

Cryo-EM Images are Intrinsically Low Dimensional

Luke Evans^{1,*}, Octavian-Vlad Murad^{2,*}, Lars Dingeldein³, Pilar Cossio^{1,4},
Roberto Covino^{5,3} and Marina Meila²

¹Center for Computational Mathematics, Flatiron Institute, New York, New York, USA

²Department of Statistics, University of Washington, Seattle, Washington, USA

³Frankfurt Institute for Advanced Study, Frankfurt, Hesse, Germany

⁴Center for Computational Biology, Flatiron Institute, New York, New York, USA

⁵Institute of Computer Science, Goethe University Frankfurt, Frankfurt, Hesse, Germany



(Received 16 April 2025; accepted 13 August 2025; published 22 September 2025)

Simulation-based inference provides a powerful framework for cryoelectron microscopy, employing neural networks in methods like CryoSBI to infer biomolecular conformations via learned latent representations. This latent space represents a rich opportunity, encoding valuable information about the physical system and the inference process. Harnessing this potential hinges on understanding the underlying geometric structure of these representations. We investigate this structure by applying manifold learning techniques to CryoSBI representations of a simulated benchmark dataset and both simulated and experimental images of hemagglutinin. We reveal that these high-dimensional data inherently populate low-dimensional, smooth manifolds, with simulated data effectively covering the experimental counterpart. By characterizing the manifold's geometry using Diffusion Maps and identifying its principal axes of variation via coordinate interpretation methods, we establish a direct link between the latent structure and key physical parameters. Discovering this intrinsic low-dimensionality and interpretable geometric organization not only validates the CryoSBI approach but also enables us to learn more from the data structure and provides opportunities for improving future inference strategies by exploiting this revealed manifold geometry.

DOI: [10.1103/txrb-fw3z](https://doi.org/10.1103/txrb-fw3z)

I. INTRODUCTION

Cryogenic electron microscopy (cryo-EM) is a structural biology technique for imaging individual biomolecules at atomic resolution. In a cryo-EM experiment, a biomolecular sample is imaged with a transmission electron microscope, and the resulting data are processed to yield a large dataset of unlabeled two-dimensional (2D) images with one molecule per image (particles). Reconstruction algorithms [1] can estimate the 3D structure of the biomolecule from the 2D particles. In many cases, biomolecules coexist in different conformational states in the sample.

Machine learning methods, including Diffusion Maps [2] and deep-generative models [3–5], have become central in cryo-EM for reconstructing heterogeneous conformations of biomolecules [6,7]. These methods project the high-dimensional conformational space on to a low-dimensional latent representation, but these latent spaces lack interpretability [8]. Some of these challenges can be addressed by incorporating physical interpretability during training [9] or by using physics-based priors in the model [10]. Additionally, analyses based solely on simulated data have also provided

valuable insights [11]. However, extracting physical and geometrical information from the featured images remains challenging due to nonlinear feature mapping, low signal-to-noise ratio (SNR) and uncertainty in pose assignment, which can be confused with conformational changes.

Recent simulation-based techniques from integrative structural biology [12] and probabilistic machine learning [13] hold great promise for analyzing cryo-EM data. CryoSBI [14] is an emerging paradigm using simulation-based inference [15,16] (SBI) to infer conformations and uncertainties from cryo-EM particles by training neural networks producing a latent representation and a density estimator with simulated cryo-EM experiments (Fig. 1). The trained networks can be quickly evaluated on large experimental particle datasets. Because the training is only done with simulated data, a key feature of cryoSBI is that it enables linking of physical properties of the molecules and the experiment to experimental data.

Supported by preliminary evidence [14], we hypothesize that the representations learned by the neural network are near low-dimensional manifolds inside the latent space. The objective of this work is to study the geometry of the data using manifold learning techniques [17–20]. First, we will seek to ascertain whether the learned representations correspond to well-behaved low-dimensional manifolds and, second, whether these are parameterized by generative variables important in predicting the posterior over the conformation. Our analysis quantitatively validates the latent space of cryoSBI and leads to a general computational workflow both for interpreting latent spaces of cryo-EM heterogeneity analysis

*These authors contributed equally to this work.

Published by the American Physical Society under the terms of the Creative Commons Attribution 4.0 International license. Further distribution of this work must maintain attribution to the author(s) and the published article's title, journal citation, and DOI.

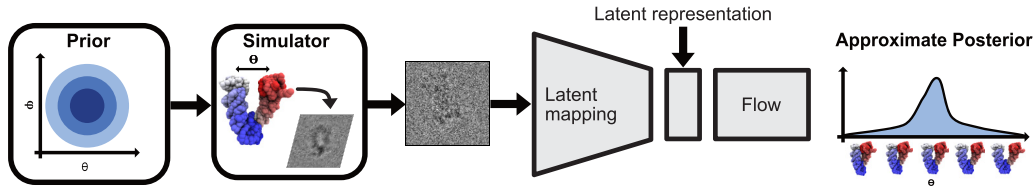


FIG. 1. Schematic workflow for learning the surrogate posterior with cryoSBI. Parameter samples are drawn from the prior to simulate synthetic cryo-EM images. These images are then used to approximate the posterior by jointly training a summary network and a normalizing flow.

methods and more broadly for learned summary statistics in simulation-based inference.

II. CRYOSBI AND LATENT SPACES

CryoSBI [14] is a new method to quantify the probability that a given image I depicts a molecular conformation θ . We assume to have a set of structures, e.g., from molecular simulations or AI methods [21], which we expect to find in the sample. For simplicity, we also assume that θ is a one-dimensional parameter, and we aim to infer the conformation θ of the molecule observed in the image, i.e., compute the Bayesian posterior $p(\theta|I)$. The posterior quantifies how compatible θ is with the observed image I .

To model the image formation process, one must consider experimental details such as microscope aberration, noise, and random orientation of the molecule. To simulate a cryo-EM image, one samples conformations from the prior $\theta_i \sim p(\theta)$, and imaging parameters from $\phi_i \sim p(\phi)$ and then generates a synthetic image $I_i \sim p(I|\theta_i, \phi_i)$ using a forward model of the imaging process (Appendix A), accumulating a dataset of simulated images and ground-truth parameters $\mathcal{D} = \{\theta_i, \phi_i, I_i\}_{i=1}^N$. The nuisance parameter vector ϕ_i includes random orientations, a wide range of defocus values, center translations, and SNRs.

A. Feature latent representation and neural posterior estimation

CryoSBI builds on the Neural Posterior Estimation framework [22,23], jointly training a latent representation network $S_\psi(\cdot)$ to extract summary statistics and a normalizing flow $q_\phi(\cdot)$ as surrogate model of the posterior $q_\phi(\theta|S_\psi(I)) \approx p(\theta|I)$. This is done by maximizing the average log-likelihood $\mathcal{L}(\varphi, \psi) = \frac{1}{N} \sum_{i=1}^N \log q_\phi(\theta_i|S_\psi(I_i))$ of the posterior probability under the training samples $\mathcal{D}$ (Appendix B). In principle, after training, S_ψ should (i) compress images to predict the relevant features and (ii) enable efficient comparison of simulated images to “nearby” experimental images. For example, the latent representation should distinguish images due to conformation, SNR, and projection direction, as these are the primary experimental factors determining how precisely we can estimate a molecular configuration from a single image. Distinguishing these factors is another step towards indicating physical properties of the molecule, such as symmetries affecting the pose or conformation estimates. In practice, while the feature representation for cryoSBI [14]—and more generally for Neural Posterior Estimation—offers powerful inference capabilities, it is not immediately interpretable, making it challenging to check for model misspecification [24].

B. Datasets

We consider latent representations from two different biomolecular systems: the human immunoglobulin G antibody (IgG) and influenza hemagglutinin. The IgG dataset was recently utilized as an example of large conformational changes (the “IgG-1D” dataset) in the cryo-EM heterogeneity benchmark CryoBench [11]. We applied cryoSBI to this benchmark system and analyzed the latent representations of simulated images derived from the IgG atomic models. CryoSBI training was performed as in Ref. [14] using cryo-EM simulations by sampling the priors with ranges specified in (Appendix C). After training, we created a simulated dataset $\mathcal{D}_s$ consisting of $N_s = 100\,000$ feature vectors with i th datapoint $x_i = S_\psi(I_i) \in \mathbb{R}^{256}$, nuisance parameters ϕ_i , ground-truth conformation parameter θ_i , posterior mean $\hat{\theta}_i$, and width σ_i of the posterior $q_\phi(\cdot|x_i)$, so that $\mathcal{D}_s = \{x_i, \hat{\theta}_i, \sigma_i, \theta_i, \phi_i\}_{i=1}^{N_s}$. We denote the representations learned by S_ψ , $\mathcal{X}_s = \{x_i\}_{i=1}^{N_s}$.

