ESTIMATING THE INTRINSIC DIMENSION OF NEURAL MANIFOLDS FROM VARIATIONAL AUTOENCODERS

Anonymous authors
Paper under double-blind review

ABSTRACT

Neural population activity is often conceptualized as lying on low-dimensional manifolds. Yet estimating the intrinsic dimension of these manifolds is difficult in noisy, curved neural data, and existing estimators are often not reliable enough to serve as a scientific yardstick. We therefore investigate a method based on fitting a variational autoencoder (VAE) to the neural data and estimating dimension from the effective rank of the decoder Jacobian. We explain the effectiveness of this approach by linking the VAE to local probabilistic principal component analysis, showing that the variational objective penalizes decoders that use unnecessary latent dimensions. In neuroscience-inspired simulations with highly noisy and curved data, our method accurately recovers intrinsic dimension where alternative methods fail. Applied to population recordings from the mouse head direction system, it identifies two intrinsic dimensions. Further analysis suggests that one of these dimensions may reflect gain modulation, even in the absence of explicit experimental manipulation. Applied to population recordings from a grid cell module in mouse medial entorhinal cortex, our method identifies 4 dimensions, related to x and y position, head direction, speed, and gain. Since VAE models are common in neuroscience, our method can be applied to previously fit models. We demonstrate this on a VAE fit to juvenile male zebra finch songs over weeks of vocal practice, identifying an increase in dimensionality over time.

1 INTRODUCTION

A central goal in systems neuroscience is to identify the variables represented by population activity. Yet encoding analyses are constrained by the variables chosen a priori and can miss structure associated with unmeasured or unexpected factors (Stevenson, 2018; Kriegeskorte & Douglas, 2019; Rubin et al., 2019). Unsupervised neural manifold inference provides a complementary starting point by characterizing the geometry of population activity directly (Gallego et al., 2017; Perich et al., 2025; Langdon et al., 2023; Mitchell-Heggs et al., 2023). One of its most basic properties is intrinsic dimension (ID): the number of independent degrees of freedom needed to describe the potentially non-linear activity manifold (Jazayeri & Ostojic, 2021; De & Chaudhuri, 2023). ID is informative because it can constrain the space of possible mechanisms, test whether familiar low-dimensional descriptions are sufficient, and guide subsequent interpretation.

Estimating intrinsic dimension from neural data, however, is difficult due to noise/variability in firing and curvature of the embedding (i.e. neural firing is usually not a linear function of an encoded variable) (Churchland et al., 2010; Pandarinath et al., 2018; De & Chaudhuri, 2023; Langdon et al., 2023; Facco et al., 2017). Most existing estimators rely, either explicitly or implicitly, on the local Euclidean property of a manifold (Grassberger & Procaccia, 1983; Levina & Bickel, 2004; Facco et al., 2017; Thordsen & Schubert, 2020; Fukunaga & Olsen, 1971). For instance, a commonly used method in neuroscience (Nieh et al., 2021), fractal dimension relies on the power law scaling of the number of neighbors within a radius with the radius in the euclidean space. These assumptions immediately raise a problem: one must choose a neighborhood scale at which the Euclidean approximation is supposed to hold. If the neighborhood is too small, the estimate becomes dominated by noise. If it is too large, curvature distorts the local geometry so that these Euclidean properties don't hold anymore. There is not a principled or even practically effective procedure to select these hyperparameters. Since the answer can change substantially with this choice, the estimate is fragile and difficult to interpret.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

On the other hand, deep generative models are a popular and flexible framework for modeling neural spiking data. Many popular approaches are variants of variational autoencoders (VAE) (Kingma & Welling, 2013), including LFADS (Pandarinath et al., 2018), iLQR-VAE (Schimel et al., 2022), and pi-VAE (Zhou & Wei, 2020) among others. The VAE maps the input (e.g., population vector of spike counts) onto a distribution on the latent space, which is then decoded (into firing rates) to reconstruct the input (via a Poisson observation model). Prior work suggests the trained decoder (the map from latent to observation) contains information about the geometry of the data manifold (Arvanitidis et al., 2017). However, the dimensionality of the latent space in VAE architectures is typically larger than the dimensionality of the underlying data manifold due to practical considerations related to model training (He et al., 2019; Dai & Wipf, 2019; Martinez & Williams, 2026). Here, we investigate how to leverage these models (in particular, VAE) to infer estimates of intrinsic dimensionality about the neural manifold.

We first explain why the Jacobian of the decoder should encode the correct ID, and not fill up all available latent dimensions, by linking the VAE to probabalistic principle component analysis (PPCA) and show how that objective penalizes unnecessary dimensions (Kumar & Poole, 2020; Dai et al., 2018). Intuitively, VAE models the uncertainty of the latent variable, and the latent is uncertain in the direction where the data is scarce. Along these uncertain directions, the variational inference objective penalizes decoder sensitivity, such that the decoder learns to ignore these data-poor directions. We show on synthetic datasets that this approach is more accurate and more robust than exiting estimators, even in the presence of noise and high curvature, where neighborhood methods are most vulnerable. Finally, we demonstrate the value of the estimator on real neural recordings by revisiting datasets from the rodent head direction system, grid cell modules, and zebra finch vocalizations.

1.1 Related work

ID estimation in neuroscience is mostly performed via neighborhood-based methods, or applying those methods to learned latent variables. In addition, (Altan et al., 2021) proposes to use a latent variable model to denoise the spiking observations first, and then applies the aforementioned estimators either to the denoised observations or the latent variables.

Prior analyses of VAE have shown that the variational objective can suppress superfluous latent directions in the setting of diagonal posterior. For instance, a theorem shows the number of "active dimensions" (latent dimensions with low uncertainty) can be used for ID estimation (Zheng et al., 2022). Rolinek et al. (2019); Kumar & Poole (2020) analyzed VAE from the lens of probabilistic PCA (PPCA), and performed local Taylor expansion to link the properties of the decoder Jacobian and the posterior covariance. Dai & Wipf (2019) performed a similar analysis, using a global affine decoder to show that posterior covariance of the latent can help prune superfluous latent coordinates and thereby act as a rough indication of global manifold dimension. They used the number of non-zero columns in the first decoder-layer weights. That heuristic depends on a diagonal encoder posterior, which we will show performs worse than a full covariance. (Rolinek et al., 2019) also performed local Taylor expansion to show how the stochastic part of the VAE objective promotes local orthogonality of the decoder.

Causin & Marta (2025) recently proposed a similar approach of using VAE and the pull-back geometry to estimate the ID. However it is taken for granted that the learned geometry encodes the data manifold, without any demonstration of the mechanism. Another recent approach uses deep generative models to fit the score function of the data likelihood to identify data-rich and -poor directions, from which information about the intrinsic dimension can be obtained (Stanczuk et al., 2024).

2 Theory and Methodology

2.1 ID estimator via the Jacobian rank of the VAE decoder

A variational autoencoder (VAE) models the data as generated from a lower-dimensional latent variable $z \in \mathbb{R}^D$, with prior $z \sim p(z)$. The observation y (e.g. neural spikes at one time point) is generated by a decoder network $f_\theta(z)$, with likelihood $p_\theta(y \mid z)$ given by a noise model. We use a Gaussian likelihood for the local theoretical analysis below, and a Poisson likelihood for simulations

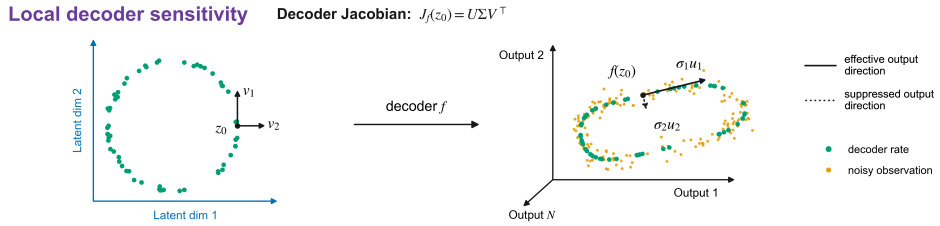

Figure 1: **Schematic of VAE-ID.** The VAE can learn to ignore irrelevant variations in the data manifold in its decoder. Its Jacobian has large singular values on the tangential directions and small singular values on the normal directions.

and real spike data. Since the marginal likelihood

$$p_\theta(y) = \int p_\theta(y \mid z) p(z)\, dz$$

is intractable, the latent posterior is approximated by a variational distribution $q_\phi(z \mid y)$, parameterized by an encoder network $g_\phi(\cdot)$. The training objective is the negative evidence lower bound (ELBO):

$$\mathrm{ELBO}(y) = -\mathbb{E}_{q_\phi(z|y)}\left[\log p_\theta(y \mid z)\right] + \mathrm{KL}(q_\phi(z \mid y) \,\|\, p(z))\,, \tag{1}$$

where the variational distribution and the prior are:

$$q_\phi(z \mid y) = \mathcal{N}(\mu, \Sigma), \qquad p(z) = \mathcal{N}(0, I).$$

For the Gaussian case with

$$p_\theta(y \mid z) = \mathcal{N}\big(f_\theta(z), \sigma^2 I\big)\,,$$

this becomes, up to constants independent of the neural network weights (θ, ϕ),

$$\mathrm{ELBO}(y) = \frac{1}{2\sigma^2}\, \mathbb{E}_{q_\phi(z|y)}\left[\|y - f_\theta(z)\|_2^2\right] + \mathrm{KL}(q_\phi(z \mid y) \,\|\, p(z))\,. \tag{2}$$

Our proposed ID estimator is based on the effective rank of the decoder Jacobian

$$J_\theta(z) = \frac{\partial f_\theta(z)}{\partial z}.$$

At each data point, we evaluate the Jacobian at the encoder mean $z_0 = \mu_\phi(y)$, and define the local ID estimate by the smallest k such that

$$\frac{\sum_{i=1}^{k} \lambda_i(J^\top J)}{\sum_{i=1}^{D} \lambda_i(J^\top J)} \geq 0.95, \tag{3}$$

where $\lambda_1 \geq \cdots \geq \lambda_D$ are the eigenvalues of $J^\top J$. We also experimented with other practical definitions, such as the participation ratio, and found the 95% criterion to be the most accurate on simulated data. Admittedly the 95% is a heuristic threshold determined by us, we showed it produces accurate result on simulation and reasonable result on real data, whereas it is difficult to determine a practically accurate rule for the neighborhood-based methods (Appendix A.4).

