

Supplemental Information:
Geometry-preserving vector field reconstruction of
high-dimensional cell-state dynamics using ddHodge

Kazumitsu Maehara^{1,2} and Yasuyuki Ohkawa^{2,*}

¹Department of Multi-Omics, Graduate School of Medical Sciences

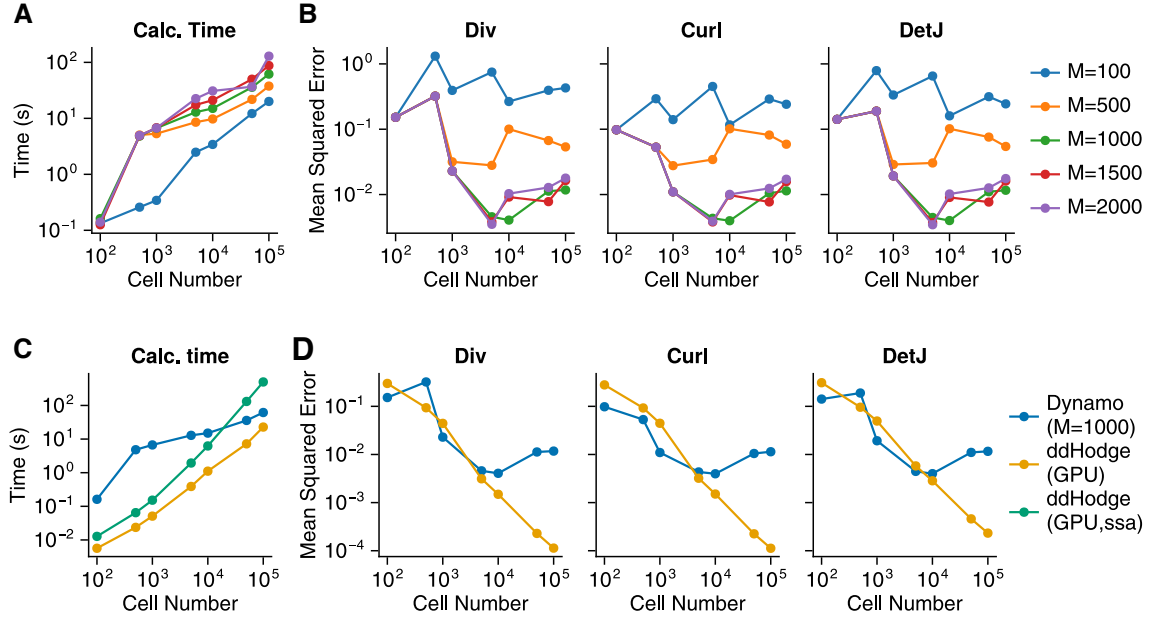
²Division of Transcriptomics, Medical Institute of Bioregulation, Kyushu
University, 3-1-1 Maidashi, Higashi-ku, Fukuoka 812-0054, Japan