The hemagglutinin data corresponds to the hemagglutinin dataset considered in Ref. [14]; it consists of the CryoSBI latent representations of the simulated and experimental images. CryoSBI training was performed as in Ref. [14] by sampling the priors (Appendix C) and simulating from them. Likewise, we created a simulated dataset $\mathcal{D}_s$ consisting of $N_s = 100\,000$ feature vectors with i th datapoint $x_i = S_\psi(I_i) \in \mathbb{R}^{256}$, nuisance parameters ϕ_i , ground-truth conformation parameter θ_i , posterior mean $\hat{\theta}_i$, and width σ_i of the posterior $q_\phi(\cdot|x_i)$, so that $\mathcal{D}_s = \{x_i, \hat{\theta}_i, \sigma_i, \theta_i, \phi_i\}_{i=1}^{N_s}$. The experimental dataset $\mathcal{D}_e$ consists of $N_e = 271558$ tuples $\mathcal{D}_e = \{\tilde{x}_i, \hat{\theta}_i, \sigma_i\}_{i=1}^{N_e}$ with $\tilde{x}_i = S_\psi(\tilde{I}_i)$, for whitened single particle-images $\{\tilde{I}_i\}_{i=1}^{N_e}$ from EMPIAR 10532 [25], where $\hat{\theta}_i, \sigma_i$ are the inferred posterior parameters (note that the experimental images have no ground truth θ or ϕ). We denote the representations learned by S_ψ , $\mathcal{X}_s = \{x_i\}_{i=1}^{N_s}$ and $\mathcal{X}_e = \{\tilde{x}_i\}_{i=1}^{N_e}$.

III. MANIFOLD ANALYSIS PIPELINE

To study the representations learned by the feature mapping S_ψ , we apply a framework of manifold learning techniques to the datasets $\mathcal{X}_s$ and $\mathcal{X}_e$ (Appendix D). We proceed by a pipeline of preprocessing the data, estimating the intrinsic dimensionality, and then embedding to a lower-dimensional manifold.

A. Data preprocessing

The data preprocessing consists of subsampling, removing outliers, and resampling the data to avoid large variation in density. The data subsampling and outlier removal ensure

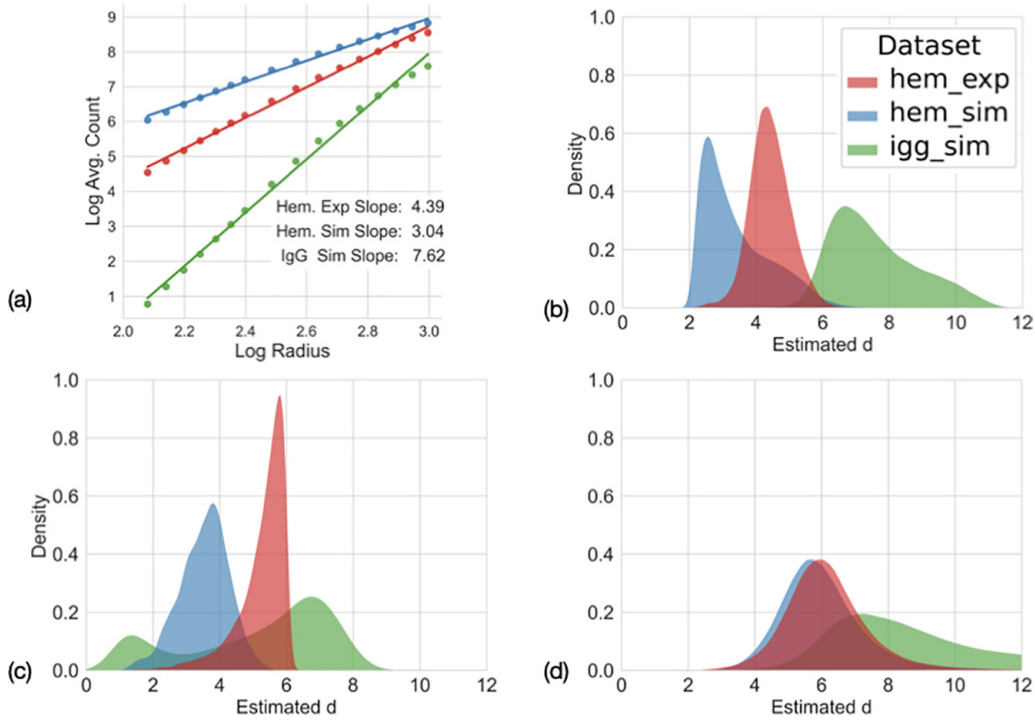


FIG. 2. Intrinsic dimensionality estimation results for the experimental (red) and synthetic (blue) hemagglutinin datasets, and the synthetic IgG dataset (green). (a) *Correlation dimension*: The slope of the fitted line yields a single global prediction $\hat{d}$ for each dataset. (b) *Doubling dimension*: Distribution of the pointwise estimates $\hat{d}(x)$ averaged over a range of radii R . (c) *Eigengap*: distribution of the pointwise intrinsic dimension estimates $\hat{d}(x)$. (d) *Levina-Bickel*: distribution of the pointwise estimates $\hat{d}(x)$ averaged over a range of nearest neighbors K .

both computational scalability and statistical robustness in our methods, and the resampling encourages a more uniform distribution of data, which is known to help debias both dimension estimation and manifold learning techniques [20]. Our full process is described in detail in Appendix E.

B. Dimensionality estimation

Intrinsic dimension estimation is challenging when the data contain noise in the high-dimensional ambient space. To ensure robustness of our findings, we employ four different estimators. The first returns a global dimension estimate $\hat{d}$, and the latter three return a local dimension estimate $\hat{d}(x)$ for any given data point x ; in this second case, the estimate $\hat{d}$ is the mode of the $\hat{d}(x)$ distribution.

The four algorithms we use are as follows:

Correlation dimension [26]. The number of neighbors within a radius r of a datapoint x satisfies $n_r(x) \propto r^d$, implying $\log n_r(x) \propto d \cdot \log r$. For a range of radii $R = \{r_1, r_2, \dots, r_m\}$, we compute the average number of neighbors $\bar{n}_r$ and fit a linear model to $\{(\log r, \log \bar{n}_r)\}_{r \in R}$. The slope yields a single global estimate $\hat{d}$ [Fig. 2(a)].

Eigengap method [27]. For each x , we compute a weighted local covariance matrix and obtain its truncated spectrum $\lambda_1 \geq \dots \geq \lambda_D$ for some conservatively large $D > d$. Then $\hat{d}(x)$ is given by the largest gap between consecutive eigenvalues. This method is the only one of the four designed to take into account the presence of noise [Fig. 2(b)].

Doubling dimension [28]. By the same scaling principle as correlation dimension, we can instead compute the local statistic $\hat{d}_r(x) = \log_2(n_{2r}(x)/n_r(x))$ for some fixed $r \in \mathbb{R}^+$

(Fig. 4 of the Supplemental Material [35]). These local estimates can be averaged over a range of radii R to yield more robust pointwise predictions [Fig. 2(c)].

Levina-Bickel MLE [29]. This method infers d from the top k -nearest neighbor distances as $\hat{d}_k(x) = [\frac{1}{k-1} \sum_{j=1}^{k-1} \log(\text{dist}_k(x)/\text{dist}_j(x))]^{-1}$ for some fixed integer k (Fig. 5 of the Supplemental Material [35]). These local estimates can be averaged over a range of numbers of neighbors K to yield more robust pointwise predictions [Fig. 2(d)].

In Appendix F and Appendix G we provide details about the algorithms, their implementation in this work, and practical guidelines for their use.

C. Manifold embedding

For embedding the high-dimensional datasets to lower dimensions, we apply Diffusion Maps [17] to our preprocessed data. To refine our initial embedding, we remove redundant embedding directions by applying independent eigencoordinate selection (IES) [18]. We then correct the distortion introduced by the embedding algorithm to the original distances within the manifold, through the procedure known as Riemannian relaxation [30]. Appendix H details our embedding pipeline.

IV. GEOMETRIC ANALYSIS OF THE CRYOSBI LATENT SPACE

Now, we proceed to study the shape of the data cloud $\mathcal{X}_e$, determine the intrinsic dimensionality of the datasets,

TABLE I. Test data negative log-likelihoods under $\hat{p}_e$ and $\hat{p}_s$.

	$-\frac{1}{n} \log \hat{p}_m(\mathcal{X}_m^{\text{test}})$	
	$\mathcal{X}_e^{\text{test}}$	$\mathcal{X}_s^{\text{test}}$
$\hat{p}_e$	86.6	2604.3
$\hat{p}_s$	175.6	186.4

assess how well the simulated data covers the experimental space, and uncover the physical interpretation of the latent representations.

A. Are the data low dimensional?

We first estimate the intrinsic dimension for the synthetic IgG dataset and the experimental and synthetic hemagglutinin datasets. Figure 2 summarizes our main result: All four methods indicate that the learned cryo-EM embeddings have *low intrinsic dimension*.

Specifically, we find that $6 \leq d_s \leq 10$ for the synthetic IgG dataset and $2 \leq d_s \leq d_e \leq 6$ for the hemagglutinin datasets. The higher dimensionality for simulated IgG data than the simulated hemagglutinin is likely due to the larger conformational change in the IgG pdb. The slight difference between d_s and d_e for hemagglutinin likely reflects experimental noise and other imaging factors not captured by the simulated model. Taken together, these results strongly support the conclusion that the manifold assumption holds for our datasets.

B. Does the simulated hemagglutinin data cover the experimental data well?

For the amortized simulation-based-inference in CryoSBI, the simulator must be able to generate many relevant experimental realizations, so that a particular experiment can be accurately analyzed without retraining. Our dimensionality results above indicate that the simulated and experimental hemagglutinin data lie on low-dimensional manifolds but do not inform whether the manifolds are close to each other or if the simulated data covers the experimental; in other words, if the experimental hemagglutinin data are in the distribution of simulated ones.