2.2 Why does the Jacobian rank capture manifold dimension

We assume that the manifolds of interest to us can be expressed locally by a smooth embedding f through $y = f(z)$ (In the VAE case this function is the decoder function). The columns of the Jacobian $J_f(z)$ span the tangent space (i.e. collection of all tangent vectors) at the point $y = f(z)$. The dimension of the tangent space is exactly the manifold's intrinsic dimension, since it specifies the degrees of freedom present in moving along the manifold.

As a simple example, suppose a 1D ring is described using a 2D latent space (without the origin) and embedded in a 3D observation space:

$$f(z_1, z_2) = \begin{bmatrix} z_1/r \\ z_2/r \\ 0 \end{bmatrix}, \tag{4}$$

where $r = \sqrt{z_1^2 + z_2^2}$. Then the Jacobian is

$$J_f(z_1, z_2) = \begin{bmatrix} z_2^2/r^3 & -z_1 z_2/r^3 \\ -z_1 z_2/r^3 & z_1^2/r^3 \\ 0 & 0 \end{bmatrix}. \tag{5}$$

Even though this matrix has two columns, it is rank-1 ($J_f(z_1, z_2)_{:,1} = -\frac{z_2}{z_1} J_f(z_1, z_2)_{:,2}$), corresponding to the fact that there is only one degree of freedom moving along the manifold. This illustrates an important detail: the latent dimension can be larger than necessary, since the Jacobian rank can still reflect the true intrinsic dimension, provided the redundant latent directions are locally ignored by the decoder near the data manifold.

2.3 Why does the VAE not overfit: a local PPCA perspective

The key question is then why the latent Jacobian should ignore redundant latent directions. To understand the implicit regularization imposed by the VAE, we linearize the decoder around the latent posterior mean μ of one data point y:

$$f_\theta(z) \approx f_\theta(\mu) + J(z - \mu), \tag{6}$$

where $J = J_\theta(\mu)$ is the decoder Jacobian at μ.

Under this local linearization, after optimizing over the Gaussian posterior covariance (Appendix A.1) and retaining only terms that depend on the decoder parameters θ (i.e., treating the posterior mean μ as fixed), the negative ELBO is approximately

$$\mathrm{nELBO}(y) \approx \frac{1}{2\sigma^2}\|y - f_\theta(\mu)\|_2^2 + \frac{1}{2}\sum_{i=1}^{D} \log\left(1 + \frac{\lambda_i(J^\top J)}{\sigma^2}\right), \tag{7}$$

where $\lambda_i(J^\top J)$ is the square of the i-th singular value of J. The first term is a point-wise reconstruction error, and the second term penalizes the scale of J. The approximation makes the regularization more explicit: large singular values of the decoder Jacobian are penalized unless they improve reconstruction enough to justify the cost. Different singular values are discouraged to be simultaneously large, since the penalty is sum of logs, which induces a penalty on their product.

This is closely related to probabilistic PCA (PPCA), which has generative model

$$z \sim \mathrm{Normal}(0, I), \tag{8}$$

$$y = \bar{y} + Jz + \epsilon, \tag{9}$$

$$\epsilon \sim \mathrm{Normal}(0, \sigma^2 I). \tag{10}$$

After centering the data as $x = y - \bar{y}$, and using a Gaussian variational posterior $q(z) = \mathcal{N}(\mu, \Sigma)$, the negative ELBO is, up to constants (Appendix A.1),

$$\mathrm{nELBO}(x) = \frac{1}{2\sigma^2}\|x - J\mu\|_2^2 + \frac{1}{2}\sum_{i=1}^{D} \log\left(1 + \frac{\lambda_i(J^\top J)}{\sigma^2}\right). \tag{11}$$

Thus, after first-order linearization, the local VAE objective takes the same form as the variational objective for PPCA. In this sense, the VAE behaves like a local PPCA model whose local linear map is given by the decoder Jacobian. The maximum likelihood solution of PPCA zeros the eigenvalues of $J^T J$ below the observation noise variance, thereby pruning the directions (Tipping & Bishop, 1999).

Local PCA has been proposed as an ID estimator (Fukunaga & Olsen, 1971), but it can perform poorly, possibly because it treats each neighborhood separately and suffers the tradeoff of choosing the proper neighborhood size (Figure A.1). In contrast, the VAE learns one global generative model and pools across all data to boost statistical efficiency. Locality is then represented through the learned decoder, rather than imposed by a fixed neighborhood radius.

Although VAEs have several well-known shortcomings from a machine learning perspective, such as imperfect data generation, unreliable disentanglement, and non-identifiability (Dai & Wipf, 2019; Locatello et al., 2019; Khemakhem et al., 2020), they don't affect its suitability for ID estimation.

Since our goal is not to generate realistic samples or assign semantic meaning to each latent coordinate, the first two concerns are irrelevant. Regarding non-identifiability, consider a smooth invertible reparameterization $z' = h(z)$ and define a new decoder

$$g = f \circ h^{-1},$$

so that $g(z') = f(z)$. Then the Jacobian transforms as

$$J_g(z') = J_f(z)\, J_{h^{-1}}(z'). \tag{12}$$

Since $J_{h^{-1}}(z')$ is invertible, the exact rank of the Jacobian is unchanged:

$$\operatorname{rank}(J_g) = \operatorname{rank}(J_f). \tag{13}$$

Thus, the Jacobian rank is invariant to smooth changes of latent coordinates. In practice, however, threshold-based effective ranks can still depend on the singular value spectrum, so this invariance is exact for rank, but only approximate for finite-sample estimators based on spectral thresholds.

While our chosen VAE architecture was accurate across a wide range of conditions in simulation, different architectures could produce different estimates. We therefore proposed a correction method to mitigate this situation in Appendix A.5. For the main results we stick to the well-tested architecture.

2.4 Alternative estimators

Here we give a brief description of the main alternative ID estimators used in this paper. Correlation integral (CorrInt, or fractal dimension, Grassberger & Procaccia, 1983) is based on the intuition that if points are uniformly distributed in $\mathbb{R}^d$, then the number of points $C(r)$ within a ball of radius r should scale with the neighborhood volume, and thus:

$$C(r) \sim r^d. \tag{14}$$

Near each point, one could treat the manifold as Euclidean where the same scaling relationship applies. The intrinsic dimension is thus estimated by picking a minimum and a maximum neighborhood size and estimating the slope in log space:

$$\hat{d} = \frac{\log C(r_{max}) - \log C(r_{min})}{r_{max} - r_{min}}. \tag{15}$$

Angle basied intrisic dimension (ABID, Thordsen & Schubert, 2022) asks the question, in a truly d-dimensional local cloud, what the distribution of angles between neighbor directions should look like. Around a point x, one takes neighboring displacement vectors and normalize: $v_i = x_i - x$, $u_i = v_i/\|v_i\|$. Again, if the local neighborhood fills a d-dimensional space roughly uniformly, then the directions u_i behave like uniformly random directions on the unit sphere S^{d-1}. The crucial relation is:

$$\mathbb{E}[(\boldsymbol{u}^T \boldsymbol{v})^2] = 1/d, \tag{16}$$

for random directions u, v on the unit sphere S^{d-1}. The proposed benefit of ABID is that it ignores the length and can be more noise robust.

3 Results

3.1 VAE-ID outperforms alternative methods on difficult simulations of neural activity

Neural data is often viewed as noisy observations of smooth tuning curves on some latent variables. We thus simulate neural data as noisy observations of a D dimensional torus, while explicitly controlling the noise level and the smoothness of the tuning (Figure 2A). While CorrInt performs decently on the low noise, high smoothness and low ID cases, it fails drastically in the case of lower smoothness, higher ID, or higher Gaussian noise or Poisson noise (Figure 2B-C). ABID has a strange tendency of outputting ~ 4 dimensions in 46% of all simulations, even when the underlying manifold is 1 or 2 dimensional and there are 25% of each dimension (Figure 2B-C). Choosing different neighborhood sizes did not consistently improve the neighborhood-based methods, especially in the