December 3, 2025

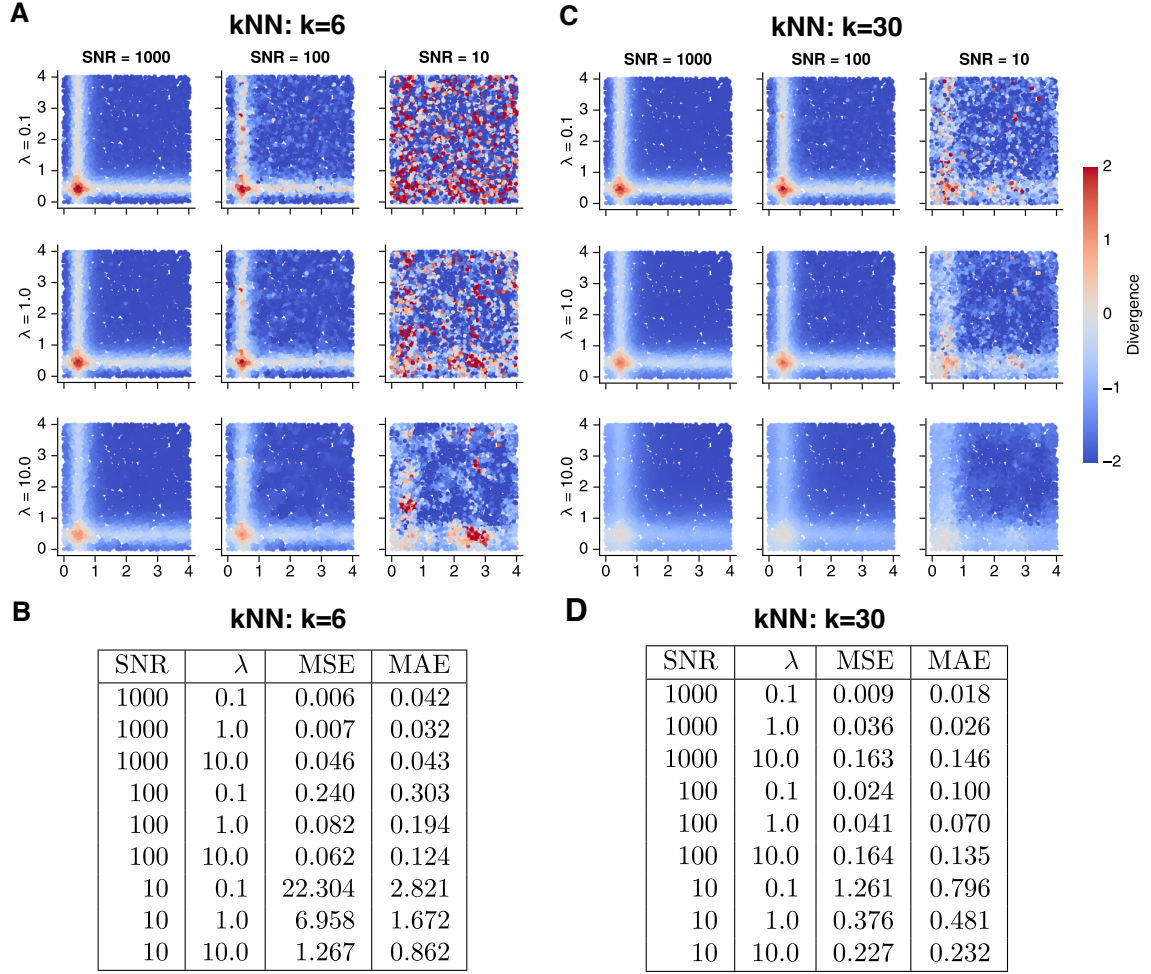
Contents

1	Supplemental Figures	2
2	Supplemental Notes	12
2.1	Hodge decomposition	12
2.2	Defining edge weights from a given velocity	14
2.3	Reconstruction of the Hessian matrix of U	15
2.4	$\mathbf{l}_2$ regularization on a graph	17
2.5	Reconstruction of the curl	18
2.6	Reconstruction of the Jacobian matrix	18
2.7	Simulating tangent-space dynamics in parameter space	19
2.8	Setting gradient and rotational components on the sphere	20
2.9	Designing the parallel coordinate system	20
2.10	Using near-parallel coordinates	22
2.11	Basic idea of sheaves on graphs	23