We investigate the covering by density estimation of both hemagglutinin datasets. For this, we first estimate the data densities p_e and p_s in $\mathbb{R}^{256}$ by kernel density estimators (KDE) [31] $\hat{p}_e$ and $\hat{p}_s$.

The bandwidths $h_e = 0.31$ and $h_s = 0.48$ are obtained by cross-validation, as detailed in Appendix G. While it is known that KDE is poor in high dimensions, the method is *adaptive*, meaning that it will work when the intrinsic dimension is low, as in this case. We use samples of size 17 000 for fitting $\hat{p}_e$ and $\hat{p}_s$. We do not expect p_e to equal p_s , but we would like to confirm that p_s is predictive of the experimental data. Thus, on two held out datasets $\mathcal{X}_e^{\text{test}}$ and $\mathcal{X}_s^{\text{test}}$, with $|\mathcal{X}_s^{\text{test}}| = |\mathcal{X}_e^{\text{test}}| = n^{\text{test}} = 3000$, we calculate the negative log-likelihoods (i.e., cross-entropies) $-\frac{1}{n^{\text{test}}} \log \hat{p}_m(\mathcal{X}_m^{\text{test}})$ for $m \in \{e, s\}$ (in Table I) and the estimated Kullback-Leibler divergences $D_{\text{KL}}(\hat{p}_e || \hat{p}_s) = 89.0$, $D_{\text{KL}}(\hat{p}_s || \hat{p}_e) = 2417.9$. These show that the simulated data can predict the experimental data well; meanwhile, the experimental data does not completely

cover the simulated data. For further analysis, we retain in $\mathcal{X}_s$ only the samples that are near the experimental data. The hypothesis that we can infer what generative parameters best describe the experimental data, is so far supported since we can, for most experimental $\tilde{x} \in \mathcal{X}_e$, find enough near-by synthetic $x \in \mathcal{X}_s$ to perform this prediction in a robust manner.

C. Modeling the low-dimensional cryo-EM images manifold

We use a suite of manifold learning techniques [17,18,30,32] to map the neural representations $\mathcal{X}_e \subseteq \mathbb{R}^{256}$ of the experimental hemagglutinin data down to a much lower-dimensional embedding Φ_e , which we here interpret geometrically and in the following section from the physical point of view, in relation to the simulated hemagglutinin data $\mathcal{X}_s$.

We use Diffusion Maps [17] with a kernel width parameter ϵ selected by the method of Ref. [32] to compute the low-dimensional embedding $\Phi_e \in \mathbb{R}^d$ of $\mathcal{X}_e$ (Fig. 3); similarly, we compute Φ_s for the filtered $\mathcal{X}_s$ data for both the simulated IgG data and simulated hemagglutinin (Fig. 3, and Fig. 6 of the Supplemental Material [35], respectively). The embeddings indicate that conformation and SNR are primary factors influencing the structure of the data manifolds, and they vary in regions that closely align with the posterior mean and posterior width, respectively. See Appendix H for further practical details on the preembedding filtering and parameter selection, and Supplemental Material [35], Figs. S7 and S8 for further embeddings.

The Diffusion Maps embedding is based on the eigendecomposition of the Laplacian matrix $\mathbf{L}$ [17], and in a first stage we compute it up to the m' th nonzero eigenvalue, for $m = 20$, and denote these coordinates with $\Phi_{1:m} \in \mathbb{R}^n$, with $n = |\mathcal{X}_e|$. The analysis of the principal eigenvalues of $\mathbf{L}$, which are slowly growing and well above 0 (Supplemental Material [35], Fig. S9), indicates that the manifold $\mathcal{M}_e$ is connected and likewise for the IgG manifold (Supplemental Material [35], Fig. S10). That is, there are no isolated clusters and no outliers for the postprocessed data. However, this analysis does not rule out the presence of clusters as high-density regions, which could occur from data that are not preprocessed. As a visual example of the effect of preprocessing (Appendix E), Supplemental Material [35], Fig. S11 maps a sample from the original probability density p_e into $\mathcal{M}_e$.

Next, we perform IES [18] to select $d = 3$ independent and low-frequency coordinates from $\Phi_{1:20}$. We use these coordinates, denoted Φ_e , to visualize and interpret the experimental data (see Appendix H for more details). As shown previously, $d = 3$ is likely close to the true intrinsic dimension of $\mathcal{M}_s$ and $\mathcal{M}_e$, meaning we can expect to capture most of the relevant structure of the experimental data by analyzing these Φ_e coordinates. We apply Riemannian relaxation [30] to Φ_e to make it closer to being isometric to $\mathcal{X}_e$ for hemagglutinin. The resulting embedding is shown in Supplemental Material [35], Fig. S8. We perform similar steps with the simulated data $\mathcal{X}_s$ for hemagglutinin (Appendix H).

D. Physical interpretation of the experimental data manifold

In the absence of ground-truth generative parameters for the experimental hemagglutinin data, we have to find

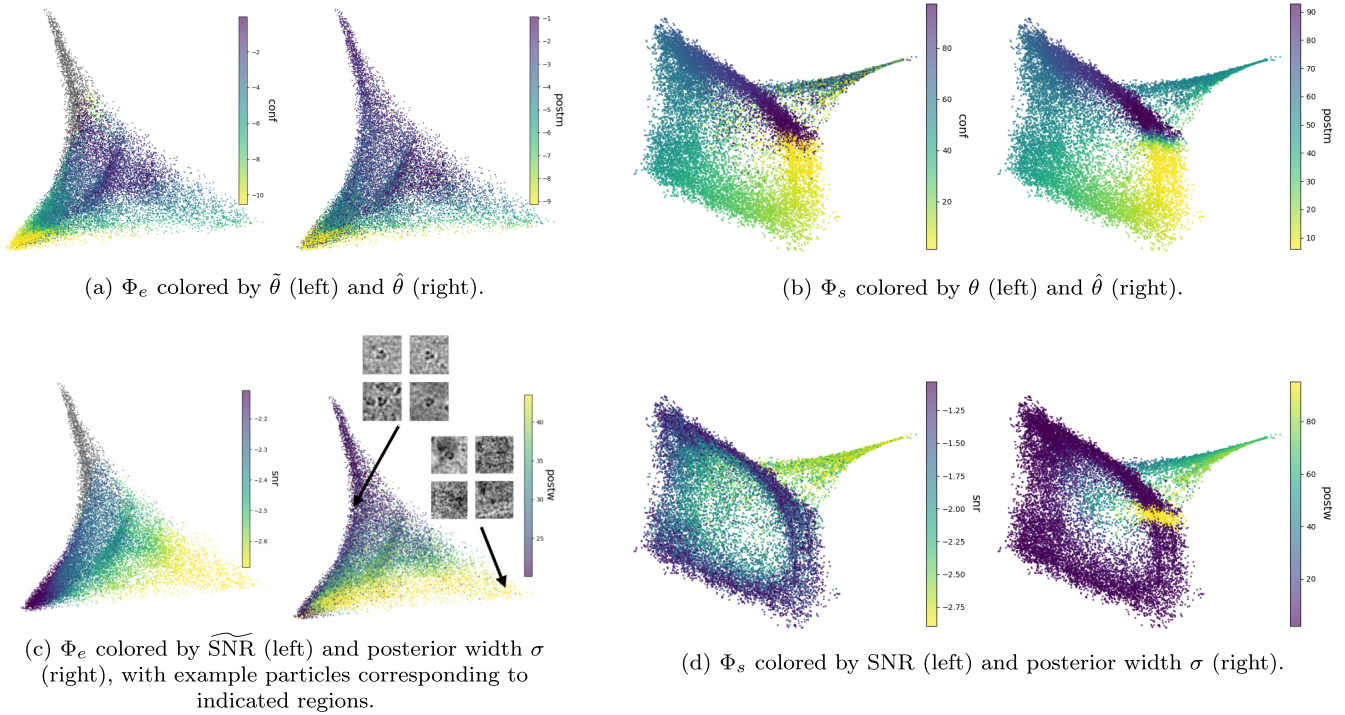


FIG. 3. Diffusion Maps embedding Φ_e learned from the experimental hemagglutinin dataset [(a) and (c)] and Φ_s learned from the simulated IgG dataset [(b) and (d)]. We display the lowest frequency $d = 3$ coordinates selected by IES and rotate the plots to best display the embedding. Importantly, the neural network predicted conformation $\hat{\theta}$ and uncertainty σ vary smoothly over the manifolds and agree with the true generative conformation θ and SNR over the embedded points. In the absence of the true generative conformation θ and SNR for the experimental data, we use $\hat{\theta}$ and $\widetilde{\text{SNR}}$ to color Φ_e , values which we infer via kernel interpolation (see Appendix I) from the simulated hemagglutinin dataset. Plots (c) and (d) have opposing color scales to emphasize regions where (high/low) SNR correspond to (low/high) posterior width.