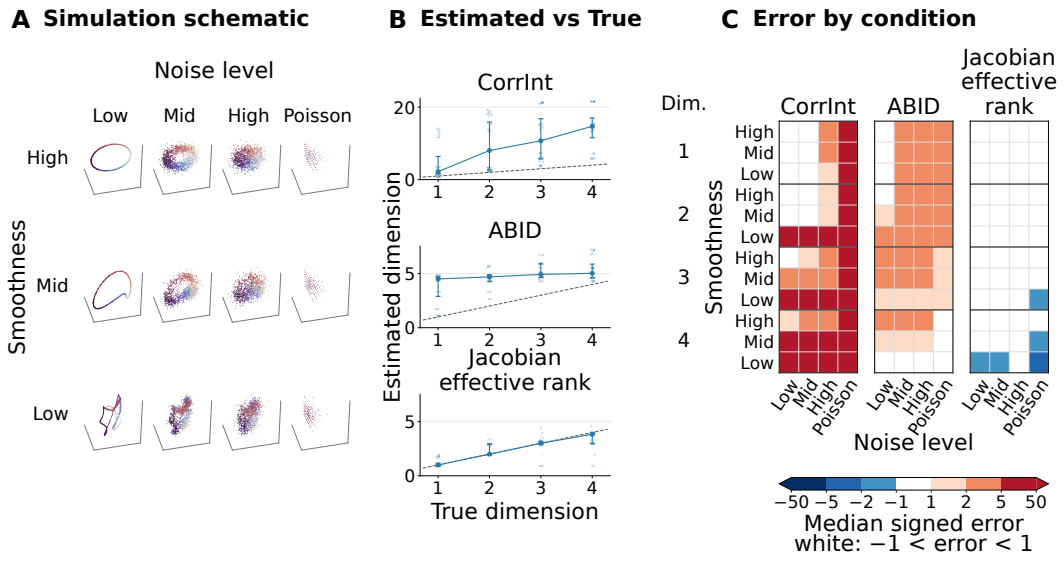

Figure 2: **VAE-ID estimates ID accurately on synthetic neural data.** **(A)** Schematic for the neural-like simulation in 1D, of varying noise level and tuning smoothness. The axes are example neurons' observations. The first three columns show Gaussian noise and the last one Poisson. **(B)** Estimated vs true intrinsic dimensions for different methods. Each dot is one simulation. Blue solid lines mark the median. Error bars mark the interquartile range. The dotted line marks the identity line. **(C)** Heatmaps of the median signed error of different methods and simulation conditions.

Table 1: Accuracy of different ID estimators

Input / model	Estimator	MAE ↓	Prediction R^2 ↑
Observations	CorrInt	7.556	−77.785
	ABID	2.207	−3.953
	MLE	9.746	−140.125
	TwoNN	21.953	−576.152
	Local PCA	40.809	−2557.788
Full-posterior VAE	**Jacobian effective rank (95% var.)**	**0.403**	**0.590**
	Jacobian effective rank (97% var.)	0.449	0.527
	Jacobian participation ratio	0.545	0.479
	Latent CorrInt	0.552	0.551
	Latent ABID	0.650	0.502
Diagonal-posterior VAE	Jacobian effective rank (95% var.)	0.697	0.038
	Jacobian effective rank (97% var.)	0.809	−0.263
	Jacobian participation ratio	0.523	0.623
	Latent CorrInt	0.701	0.308
	Latent ABID	0.966	−0.181

high noise condition (Figure A.2E-F). VAE based methods generally have negligible errors, except for the really difficult low smoothness cases in higher dimensions (Figure 2B-C). In addition to being more accurate, the VAE-ID also benefits from more observations and more neurons, compared to neighborhood-based methods Figure A.2A-D. Furthermore, other estimators from scikit-dimension (Bac et al., 2021) also performed poorly (Figure A.1A, Table 1). The full covariance VAE performs better than the diagonal covariance VAE (Figure A.1B-C, Table 1). Applying the neighborhood-based estimators on the latent means also largely improved accuracy, although still worse than the VAE-ID (Figure A.1B-C, Table 1).

3.2 Mouse head direction activity exhibits two intrinsic dimensions

We next applied the estimator to population recordings of the post-subiculum neurons in mice navigating in different open-field environments (71 head direction units, 40 minutes, with 100ms bins,

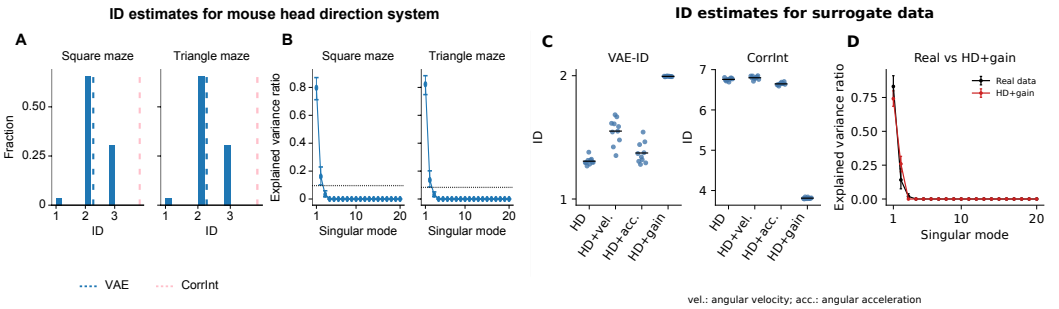

Figure 3: **VAE-ID identifies two intrinsic dimensions in mouse head direction population.** **(A)** Histograms of ID estimates for different environments. Vertical lines mark the average. **(B)** The explained variance ratio of the singular mode of the decoder Jacobians. Blue line is the median across time bins. The error bars span the 25th-75th percentiles across time bins. The dotted horizontal line marks the threshold for 95% variance explained. **(C)** ID estimates for surrogate data. Each dot is a surrogate dataset, generated from the corresponding tuning model labeled on the X axis. **(D)** Similar to (B), The explained variance ratio of the singular mode of the decoder Jacobians, median and IQR, for the real data (black) and one HD+gain surrogate dataset (red).

yielding 25000 bins) (Duszkiewicz et al., 2024). The VAE ID estimator estimates an average intrinsic dimension of 2 (Figure 3A-B). The estimates from the two environments are consistent with each other (Figure 3A) [1]. This estimate goes against the conventional belief that the head direction circuit is a one-dimensional ring attractor (Zhang, 1996; Knierim & Zhang, 2012). The CorrInt gives close to 4D, which is likely an overestimation given its performance on the simulation.

Could it be that VAE-ID still overestimates the dimension in real data? We first confirmed that it can recover a 1D manifold using a surrogate dataset. We generated data by sampling Poisson spike counts based on the empirical head direction tuning curves independently across neurons (Appendix A.2). This generates noisy observations around a 1D manifold by construction, and VAE-ID indeed recovers the correct dimensionality (Figure 3C). We then looked for a second covariate that could explain the second manifold dimension. However, neither angular velocity nor angular acceleration added sufficient degrees of freedom to make the simulated activity 2D (Figure 3C). We finally took inspiration from (Ajabi et al. 2023; see also Goris et al. 2014), and fit a model where a common gain is multiplied to all neurons' head direction tuning curves. The gain varies in time and is jointly inferred with the head direction tuning. In (Ajabi et al., 2023) the gain model was introduced to account for the change in the activity of head direction neurons in response to changes in lighting and cue configuration. Even though these manipulations were absent, we found that surrogate data generated with animal's head direction and the fitted instantaneous gain was able to reproduce the 2D intrinsic dimension as in real data (Figure 3C-D). Across these simulations, CorrInt consistently gave inflated estimates of dimensionality, which sometimes decreased after introducing additional variables into the surrogate data (Figure 3C).

The analysis of the surrogate datasets suggests that the 2D result is consistent with a manifold of head direction plus gain. But our VAE-ID further offers a way to directly relate the directions on the manifold to labels of interest. To do so, we examine the geometry of the supervised model of population spikes, i.e. the head direction tuning with population gain model. If the two dimensions of the neural manifold defined by the VAE truly correspond to head direction and gain, the partial derivatives of the supervised model with respect to head direction and gain should lie in the subspace spanned by the top two singular vectors of the VAE decoder Jacobian. We first construct a tuning manifold, i.e. the collection of population firing rates at the grids of HD values and gains, and compute the partial derivatives with respect to gain and head directionat the grid points. We then input these rates into the trained VAE (Figure 4A), and query the decoder Jacobians (Figure 4B), perform SVD, and measure the absolute cosine angle between the partial derivatives and the subspace spanned by the top 2 singular vectors of the decoder Jacobians. On average, the alignment is $0.76/0.77$ for HD/gain, compared to the 97.5% of the null from the random $2D$ subspaces: $0.32/0.32$ (Figure 4C). Together with the fact that the cosine angle between HD and gain partial derivatives is 0.01 on average, this analysis suggests HD and gain indeed form the two directions spanning the local tangent space of the neural manifold.

[1] Mathematically speaking, a smooth manifold has the same intrinsic dimension everywhere. However, on noisy data, it is reasonable to allow for varying ID.

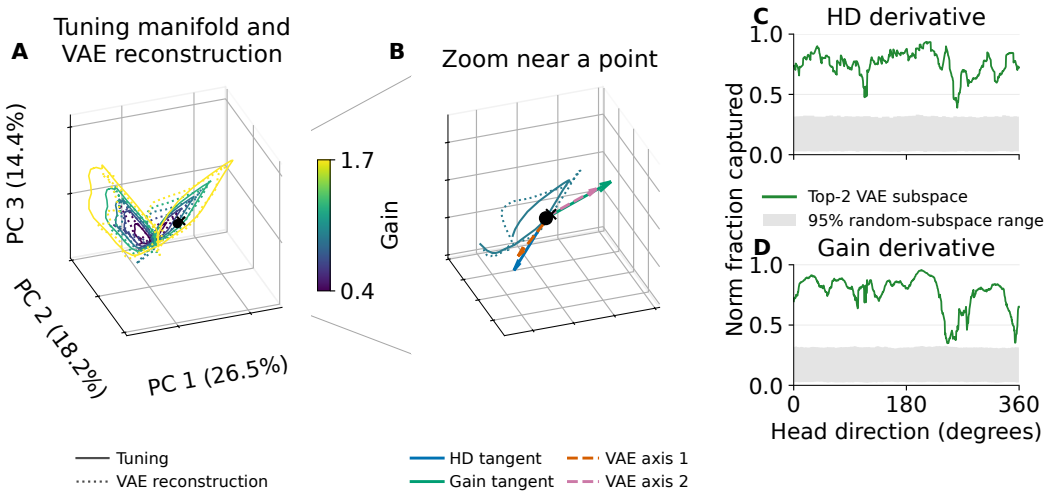

Figure 4: **Interpreting manifold dimensions with Jacobian axis alignment (A)** Tuning manifold and VAE reconstruction in principal components space. Tuning manifold is the collection of population firing rates from the supervised tuning model evaluated on a grid of head directions and gains. Axis labels show the fraction of variance explained. **(B)** Zoom near the black dot in (A), to show the tuning partial derivatives (solid arrows) and the top 2 VAE decoder Jacobian left singular vectors (dotted arrows), all projected onto the principal components space. The curves are the tuning manifold (solid) and VAE reconstruction (dotted) near the point. **(C-D)** The alignment of the partial derivatives of the tuning model with the top-2 VAE decoder Jacobian singular subspace, as a function of head direction.