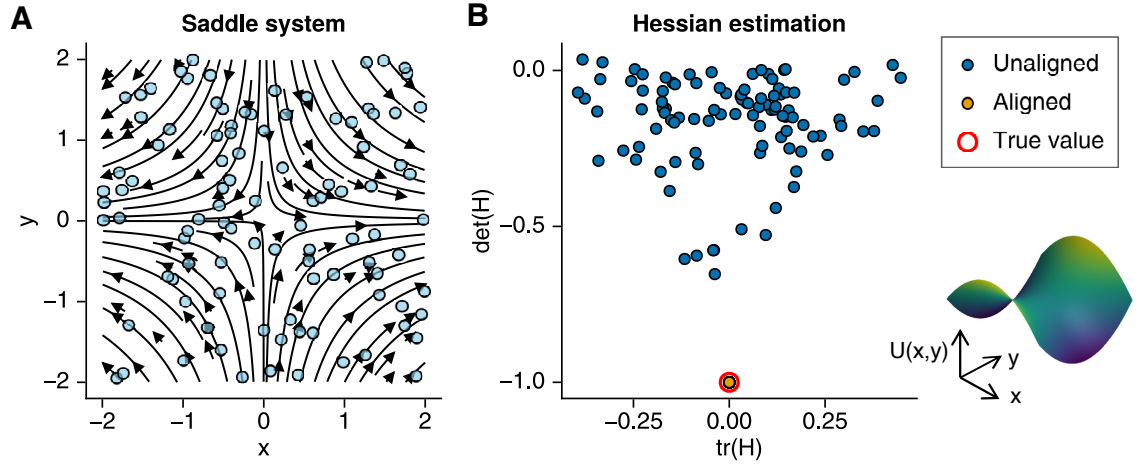
1 Supplemental Figures



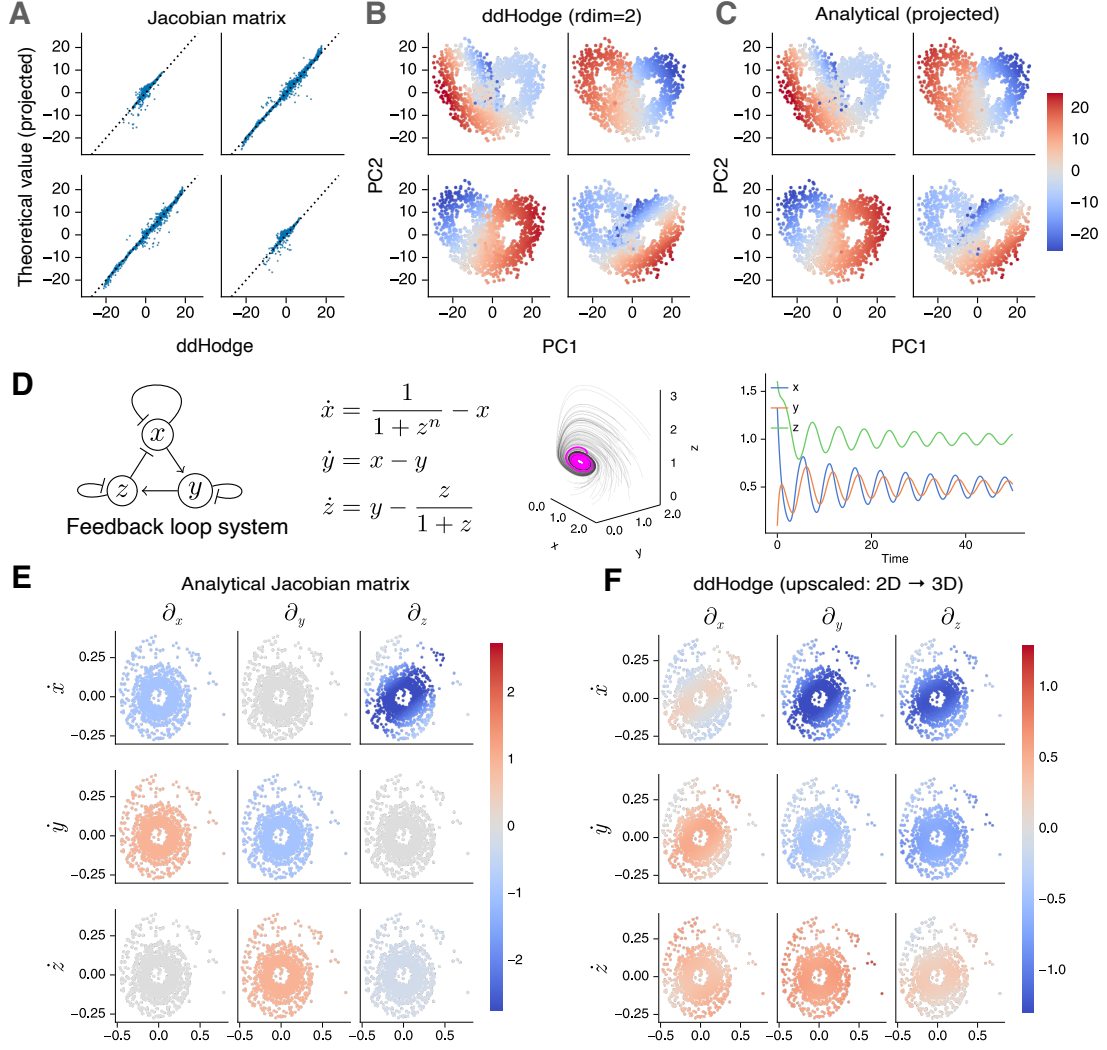
Supplementary Figure S1. Benchmark results. (A, B) Performance comparison of Dynamo with varying basis function numbers ($M = 100$ to 2000) showing computation time and reconstruction accuracy for divergence (Div), curl, and Jacobian determinant (DetJ). To obtain these result from Dynamo, setting the option `lstsq_method='scipy'` in `dynamo.vf.VectorField` is required. (C, D) Direct comparison between optimized Dynamo ($M = 1000$) and ddHodge with options (GPU, GPU+SSA) across different cell numbers, demonstrating computational efficiency and reconstruction accuracy.



