\tilde{x}_{j;j\notin\{i,i-1,...,1\}})\right)}\right],\end{aligned}$$

where optimization parameters are learning weights in f_{emb}, f_{pro} and g. In practical implementation, we also add three sigmoid functions σ to bound three component losses to $[0, 1]$ to ease the optimization.

Algorithmic view

We present a step-wise implementation of ISED to provide a clear view of its algorithmic components, as detailed in Algorithm 1.

Partial ISED implementation: ISED sub-learners

Depending on the inclusion of smoothing operations in ISED, i.e., whether local and/or global smoothing operations are incorporated, we can derive three ISED sub-learners to examine their respective contributions to temporal smoothing, which will be evaluated in the later ablation study.

The ISED sub-learners are:

- ISED-I: Based solely on mutual information (loss function: $\max \mathrm{I}(x; z)$).
  This sub-learner focuses only on preserving core information of the original feature dynamics in learning embedding dynamics. Apart from the subsequence operation, no explicit temporal smoothing operations are applied. As such, we expect this ISED sub-learner to produce the least smooth embedding dynamics.

- ISED-II: Incorporates mutual information and global smoothness (loss function: $\max \mathrm{I}(x; z) + \max I_{\text{pred}}(z)$).

Algorithm 1 Step-wise implementation of ISED

Input: Feature dynamics (System) $x \in \mathbb{R}^{T \times D}$
Output: Embedding dynamics (Trajectories) $z \in \mathbb{R}^{T \times d}$
Require: An embedding function $f_{\text{emb}} : \mathbb{R}^D \to \mathbb{R}^d$
Require: A projection function $f_{\text{pro}} : \mathbb{R}^d \to \mathbb{R}^d$
Require: A generative model (decoder) $g : \mathbb{R}^d \to \mathbb{R}^D$

Initialization:
Initialize f_{emb} with input dimension D and output dimension d
Initialize f_{pro} with input dimension d and output dimension d
Initialize g with input dimension d and output dimension D
Step 1: Obtain feature subsequences
Attain feature subsequences $\tilde{x} \in \mathbb{R}^{T \times k \times D}$ from the input x with determined subsequence length k
Step 2: Encode feature subsequences
Encode subsequences using f_{emb}
$\tilde{z} \leftarrow f_{\text{emb}}(\tilde{x})$
Step 3: Compute local smoothness
Calculate the Jacobian $J(\tilde{z}) \leftarrow \frac{\partial \tilde{z}_k}{\partial \tilde{z}_{k-1}}$ for each embedding subsequence $\tilde{z}$
Step 4: Project embeddings
Project the embeddings using f_{pro}
$z \leftarrow f_{\text{pro}}(\tilde{z})$
Step 5: Calculate global smoothness
Compute the global smoothness information $I_{pred}(z) \leftarrow \frac{p(z_i|z_i')}{p(z_i)}$
Step 6: Compute mutual information
Calculate the mutual information between x and z
$I(x; z) \leftarrow \|g(f_{\text{pro}}(f_{\text{emb}}(\tilde{x}))) - x\|^2$

This sub-learner adds local smoothness by minimizing the Jacobian of the dynamics in each embedding subsequence. We hypothesize that this will result in much smoother embedding dynamics than ISED-I.

- ISED-III: mutual information and global smoothness information (loss function: $\max \mathrm{I}(x; z) + \max I_{\text{pred}}(z)$).
  This sub-learner aims to achieve smooth embedding dynamics by encouraging the subsequences to be predictive of one another along the temporal axis. We anticipate that this approach will produce smoothing effects comparable to ISED-II.

Note that sub-learners considering only local or global smoothness without mutual information were excluded due to insufficient representation power of the original feature dynamics. These ISED sub-learners were individually applied to synthetic data with varying levels of randomness. The learned embedding dynamics were evaluated based on temporal smoothness and reconstruction accuracy in both 2D latent and 100D feature spaces (Extended Data Fig. 2((C)(D)).

Conventional embedding methods

In addition to ISED, we implement a wide range of conventional embedding methods. These methods are simple linear approaches, such as the standard linear PCA, to more complex non-linear neighborhood-based methods, such as UMAP [17] and t-SNE [16]. We also include methods that effectively incorporate temporal information, such as deep recurrent neural networks with autoencoders (AE+RNN) and the Gaussian Process Latent Variable Model (GP-LVM). Furthermore, we explore recently proposed methods that are either tailored to specific neuroimaging modalities, such as PHATE and T-PHATE [46], or those that correlate with behavioral labels, such as Cebra [20] and PSID [19].

Interestingly, conventional embedding methods can be reinterpreted through our manifold-dynamics framework, with nomenclature provided in Supplementary Note §A (with attached Table 1). Key implementation details are outlined in the following sections.

To further illustrate, smoothness shapes the embedding performance of other embedding methods, we present four transformation cases: (1) from PHATE to T-PHATE, (2) Cebra-time with varying Δ sizes, (3) from PCA to PCA-smooth, and (4) from AE+RNN to its subsequence version, AE+RNN-sub. The first two cases have already documented in existing literature, while the latter two are introduced by the authors.

PCA

Conventional PCA

PCA is one of the most widely used embedding methods, valued for its simplicity and linear interpretability. For the conventional PCA implementation, we used the off-the-shelf API from `sklearn.decomposition.PCA`. The number of components, `n_components`, was fixed at 2 for simulation data and 3 for CA1 data and resting-state functional MRI data. All other hyperparameters were left at their default settings for both synthetic and real neural systems.

PCA-smooth

Conventional PCA is a domain-general embedding method that does not effectively model temporal information, as it lacks inherent temporal smoothing operations. To address this, we propose a smooth version of PCA (PCA-smooth) that incorporates an autoregressive function to introduce temporal smoothing into the model.

The standard PCA objective is to find a set of orthogonal principal components that maximize the variance of the projected data. Let $X \in \mathbb{R}^{T\times D}$ represent the centered data matrix with T time steps and D feature dimensions, and let $W \in \mathbb{R}^{D\times d}$ be the transformation matrix containing the d principal components. The conventional PCA objective can be expressed as:

$$\min_{W} -\mathrm{Tr}(W^T X^T X W),$$

where $\mathrm{Tr}(\cdot)$ represents the trace of a matrix.

To incorporate the principle of temporal smoothness, similar to the global smoothness used in ISED, we introduce an autoregressive (AR) function into the PCA framework. Specifically, we apply a first-order autoregressive function (AR(1)), defined as $Z_{i,t} = \phi_{i,0} + \phi_{i,1} Z_{i,t-1}$, where $Z_i = XW_i$ denotes the i-th PCA-transformed component at time t, and $\phi_{i,0}$ and $\phi_{i,1}$ are the autoregressive coefficients. This AR(1) function enhances the smoothness of the learned embedding dynamics. The overall optimization objective of PCA-smooth is then given by,

$$\min_{W,\phi} -\mathrm{Tr}(W^T X^T X W) + \sum_{i=1}^{k} \sum_{t=2}^{n} (Z_{i,t} - \phi_{i,0} - \phi_{i,1} Z_{i,t-1})^2.$$

Thus, PCA-smooth can be viewed as a modified version of PCA that incorporates the principle of temporal smoothness.