alternative ways to determine whether S_ψ is a good predictor for the true conformational parameter θ , and the noise level, an important nuisance parameter. While this can be done with a manually labeled test set, we focus on indirect geometric methods that do not require scientific labeling. We first use a statistical method, TSLasso [19] to interpret the embedding Φ_e . Afterwards, we support its results and expand the analysis with visualizations. TSLasso searches for the optimal interpretation of an embedding in a *dictionary* $\mathcal{F} = \{f_k : \mathcal{M}_e \rightarrow \mathbb{R}, k = 1 : p\}$ of (smooth) potential coordinate functions on $\mathcal{M}_e$. Here each $f_k \in \mathcal{F}$ represents one of the simulation parameters (the conformation θ or one of the nuisance parameters in ϕ) and hence $|\mathcal{F}| = 10 = p$. TSLasso recovers a subset f_s of $\mathcal{F}$ which parametrizes $\mathcal{M}_e$, by selecting d functions whose gradients “most economically” span the tangent spaces of the manifold at a sample of the data. Since the functions f_k are unknown on the experimental data, we infer them by interpolation (Appendix I), obtaining $\hat{\theta}$ and $\hat{\phi}$ for the experimental data. We also estimate the gradients ∇f_k (Appendix J). TSLasso is run 20 times using random subsets of 500 data points. We find that f_s almost always consists of conformation θ , SNR, and one of the rotation coordinates in $\phi/\hat{\phi}$ (albeit not always the same one). The full results are presented in Appendix K. For completeness, we apply the same algorithm to the simulated data of both hemagglutinin and IgG. Our results show that this combination of functions parametrizes both $\mathcal{M}_s$ and $\mathcal{M}_e$. We have confirmed statis-

tically, without any visualization, that the two parameters θ and SNR inferred from nearby simulated data, vary smoothly along the experimental data manifold $\mathcal{M}_e$ (as well as along $\mathcal{M}_s$), therefore, supporting the neural network predictions for $\mathcal{X}_e$. The visualizations are shown in Figs. 3(a) and 3(b).

V. DISCUSSION

In summary, our study of the latent embedding representations of IgG and hemagglutinin cryo-EM data from cryoSBI has revealed that these live near a well-behaved low-dimensional manifold in $\mathbb{R}^{256}$ space. The difference in intrinsic dimensionality of the simulated data manifolds, from hemagglutinin to the more heterogeneous IgG, indicate that conformational variability affects dimensionality of the manifold. For the hemagglutinin data, the simulated hemagglutinin images cover (almost entirely) the experimental ones. Therefore, we can use the simulated data (on which we have full control) to interpret the experimental data in the latent space. Furthermore, we have identified the physical and geometrical features that explain the different directions in the latent space.

We presented visualizations (e.g., by postprocessed Diffusion Maps embedding) that accurately display the data shape by being almost isometric. We are also excited by the possibilities of replacing visual analysis with quantitative measures, and principled algorithms in creating and validating low-dimensional models of cryo-EM data. Examples of

such tasks include detecting the intrinsic dimensionality, interpreting the manifold by physical coordinates, measuring the smoothness of functions over the data manifold (not included here but straightforward via the Laplacian operator), detecting if clusters exist, and measuring local distortion [33].

From the methodological point of view, we present a pipeline for analyzing, exploring and visualizing high-dimensional data presumably living near a smooth manifold. The pipeline components integrate state-of-the-art geometric algorithms and theoretical results. However, we note that we do not propose to replace the trained neural network predictor with (a variant of) the methods presented here. Typically, dimension reduction methods do not outperform a neural network trained in supervised mode. What our method offers is interpretability of the latent representations and a connection of the experimental data to the physical simulator.

At the same time, we acknowledge that the data might not align perfectly with the manifold hypothesis. Our current understanding does not yet enable us to predict, comprehend, or control how finer-scale data structures—e.g., what we consider “noise”—affect geometric algorithms, which should be a matter of further investigation.

We note that our work here investigates only one experimental dataset, and only a synthetic example of a large conformational change. We showed that large conformational change appears to increase the intrinsic dimensionality, but other factors such as nonwhite noise and background effects could play a role. We anticipate that our protocols will be robust when applied to data with nonuniform pose distributions or nonuniform conformational distributions.

In future work, one may investigate if the settings of cryoSBI, such as the chosen dimension of the latent space, choice of priors on imaging parameters, and size of the training set affect the geometry, topology, and intrinsic dimension of the embedding.

ACKNOWLEDGMENTS

The Flatiron Institute is a division of the Simons Foundation. L.D. and R.C. acknowledge the support of Goethe University Frankfurt, the Frankfurt Institute of Advanced Studies, the LOEWE Center for Multiscale Modelling in Life Sciences of the state of Hesse, the CRC 1507: Membrane-associated Protein Assemblies, Machineries, and Supercomplexes (P09), and the International Max Planck Research School on Cellular Biophysics. L.D. and R.C. thank the Flatiron Institute, and M.M. gratefully acknowledges the DataShape Group at INRIA Saclay for hospitality while a portion this research was carried out. This project was initiated out of discussions at the “Data Driven Materials Informatics” long program at the Institute for Mathematical and Statistical Innovation (IMSI) and was further motivated by the NeurIPS workshop “Machine Learning and Structural Biology.” The authors would like to thank the organizers and IMSI for fostering this collaboration.

DATA AVAILABILITY

The data that support the findings of this article are openly available [34].

APPENDIX A: CRYO-EM IMAGE FORMATION FORWARD MODEL

We simulate cryo-EM particles from 3D molecular structures with the forward model in Refs. [36,37]. The electron density $\rho(X)$ of a given structure X is approximated as a Gaussian mixture model with centers on the positions of the C_α atoms, and standard deviations γ . Then, we apply a rotation R_q with quaternion q and projection P_z onto the z axis to $\rho(X)$ and then convolve with a point-spread function (PSF), which incorporates the microscope defocus and aberration. The PSF is more straightforward to apply in Fourier space, where the convolution becomes a pointwise multiplication with the Fourier transform of the point-spread function, known as the contrast transfer function (CTF). The CTF is defined as $\text{CTF}_{A,b,\Delta z}(s) = e^{-bs^2/2} [A \cos(\pi \Delta z \lambda_e s^2) - \sqrt{1 - A^2} \sin(\pi \Delta z \lambda_e s^2)]$, with reciprocal radius component $s = 2\pi / \sqrt{x^2 + y^2}$, amplitude A , b -factor b , defocus Δz and electron wavelength λ_e . After applying the point-spread function, we translate the image by τ and add Gaussian noise with variance $\sigma_{\text{noise}}^2 = \sigma_{\text{signal}}^2 / \text{SNR}$, where σ_{signal}^2 is the variance of the signal and SNR is the signal-to-noise ratio. The variance of the signal σ_{signal}^2 is computed by applying a circular mask with a predefined radius on the noiseless image and then calculating the mean-squared intensity. The image formation forward model is then

$$I(x, y | \phi, \rho) = \text{PSF}_{A,b,\Delta z} * (P_z R_q \rho(X) + \tau) + \epsilon, \\ \epsilon \sim \mathcal{N}(0, \sigma_{\text{noise}}^2), \quad (\text{A1})$$

where $*$ denotes convolution. The imaging parameters utilized for simulating cryo-EM images in CryoSBI are the Gaussian mixture width γ , quaternion q , translation τ , noise level σ_{noise} , and PSF parameters $A, b, \Delta z$, with $\phi = \{\gamma, q, \tau, A, b, \Delta z, \sigma_{\text{noise}}\}$.

APPENDIX B: CRYOSBI FEATURE LATENT NETWORK AND CONDITIONAL DENSITY ESTIMATION

The latent network S_ψ follows a ResNet-18 architecture [38] as implemented in Ref. [14], with modifications for grayscale image input and 256-dimensional feature vector output. For the density estimator q_ϕ , we implement a neural spline flow [39] with the same architecture and training as utilized in Ref. [14], and likewise generating each batch of synthetic images on demand in training.

APPENDIX C: CRYOSBI PRIORS

All data processing and cryoSBI procedures for the IgG data were carried out as in Ref. [14]. The 100 atomic models were generated from an initial structure (PDB id:1HZH) by rotation of a dihedral connecting the fragment antibody domain to the rest of the structure (Supplemental Material [35], Fig. S12), as outlined in Ref. [11]. The conformations are indexed by the dihedral angle θ_i , with displacement $\theta_{i+1} - \theta_i = 3.6^\circ$.

All processing and SBI for hemagglutinin data were carried out as in Ref. [14], with experimental hemagglutinin images obtained from EMPIAR 10532 [25] and whitened using ASPIRE [40] [Supplemental Material [35], Fig. S13(a)].

The conformations from hemagglutinin were obtained from a normal mode analysis on atomic structure built from a 3 Å reconstruction (PDB id: 6wxh), resulting in 20 conformations indexed by RMSD displacement θ_i , $i = 1, \dots, 20$ [Supplemental Material [35], Fig. S13(b)].

For both datasets, the conformation prior $p(\theta)$ was taken as a uniform distribution over the possible conformational displacements $\{\theta_i\}$, and the logarithm of the SNR was sampled from a uniform distribution values between $\log 10^{-1}$ and $\log 10^{-3}$. Likewise, for both datasets the prior on the quaternions q was chosen so that rotations R_q were sampled uniformly in $SO(3)$ [41]. The other imaging parameters were sampled from uniform distributions in each parameter within bounds chosen in Ref. [14]. All nuisance parameters comprising ϕ were assumed independent and sampled independently from their respective priors.