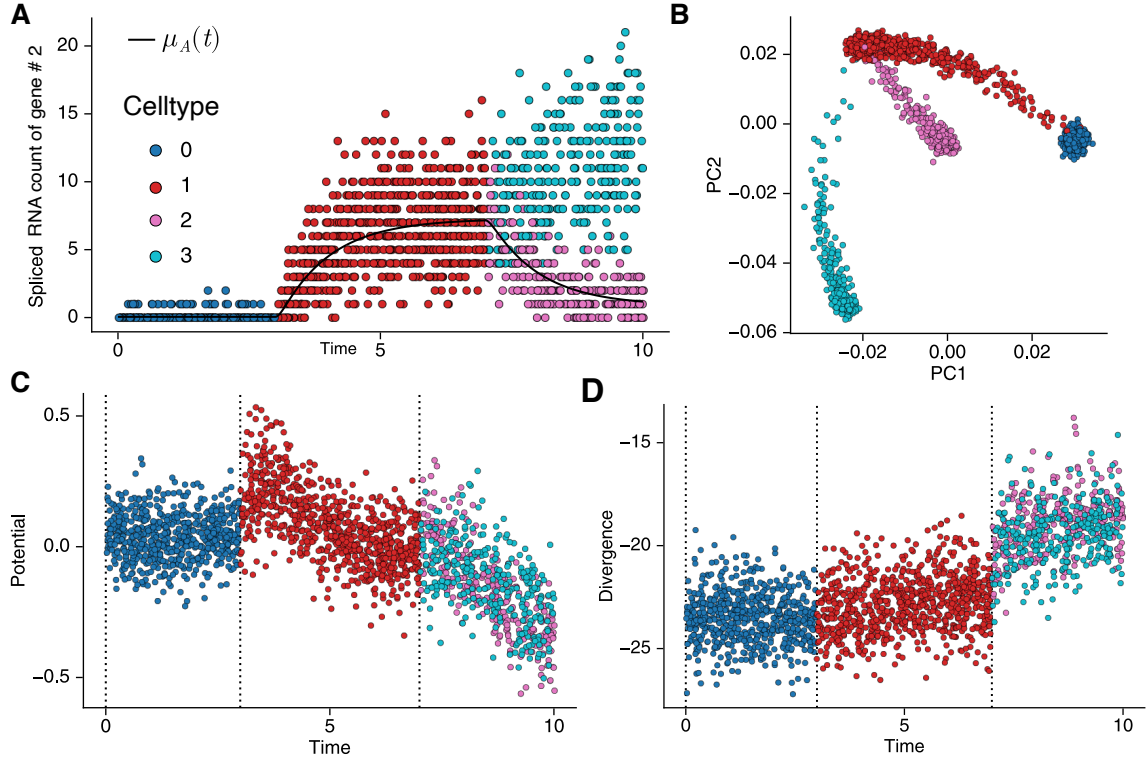
Supplementary Figure S2. Impacts of smoothing parameters in noisy simulation. Performance evaluation on 2D toggle switch simulation with Gaussian noise. (A, C) Reconstruction results of divergence for $k = 6$ and $k = 30$ across different noise levels (SNR = 1000 to 10) and regularization parameters λ (ℓ_2 penalty in Hessian/Jacobian estimation; see Methods and Notes). (B, D) Quantitative evaluation showing mean squared error (MSE) and mean absolute error (MAE) for each condition.