Implementation of PCA-smooth with varying AR orders

We implemented both conventional PCA and PCA-smooth on the synthetic system (data). To explore the effect of varying levels of smoothing on the learned embedding dynamics, we derived multiple versions of PCA-smooth by adjusting the AR order. This approach generated a range of PCA-Smooth models, each producing different embedding dynamics. For simplicity, we varied the AR order from 0 (equivalent to conventional PCA) to 100, creating an array of models that were evaluated on both smoothness and reconstruction accuracy in the 2D latent space. The implementation code for PCA-smooth is available at https://github.com/LeonBai/ISED.

t-SNE

t-SNE is a widely known non-linear embedding method, but it is more accurately described as a visualization tool rather than a learning method, as its learning and generalization are restricted to the same dataset. Additionally, since t-SNE is a domain-general embedding method, it neither leverages temporal information effectively nor includes any smoothing operations.

For the implementation of t-SNE on both simulation and real systems, we used the `sklearn.manifold.TSNE` API. The `n_components` was set to 2 or 3, `perplexity` to 10, and `n_iter` to 10,000. All other hyperparameters were kept at their default settings.

UMAP

UMAP is another well-known non-linear based embedding method that projects data onto the Non-Euclidean embedding space. Differ to the non-generalizable t-SNE, UMAP does allow for separate learning and testing stages on different data. For UMAP implementation on both synthetic and real systems, we used off-the-shelf API from `umap-learn` python library. We fixed the `n_components` to 2 or 3, respectively, while all other parameters were kept at their default settings.

GP-LVM

Unlike the previous three domain-general embedding methods, state-space models like GP-LVM effectively use time information in learning embedding dynamics and leverage Gaussian processes to model latent dynamics.

For embedding approaches that assume latent dynamics follow a correlation-invariant Gaussian process, we implemented the well-known GP-LVM (Gaussian Process Latent Variable Model) [58], which is suitable for any type of continuous time series (as opposed to GPFA [39], which is specific to spike train data). In essence, GP-LVM is a state-space model that assumes that observed data points are generated from a low-dimensional latent space via a non-linear mapping function parameterized by Gaussian processes. Since the Gaussian process naturally induces a temporal smoothing effect [59], GP-LVM can be regarded as an embedding method with built-in temporal smoothing.

For practical implementation, we used the off-the-shelf `GPy` library (https://gpy.readthedocs.io/en/deploy/) to instantiate GP-LVM for learning low-dimensional embedding dynamics from both simulation and real datasets.

AE+RNN

RNNs, with their numerous variations, are widely used for learning embedding dynamics in many scientific fields due to their strong representational capacity. In this study, we add an autoencoder (AE) on top of the RNN, allowing us to compute the reconstruction error, i.e., the decoding loss between the original and reconstructed dynamics.

For implementation, we used the `tensorflow` Python library to instantiate the AE+RNN model. Detailed model structures and layer-wise configurations can be found at https://github.com/LeonBai/ISED.

AE+RNN-sub

As demonstrated in [51], introducing subsequences in AE+RNN induces a smoothing effect on the reconstructed time series, leading to development of AE+RNN-sub. By receiving subsequences as input, the subsequence length k in $x \in \mathbb{R}^{T \times k \times D}$ becomes a key parameter related to the smoothing effect. Different values of k are expected to induce varying degrees of smoothness in the learned embedding dynamics. To explore this, we implemented various AE+RNN-sub models with k ranging from 5 to 150 for modeling the synthetic data.

For AE+RNN-sub implementation, we also used the `tensorflow` library to build multiple AE+RNN-sub models with varying k values and custom layer-wise configurations.

PHATE

A novel embedding-based visualization method, PHATE, utilizes both local and global data structures to map high-dimensional neural data onto a low-dimensional embedding manifold. Compared to its predecessor, Multi-Dimensional Scaling (MDS),

PHATE excels at preserving data continuity by maintaining the distance between data points in both local and global contexts. However, as a visualization tool rather than a learning-based method, the learned embedding function of PHATE cannot be generalized to new datasets.

For implementation, we used the released API (https://phate.readthedocs.io/en/stable/api.html) to instantiate PHATE models for both simulation and real systems.

T-PHATE

T-PHATE (the temporal potential of heat-diffusion for affinity-based transition embedding method) is a recently proposed modification to PHATE that incorporates both MDS-like distance preserving characters and data's auto-correlative structure [46]. A critical temporal smoothing related operation in deriving T-PHATE embedding dynamics is the computation of the kernel $D(x_i, x_j)$ (where i, j is the time index to indicate the sample in i and j time step) with rolling average over ω numbers of time points, i.e., $\frac{1}{\omega}\Sigma_{j=1}^{i+w-1} D(x_i, x_j)$. Therefore, we refer this ω as the adjustable smoothing parameter that exert direct control on the smoothness of learned embedding dynamics. For this, we choose the wide rang of ω sizes from 1 to 500 to produce a series of embedding dynamics, and evaluated their smoothness and reconstruction accuracy on the 2D latent space, respectively. However, similar to previous PHATE, T-PHATE is primarily a visualization tool rather than a learning-based embedding method.

T-PHATE is specifically designed to uncover low-dimensional dynamics from MRI time series by incorporating dual-view diffusion operators, which preserve both pairwise distances between samples and the data's autocorrelation function [46]. For practical implementation, we used the open-source Python package for T-PHATE (https://github.com/KrishnaswamyLab/TPHATE).

PSID

The preferential subspace identification Method (PSID) aims to identify behavior-related (continuous) embedding dynamics from recorded data by leveraging linear SVD and predictive modeling to find behavior-relevant dimensions [19]. We implemented the off-the-shelf PSID algorithm (https://github.com/ShanechiLab/PyPSID/blob/main/source/PSID) on CA1 data to cross-check whether our smoothness-enhanced embedding dynamics align with the PSID-derived behavior-related latents.

For implementation, we set both `nx` and `n1` to 10 to ensure the same dimensionality as our smooth embedding dynamics. The input data for neural activity (`y`) and behavior (`z`) consisted of mouse-specific CA1 memory cell activities and exploration behavior, respectively.

Cebra

Cebra stands for the consistent embeddings of high-dimensional recordings using auxiliary variables. It is a recently proposed contrastive learning-based embedding method [20]. Depending on its reliance on external labels, such as correlated continuous or discrete behavior outputs, Cebra comes in two versions: Cebra-behavior and Cebra-time. In this study, we opted for Cebra-time due to its lower dependency on behavior

labels. Cebra-time uses only time information, not behavior labels, in learning embedding dynamics. The essential step to construct the contrastive samples for learning is defined as: $x^+ \sim D = \{t \in [T], \tau \in \Delta : x_{t+\tau}\}$, where Δ marks the length of time windows. Thus, the varying value of Δ should leads to different smoothing effects onto learned neural embedding dynamics. We then refer Δ as the adjustable smoothing parameter in Cebra-time. In a similar vein to the implementation of T-PHATE with different choices for its smoothing parameters, we also implement Cebra-time with a wide range of Δ from 1 to 250 to producing a panoply of embedding dynamics. These Cebra-time embedding dynamics were also evaluated on our smoothness and reconstruction accuracy metrics.