3.3 MEC DATA

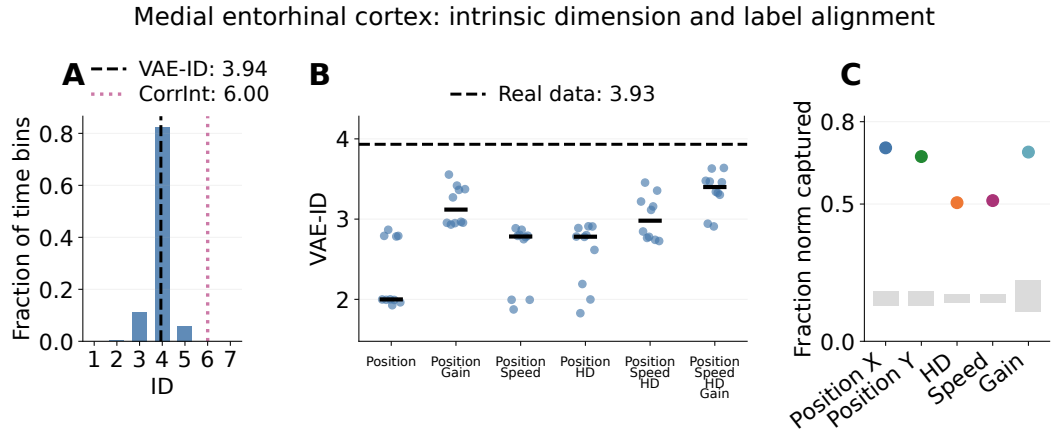

Figure 5: **VAE-ID identifies four intrinsic dimensions in a mouse grid cell module. (A)** Histogram of the ID estimates for a grid cell module in the mouse MEC during open field exploration. Vertical lines mark the averages. **(B)** ID estimates for surrogate data. Each dot is a surrogate dataset, generated from the corresponding tuning model labeled on the X axis. Short horizontal bars are medians. Dotted horizonal line marks the VAE-ID from real data. **(C)** The alignment of the partial derivatives of the tuning model with the top-4 VAE decoder Jacobian singular subspace, averaged over time. Grey shades mark the central 95% range of the alignment with 2000 random 4D subspaces, averaged over time.

We next applied VAE-ID to population recordings of the mouse medial enthorinal cortex (MEC) during open-field exploration (2.1 hours, with 100ms bins, yielding ∼76000 bins)(Gardner et al., 2022). We restricted our analysis to one grid module (149 units) where the original paper reported little head direction tuning. We found an average ID close to 4 (as opposed to 6 from CorrInt) (Figure 5A). Similar to HD data, we constructed surrogate datasets with tuning models to known variables. The position only model has a median of 2, as expected (Figure 5B). Adding gain increases the ID to above 3. Further adding speed and head direction increases the estimate to 3.5, suggesting while speed and head direction adds more degrees of freedom to the neural activity, they are not strong enough to count as an extra dimension (Figure 5B). The alignment analysis further supports

this conclusion (Figure 5C). For all of the listed variables, the tuning sensitivity (partial derivatives of tuning with respect to the variables) align well with the top-4 VAE decoder Jacobian subspaces compared with the random 4D subspace null. Alignment with position and gain is higher than that of speed and head direction. Together, the result suggests one grid cell module is modulated beyond the 2D position, speed and head direction. Further analysis is needed to fully explain the gap between the best surrogate and the real data estimate of 4D.

3.4 BIRD SONG DATA

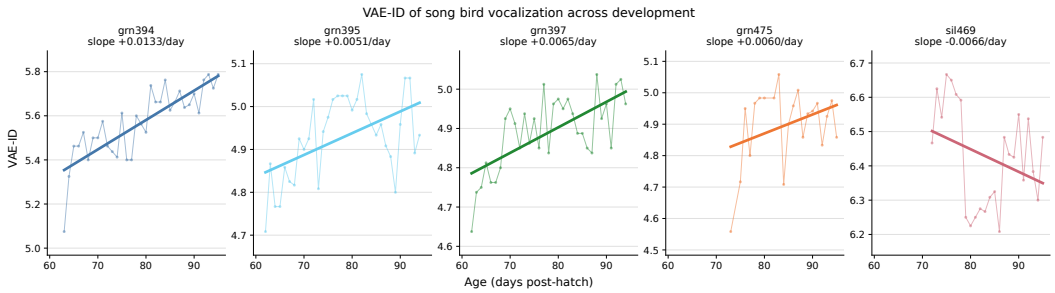

Figure 6: **VAE-ID shows an increase in the ID of bird songs across development** Faint curves: average VAE-ID for all vocalizations within a day (Y axis) vs age (X axis). Each panel is one animal. Solid straight lines: linear regression fits.

Lastly, we demonstrate how this framework can take advantage of existing work that already uses VAE in the broader systems neuroscience, beyond neural data. VAE has been used to characterize features of the zebra finch vocalization (Brudner et al., 2023). We analyzed these fitted VAE using our VAE-ID approach and found that the average ID of all vocalizations increase with age in four out of five animals (Figure 6) [2]. This result opens up future directions for interpreting the increased degrees of freedom and relating it to learning.

4 DISCUSSION

We introduced a framework for extracting manifold intrinsic dimension from trained VAEs, explained how VAE objective is conducive to ID estimation, and demonstrated its utility in rodent extracellular population recordings and bird song spectrograms. Our work makes the concept of "neural manifold" more rigorous by offering a way to probe the nonlinear geometric structure of the neural manifold and interpreting it with interpretable labels.

More importantly, it marks a step towards the more general philosophy of training an expressive model of neural data and interrogating its structure. Such an approach combines the benefit of the expressive power of deep learning to explain more neural variance than traditional hand-crafted or linear models, and the interpretability of linear models (e.g. using local linear approximations).

One limitation of the work is that the choice of the VAE architecture requires some heuristics. We discuss methods to mitigate the impact of having different choices in Appendix A.5. Future work could develop a more principled way of selecting the architecture. Furthermore, we link the dimension penalty of VAE to local PPCA, whose dimension penalty only occurs given the observation noise. Bayesian PCA (Bishop, 1998) with automatic relevance determination (ARD) can automatically infer the effective number of PCs. Similar ideas could be incorporated into VAE-ID to reduce its reliance on a pre-specified noise hyperparameters.

AI USE STATEMENT

In this work, we used generative AI tools to facilitate coding, generate figures, check mathematical claims and brainstorm in general. We did not use it to come up with the question, method or analysis in the current work. It is only used to refine writing, not generate bulks of writing directly.

[2] While our approach obtains an integer at each data point, their average could reflect an overall levels of degrees of freedom that could be decimal.

REFERENCES

Zaki Ajabi, Alexandra T Keinath, Xue-Xin Wei, and Mark P Brandon. Population dynamics of head-direction neurons during drift and reorientation. *Nature*, 615(7954):892–899, 2023.

Ege Altan, Sara A Solla, Lee E Miller, and Eric J Perreault. Estimating the dimensionality of the manifold underlying multi-electrode neural recordings. *PLoS computational biology*, 17(11): e1008591, 2021.

Georgios Arvanitidis, Lars Kai Hansen, and Søren Hauberg. Latent space oddity: on the curvature of deep generative models. *arXiv preprint arXiv:1710.11379*, 2017.

Jonathan Bac, Evgeny M Mirkes, Alexander N Gorban, Ivan Tyukin, and Andrei Zinovyev. Scikit-dimension: a python package for intrinsic dimension estimation. *Entropy*, 23(10):1368, 2021.

Christopher Bishop. Bayesian pca. *Advances in neural information processing systems*, 11, 1998.

Samuel Brudner, John Pearson, and Richard Mooney. Generative models of birdsong learning link circadian fluctuations in song variability to changes in performance. *PLOS Computational Biology*, 19(5):e1011051, 2023.

Paola Causin and Alessio Marta. Estimating dataset dimension via singular metrics under the manifold hypothesis: Application to inverse problems. *arXiv preprint arXiv:2507.07291*, 2025.

Mark M Churchland, Byron M Yu, John P Cunningham, Leo P Sugrue, Marlene R Cohen, Greg S Corrado, William T Newsome, Andrew M Clark, Paymon Hosseini, Benjamin B Scott, et al. Stimulus onset quenches neural variability: a widespread cortical phenomenon. *Nature neuroscience*, 13(3):369–378, 2010.

Bin Dai and David Wipf. Diagnosing and enhancing vae models. *arXiv preprint arXiv:1903.05789*, 2019.

Bin Dai, Yu Wang, John Aston, Gang Hua, and David Wipf. Connections with robust pca and the role of emergent sparsity in variational autoencoder models. *Journal of Machine Learning Research*, 19(41):1–42, 2018.

Anandita De and Rishidev Chaudhuri. Common population codes produce extremely nonlinear neural manifolds. *Proceedings of the National Academy of Sciences*, 120(39):e2305853120, 2023.