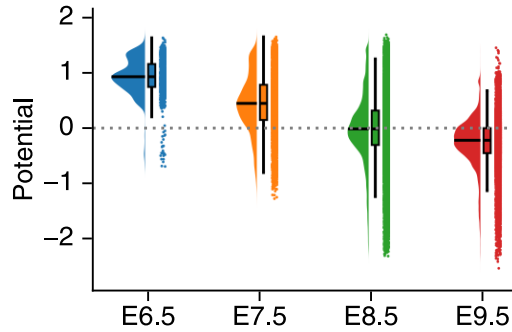
Supplementary Figure S3. Coordinate alignment importance demonstrated with a saddle-type potential. (A) Streamlines and sample points (circles) for the saddle-type gradient system: $U(x, y) = x^2 - y^2$. (B) Hessian-matrix trace (divergence) and determinant estimates. Orange points: smoothing results with aligned axes, blue points: results with unaligned axes. Red circle: theoretical value; $\text{tr}(H) = 0, \det(H) = -1$. The inset shows the surface of the saddle potential U .



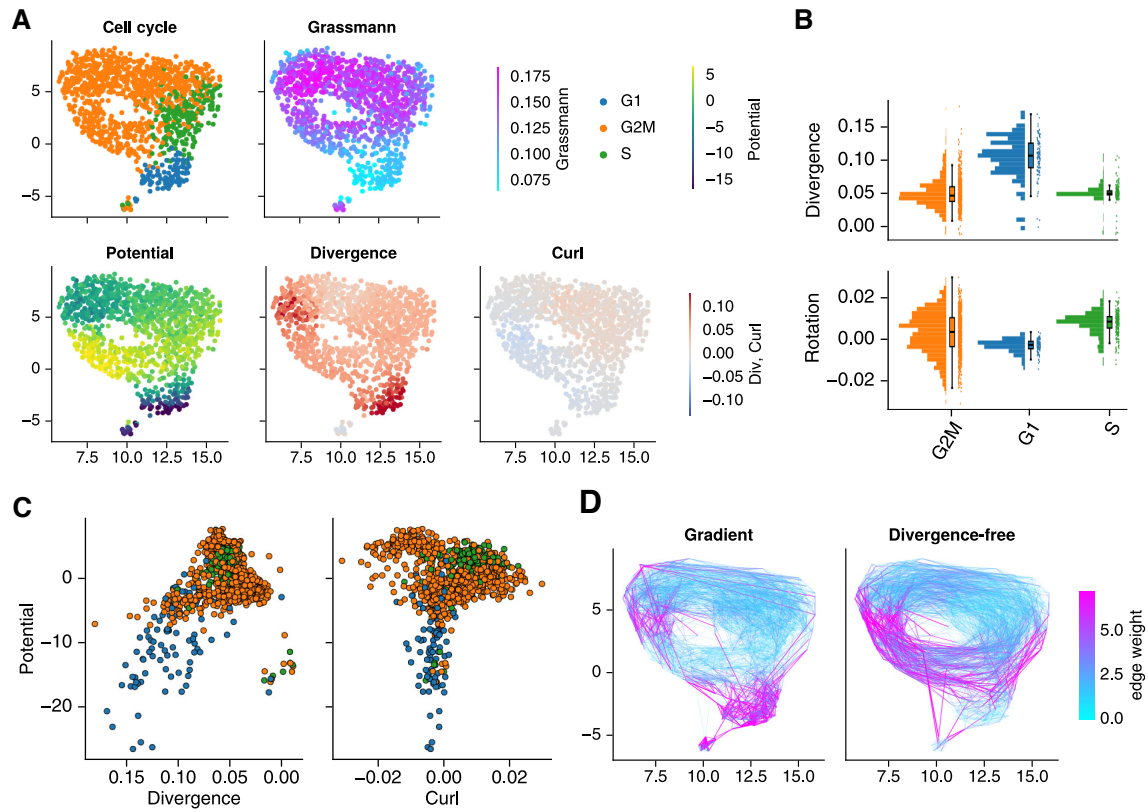
Supplementary Figure S4. Jacobian matrix approximation accuracy on tangent space. (A–C) Lorenz system example: (A) comparison of ddHodge-estimated versus theoretical Jacobian elements projected onto tangent space; (B, C) dynamics visualization in the PC1–PC2 plane for ddHodge-reconstructed and theoretical projected cases. (D–F) Feedback-loop system example: (D) 3D trajectory with governing equations where $n = 12$ we set; (E, F) spatial distribution of Jacobian elements for theoretical projected and ddHodge-reconstructed cases.