For implementation, we used the publicly available Python API from https://cebra.ai/docs/api to run the Cebra-time algorithm, using the default `offset-10` model and setting the optimization steps to 1000 for both simulation and real data.

Data

Synthetic system with a ground-truth 2D latent.

We crafted multiple synthetic high-dimensional (neural) systems with defined 2D latent (embedding dynamics) as follows. We started with producing the true 2D latent dynamics. Let time steps set to 1000, i.e., $t = 1000$, we generate a 2D time series T with added noise as:

$$\begin{aligned} T(t) &= S(t) + \epsilon \\ S(t) &= \begin{bmatrix} \sin(t) \\ \cos(t) \end{bmatrix} \\ \epsilon &\sim \mathcal{N}(0, 1-\alpha), \end{aligned}$$

where $T(t)$ is a 2D vector each time step, with the first dimension being $\sin(t) + \epsilon$ and the second dimension $\cos(t) + \epsilon$. Under this set-up, with tuned hyper-parameter $\alpha, \alpha \in (0, 1)$, we can generate 2D time series with various degree of randomness [60]. Higher value of α corresponding to decreased degree of randomness in generated 2D latents. For synthetic systems, we set the values of α to 0.8/0.5/0.2, it allows us to craft three sets of 2D ground-truth latent with controlled level of noise.

Armed with the crafted ground-truth 2D latent, we then applied randomly initialized RealNVP [34] to map these 2D latents onto 100D feature space (see Supplementary Notes §D for detailed implementation of RealNVP). These generated high-dimensional time series $x \in \mathbb{R}^{T=1000 \times D=100}$ were treated a synthetic (neural) system, and the inputted feature dynamics for various embedding learning methods. These feature dynamics were then split into training (50%) and testing (50%) sets for fair performance evaluation. The identical training set was utilized to train various embedding methods. Qualitative and quantitative evaluations of learned embedding dynamics were only conducted on the test set for each embedding method, except non-learnable embedding methods, such as, t-SNE, PHATE and T-PHATE.

Mouse hippocampal system from multi-cellular electrophysiological data.

The first real high-dimensional neural system comes from well-researched CA1 (hippocampus) memory cell recordings [33, 61], involving four rats running on a 1.6-meter linear track with rewards placed at both ends. For convenience, we only include a single recording session for four mice here. Neural activity traces in the CA1 area of the hippocampus were recorded, involving approximately 48 to 120 putative pyramidal neurons. We allowed packaged preprocessing steps to bin neural activity into 25-ms time intervals (no additional preprocessing steps were conducted so as to align with pre-processed data used in [20] [18]). For behavior information, the exploration positions (in the form of meters: $[m]$) were utilized for later probing the behavior interpretability of the learned embedding dynamics.

Human cortical system from resting-state functional MRI data.

We obtained the resting-state functional MRI data from the Autism Brain Imaging Data Exchange (ABIDE-I) database [12]. To reduce the multi-site scanning bias, we adopted the data from the scanning site with the longest scanning duration, i.e., University of Michigan (UM_1 in its coding convention). This led to formation of resting-state functional MRI data (N = 117; healthy control /ASD individuals: 74/43) to represent the human brain dynamical system at rest.

Since preprocessed resting-state functional MRI data from the ABIDE-I database are available, we then directly used these preprocessed data. We did not conduct any additional preprocessing on ABIDE-I functional MRI data. Each 3T scanned subject-wise data contains 296 volumes with 2000ms repetition time (TR), and 3-mm isotropic voxel dimensions with nuisance signals regressed out. We applied AAL parcellation atlas (with 116 defined cortical regions) [62] to extract the RoI-wise ABIDE time series to serve as the input feature dynamics, i.e., $x \in \Re^{T=296 \times D=116}$.

Analysis

Reporting guidelines for conducted analyses and statistical tests

We report each analysis alongside its statistical test(s) in a hyphen-connected format, e.g., `Temporal smoothness analysis − Paired t-test` indicates an assessment on temporal smoothness of two groups of trajectories and then tested on whether there is statistical significantly difference between these two groups. These analyses are highlighted in Table 2 and are unfolded in detail in following segments. For $p > 0.05$, we report the raw values, for $p < 0.05/0.01/0.001$, we indicate significance ranges as $*/**/***$ on corresponding plots.

Temporal smoothness

The core evaluation measures the temporal smoothness of learned embedding dynamics. Here, for all data, we utilize the cumulative sum over the 1st-order differences across time to quantify smoothness of a dynamic system. Considering learned embedding dynamics as z with t time length, the inverse of the cumulative sum can be computed as: $\frac{1}{\sum_{j=1}^{t-1}(||\Delta z_j||)}$, where $\Delta z_j = z_{j+1} - z_j$ for $j = 1, 2, ..., t-1$. Since this

Table 2 Summary of analyses.

Analysis	Description
Universal for all systems	
Temporal smoothness	1st order difference across embedding time length; higher values indicate smoother embedding dynamics. Higher values are preferred.
Synthetic system	
Reconstruction accuracy (2D latent)	R^2 scores between learned and ground-truth embedding dynamics. Higher values are preferred.
Reconstruction accuracy (100D feature)	R^2 scores between reconstructed (either through adding the decoding function or inherited inverse transformation) and input 100D feature dynamics. Higher values are preferred.
Mouse hippocampal system	
Intra-mouse behavior interpretability	Linear predictability R^2 of subject-wise learned embedding dynamics and its associated continuous behavior (or other external) measures. Higher values are preferred.
Behavior-aligned inter-mouse consistency	Linear predictability R^2 between pairs of mouse-wise learned embedding dynamics. Higher values are preferred.
Similarity between ISED and PSID-derived trajectories	Linear cosine similarity measure between pairs of mouse-wise trajectories learned from our ISED and PSID method [19].
Inter-mouse representation similarity	Mantel test assessed (with 10k permutation) similarity between the embedding distance matrix and the behavior distance matrix across four mice. Higher values are preferred.
Rolling variation (Supplementary Results §B)	(Temporally) fluctuation pattern of learned embedding dynamics.
Recurrence (Supplementary Results §B)	Recurrent spatial-temporal patterns (recurrence plot) with computed recurrence rate (RR) and determinism (DET) values. Higher values are preferred.
Human cortical system	
Inter-subject consistency	$1 -$ Normalized DTW distance ($[0, 1]$) between feature or learned embedding dynamics subjects. DTW: dynamic time warping. Higher values are preferred.
Phenotype decoding	Leave-one-out cross-validation (LOOCV) on decoding performance utilizing either feature or embedding dynamics in classifying HC and ASD individuals. Higher values are preferred.

metric captures the (inverse) cumulative squared changes between each pair of consecutive pointes of a dynamic system [63], it can denote local variation of the time series. Thus, a resulting high value in this smoothness metric denotes a high degree of smoothness. This measure on temporal smoothness bears the mathematical similarity to the concept of slowness [64].