Adrian J Duszkiewicz, Pierre Orhan, Sofia Skromne Carrasco, Eleanor H Brown, Eliott Owczarek, Gilberto R Vite, Emma R Wood, and Adrien Peyrache. Local origin of excitatory–inhibitory tuning equivalence in a cortical network. *Nature neuroscience*, 27(4):782–792, 2024.

Julio Esparza, Juan Pablo Quintanilla, Elena Cid, Ana C Medeiros, Juan A Gallego, and Liset Menendez de la Prida. Cell-type-specific manifold analysis discloses independent geometric transformations in the hippocampal spatial code. *Neuron*, 2025.

Elena Facco, Maria d'Errico, Alex Rodriguez, and Alessandro Laio. Estimating the intrinsic dimension of datasets by a minimal neighborhood information. *Scientific reports*, 7(1):12140, 2017.

Keinosuke Fukunaga and David R Olsen. An algorithm for finding intrinsic dimensionality of data. *IEEE Transactions on computers*, 100(2):176–183, 1971.

Juan A Gallego, Matthew G Perich, Lee E Miller, and Sara A Solla. Neural manifolds for the control of movement. *Neuron*, 94(5):978–984, 2017.

Richard J Gardner, Erik Hermansen, Marius Pachitariu, Yoram Burak, Nils A Baas, Benjamin A Dunn, May-Britt Moser, and Edvard I Moser. Toroidal topology of population activity in grid cells. *Nature*, 602(7895):123–128, 2022.

Robbe LT Goris, J Anthony Movshon, and Eero P Simoncelli. Partitioning neuronal variability. *Nature neuroscience*, 17(6):858–865, 2014.

Peter Grassberger and Itamar Procaccia. Measuring the strangeness of strange attractors. *Physica D: nonlinear phenomena*, 9(1-2):189–208, 1983.

Junxian He, Daniel Spokoyny, Graham Neubig, and Taylor Berg-Kirkpatrick. Lagging inference networks and posterior collapse in variational autoencoders. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=rylDfnCqF7.

Mehrdad Jazayeri and Srdjan Ostojic. Interpreting neural computations by examining intrinsic and embedding dimensionality of neural activity. *Current opinion in neurobiology*, 70:113–120, 2021.

Ilyes Khemakhem, Diederik Kingma, Ricardo Monti, and Aapo Hyvarinen. Variational autoencoders and nonlinear ica: A unifying framework. In *International conference on artificial intelligence and statistics*, pp. 2207–2217. PMLR, 2020.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

James J Knierim and Kechen Zhang. Attractor dynamics of spatially correlated neural activity in the limbic system. *Annual review of neuroscience*, 35:267–285, 2012.

Nikolaus Kriegeskorte and Pamela K Douglas. Interpreting encoding and decoding models. *Current opinion in neurobiology*, 55:167–179, 2019.

Abhishek Kumar and Ben Poole. On implicit regularization in β $-vaes. In$*International Conference on Machine Learning*, $pp.\ 5480 - -5490. PMLR$, 2020.

Christopher Langdon, Mikhail Genkin, and Tatiana A Engel. A unifying perspective on neural manifolds and circuits for cognition. *Nature Reviews Neuroscience*, 24(6):363–377, 2023.

Elizaveta Levina and Peter Bickel. Maximum likelihood estimation of intrinsic dimension. *Advances in neural information processing systems*, 17, 2004.

Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Raetsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. Challenging common assumptions in the unsupervised learning of disentangled representations. In *international conference on machine learning*, pp. 4114–4124. PMLR, 2019.

Miles Martinez and Alex Williams. Quasi-monte carlo methods enable extremely low-dimensional deep generative models. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 63835–63860, 2026.

Rufus Mitchell-Heggs, Seigfred Prado, Giuseppe P Gava, Mary Ann Go, and Simon R Schultz. Neural manifold analysis of brain circuit dynamics in health and disease. *Journal of computational neuroscience*, 51(1):1–21, 2023.

Edward H Nieh, Manuel Schottdorf, Nicolas W Freeman, Ryan J Low, Sam Lewallen, Sue Ann Koay, Lucas Pinto, Jeffrey L Gauthier, Carlos D Brody, and David W Tank. Geometry of abstract learned knowledge in the hippocampus. *Nature*, 595(7865):80–84, 2021.

Chethan Pandarinath, Daniel J O'Shea, Jasmine Collins, Rafal Jozefowicz, Sergey D Stavisky, Jonathan C Kao, Eric M Trautmann, Matthew T Kaufman, Stephen I Ryu, Leigh R Hochberg, et al. Inferring single-trial neural population dynamics using sequential auto-encoders. *Nature methods*, 15(10):805–815, 2018.

Matthew G Perich, Devika Narain, and Juan A Gallego. A neural manifold view of the brain. *Nature Neuroscience*, 28(8):1582–1597, 2025.

Neil C Rabinowitz, Robbe L Goris, Marlene Cohen, and Eero P Simoncelli. Attention stabilizes the shared gain of v4 populations. *Elife*, 4:e08998, 2015.

Michal Rolinek, Dominik Zietlow, and Georg Martius. Variational autoencoders pursue pca directions (by accident). In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 12406–12415, 2019.

Alon Rubin, Liron Sheintuch, Noa Brande-Eilat, Or Pinchasof, Yoav Rechavi, Nitzan Geva, and Yaniv Ziv. Revealing neural correlates of behavior without behavioral measurements. *Nature communications*, 10(1):4745, 2019.

Marine Schimel, Ta-Chu Kao, Kristopher T Jensen, and Guillaume Hennequin. ilqr-vae: Control-based learning of input-driven dynamics with applications to neural data. In *International Conference on Learning Representations*, 2022.

Jan Pawel Stanczuk, Georgios Batzolis, Teo Deveney, and Carola-Bibiane Schönlieb. Diffusion models encode the intrinsic dimension of data manifolds. In *Forty-first International Conference on Machine Learning*, 2024.

Ian H Stevenson. Omitted variable bias in glms of neural spiking activity. *Neural computation*, 30(12): 3227–3258, 2018.

Erik Thordsen and Erich Schubert. Abid: angle based intrinsic dimensionality. In *International Conference on Similarity Search and Applications*, pp. 218–232. Springer, 2020.

Erik Thordsen and Erich Schubert. Abid: angle based intrinsic dimensionality—theory and analysis. *Information Systems*, 108:101989, 2022.

Michael E Tipping and Christopher M Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 61(3):611–622, 1999.

Kechen Zhang. Representation of spatial orientation by the intrinsic dynamics of the head-direction cell ensemble: a theory. *Journal of Neuroscience*, 16(6):2112–2126, 1996.

Yijia Zheng, Tong He, Yixuan Qiu, and David P Wipf. Learning manifold dimensions with conditional variational autoencoders. *Advances in Neural Information Processing Systems*, 35:34709–34721, 2022.

Ding Zhou and Xue-Xin Wei. Learning identifiable and interpretable latent models of high-dimensional neural activity using pi-vae. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

A Appendix

A.1 Derivation for why the VAE objective penalizes redundant latent dimension

The key question is then why the latent Jacobian should ignore redundant latent directions. To understand the implicit regularization imposed by the VAE, we linearize the decoder around the latent posterior mean μ of one data point y:

$$f_\theta(z) \approx f_\theta(\mu) + J(z-\mu), \tag{17}$$

where $J = J_\theta(\mu)$ is the decoder Jacobian at μ.

Then the Gaussian negative ELBO is approximately

$$\mathrm{nELBO}(y) \approx \frac{1}{2\sigma^2}\mathbb{E}_{q_\phi(z|y)}\Big[\big\|y - \big(f_\theta(\mu) + J(z-\mu)\big)\big\|_2^2\Big] + \frac{1}{2}\Big(\mathrm{tr}(\Sigma) + \|\mu\|_2^2 - \log\det\Sigma - D\Big) \tag{18}$$

$$\approx \frac{1}{2\sigma^2}\big\|y - f_\theta(\mu)\big\|_2^2 + \frac{1}{2\sigma^2}\mathrm{tr}(J^\top J\,\Sigma) + \frac{1}{2}\Big(\mathrm{tr}(\Sigma) + \|\mu\|_2^2 - \log\det\Sigma - D\Big) \tag{19}$$

$$= \frac{1}{2\sigma^2}\|y - f_\theta(\mu)\|_2^2 + R(J,\Sigma) + \mathrm{const}, \tag{20}$$

where the first term is a point-wise reconstruction, and the second term:

$$R(J,\Sigma) = \frac{1}{2\sigma^2}\mathrm{tr}(J^\top J\,\Sigma) + \frac{1}{2}\Big(\mathrm{tr}(\Sigma) + \|\mu\|_2^2 - \log\det\Sigma - D\Big) \tag{21}$$

is a penalty term for complexity. The term

$$\frac{1}{2\sigma^2}\mathrm{tr}(J^\top J\,\Sigma)$$

already shows a penalty on the singular values of the decoder Jacobian, given a fixed posterior covariance Σ. We can show the penalty more explicitly by first optimizing the above objective over Σ:

$$\Sigma^* = \left(I + \frac{1}{\sigma^2} J^\top J \right)^{-1}. \tag{22}$$

This inverse relationship is intuitive: a direction with high posterior variance will increase the expected reconstruction error unless the decoder is nearly insensitive to that direction, that is, unless the corresponding singular value of J is small.