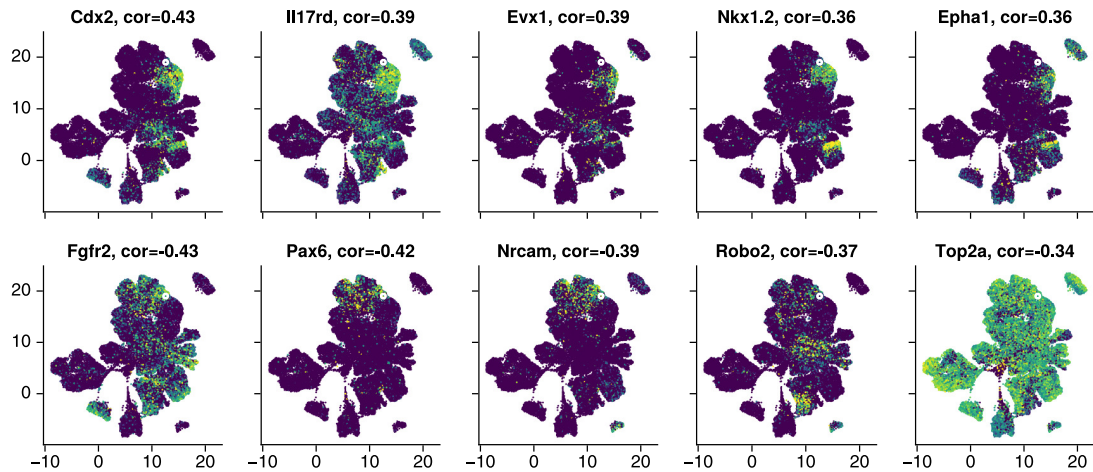
Supplementary Figure S5. ddHodge performance on a 100-dimensional RNA-velocity simulation. (A) Poisson sampling (points) with the mean spliced RNA count $\mu_A(t)$ (trajectory of the mean; black line) for a representative gene across cells. To emulate driver-gene changes, four cell types (color-coded) were assigned different transcription rates α , with switches at $t = 3$ and $t = 7$. (B) PCA visualization in the PC1–PC2 plane of the 100-dimensional gene space. (C) Potential values estimated by ddHodge over time. (D) Divergence values over time.



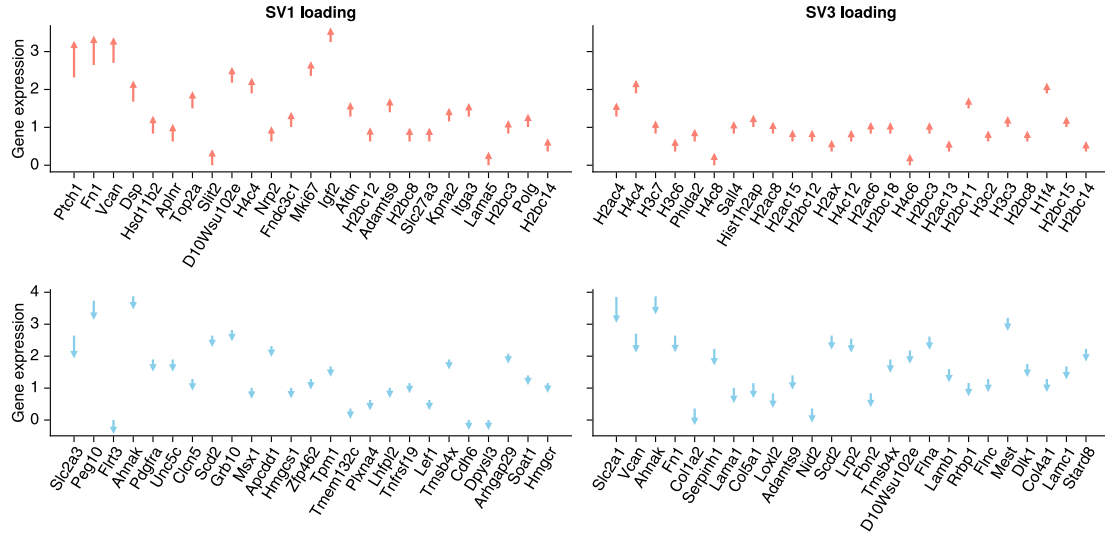
Supplementary Figure S6. Developmental stage and potential corresponding to Figure 4D.