APPENDIX D: MANIFOLD ANALYSIS FRAMEWORK

Our manifold analysis as outlined in Sec. II B follows a general framework that validates whether S_ψ has learned a representation which exhibits the following desirable properties:

Low intrinsic dimensionality. The encoder should compress images into a low-dimensional manifold with intrinsic dimension $d \ll 256$.

Sufficiency for posterior prediction. The embedding $x = S_\psi(I)$ should retain all relevant information needed by the posterior network q_ϕ to produce accurate estimates of the posterior mean and uncertainty.

Invariance to nuisance variation. The latent representation should be invariant to uninformative generative factors which do not significantly affect the posterior over θ .

Disentanglement of relevant factors. The embedding should organize generative factors such as conformation or SNR into independent directions in the latent space.

Coverage of and fidelity to the experimental data. The simulator should be able to generate realistic synthetic data that spans the variability observed in experimental images, allowing the neural network ensemble to generalize to new experimental inputs without requiring retraining.

Our analysis focuses on the statistical and geometric properties of the embedding produced by S_ψ , offering a complementary validation procedure to works such as Ref. [42] which focus on the calibration of the estimated posterior distribution. Our framework could be viewed as a model misspecification detection protocol, and in future work could be synergized with other analysis of latent embeddings in neural posterior estimation [24,43].

An overview of our pipeline can be found in Supplemental Material [35], Fig. S14. If a step in our framework is applicable to both synthetic and experimental data, we suppress s or e from the subscript (e.g., we write $\mathcal{M}$ instead of $\mathcal{M}_e$).

APPENDIX E: DATA PREPROCESSING

Our data preprocessing consists of three steps: subsampling, outlier removal and resampling the data. Here we provide information on each step, with general guidelines and implementation details for the datasets in this work.

1. Data subsampling

Most stages of our framework require computations that scale as $\mathcal{O}(N^2)$, primarily due to pairwise distance or neighbor graph construction which can be prohibitive for large N . Thus, if needed, we can randomly subsample the data down to a more manageable size. However, the subsample must be large enough to capture the underlying geometric structure, maintain statistical robustness, and ensure that we can uniformly subsample the data later in the pipeline but small enough to remain tractable given available computational resources. In practice, we find that selecting approximately three times the number of points that will be used in the final manifold learning step is a reasonable balance. The manifold embedding itself should be performed with as many points as the practitioner's computational budget allows, since more points typically yield a better approximation of the underlying geometry. In our experiments, we use 20 000 points for computing the Diffusion Maps embedding, so we subsample all datasets down to 60 000 points.

2. Outlier removal

Outlier detection serves a critical function in our validation framework as it removes pathological or noisy points that can distort downstream geometric analysis (e.g., spectral embedding, intrinsic dimension estimation). Thus, for each dataset $\mathcal{X}$, we train an outlier detector $\hat{y} : \mathbb{R}[256] \rightarrow \{0, 1\}$, which classifies input neural embeddings as either in the distribution p or out of the distribution p . While various methods exist—including one-class SVMs [44], isolation forests [45], or KDE-based detectors [46]—we favor *minimum volume sets* (MVS) [47,48] for their conceptual simplicity, strong theoretical guarantees, and alignment with the local neighborhood statistics used throughout our pipeline.

An α -MVS is the smallest measurable set U_α containing at least a fraction α of the probability mass of p . Points lying outside of U_α are flagged as outliers, with the magnitude of $1 - \alpha$ determining the stringency of the anomaly filtering. As shown in Ref. [48], one can use an empirical sample $\mathcal{X} \sim p$ to construct a simple membership test that, in the limit, consistently approximates U_α . Specifically, we approximate the local density at each point using either the inverse k -nearest neighbor distance, $1/\text{dist}_x$, or the neighbor count $n_r(x)$ within radius r of x , both of which are proportional to the true local density. Then, given a fixed number of neighbors k , the classifier (i.e., the U_α membership test) is defined as:

$$\hat{y}_k(x') := \begin{cases} 1, & \text{if } 1 - \alpha \leq s_k(x') \\ 0, & \text{otherwise} \end{cases}$$

for $s_k(x') := \frac{1}{N} \sum_{x \in \mathcal{X}} \mathbb{I}[\text{dist}_k(x') \leq \text{dist}_k(x)]$ (or an analogous formula if using n_r with fixed radius r and with reversed inequality signs). Intuitively, this means that a point is deemed in-distribution if its estimated local density exceeds that of at least a fraction $1 - \alpha$ of the training points. For robustness, we can instead use the classifier:

$$\hat{y}(x') := \begin{cases} 1, & \text{if } 1 - \alpha \leq \frac{1}{|K|} \sum_{k \in K} s_k(x') \\ 0, & \text{otherwise} \end{cases},$$

which averages the score $s_k(x')$ over multiple values of $k \in K$ (and similarly for n_r by averaging over multiple radii $r \in R$).

We obtain the "clean" dataset $\mathcal{X}^{\text{clean}} := \{x \in \mathcal{X} \mid \hat{y}(x) = 1\}$ of size $N^{\text{clean}} = \text{round}(N \cdot \alpha)$.

In our experiments, we apply the robust MVS outlier filter defined above to all three of our datasets. For the IgG dataset, we use $\alpha = 0.2$. For the hemagglutinin datasets, we use $\alpha = 0.3$. For all datasets we average the scores for $K \in \{10, 20, \dots, 120\}$. In Supplemental Material [35], Fig. S15 we provide practical guidance on selecting α .

3. Uniform resampling

Many manifold learning methods, including our intrinsic dimensionality estimators (Sec. IV A) and Diffusion Maps (Sec. IV C), either explicitly assume or empirically perform better when the distribution over the underlying manifold is uniform. For example, a nonuniform distribution p over $\mathcal{M}$ introduces bias in the estimation of the Laplace-Beltrami operator $\Delta_{\mathcal{M}}$ via Laplacian eigenmaps [49]. While the normalization scheme used by Diffusion Maps [17] theoretically corrects this bias, the result is asymptotic and relies on mild density variations, which rarely hold in practice.

To address this, we perform inverse density-weighted resampling to produce a subsample that approximates uniform coverage of the manifold. Empirically, we observe a significant improvement in performance after implementing this technique. Additionally, a uniform distribution reduces the need for adaptive bandwidths in local neighborhood computations and enables reliable use of a single global kernel width ϵ .

For fixed $k \in \mathbb{N}$, we estimate the local density at each $x \in \mathcal{X}^{\text{clean}}$ using the nonparametric estimator in Ref. [50]:

$$\hat{p}_k(x) = \frac{k}{N^{\text{clean}} \cdot V_d(\text{dist}_k(x))},$$

which is consistent under mild conditions [51]. Here $\text{dist}_k(x)$ is the distance to the k th nearest neighbor, d is the intrinsic dimension of $\mathcal{M}$, and $V_d(r)$ is the volume of a d -dimensional ball of radius r . We obtain local estimates of d by the *doubling dimension* method [28] described in Sec. IV A. For robustness, we compute density estimates for a range of $k \in K$ values and average the results obtaining the density estimate:

$$\hat{p}(x) = \frac{1}{|K|} \sum_{k \in K} \hat{p}_k(x).$$

In our experiments we use $K \in \{10, 20, \dots, 120\}$. Practitioners are free to choose any local density estimator they prefer; our choice is motivated by conceptual simplicity, effectiveness, and by the fact that we end up using the same local statistics through out our preprocessing routine.

After estimating local densities, we assign sampling weights proportional to the inverse density. Normalizing these weights yields a probability distribution over $\mathcal{X}^{\text{clean}}$ from which we draw, without replacement, a new subset $\mathcal{X}^{\text{unif}}$ of size N^{unif} . This resampled dataset approximates a uniform sample on the manifold, effectively disentangling geometry from the sampling distribution. For all our experiments we use $N^{\text{unif}} = 20\,000$. It is very important that N^{clean} is large enough relative to N^{unif} in order to allow the subsampled distribution to shift towards a uniform one. As an extreme example, if $N^{\text{clean}} = N^{\text{unif}}$, then the subsample will have exactly the same

distribution p as $\mathcal{X}^{\text{clean}}$ as we are sampling without replacement.

APPENDIX F: ADDITIONAL DETAILS FOR INTRINSIC DIMENSION ESTIMATION

This section provides more detail about the four intrinsic dimensionality algorithms employed in our study. We will focus more on practical implementation specifics, including hyperparameter selection and diagnostic procedures used in our experiments, and recommend the review in Ref. [52] for broader theoretical context.

1. Correlation dimension [26]

This method relies on the observation that the number of neighbors within a radius r scales proportionally to r^d , where d is the intrinsic dimension. As such, $\log n_r(x) = d \cdot \log r + \text{const}$. So, if we compute $\bar{n}_r = \frac{1}{N^{\text{unif}}} \sum_{x \in \mathcal{X}^{\text{unif}}} n_r(x)$, the average number of neighbors over the whole dataset for r in a range of radii R , we can globally estimate $\hat{d}$ as the slope of the line fitted to $\log \bar{n}_r$ versus $\log r$.