We then plug Σ^* back into the covariance-dependent part (eq.21) of the objective gives

$$R(J) = \frac{1}{2} \log\det \left(I + \frac{1}{\sigma^2} J^\top J \right) \tag{23}$$

$$= \frac{1}{2} \sum_{i=1}^{D} \log \left(1 + \frac{\lambda_i(J^\top J)}{\sigma^2} \right), \tag{24}$$

where $\lambda_i(J^\top J)$ is the square of the $i-th$ singular value of J. This makes the regularization more explicit: large singular values of the decoder are penalized unless they improve reconstruction enough to justify the cost. Different singular values are discouraged to be simultaneously large, since the penalty is sum of log induces a penalty on their product.

This is closely related to probabilistic PCA (PPCA), which has generative model

$$z \sim \mathrm{Normal}(0, I), \tag{25}$$

$$y = \bar{y} + Jz + \epsilon, \tag{26}$$

$$\epsilon \sim \mathrm{Normal}(0, \sigma^2 I). \tag{27}$$

After centering the data as $x = y - \bar{y}$, and using a Gaussian variational posterior $q(z) = \mathcal{N}(\mu, \Sigma)$, the negative ELBO is, up to constants,

$$\mathrm{nELBO}(x) = \frac{1}{2\sigma^2} \mathbb{E}_{q(z)} \left[\|x - Jz\|_2^2 \right] + \mathrm{KL}(q(z) \,\|\, p(z)) \tag{28}$$

$$= \frac{1}{2\sigma^2} \|x - J\mu\|_2^2 + \frac{1}{2\sigma^2} \mathrm{tr}(J^\top J \, \Sigma) + \mathrm{KL}(q(z) \,\|\, p(z)) \tag{29}$$

$$= \frac{1}{2\sigma^2} \|x - J\mu\|_2^2 + \frac{1}{2} \sum_{i=1}^{D} \log \left(1 + \frac{\lambda_i(J^\top J)}{\sigma^2} \right). \tag{30}$$

The last line is obtained in the same way as for VAE (Eqs. 18–24). Thus, after first-order linearization, the local VAE objective takes the same form as the variational objective for PPCA. In this sense, the VAE behaves like a local PPCA model whose local linear map is given by the decoder Jacobian.

A.2 Implementation details

VAE architecture. The encoder and decoder were multilayer perceptrons with three hidden layers (default width 512 per layer) and ReLU nonlinearities. More layers tend to result in difficulty in training and fewer layers can underfit in certain simulations. The encoder mapped the observed activity vector to either a diagonal-covariance Gaussian posterior or a full-covariance Gaussian posterior in a latent space of dimension 10 for the simulation, and 20 for the real data. For the full-covariance variant, the posterior covariance was parameterized by a Cholesky factor. The decoder mapped latent variables back to neural activity space, with the final output layer chosen to match the observation model. For Gaussian simulations, we fixed the variance at 0.5. For the Poisson observation model, we passed the neural network output through a softplus to ensure nonnegative firing rates.

Models were trained with Adam using learning rate 1e-3, weight decay 1e-4, batch size 256, and gradient clipping at norm 1.0. Training used a 90/10 train-validation split. We found that the standard VAE would not learn in some Poisson simulations and suspected the KL penalty was too strong,

so we chose to do a warm-up of the KL weight from 0.5 to 1.0 over the first 100 epochs, and training continued for up to 500 epochs. Early stopping monitored validation loss, used patience 20 and minimum fractional improvement 1e-4, and restored the best checkpoint at the end. To avoid stopping during the KL transient, early-stop counting began only after warmup and an additional minimum-training window of 100 epochs.

Alternative ID estimators We benchmarked several neighborhood-based intrinsic-dimension estimators: correlation dimension (CorrInt), angle-based intrinsic dimension (ABID), two-nearest-neighbor estimation (TwoNN), local PCA (lPCA), and maximum-likelihood estimation (MLE). All but ABID were implemented using `scikit-dimension` Bac et al. (2021).

For ABID, lPCA, and MLE, we evaluated each method over five neighborhood sizes spanning 5% to 20% of the sample size T, with a minimum neighborhood size of 10. ABID estimates local dimension from the angular structure of the k-nearest-neighbor cloud; we averaged the local estimates across samples for each k and took the median across neighborhood sizes as the final estimate. For lPCA, we computed pointwise local PCA-based dimension estimates for each neighborhood size, averaged across samples for each k, and then averaged across neighborhood sizes. For MLE, we computed one estimate per neighborhood size and averaged across neighborhood sizes.

For CorrInt, we used neighborhood ranks k_1 and k_2 set to 5% and 20% of T, respectively. TwoNN was applied once per dataset with default settings. Each method yielded a single scalar estimate for each simulated dataset, which we compared with the ground-truth latent dimensionality. All but ABID were performed with the package `scikit-dimension` Bac et al. (2021). ABID was adapted from the repository `hippocampal_manifolds`Esparza et al. (2025).

Generate synthetic data via a Latent Variable Model The generative model for neural activities is given by the following:

$$\text{(d-dimensional Latent variable)} \quad \mathbf{x}_t \sim \mathrm{Unif}([0,1]^d) \tag{31}$$

$$\text{(Weights)} \quad w_b^n \sim \mathcal{N}(0,1), \quad \epsilon_t^n \sim \mathcal{N}(0,\sigma^2) \tag{32}$$

$$\text{(Firing rates)} \quad r_t^n = g\Big(\sum_{b=1}^{2B} \Phi^b(\mathbf{x}_t)\, w_b^n\Big) \quad g(\cdot) = \begin{cases} \text{Identity} & \text{if Normal} \\ \text{Softplus} & \text{if Poisson} \end{cases} \tag{33}$$

$$\text{(Spikes)} \quad y_t^n \sim \text{Normal}(r_t^n, \sigma^2) \quad \text{or Poisson(rate=}r_t^n\text{)} \tag{34}$$

$$\text{(Basis functions)} \quad \Phi^b(\mathbf{x}_t) = \begin{cases} \sqrt{\rho_m}\, \sin(2\pi \boldsymbol{f}_b \cdot \mathbf{x}_t) & \text{for } b = 2m, \\ \sqrt{\rho_m}\, \cos(2\pi \boldsymbol{f}_b \cdot \mathbf{x}_t) & \text{for } b = 2m+1 \end{cases} \tag{35}$$

$$\text{(Kernel eigenvalues)} \quad \rho_m = s^2 \frac{e^{-2\pi^2 \|\boldsymbol{f}_b\|^2\, l^2}}{C}, \tag{36}$$

where $\boldsymbol{f}_b$ is the b-th frequency vector, s is the output scale of the kernel, and C is the normalizer of the basis such that the eigenvalues sum to s^2. The simulation used approximately 64,000 Fourier frequencies per condition, each represented by paired sine and cosine functions.

Tuning curve and surrogate analyses. To build interpretable controls, we estimated tuning curves from binned spike counts and aligned behavioral labels using a supervised tuning-curve estimator. Tuning curves were computed as Gaussian-smoothed spike-count sums divided by Gaussian-smoothed occupancy. For the head direction (HD) data, HD was divided into 60 circular bins of width 6°. The standard deviation for smoothing was selected using two-fold cross-validation, with successive 10-s blocks assigned alternately to the two folds and held-out Poisson log likelihood used for selection. The selected widths were 6° for HD, 9° for HD plus angular velocity, and 6° for HD plus angular acceleration; the HD-plus-gain model inherited the HD-only width. For the HD + angular acceleration model, the angular acceleration was binarized into negative versus nonnegative sign and treated as categorical. For the HD + angular velocity model, smoothed angular velocity was discretized into six signed categories (three negative, three positive), with a small range around zero excluded. For the medial entorhinal cortex (MEC) controls, we fit multidimensional tuning maps using position, speed, head direction, and population gain. Position was divided into a 24×24

grid spanning the observed x and y ranges; speed used 16 bins, and HD used 36 circular bins. The selected standard deviations for smoothing were approximately 4 cm for each position coordinate, 10.74 cm/s for speed, 8.31 and 8.91 cm/s for the two velocity components, and 40° for HD.

Surrogate spike trains were then generated by evaluating the fitted tuning curves at the observed labels and sampling Poisson counts from the resulting expected rates.

Gain model. For the gain-control surrogate for head direction, we fit a multiplicative HD tuning model (Ajabi et al., 2023; Goris et al., 2014; Rabinowitz et al., 2015)

$$y_{t,n} \sim \mathrm{Poisson}(\lambda_{t,n}), \qquad \lambda_{t,n} = g_t f_n(x_t),$$

where $g_t > 0$ is a time-varying population gain, $f_n(\cdot)$ is the tuning curve of neuron n, and x_t denotes the head direction at time t. Inference alternated between updating gain and tuning. Given tuning, gain was updated from the ratio of total population count to total predicted population drive at the current HD bin, then normalized to have mean 1 to remove the gain–tuning scale degeneracy. Given gain, tuning curves were recomputed similarly as before, with the only difference that the occupancy now is weighted by the gain. The gain-weighted occupancy of each HD bin E_b was formed by summing the gain from times when the HD fall into bin b. The occupancy was then smoothed across HD bins into $\tilde{E}_b$, using the same circular Gaussian smoother as for the spike counts. The rest of the steps were the same. After several alternating updates, surrogate spikes were sampled from the fitted rates $\hat{\lambda}_{t,n} = g_t f_n(x_t)$. For MEC data, gain was estiamted from the full position + speed + HD model, then held fixed across gain-containing surrogate conditions.

Tuning curve alignment with VAE decoder directions. We evaluated whether directions predicted by the fitted tuning models were contained in the leading output directions of VAEs. For the HD analysis, the fitted HD plus gain tuning curve was evaluated on a uniform grid of 720 head directions at 0.5° spacing and at five gain values given by the 10th, 30th, 50th, 70th, and 90th percentiles of the fitted gain trace. For MEC, the joint position–speed–HD tuning map was evaluated on a preselected interior grid spanning both position coordinates, speeds of 5, 10, 20, 30, and 40 cm/s, 18 regularly spaced head directions, and five fitted gain quantiles. MEC grid points were retained only when all 16 surrounding grid points had sufficient behavioral occupancy.