Reconstruction accuracies on 2D latent and 100D feature dynamics

For evaluating the reconstruction accuracy of learned embedding dynamics, i.e., whether the true 2D latent and generated 100D feature dynamics were recovered, we measured the linear fitting performance between the learned and true 2D latents in terms of their respective reconstruction R^2 scores on both latents and features. Importantly, as some embedding approaches are not instantiated with innate decoders (or generative functions), e.g., PHATE, T-PHATE, Cebra-time, and t-SNE, their reconstruction R^2 on feature was not assessed for fairness in comparison.

Intra-mouse behavior interpretability

When it comes to evaluation of learned trajectories (embedding dynamics) of real neural systems, the most important criterion is to assess their behavior interpretability. In CA1 memory cell activities of four mice, we chiefly evaluate whether learned trajectories can be explained by mouse-wise explorative behavior. This intra-mouse behavior interpretability is measured independently for each mouse by fitting a linear regression model with training-testing ratio set to 80/20 to evaluate the fitting performance of learned trajectories (indicated as R^2 value).

Behavior-aligned inter-mouse consistency

To quantify the embedding (trajectory) consistency between four mice in CA1 memory cell data, we follow the alignment and down-sampling procedure in [20], in which we first align the learned embeddings with mouse-wise behavior, and down-sampled to the same length ($T = 100$) for all mice for comparative purposes.We then adopt a linear predictability metric to assess inter-mouse embedding consistency. This approach involves fitting a linear regression model with the training-testing ratio set to 80/20, for each pair of embeddings to assess their linear relationship. By calculating R^2 for each pair, excluding self-comparisons, we generate a matrix that quantifies the cross-predictive accuracy of embeddings

Similarity between ISED and PSID-derived trajectories

To assess similarity between ISED-learned neural trajectories and behaviorally relevant subspaces identified by PSID, we compared ISED-learned smoothness-enhanced trajectories from CA1 data with PSID-identified behavior-related subspaces. As shown in Extended Data Fig. 3(A), despite methodological differences, PSID being a supervised embedding approach that leverages continuous behavioral information, e.g., exploratory distance, to guide subspace identification, and ISED emphasizing generation of smoothness-enhanced neural trajectories for each mouse, embeddings derived by ISED showed a high degree of similarity to those produced by PSID.

Inter-mouse representation similarity

To relate inter-mouse consistency with behavioral differences among mice, we conducted an inter-mouse representational similarity analysis (RSA) [35] to explore whether similarity in embedding dynamics among mice corresponds to their respective behavioral similarity. Specifically, we computed (normalized) dynamic time warping

(DTW) distance matrices from both embedding dynamics and behavior data, i.e., the continuous exploratory patterns. A Mantel test with 10,000 permutations was used to calculate the inter-subject representational similarity, measured as a correlation coefficient (r) between brain and behavior data.

As illustrated in Extended Data Fig. 3(B), we applied the inter-mouse RSA to compare inter-mouse brain signature differences and behavior differences. DTW distance was used to derive the matrices, which capture inter-mouse brain and behavior differences. The analysis was performed separately for both original trajectories (feature dynamics) and ISED-learned trajectories (embedding dynamics), providing insight into the relationship between brain dynamics and behavior across subjects.

Inter-subject consistency

Unlike the foregoing behavior-aligned inter-mouse consistency analysis for CA1 data, for resting-state functional MRI data from the ABIDE-I database, there is no need for behavior alignment prior to consistency computation. Here, to capture the possibly non-linear relationship between individuals in terms of their original features or learned trajectories, we derived a dynamic time warping (DTW)-based consistency metric: $1-$normalized DTW, to evaluate the inter-subject consistency across individuals with and without phenotypic grouping.

Phenotype decoding

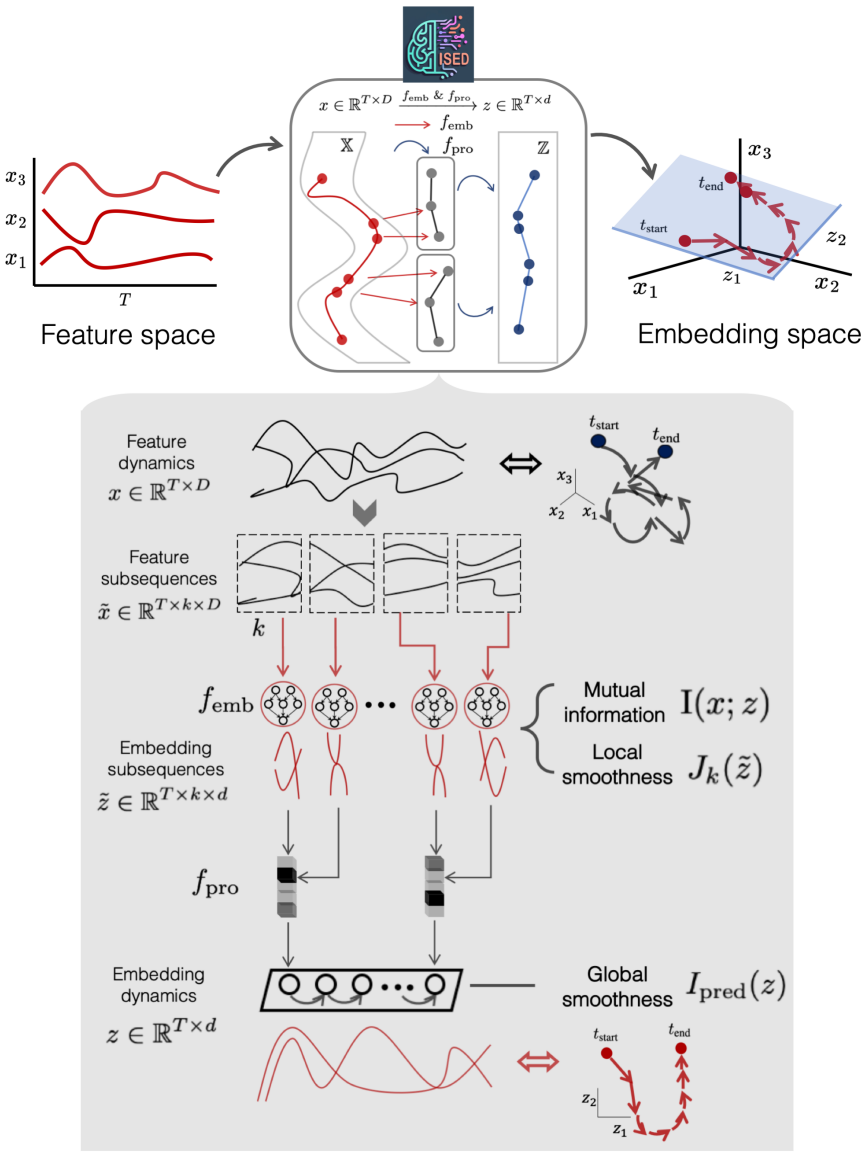
To evaluate the discriminative capacity of the original features and learned embedding dynamics in distinguishing individuals with autism spectrum disorder (ASD) from healthy controls (HC), we conduct a decoding analysis. Specifically, we assessed whether the original features or embedding dynamics provided superior information for classification, and further, to determine which aspects of embedding dynamics are most effective for this task. To ensure a fair comparison, we controlled for identical dimensionality across feature sets, maintaining an equal number of discriminative features available for classification. Decoding was performed using a linear classifier, specifically logistic regression, a widely used method for binary classification. Both the original feature set and learned embedding dynamics served as input features for the decoding analysis, with the classification task being evaluated using leave-one-out cross-validation (LOOCV). This rigorous validation framework systematically trained the model on all samples except one, using the withheld sample for testing, and repeated the process for every individual in the dataset. By applying LOOCV, we robustly compared the classification performance of logistic regression models trained on different feature representations.