In our experiments, we use $R = \{8.0, 8.5, 9.0, \dots, 18.0\}$. The log-log plot should appear linear as in [Fig. 2(a)]. Downward curvature at the ends indicates R is too wide, and extreme values should be removed from the range R . Otherwise, the predicted $\hat{d}$ will likely be biased too low.

2. Eigengap method [27]

This approach exploits the idea that the eigenvalues $\{\lambda_k\}_{k=1}^{256}$ of the local covariance matrix are computed from data sampled from a d -dimensional manifold embedded in $\mathbb{R}^{256}$ and thus should exhibit a significant gap between λ_d and λ_{d+1} . The weighted local covariance at a point $x_i \in \mathcal{X}^{\text{unif}}$ is computed as:

$$C_{x_i} = \sum_{x_j \in \mathcal{N}_r(x_i)} \frac{w_{ij}}{w_i} (x_j - x_i)^\top (x_j - x_i),$$

where $\mathcal{N}_r(x_i)$ denotes neighbors within a radius r of x_i , $w_{ij} = \exp(-\frac{\|x_i - x_j\|_2^2}{\epsilon^2})$, and $w_i = \sum_{x_j \in \mathcal{N}_r(x_i)} w_{ij}$. We perform the truncated SVD of C_{x_i} for a conservatively high upper bound $D > d$ in order to obtain the eigenvalues. In our experiments we use $D = 20$. Parameters ϵ and r are selected via the distortion-minimization procedure in Ref. [32] which in our experiments we run over values $r \in \{8.0, 9.0, \dots, 22.0\}$ with $\epsilon = r/3$, a typical default bandwidth. For computational efficiency, we compute the local dimension at a subsampled set of 5000 data points for each dataset. The local dimension is estimated using either the maximum gap or the softmax gap:

$$\hat{d}(x_i) = \arg \max_k g_k \quad \text{or} \quad \hat{d}(x_i) = \frac{\sum_{k=1}^{D-1} k \cdot e^{g_k}}{\sum_{k=1}^{D-1} e^{g_k}}$$

for $g_k = \lambda_k - \lambda_{k+1}$. We prefer the softmax-weighted version as it smooths out the prediction, making it more robust to noise. A global prediction $\hat{d}$ is obtained by averaging local estimates $\hat{d} = \frac{1}{N^{\text{unif}}} \sum_{x \in \mathcal{X}^{\text{unif}}} \hat{d}(x)$.

3. Doubling dimension [28]

Similarly to correlation dimension, this method examines how point counts scale with radius but focuses specifically on the correlation when doubling the radius. The local estimate for a fixed radius $r \in \mathbb{R}^+$ is

$$\hat{d}_r(x) = \log_2 \left(\frac{n_{2r}(x)}{n_r(x)} \right),$$

which we can average over a range of radii R as $\hat{d}(x) = \frac{1}{|R|} \sum_{r \in R} \hat{d}_r(x)$ in order to improve robustness. A global prediction $\hat{d}$ is obtained by averaging local estimates $\hat{d} = \frac{1}{N^{\text{unif}}} \sum_{x \in \mathcal{X}^{\text{unif}}} \hat{d}(x)$.

In our experiments we use $R = \{8.5, 9.0, \dots, 11.0\}$. An important practical aspect is that the estimates should be stable across the chosen range R as seen in Supplemental Material [35], Fig. S4. If there is significant variation, then the range R should be tightened.

4. Levina-Bickel MLE [29]

This estimator uses the scaling of distances to nearest neighbors. For a fixed $k \in \mathbb{N}$ we have the local estimates:

$$\hat{d}_k(x) = \left[\frac{1}{k-1} \sum_{j=1}^{k-1} \log \frac{\text{dist}_k(x)}{\text{dist}_j(x)} \right]^{-1},$$

where $\text{dist}_j(x)$ is the distance to the j th nearest neighbor. We replace the $k-1$ denominator with $k-2$ in order to obtain an unbiased estimator as shown in the original paper. Similarly to previous methods, we can obtain a robust estimator $\hat{d}(x)$ by averaging $\hat{d}_k(x)$ over multiple values of k in a range K and a global estimate $\hat{d}$ by averaging these values in turn.

We use $K = \{10, 20, \dots, 120\}$. As with the doubling dimension, in practice the estimates should be stable across K as in Supplemental Material [35], Fig. S5, and if one sees significant variation, the range of K should be smaller.

APPENDIX G: ADDITIONAL DETAILS ON KDE

For KDE, we sample disjoint training, validation, and testing datasets from $\mathcal{X}_e^{\text{clean}}$ and $\mathcal{X}_s^{\text{clean}}$ as obtained by outlier removal in Appendix E 1. Note that we are not using the uniformly resampled $\mathcal{X}_e^{\text{unif}}$, $\mathcal{X}_s^{\text{unif}}$ for this purpose as they are not samples from the data distributions p_e and p_s . The training and validation sets will be used to fit the KDE models and to select appropriate kernel bandwidths, while the test subset will be used in the Monte Carlo estimation of the divergence. In order to train the KDEs and evaluate the Monte Carlo estimated KL divergences, we sample the following sample subsets from $\mathcal{X}_s^{\text{clean}}$ and $\mathcal{X}_e^{\text{clean}}$:

Training set: used to fit each KDE. Size 17 000 in our experiments.

Validation set: used for bandwidth selection. Size 3000 points in our experiments. We select the kernel bandwidths h_s and h_e that maximize the log-likelihood of the validation set. We use a single validation run on the whole validation set to minimize. We initially search over a wide exponential grid to find a coarse range, then tighten the search interval. The final bandwidths used are as follows: $h_s = 0.48$ and $h_e = 0.31$.

Test set: an independent subset for Monte Carlo estimation of the divergences. Size 3000 points in our experiments.

APPENDIX H: MANIFOLD LEARNING DETAILS

Our manifold learning step first requires construction of an affinity matrix. We follow the method in Ref. [32] to estimate the kernel bandwidth ϵ and cutoff radius r that best preserve the geometry of the manifold when constructing the Gaussian affinity matrix. We perform the search over radii $r \in \{8.0, 9.0, \dots, 22.0\}$ and set $\epsilon = r/3$, a typical default value. For the hemagglutinin dataset we find $r_e = 14.0$ for the experimental data and $r_s = 13.0$ for the simulated data. For the IgG dataset, we obtain $r_s = 15.0$. After the initial estimation, we remove all latent data vectors whose degrees in the affinity matrix fall in the bottom 5%. The outlier removal step improves the stability of the eigendecomposition in the Diffusion Maps algorithm and is generally a useful step for whatever embedding algorithm is used. Alternatively, the user could check a histogram of the degrees in the affinity matrix and select a threshold that removes a long tail of nodes with very small degrees.

Given the filtered affinity matrix, we compute the Diffusion Maps embedding [17] by performing an eigendecomposition of the geometric Laplacian $\mathbf{L}$, which, in the limit, estimates the Laplace-Beltrami operator $\Delta_{\mathcal{M}}$. In our experiments we retain the top $m = 20$ nontrivial eigenvectors to form the initial embedding matrix Φ . The value of m should be comfortably larger than what we expect the real d to be, but not excessively so. The eigenvalue spectra, shown in Supplemental Material [35], Figs. S9 and S10, confirm that only the first eigenvalue is zero and that the spectra grow slowly, indicating that the manifolds are smooth, connected, and free of isolated clusters.

To ensure that the final coordinates used for visualization and interpretation are truly informative and nonredundant, we apply IES [18] on the initial spectral embedding. The key problem IES addresses is that Diffusion Maps (or any spectral embedding) may yield embedding coordinate functions that are dependent. For example, a pair like $(x, \sin(x))$ varies in two directions but really contains only one independent degree of freedom. Visualizing both would be misleading or wasteful if the goal is to reveal distinct generative factors. To solve the problem, IES searches for a subset S of the initial embedding coordinates Φ that maintains full local rank d everywhere on the manifold, thus avoiding singularities. In our usage, we set the subset size $|S|$ equal to our estimated intrinsic dimension $\hat{d}$, but choose the target rank d to match the number of known, highly relevant generative factors we expect the manifold to express (θ and SNR for our datasets). For computational efficiency we subsample all input datasets to a set of 500 points for computing IES. Our final selected subsets of embedding coordinates are $S_s = \{0, 2, 3\}$ for the experimental hemagglutinin data and $S_e = \{0, 1, 2, 3, 4\}$ for the simulated version, while for the IgG dataset we obtain $S_s = \{0, 1, 2, 5, 6, 14\}$. We then display the first three coordinates which for DM embeddings are the lowest frequency ones.

As an optional refinement, we apply Riemannian relaxation [30] to push the embeddings closer to being isometric to their respective neural representations. To do this, Riemannian relaxation starts from the initial embeddings Φ^{indep} and

iteratively modifies them via gradient descent with respect to a loss function which penalizes local distortions. For this work, we only applied Riemannian relaxation to the hemagglutinin datasets. In Supplemental Material [35], Fig. S16, we display “relaxed” versus “unrelaxed” versions of Φ_e and Φ_s for the hemagglutinin data. For the hemagglutinin datasets, we use $d = 3$ and $\epsilon_{\text{orth}} = 0.5$, and run Riemannian relaxation until convergence.