The tuning values needed at these evaluation points generally fell between the centers of the bins used to estimate the tuning curves. For HD, let $\boldsymbol{f}_b \in \mathbb{R}^N$ denote the population tuning vector at HD bin center θ_b, where N is the number of neurons. For θ between θ_b and θ_{b+1}, the tuning value was obtained by periodic linear interpolation:

$$\boldsymbol{f}(\theta) = (1-\alpha)\boldsymbol{f}_b + \alpha \boldsymbol{f}_{b+1}, \qquad \alpha = \frac{\theta - \theta_b}{\theta_{b+1} - \theta_b},$$

where the bin index wraps circularly at 2π.

For MEC, let $\boldsymbol{f}_{i,j,k,l} \in \mathbb{R}^N$ denote the population tuning value at position bins i and j, speed bin k, and HD bin l. Within a grid cell, define the fractional positions

$$\alpha_x = \frac{x - x_i}{\Delta x}, \qquad \alpha_y = \frac{y - y_j}{\Delta y}, \qquad \alpha_s = \frac{s - s_k}{\Delta s}, \qquad \alpha_\theta = \frac{\theta - \theta_l}{\Delta \theta},$$

where x_i, y_j, s_k, and θ_l are the lower bin centers. For each coordinate, define $w_x(0) = 1 - \alpha_x$ and $w_x(1) = \alpha_x$, and analogously for w_y, w_s, and w_θ. Multilinear interpolation then gives

$$\boldsymbol{f}(x, y, s, \theta) = \sum_{a,b,c,d \in \{0,1\}} w_x(a) w_y(b) w_s(c) w_\theta(d) \boldsymbol{f}_{i+a, j+b, k+c, l+d} \in \mathbb{R}^N.$$

HD indices were wrapped periodically.

At each evaluation point, the tuning model produced the mean population activity

$$\boldsymbol{m}(\boldsymbol{q}, g) = g\, \boldsymbol{f}(\boldsymbol{q}) \in \mathbb{R}^N,$$

where $\boldsymbol{q} = \theta$ for HD and $\boldsymbol{q} = (x, y, s, \theta)$ for MEC. This mean activity was passed through the encoder of the VAE trained on real data, and the posterior mean was used as the latent representation.

We evaluated the decoder Jacobian at that latent point and used its left singular vectors as the local output directions.

Derivatives of the tuning model were computed directly from differences between neighboring tuning bins. For HD, the derivative was constant between adjacent bin centers:

$$\boldsymbol{f}'(\theta) = \frac{\boldsymbol{f}_{b+1} - \boldsymbol{f}_b}{\theta_{b+1} - \theta_b} \in \mathbb{R}^N,$$

with circular wrapping between the final and first bins.

For MEC, each partial derivative was evaluated by varying one coordinate while holding the others fixed. For example, to compute the derivative with respect to x at an evaluation point (x, y, s, θ), we first interpolated over y, speed, and HD on each of the two neighboring x bin centers, producing population tuning vectors $\boldsymbol{f}_{\mathrm{L}}(y, s, \theta)$ and $\boldsymbol{f}_{\mathrm{R}}(y, s, \theta)$. The derivative was then

$$\frac{\partial \boldsymbol{f}}{\partial x}(x, y, s, \theta) = \frac{\boldsymbol{f}_{\mathrm{R}}(y, s, \theta) - \boldsymbol{f}_{\mathrm{L}}(y, s, \theta)}{\Delta x} \in \mathbb{R}^N.$$

Derivatives with respect to y, speed, and HD were computed analogously, with circular wrapping for HD.

For the multiplicative mean model, the behavioral and gain directions were

$$\boldsymbol{d}_{q_j} = \frac{\partial \boldsymbol{m}}{\partial q_j} = g\,\frac{\partial \boldsymbol{f}}{\partial q_j} \in \mathbb{R}^N, \qquad \boldsymbol{d}_g = \frac{\partial \boldsymbol{m}}{\partial g} = \boldsymbol{f}(\boldsymbol{q}) \in \mathbb{R}^N.$$

For each derivative vector $\boldsymbol{d} \in \mathbb{R}^N$, alignment with the leading k decoder directions was quantified as

$$A_k(\boldsymbol{d}) = \frac{\|\boldsymbol{U}_k^\top \boldsymbol{d}\|_2}{\|\boldsymbol{d}\|_2},$$

where $\boldsymbol{U}_k \in \mathbb{R}^{N \times k}$ contains the leading k left singular vectors of the decoder Jacobian. Alignment was computed separately at each behavioral grid point and gain level before averaging.

As a reference, we generated 2,000 random subspaces of the same ranks. Each random orthonormal basis was obtained by applying QR decomposition to a Gaussian random matrix. Within each null draw, the same random basis was used across all behavioral locations and variables, and the projection statistic was averaged in the same way as the observed alignment. This null measures the alignment expected for a randomly oriented subspace of the same dimension and does not include uncertainty from fitting the tuning curves or the VAE.

Bird song analysis We analyzed the zebra finch song dataset of Brudner et al. (2023), available through the Duke Research Data Repository (`https://doi.org/10.7924/r4j38x43h`). The authors recorded five juvenile males during song development after paternal tutoring, housing each bird individually in a microphone-equipped sound box. They segmented recordings into individual sounds and converted them into normalized 128×128 log-magnitude spectrograms with mel-spaced frequency bins. They used the convolutional VAE comprising a convolutional encoder and a decoder with fully connected and transposed-convolution layers, using ReLU activations and batch normalization. The model used 32 latent dimensions, a Gaussian posterior with diagonal-plus-rank-one covariance, a standard normal prior, and an isotropic Gaussian observation model with fixed pixel variance 0.1, trained with the standard evidence lower bound. A separate VAE was trained for each bird on 30,000 sounds sampled across development. We reused these pretrained models without fine-tuning. We reused the released pretrained VAEs, posterior latent means, and syllable labels to evaluate developmental changes in VAE-ID across 13 bird-specific syllable types between 60 and 95 days post-hatch. We retained syllable-days containing at least 80 renditions and sampled 40 renditions per syllable-day, balanced across four equal-count groups ordered by recording time. This yielded 14,600 evaluation points across 365 syllable-days. We computed VAE-ID for each sampled rendition and averaged across renditions and syllable types to obtain one daily value per bird. Bird-specific developmental trends were summarized by linear regression of daily mean VAE-ID against age.

A.3 ADDITIONAL ID ESTIMATORS ON SIMULATED DATA

We tested additional estimators in the following categories: 1) directly computing statistics from the observations (Figure A.1A); 2) fitting a diagonal-posterior VAE first, then either estimating from the decoder Jacobian or the latent posterior mean ((Figure A.1B); 3) similar to 2), but using a full-posterior VAE. The observation-based ones perform poorly overall, grossly overestimating the dimension in the presence of noise. Full-posterior VAE performs better than diagonal-posterior VAE, both performing much better. Applying the neighborhood-based methods like CorrInt and ABID on VAE posterior means also perform decently (although still underperforming compared to the proposed method Jacobian effective rank), highlighting the role of a latent variable in denoising and aiding the discovery of structure.

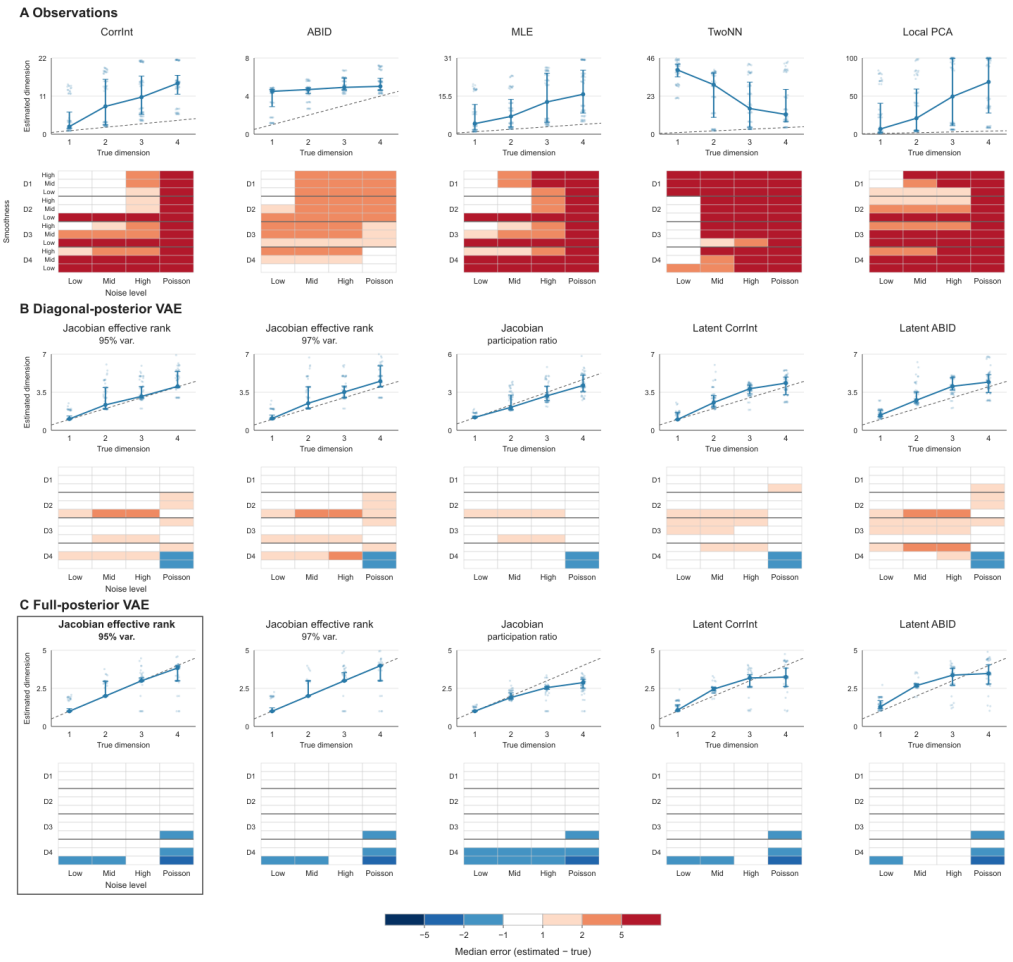

Figure A.1: **Additional estimators on simulations of neural data.** Similar to Figure 2B-C. Each column is one estimator, grouped into blocks: (A) estimators that directly use the observations, (B) estimators that fit a diagonal-posterior VAE first, and (C) estimators that fit a full-posterior VAE first. Top row: Estimated dimension vs true dimension. Each dot is one simulation. Blue solid lines mark the median. Error bars mark the interquartile range. The dotted line marks the identity line. Bottom row: Heatmaps of the median signed error of different methods and simulation conditions.