Declarations

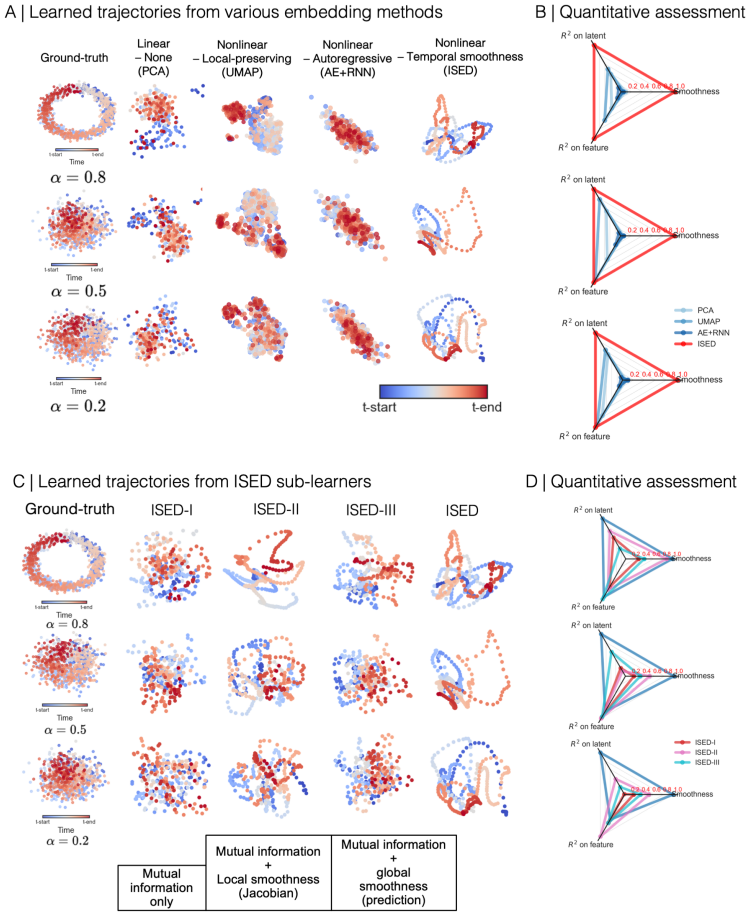
Supplementary Information. Extended Data Fig. 1 – 4; Supplementary Notes §A – §D; Supplementary Results §A – §C; Supplementary Fig. 1 – 7 and Supplementary Table 1 in Supplementary Results.

Data Availability. Simulation data: The complete simulation dataset, including 2D latent and 100D feature dynamics, is available for download at https://github.com/LeonBai/ISED. CA1 hippocampal neuron data: Accessible from the CRCNS repository at https://crcns.org/data-sets/hc/hc-11/about-hc-11. Packaged and preprocessed data are provided via the `Cebra` Python library [20]: https://github.com/AdaptiveMotorControlLab/CEBRA/blob/main/cebra/data/datasets.py. Resting-state functional MRI data: Sourced from the ABIDE-I database, available at http://preprocessed-connectomes-project.org/abide/.

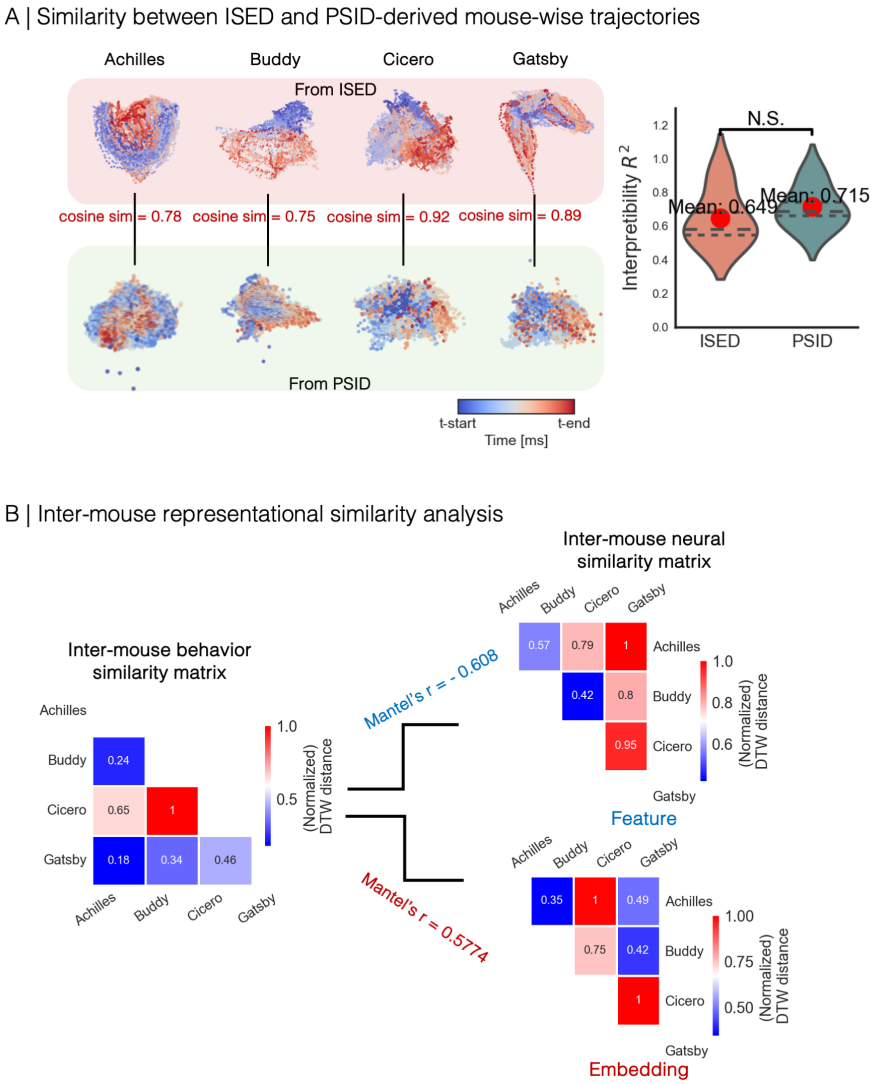
Code Availability. Complete Python code for implementing ISED and other embedding methods on simulation data is publicly available at https://github.com/LeonBai/ISED for academic use. All analyses listed in Table 2 are included in the repository. Additionally, the `ISED` package can be installed via `pip install ISED_learner`. Other materials and analyses are available from the corresponding author upon request.