APPENDIX I: ESTIMATING THE PARAMETERS OF THE EXPERIMENTAL DATA BY INTERPOLATION

In this section, we explain how we infer the generative parameters $\tilde{\theta}$ and $\tilde{\phi}$ for the experimental data and how we embed a new sample from $\mathcal{X}_e$ into the embedding space Φ_e as in Supplemental Material [35], Fig. S11. This is done via Nadaraya-Watson kernel regression [13] in the neural embedding space. More specifically, for every $\tilde{x}_i \in \mathcal{X}_e$, we estimate the conformation $\tilde{\theta}_i = \frac{\sum_{x_j \in \mathcal{X}_s} K(\tilde{x}_i, x_j) \theta_j}{\sum_{x_j \in \mathcal{X}_s} K(\tilde{x}_i, x_j)}$. Similarly, we obtain estimated nuisance parameters $\tilde{\phi}_i$. To embed a new point $\hat{x}_i \in \mathcal{X}_e$ in the embedding space Φ_e , we compute the c th coordinate of $\Phi_e(\hat{x}_i)$ as $\Phi_e^c(\hat{x}_i) = \frac{\sum_{\tilde{x}_j \in \mathcal{X}_e} K(\hat{x}_i, \tilde{x}_j) \Phi_e^c(\tilde{x}_j)}{\sum_{\tilde{x}_j \in \mathcal{X}_e} K(\hat{x}_i, \tilde{x}_j)}$.

APPENDIX J: GRADIENT ESTIMATION

Since the gradients ∇f_k are not analytically known, we estimate them using a local finite-difference procedure. For each point $x \in \mathcal{X}^{\text{unif}}$, we compute a weighted local neighborhood based on the same kernel matrix used for Diffusion Maps. We perform local weighted PCA to obtain a local basis $U(x) \in \mathbb{R}^{256 \times d'}$, where we use $d' = 10$ principal directions. We then construct local differences of coordinates and function values for neighbors x' of x respectively as:

$$\begin{aligned} \Delta_x(x') &= w(x')(x' - x) \in \mathbb{R}^{[256]} \\ \Delta_{f_k}(x') &= w(x')(f_k(x') - f_k(x)) \in \mathbb{R}, \end{aligned}$$

where $w(x')$ is the kernel weight for neighbor x' . Next, we project the coordinate differences onto the span of $U(x)$ as $\tilde{\Delta}_x(x') = [U(x)]^\top \Delta_x(x') \in \mathbb{R}^{[d']}$. From all neighbors of x' of x , we gather all $\tilde{\Delta}_x(x')$ into a design matrix and all $\Delta_{f_k}(x')$ into the target vector and solve the resulting weighted least-squares problem for $g \in \mathbb{R}^{[d']}$. This is an estimate of the gradient represented in $U(x)$ coordinates which we transform into an estimate in the original space by $\nabla f_k(x) = U(x)g$.

The resulting gradients are then used as input to TSLasso. We note that the projection step is optional, but we found it useful both in terms of robustness and computational efficiency as it reduces the dimensionality of the least-squares problem from 256 features to $d' \ll 256$.

APPENDIX K: TSLASSO DETAILS

We use TSLasso [19] to validate whether the embeddings learned by S_ψ are predictive for inferring the conformation posterior while remaining robust to nuisance variation. TSLasso performs feature selection on manifolds: Given a dictionary $\mathcal{F} = \{f_k : \mathcal{M} \rightarrow \mathbb{R}\}_{k=1}^p$ of candidate functions—here the simulated generative parameters such as conforma-

tion θ and nuisance factors ϕ —it searches for a subset of $\mathcal{F}$ whose gradients $\{\nabla f_k\}_{k=1}^p$ best span the tangent bundle of the manifold $\mathcal{M}$.

First, TSLasso estimates local tangent spaces from the data using local PCA. Then it projects the gradients ∇f_k onto these tangent spaces and finally reconstructs local bases as sparse linear combinations of the projected gradients. The magnitudes of the linear coefficients B_k are regularized with a Group Lasso penalty to encourage sparsity across all samples. We sweep the regularization parameter λ to find a value for which exactly d dictionary elements have corresponding nonzero coefficients B_k . The average magnitudes of B_k as λ varies indicate how strongly each function contributes to spanning the local geometry.

In our experiments, each f_k corresponds to one of the known generative parameters (the conformation θ or one of the nuisance factors in ϕ), giving $|\mathcal{F}| = 10$ for all datasets. For the hemagglutinin dataset we use $|\mathcal{S}| = 4$; for IgG we use $|\mathcal{S}| = 5$, values which are slightly lower than the estimated $\hat{d}$'s for the experimental hemagglutinin dataset and the synthetic IgG datasets but within a reasonable range. The true generative factors are unknown for experimental data, so we infer them by kernel interpolation from nearby simulated data, yielding $\tilde{\theta}$ and $\tilde{\phi}$ (Appendix I).

We run TSLasso 20 times using random subsets of 500 points drawn from the $\mathcal{X}^{\text{unif}}$ sample. To test robustness to noise, each run samples points whose SNR (inferred for experimental data) values are within the top q th percentile of all points, with $q \in \{0, 5, \dots, 95\}$. The results are summarized in Supplemental Material [35], Fig. S17. We find that TSLasso almost always selects conformation θ (or $\tilde{\theta}$ for experimental data) and SNR(SNR for experimental data) as the dominant factors parametrizing the embedded manifold.

APPENDIX L: RUNTIME CONSIDERATIONS

All operations in our framework that involve pairs of neighbors (neighbor counts, distances to the k th nearest neighbor, KDE distances, etc.), as well as the spectral decomposition of the Laplacian matrix $\mathbf{L}$ required to compute the embeddings, scale as $\mathcal{O}(N^2)$ for N the number of points used for that computation. Our experiments start at $N = 60\,000$ and after uniform subsampling reduces to $N = 20\,000$. We implement these operations using sparse linear algebra NumPy routines, which greatly reduces both time and space complexity. The experiments involving the computation of neighbor counts, neighbor distances and KDE distances as described in Appendix E, were all performed in under 10 min on a personal laptop.

The remaining operations of gradient estimation, eigengap dimensionality estimation, IES, and TSLasso, require point-wise estimations of tangent spaces which are obtained by spectral decompositions of local covariance matrices. Thus, these operations scale as $\mathcal{O}(N^3)$. However, we need not compute these local decompositions on the whole dataset but only on a representative subsample. For the eigengap dimension estimate and IES, we used 5000 points, while for IES and TSLasso we used 500 points. Using personal laptops, these operations amounted to under 20 min.

Finally, Riemannian relaxation involves a heavy iterative optimization procedure and is by some margin the most

expensive operation in our framework taking up to 3 h. Fortunately, this procedure is an optional step for the pipeline, and

future work can investigate the use of early stopping to reduce the computational load.