A.4 Robustness to downsampling

To answer the question how VAE-ID performs with different numbers of neuron or time samples, we downsampled time or neurons in two simulations (3D, middle smoothness, high vs low Guassian observation noise). The performance improves with increasing the number of observations or neurons, which is desirable, as collecting more data can help with the estimate (Figure A.1A-D). In contrast, the neighborhood-based methods CorrInt and ABID do not show a scaling law (Figure A.1A-D). For VAE-ID, the performance saturates with 30 neurons or 10000 neurons in the current simulations, both of which are usually achieved in today's population recordings.

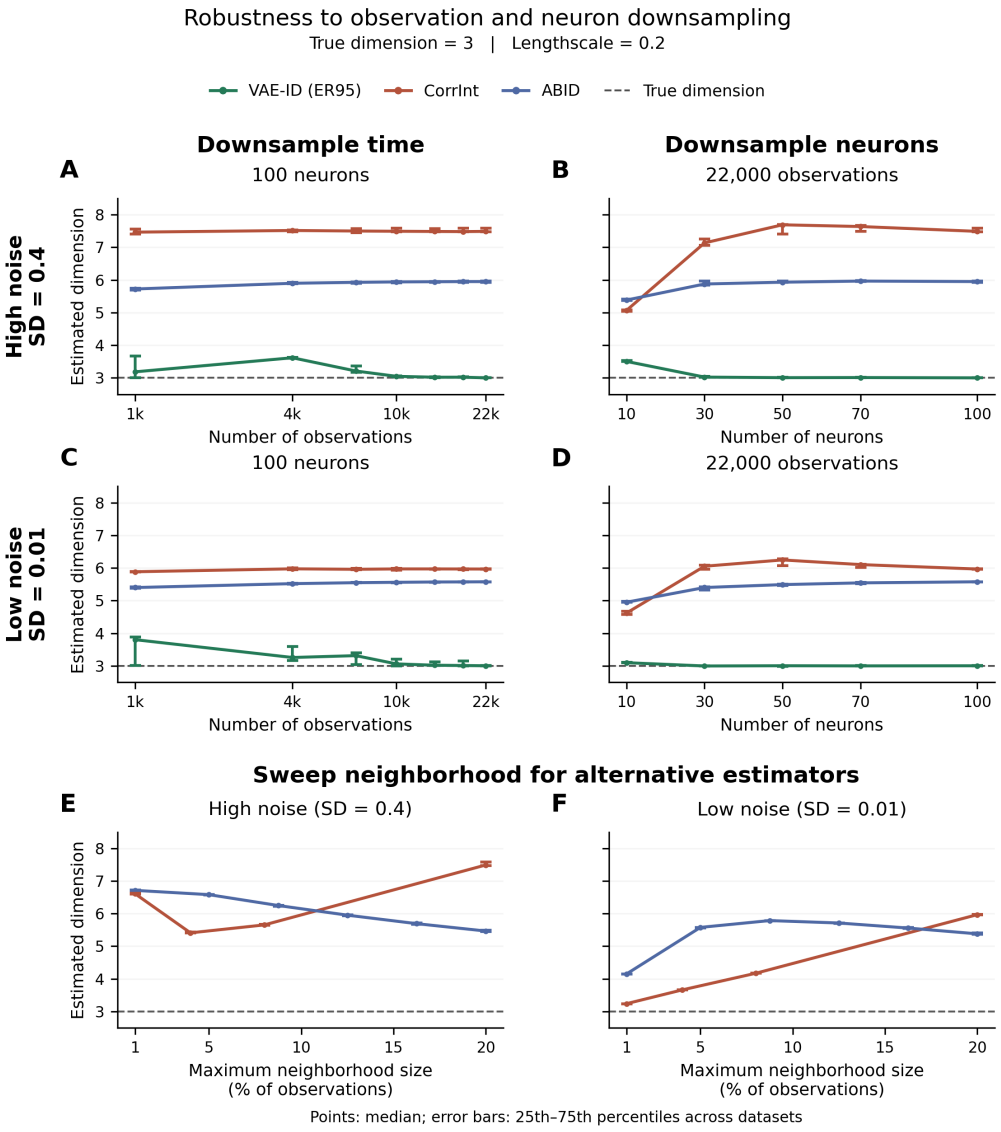

Figure A.2: **Downsampling and neighborhood sweep (A-D)** Dimension estimates vs the number of observations (A and C) or neurons (B and D), for simulated 3D data with high noise (A and B) or low noise (C and D). **(E-F)** Dimension estimates vs maximum neighborhood size for the neighborhood-based estimators, for high noise (E) and low noise (F) simulations. For CorrInt, the minimum neighborhood size is chosen to be $1/4$ of the maximum size.

A.5 MITIGATING ARCHITECTURAL DEPENDENCY

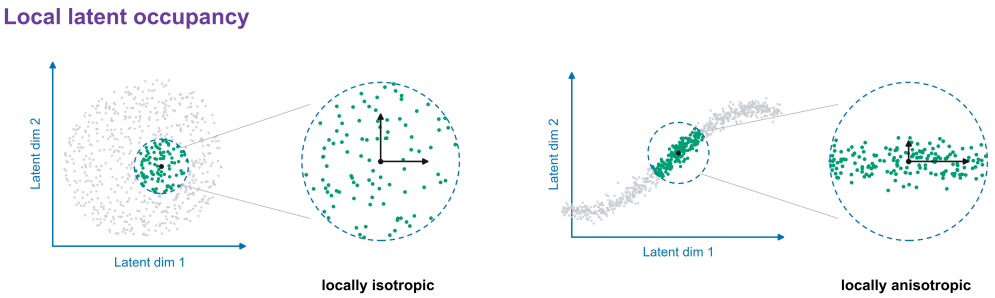

Figure A.3: **Additional correction for VAE-ID.** The latent posterior means might not occupy the full space, limiting the effective output dimensions.

We chose the current VAE architecture based on exploration on a subset of simulations and found that it was accurate across a wide range of conditions. Nevertheless, different architectures, including changes in network depth or activation function, produced different ID estimates on the real HD data (Table 2). One possible explanation is that, although the VAE objective implicitly discourages redundant latent dimensions, certain architectures impose inductive biases that can still favor solutions with additional decoder-output directions. We therefore developed a local covariance correction to reduce the dependence of the ID estimate on VAE architecture.

The uncorrected estimator examines the decoder Jacobian J_i at a latent point z_i and asks how many orthogonal directions in neural activity are needed to describe its local output variation. This calculation treats every latent direction as available. It therefore characterizes what the decoder *could* express locally, rather than what the encoded data actually explore. A decoder may, for example, be sensitive to a particular latent direction even though nearby observations vary little along that direction (Figure A.3).

The covariance correction accounts for this distinction between possible and occupied variation. Around each z_i, we identify a fixed local neighborhood of encoded observations—1% of the data in our analyses—and compute the covariance C_i of their posterior means. The eigenvectors of this covariance identify the latent directions traversed by nearby data, while its eigenvalues quantify the amount of variation along those directions. Rather than computing the effective rank of J_i alone, we compute it from the composed map

$$J_i \left(C_i\right)^{1/2}. \tag{37}$$

This expression has a simple interpretation. The covariance factor first converts an isotropic perturbation into one with the shape and scale of the local latent cloud. The Jacobian then maps that locally occupied perturbation into neural activity space. Consequently, a direction to which the decoder is highly sensitive contributes little if nearby data scarcely vary along it. A direction contributes strongly only when it is both occupied in latent space and amplified by the decoder. We define the corrected ID as the smallest number of singular directions that account for 95% of the total squared singular-value of this composed map. This correction reduces the architectural variation of ID estimate on the HD data by 43%. Note that the corrections did not meaningfully change the estimates from default 3-layer ReLU network, supporting our choice of this architecture as the default.

Table 2: Intrinsic-dimension estimates on real HD data across architectures and correction methods.

		ID estimate		
Activation	Hidden layers	Jacobian	Covariance correction	Validation ELBO
ReLU	1	4.3	3.4	-0.392
ReLU	2	3.3	3.0	-0.389
ReLU	3	2.1	2.0	-0.392
SiLU	1	4.7	3.5	-0.395
SiLU	2	3.7	3.1	-0.387
SiLU	3	2.8	2.6	-0.389
Range		2.1–4.7	2.0–3.5 (43.6%)	—