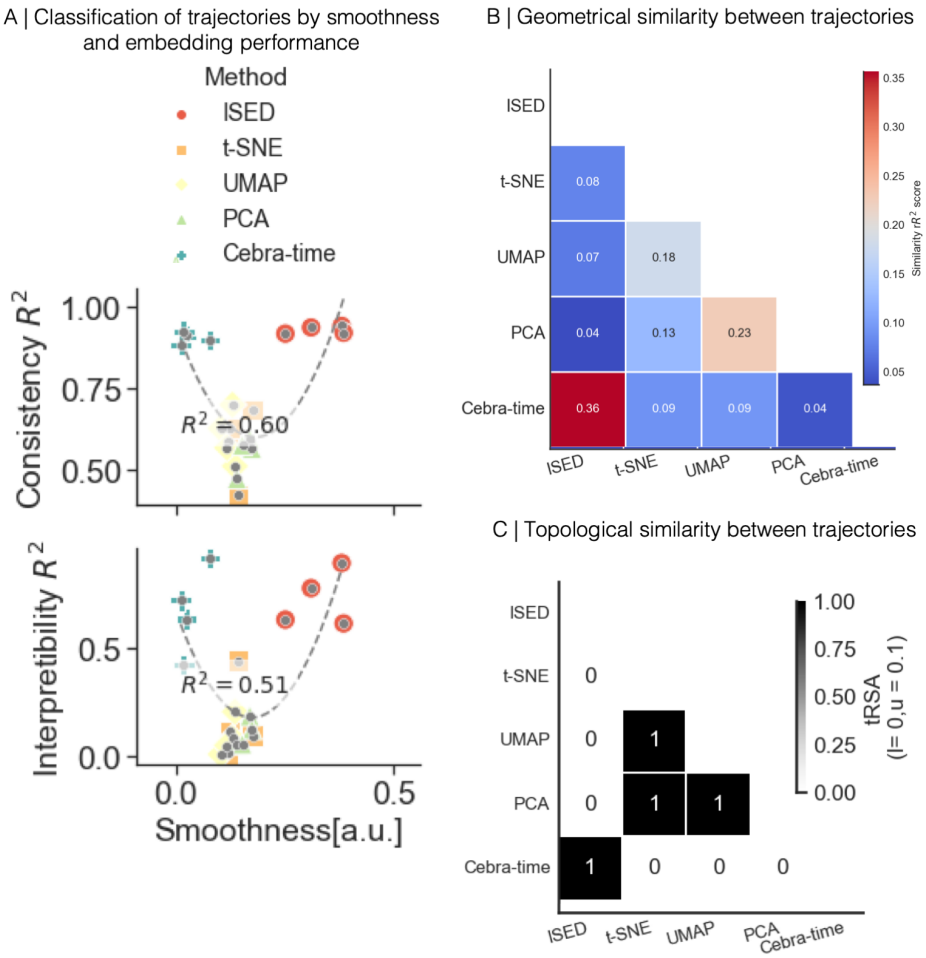
Extended Data Fig. 1 Schematic of ISED.
ISED is a subsequence-based, two-stage embedding method designed to capture temporally smooth trajectories. First, input feature dynamics x are segmented into subsequences $\tilde{x}$ (left, "Feature space" panel). These subsequences are then embedded into a low-dimensional space via a learnable embedding function f_{emb}, producing embedding subsequences $\tilde{z}$ (middle panel, "Embedding space"). In the second stage, $\tilde{z}$ is projected and summarized into the final embedding dynamics z using a learnable projection function f_{pro}. To capture the essence of the original dynamics, ISED maximizes the mutual information between the embedding z and the feature dynamics x ($\mathrm{I}(x; z)$). Temporal smoothness is achieved by minimizing the Jacobian of the embedding subsequences for local smoothness ($J_k(\tilde{z})$) and by enhancing temporal predictability for global smoothness ($I_{\text{pred}}(z)$). The combined loss function integrates $\mathrm{I}(x; z)$, $J_k(\tilde{z})$, and $I_{\text{pred}}(z)$, guiding the learning of f_{emb} and f_{pro} on the original data. Once trained, these functions are applied to generate temporally smoothed trajectories or to validate performance on new data.



Extended Data Fig. 2 Evaluation of ISED (and its sub-learners) and other embedding methods on synthetic systems with various noise levels.
(A) Learned trajectories from various embedding methods. This panel shows the learned 2D trajectories from different embedding methods, including PCA, UMAP, AE+RNN, and ISED, using synthetic input dynamics with varying levels of randomness ($\alpha = 0.8, 0.5, 0.2$). The scatter plots are color-coded by time (from blue at t−start to red at t−end), illustrating temporal progression in each learned trajectory.
(B) Quantitative assessment of embedding methods. We quantitatively evaluated each method on three metrics: temporal smoothness, R^2 on the latent space, and R^2 on feature dynamics. The results are displayed in triangular spider plots, with normalized values for each metric. This comparison highlights the performance of ISED in smoothness and reconstruction accuracy, compared to PCA, UMAP, and AE+RNN.
(C) Learned trajectories from ISED sub-Learners. This panel displays trajectories generated by three variations of ISED (ISED-I, ISED-II, ISED-III) and the original ISED, using the same synthetic input dynamics. Each variation applies a different smoothness criterion: ISED-I uses mutual information only, ISED-II adds local smoothness (Jacobian), ISED-III incorporates global smoothness (prediction), and ISED combines all these criteria. Scatter plots are color-coded by time for visualizing temporal progression.
(D) Quantitative assessment of ISED variations. The four ISED variations were evaluated for temporal smoothness, R^2 on latent dynamics, and R^2 on feature dynamics. The triangular spider plots show normalized performance across these metrics, highlighting the contributions of each smoothness criterion in enhancing trajectory smoothness and embedding fidelity.



Extended Data Fig. 3 Smoothness-enhanced neural trajectories are behaviorally relevant.
(A) Comparison of ISED and PSID-derived neural trajectories. The left panel illustrates the similarity between ISED-learned smoothness-enhanced neural trajectories and PSID-identified behavior-related subspaces in CA1 data (see Methods). For each mouse, ISED trajectories align closely with PSID-derived embeddings, underscoring the impact of temporal smoothness in embedding learning. Unlike ISED, PSID is a supervised approach guided by continuous behavioral data. Viewing angles of both ISED and PSID trajectories are optimized for clarity. We observed no significant differences between ISED-derived and PSID-derived trajectories with respect to behavioral interpretability (the right panel).
(B) Inter-mouse representational similarity analysis. Extending inter-subject representational similarity analysis [35] to an inter-mouse framework, this panel assesses the relationship between behavioral and brain signature differences across mice. Mantel's correlation coefficient is used to compare two distance matrices, with DTW distance applied to quantify inter-mouse differences in both brain and behavior. Analyses were performed separately for original (feature dynamics) and ISED-learned trajectories (embedding dynamics).