-
- [1] E. Nogales and S. H. W. Scheres, Cryo-EM: A unique tool for the visualization of macromolecular complexity, *Mol. Cell* **58**, 677 (2015).
 - [2] A. Dashti, P. Schwander, R. Langlois, R. Fung, W. Li, A. Hosseinzadeh, H. Y. Liao, J. Pallesen, G. Sharma, V. A. Stupina, A. E. Simon, J. D. Dinman, J. Frank, and A. Ourmazd, Trajectories of the ribosome as a Brownian nanomachine, *Proc. Natl. Acad. Sci. USA* **111**, 17492 (2014).
 - [3] E. D. Zhong, T. Bepler, B. Berger, and J. H. Davis, CryoDRGN: Reconstruction of heterogeneous cryo-EM structures using neural networks, *Nat. Meth.* **18**, 176 (2021).
 - [4] M. Chen and S. J. Ludtke, Deep learning-based mixed-dimensional Gaussian mixture model for characterizing variability in cryo-EM, *Nat. Meth.* **18**, 930 (2021).
 - [5] A. Punjani and D. J. Fleet, 3DFlex: determining structure and motion of flexible proteins from cryo-EM, *Nat. Methods* **20**, 860 (2023).
 - [6] C. Donnat, A. Levy, F. Poitevin, E. D. Zhong, and N. Miolane, Deep generative modeling for volume reconstruction in cryo-electron microscopy, *J. Struct. Biol.* **214**, 107920 (2022).
 - [7] W. S. Tang, E. D. Zhong, S. M. Hanson, E. H. Thiede, and P. Cossio, Conformational heterogeneity, and probability distributions from single-particle cryo-electron microscopy, *Curr. Opin. Struct. Biol.* **81**, 102626 (2023).
 - [8] R. R. Lederman and B. Toader, On manifold learning in Plato's cave: Remarks on manifold learning and physical phenomena, in *Proceedings of the 2023 International Conference on Sampling Theory, and Applications (SampTA)* (IEEE, Los Alamitos, CA, 2023), pp. 1–7.
 - [9] D. A. Klindt, A. Hyvärinen, A. Levy, N. Miolane, and F. Poitevin, Towards interpretable Cryo-EM: Disentangling latent spaces of molecular conformations, *Front. Molec. Biosci.* **11**, 1393564 (2024).
 - [10] B. Shi, K. Zhang, D. J. Fleet, R. A. McLeod, R. J. Dwayne Miller, and J. Y. Howe, Deep generative priors for biomolecular 3D heterogeneous reconstruction from cryo-EM projections, *J. Struct. Biol.* **216**, 108073 (2024).
 - [11] M. Jeon, R. Raghu, M. Astore, G. Woollard, R. Feathers, A. Kaz, S. M. Hanson, P. Cossio, and E. D. Zhong, CryoBench: Diverse and challenging datasets for the heterogeneity problem in cryo-EM, *Adv. Neural Inf. Proc. Syst.* **37**, 89468 (2024).
 - [12] M. P. Rout and A. Sali, Principles for integrative structural biology studies, *Cell* **177**, 1384 (2019).
 - [13] K. P. Murphy, *Probabilistic Machine Learning: An Introduction* (MIT Press, Cambridge, MA, 2022).
 - [14] L. Dingeldein, D. Silva-Sánchez, L. Evans, E. D'Imprima, N. Grigorieff, R. Covino, and P. Cossio, Amortized template matching of molecular conformations from cryoelectron microscopy images using simulation-based inference, *Proc. Natl. Acad. Sci. USA* **122**, e2420158122 (2025).
 - [15] K. Cranmer, J. Brehmer, and G. Louppe, The frontier of simulation-based inference, *Proc. Natl. Acad. Sci. USA* **117**, 30055 (2020).
 - [16] G. Papamakarios and I. Murray, Fast ε -free inference of simulation models with Bayesian conditional density estimation, *Adv. Neural Inf. Proc. Syst.* **29** (2016).
 - [17] R. R. Coifman, S. Lafon, A. B. Lee, M. Maggioni, B. Nadler, F. Warner, and S. W. Zucker, Geometric diffusions as a tool for harmonic analysis, and structure definition of data: Diffusion maps, *Proc. Natl. Acad. Sci. USA* **102**, 7426 (2005).
 - [18] Y.-C. Chen and M. Meila, Selecting the independent coordinates of manifolds with large aspect ratios, *Adv. Neural Inf. Proc. Syst.* **32** (2019).
 - [19] S. J. Koelle, H. Zhang, O.-V. Murad, and M. Meila, Consistency of dictionary-based manifold learning, in *International Conference on Artificial Intelligence, and Statistics* (PMLR, Cambridge, MA, 2024), pp. 4348–4356.
 - [20] M. Meilä and H. Zhang, Manifold learning: What, how, and why, *Annu. Rev. Stat. Appl.* **11**, 393 (2024).
 - [21] G. M. da Silva, J. Y. Cui, D. C. Dalgarno, G. P. Lisi, and B. M. Rubenstein, High-throughput prediction of protein conformational distributions with subsampled AlphaFold2, *Nat. Commun.* **15**, 2464 (2024).
 - [22] J.-M. Lueckmann, J. Boelts, D. Greenberg, P. Goncalves, and J. Macke, Benchmarking simulation-based inference, in *International Conference on Artificial Intelligence, and Statistics* (PMLR, 2021), pp. 343–351.
 - [23] A. Zammit-Mangion, M. Sainsbury-Dale, and R. Huser, Neural methods for amortised parameter inference, [arXiv:2404.12484](https://arxiv.org/abs/2404.12484).
 - [24] M. Schmitt, P.-C. Bürkner, U. Köthe, and S. T. Radev, Detecting model misspecification in amortized Bayesian inference with neural networks: An extended investigation, [arXiv:2406.03154](https://arxiv.org/abs/2406.03154).
 - [25] Y. Z. Tan and J. L. Rubinstein, Through-grid wicking enables high-speed cryoem specimen preparation, *Acta Cryst. D* **76**, 1092 (2020).
 - [26] P. Grassberger and I. Procaccia, Measuring the strangeness of strange attractors, *Physica D* **9**, 189 (1983).
 - [27] G. Chen, A. V. Little, and M. Maggioni, *Multi-Resolution Geometric Analysis for Data in High Dimensions* (Birkhäuser, Boston, 2013), pp. 259–285.
 - [28] P. Assouad, Plongements lipschitziens dans $\mathbb{R}^n$, *Bull. Soc. Math. France* **111**, 429 (1983).
 - [29] E. Levina and P. Bickel, Maximum likelihood estimation of intrinsic dimension, in *Advances in Neural Information Processing Systems*, edited by L. Saul, Y. Weiss, and L. Bottou (MIT Press, Cambridge, MA, 2004), Vol. 17.
 - [30] J. McQueen, M. Meila, and D. Perrault-Joncas, Nearly isometric embedding by relaxation, *Adv. Neural Inf. Proc. Syst.* **29** (2016).
 - [31] B. W. Silverman, *Density Estimation for Statistics, and Data Analysis* (Routledge, Milton Park, UK, 2018).
 - [32] D. Joncas, M. Meila, and J. McQueen, Improved graph Laplacian via geometric self-consistency, *Advances in Neural Information Processing Systems*, edited by I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus,

- S. Vishwanathan, and R. Garnett (Curran Associates, Inc., 2017), Vol. 30.
- [33] D. Perraul-Joncas and M. Meila, Non-linear dimensionality reduction: Riemannian metric estimation, and the problem of geometric discovery, [arXiv:1305.7255](#).
- [34] L. Evans, “cryo-em images are intrinsically low dimensional”: Latent data from SBI, Zenodo (2025), <https://zenodo.org/records/15733579>.
- [35] See Supplemental Material at <http://link.aps.org/supplemental/10.1103/txrb-fw3z> for additional figures referenced in the text.
- [36] E. Seitz, F. Acosta-Reyes, P. Schwander, and J. Frank, Simulation of cryo-EM ensembles from atomic models of molecules exhibiting continuous conformations, [bioRxiv 864116 \(2019\)](#).
- [37] J. Giraldo-Barreto, S. Ortiz, E. H. Thiede, K. Palacio-Rodriguez, B. Carpenter, A. H. Barnett, and P. Cossio, A Bayesian approach to extracting free-energy profiles from cryo-electron microscopy experiments, [Sci. Rep. 11, 13657 \(2021\)](#).
- [38] K. He, X. Zhang, S. Ren, and J. Sun, Deep residual learning for image recognition, in *Proceedings of the IEEE Conference on Computer Vision, and Pattern Recognition* (IEEE, Los Alamitos, CA, 2016), pp. 770–778.
- [39] C. Durkan, A. Bekasov, I. Murray, and G. Papamakarios, Neural spline flows [arXiv:1906.04032](#).
- [40] G. Wright, J. Andén, V. Bansal, J. Xia, C. Langfield, J. Carmichael, K. Sowattanangkul, R. Brook, Y. Shi, A. Heimowitz, G. Pragier, I. Sason, A. Moscovich, Y. Shkolnisky, and A. Singer, ComputationalCryoEM/ASPIRE-Python: v0.12.0 (v0.12.0), Zenodo (2023), <https://doi.org/10.5281/zenodo.8321443>.
- [41] A. J. Hanson, *Visualizing Quaternions* (Morgan Kaufmann Publishers Inc., San Francisco, CA, 2006).
- [42] S. Talts, M. Betancourt, D. Simpson, A. Vehtari, and A. Gelman, Validating Bayesian inference algorithms with simulation-based calibration, [arXiv:1804.06788](#).
- [43] D. Huang, A. Bharti, A. Souza, L. Acerbi, and S. Kaski, Learning robust statistics for simulation-based inference under model misspecification, [arXiv:2305.15871](#).
- [44] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson, Estimating the support of a high-dimensional distribution, [Neural Comput. 13, 1443 \(2001\)](#).
- [45] F. T. Liu, K. M. Ting, and Z.-H. Zhou, Isolation forest, in *Proceedings of the 2008 Eighth IEEE International Conference on Data Mining* (IEEE, Los Alamitos, CA, 2008), pp. 413–422.
- [46] C. I. Lang, F.-K. Sun, B. Lawler, J. Dillon, A. A. Dujaili, J. Ruth, P. Cardillo, P. Alfred, A. Bowers, A. Mckiernan, and D. S. Boning, One class process anomaly detection using kernel density estimation methods, [IEEE Trans. Semicond. Manufact. 35, 457 \(2022\)](#).
- [47] C. D. Scott and R. D. Nowak, Learning minimum volume sets, *J. Mach. Learn. Res.* **7**, 665 (2006).
- [48] M. Zhao and V. Saligrama, Anomaly detection with score functions based on nearest neighbor graphs, in *Advances in Neural Information Processing Systems*, edited by Y. Bengio, D. Schuurmans, J. Lafferty, C. Williams, and A. Culotta (Curran Associates, Red Hook, NY, 2009), Vol. 22.
- [49] M. Belkin and P. Niyogi, Laplacian eigenmaps, and spectral techniques for embedding, and clustering, in *Advances in Neural Information Processing Systems*, edited by T. Dietterich, S. Becker, and Z. Ghahramani (MIT Press, Cambridge, MA, 2001), Vol. 14.
- [50] D. O. Loftsgaarden and C. P. Quesenberry, A nonparametric estimate of a multivariate density function, [Ann. Math. Statist. 36, 1049 \(1965\)](#).
- [51] G. Boente and R. Fraiman, Consistency of a nonparametric estimate of a density function for dependent variables, [J. Multivar. Anal. 25, 90 \(1988\)](#).
- [52] F. Camastra, and A. Staiano, Intrinsic dimension estimation: Advances, and open problems, [Inf. Sci. 328, 26 \(2016\)](#).