Extended Data Fig. 4 Geometric and topological similarity among smooth and non-smooth trajectories.
(A) Classification of trajectories by smoothness and embedding performance in CA1 data. Trajectories derived from various embedding methods applied to CA1 hippocampal recordings are classified based on their smoothness and interpretability. The methods produce three main groups: non-smooth but interpretable (Cebra-time), non-smooth and non-interpretable (PCA, UMAP, and t-SNE), and smooth and interpretable (ISED). A non-linear relationship is observed between trajectory smoothness and embedding performance, highlighting distinct performance patterns across methods.
(B) Geometric similarity between trajectories. Representational similarity analysis (RSA) [65] with linear interpretability was used to evaluate the geometric similarity between trajectories. Trajectories in the smooth and interpretable and non-smooth and interpretable groups demonstrate high geometric similarity, whereas those within the non-smoothnon-interpretable group are more similar to each other.
(C) Topological similarity between trajectories. Topological similarity was assessed using topology-based representational similarity analysis (tRSA) [66], with parameters $l = 0$ and $u = 0.1$. The results parallel those in (B), showing that smooth-interpretable and non-smooth and interpretable trajectories share topological similarities, while those in the non-smooth and non-interpretable group exhibit topological similarity among themselves.

References

[1] Buzsaki G. Large-scale recording of neuronal ensembles. Nature neuroscience. 2004;7(5):446–451.

[2] Hamel EJ, Grewe BF, Parker JG, Schnitzer MJ. Cellular level brain imaging in behaving mammals: an engineering approach. Neuron. 2015;86(1):140–159.

[3] Pesaran B, Vinck M, Einevoll GT, Sirota A, Fries P, Siegel M, et al. Investigating large-scale brain dynamics using field potential recordings: analysis and interpretation. Nature neuroscience. 2018;21(7):903–919.

[4] Paulk AC, Kfir Y, Khanna AR, Mustroph ML, Trautmann EM, Soper DJ, et al. Large-scale neural recordings with single neuron resolution using Neuropixels probes in human cortex. Nature Neuroscience. 2022;25(2):252–263.

[5] Zhao S, Tang X, Tian W, Partarrieu S, Liu R, Shen H, et al. Tracking neural activity from the same cells during the entire adult life of mice. Nature neuroscience. 2023;26(4):696–710.

[6] Stevenson IH, Kording KP. How advances in neural recording affect data analysis. Nature neuroscience. 2011;14(2):139–142.

[7] Churchland MM, Cunningham JP, Kaufman MT, Foster JD, Nuyujukian P, Ryu SI, et al. Neural population dynamics during reaching. Nature. 2012;487(7405):51–56.

[8] Cunningham JP, Yu BM. Dimensionality reduction for large-scale neural recordings. Nature neuroscience. 2014;17(11):1500–1509.

[9] Nieh EH, Schottdorf M, Freeman NW, Low RJ, Lewallen S, Koay SA, et al. Geometry of abstract learned knowledge in the hippocampus. Nature. 2021;595(7865):80–84.

[10] Duncker L, Sahani M. Dynamics on the manifold: Identifying computational dynamical activity from neural population recordings. Current opinion in neurobiology. 2021;70:163–170.

[11] Jaffe PI, Poldrack RA, Schafer RJ, Bissett PG. Modelling human behaviour in cognitive tasks with latent dynamical systems. Nature Human Behaviour. 2023;p. 1–15.

[12] Di Martino A, Yan CG, Li Q, Denio E, Castellanos FX, Alaerts K, et al. The autism brain imaging data exchange: towards a large-scale evaluation of the intrinsic brain architecture in autism. Molecular psychiatry. 2014;19(6):659–667.

[13] Funamizu A, Kuhn B, Doya K. Neural substrate of dynamic Bayesian inference in the cerebral cortex. Nature neuroscience. 2016;19(12):1682–1689.

[14] Roads BD, Love BC. The Dimensions of dimensionality. Trends in Cognitive Sciences. 2024;.

[15] Borg I, Groenen PJ. Modern multidimensional scaling: Theory and applications. Springer Science & Business Media; 2007.

[16] Van der Maaten L, Hinton G. Visualizing data using t-SNE. Journal of machine learning research. 2008;9(11).

[17] McInnes L, Healy J, Melville J. Umap: Uniform manifold approximation and projection for dimension reduction. arXiv preprint arXiv:180203426. 2018;.

[18] Zhou D, Wei XX. Learning identifiable and interpretable latent models of high-dimensional neural activity using pi-VAE. Advances in Neural Information Processing Systems. 2020;33:7234–7247.

[19] Sani OG, Abbaspourazad H, Wong YT, Pesaran B, Shanechi MM. Modeling behaviorally relevant neural dynamics enabled by preferential subspace identification. Nature Neuroscience. 2021;24(1):140–149.

[20] Schneider S, Lee JH, Mathis MW. Learnable latent embeddings for joint behavioural and neural analysis. Nature. 2023;p. 1–9.

[21] Einhäuser W, Hipp J, Eggert J, Körner E, König P. Learning viewpoint invariant object representations using a temporal coherence principle. Biological cybernetics. 2005;93(1):79–90.

[22] Becker S, Hinton GE. Self-organizing neural network that discovers surfaces in random-dot stereograms. Nature. 1992;355(6356):161–163.

[23] Zou WY, Ng AY, Yu K. Unsupervised learning of visual invariance with temporal coherence. In: NIPS 2011 workshop on deep learning and unsupervised feature learning. vol. 3; 2011. .

[24] Bengio Y, Courville A, Vincent P. Representation learning: A review and new perspectives. IEEE transactions on pattern analysis and machine intelligence. 2013;35(8):1798–1828.

[25] Churchland MM, Yu BM, Cunningham JP, Sugrue LP, Cohen MR, Corrado GS, et al. Stimulus onset quenches neural variability: a widespread cortical phenomenon. Nature neuroscience. 2010;13(3):369–378.

[26] Ganguli S, Sompolinsky H. Compressed sensing, sparsity, and dimensionality in neuronal information processing and data analysis. Annual review of neuroscience. 2012;35:485–508.

[27] Gao P, Ganguli S. On simplicity and complexity in the brave new world of large-scale neuroscience. Current opinion in neurobiology. 2015;32:148–155.

[28] Trautmann EM, Stavisky SD, Lahiri S, Ames KC, Kaufman MT, O'Shea DJ, et al. Accurate estimation of neural population dynamics without spike sorting. Neuron. 2019;103(2):292–308.

[29] Saxe AM, McClelland JL, Ganguli S. A mathematical theory of semantic development in deep neural networks. Proceedings of the National Academy of Sciences. 2019;116(23):11537–11546.

[30] Fontenele AJ, Sooter JS, Norman VK, Gautam SH, Shew WL. Low-dimensional criticality embedded in high-dimensional awake brain dynamics. Science Advances. 2024;10(17):eadj9303.

[31] Krakauer JW, Ghazanfar AA, Gomez-Marin A, MacIver MA, Poeppel D. Neuroscience needs behavior: correcting a reductionist bias. Neuron. 2017;93(3):480–490.

[32] Urai AE, Doiron B, Leifer AM, Churchland AK. Large-scale neural recordings call for new insights to link brain and behavior. Nature neuroscience. 2022;25(1):11–19.

[33] Grosmark A, Long J, Buzsáki G. Recordings from hippocampal area CA1, PRE, during and POST novel spatial learning. CRCNS org. 2016;10:K0862DC5.

[34] Dinh L, Sohl-Dickstein J, Bengio S. Density estimation using Real NVP. In: International Conference on Learning Representations; 2016. .

[35] Finn ES, Glerean E, Khojandi AY, Nielson D, Molfese PJ, Handwerker DA, et al. Idiosynchrony: From shared responses to individual differences during naturalistic neuroimaging. NeuroImage. 2020;215:116828.

[36] Press WH, Teukolsky SA. Savitzky-Golay smoothing filters. Computers in Physics. 1990;4(6):669–672.

[37] Box GE, Jenkins GM, Reinsel GC, Ljung GM. Time series analysis: forecasting and control. John Wiley & Sons; 2015.

[38] Chatfield C, Xing H. The analysis of time series: an introduction with R. Chapman and hall/CRC; 2019.

[39] Yu BM, Cunningham JP, Santhanam G, Ryu S, Shenoy KV, Sahani M. Gaussian-process factor analysis for low-dimensional single-trial analysis of neural population activity. Advances in neural information processing systems. 2008;21.

[40] Pandarinath C, O'Shea DJ, Collins J, Jozefowicz R, Stavisky SD, Kao JC, et al. Inferring single-trial neural population dynamics using sequential auto-encoders. Nature methods. 2018;15(10):805–815.

[41] Judd K, Mees A. Embedding as a modeling problem. Physica D: Nonlinear Phenomena. 1998;120(3-4):273–286.

[42] Tan E, Algar S, Corrêa D, Small M, Stemler T, Walker D. Selecting embedding delays: An overview of embedding techniques and a new method using persistent homology. Chaos: An Interdisciplinary Journal of Nonlinear Science. 2023;33(3).

[43] Humphries MD. Strong and weak principles of neural dimension reduction. arXiv preprint arXiv:201108088. 2020;.

[44] Jazayeri M, Ostojic S. Interpreting neural computations by examining intrinsic and embedding dimensionality of neural activity. Current opinion in neurobiology. 2021;70:113–120.

[45] Safaie M, Chang JC, Park J, Miller LE, Dudman JT, Perich MG, et al. Preserved neural dynamics across animals performing similar behaviour. Nature. 2023;623(7988):765–771.

[46] Busch EL, Huang J, Benz A, Wallenstein T, Lajoie G, Wolf G, et al. Multi-view manifold learning of human brain-state trajectories. Nature computational science. 2023;3(3):240–253.

[47] Zangrossi A, Cona G, Celli M, Zorzi M, Corbetta M. Visual exploration dynamics are low-dimensional and driven by intrinsic factors. Communications Biology. 2021;4(1):1100.

[48] Carter KM, Raich R, Hero III AO. On local intrinsic dimension estimation and its applications. IEEE Transactions on Signal Processing. 2009;58(2):650–663.

[49] Creutzig F, Sprekeler H. Predictive coding and the slowness principle: An information-theoretic approach. Neural Computation. 2008;20(4):1026–1041.

[50] Floryan D, Graham MD. Data-driven discovery of intrinsic dynamics. Nature Machine Intelligence. 2022;4(12):1113–1120.

[51] Bai W, Yamashita O, Yoshimoto J. Learning task-agnostic and interpretable subsequence-based representation of time series and its applications in fMRI analysis. Neural Networks. 2023;163:327–340.

[52] Bialek W, Nemenman I, Tishby N. Predictability, complexity, and learning. Neural computation. 2001;13(11):2409–2463.

[53] Cho K, Van Merriënboer B, Gulcehre C, Bahdanau D, Bougares F, Schwenk H, et al. Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv preprint arXiv:14061078. 2014;.

[54] Oord Avd, Dieleman S, Zen H, Simonyan K, Vinyals O, Graves A, et al. Wavenet: A generative model for raw audio. arXiv preprint arXiv:160903499. 2016;.

[55] Girin L, Leglaive S, Bie X, Diard J, Hueber T, Alameda-Pineda X. Dynamical variational autoencoders: A comprehensive review. arXiv preprint arXiv:200812595. 2020;.

[56] Nowozin S, Cseke B, Tomioka R. f-gan: Training generative neural samplers using variational divergence minimization. In: Advances in neural information processing systems; 2016. p. 271–279.

[57] Oord Avd, Li Y, Vinyals O. Representation learning with contrastive predictive coding. arXiv preprint arXiv:180703748. 2018;.

[58] Lawrence N, Hyvärinen A. Probabilistic non-linear principal component analysis with Gaussian process latent variable models. Journal of machine learning research. 2005;6(11).

[59] Deisenroth MP, Turner RD, Huber MF, Hanebeck UD, Rasmussen CE. Robust filtering and smoothing with Gaussian processes. IEEE Transactions on Automatic Control. 2011;57(7):1865–1871.

[60] Arnold L, Jones CK, Mischaikow K, Raugel G, Arnold L. Random dynamical systems. Springer; 1995.

[61] Grosmark AD, Buzsáki G. Diversity in neural firing dynamics supports both rigid and learned hippocampal sequences. Science. 2016;351(6280):1440–1443.

[62] Tzourio-Mazoyer N, Landeau B, Papathanassiou D, Crivello F, Etard O, Delcroix N, et al. Automated anatomical labeling of activations in SPM using a macroscopic anatomical parcellation of the MNI MRI single-subject brain. Neuroimage. 2002;15(1):273–289.

[63] Aminikhanghahi S, Cook DJ. A survey of methods for time series change point detection. Knowledge and information systems. 2017;51(2):339–367.

[64] Wiskott L, Sejnowski TJ. Slow feature analysis: Unsupervised learning of invariances. Neural computation. 2002;14(4):715–770.

[65] Kriegeskorte N, Mur M, Bandettini PA. Representational similarity analysis-connecting the branches of systems neuroscience. Frontiers in systems neuroscience. 2008;2:249.

[66] Lin B, Kriegeskorte N. The topology and geometry of neural representations. Proceedings of the National Academy of Sciences. 2024;121(42):e2317881121.