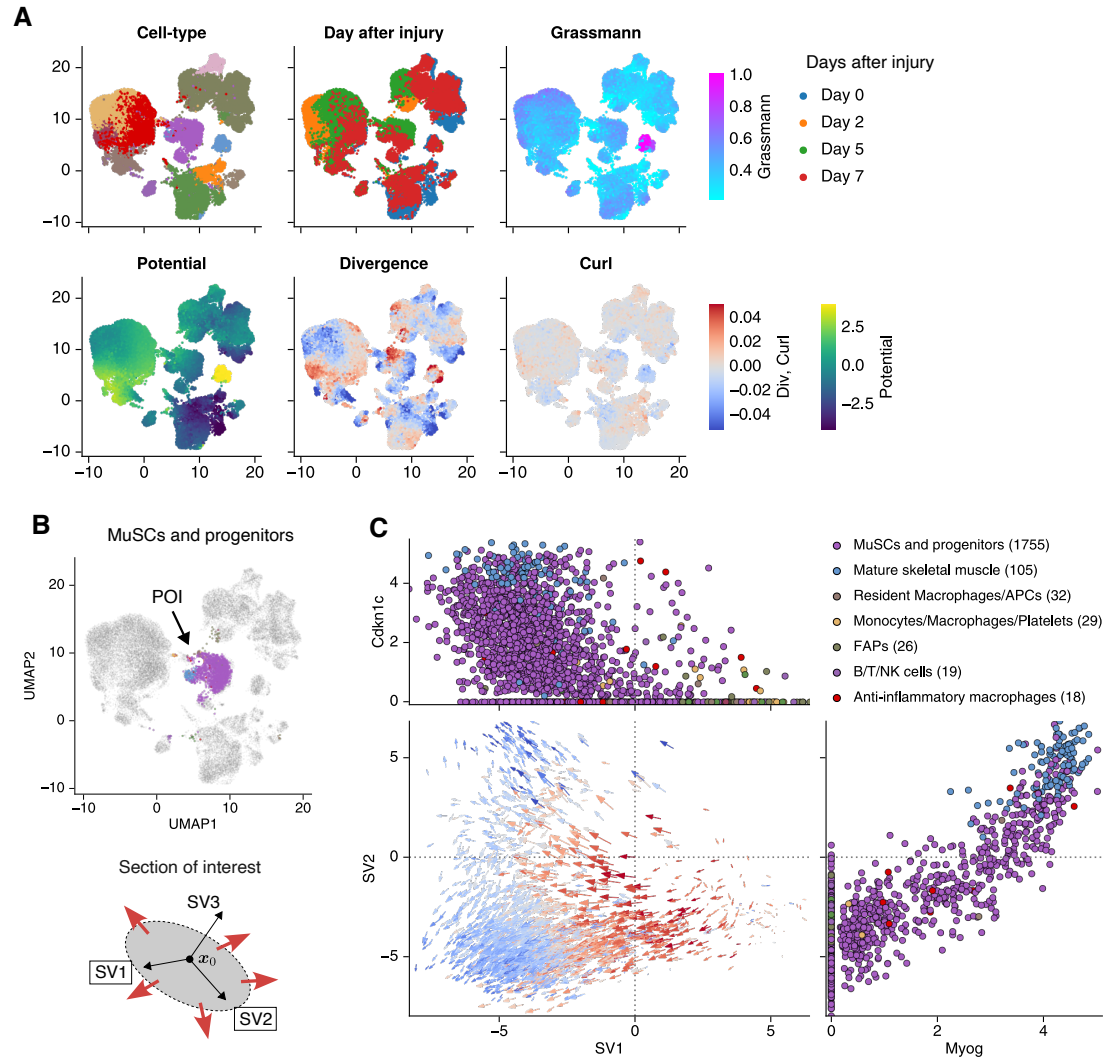
Supplementary Figure S7. ddHodge analysis of Fucci data. (A) UMAP visualization colored by cell cycle phases (G1, G2M, S) and Grassmann distance (top), potential, divergence, and curl components (bottom). (B) Distribution of divergence and rotation components across cell cycle phases. (C) Potential-divergence scatter plots. (D) Gradient (left) and divergence-free (right) components visualized by edge weights.



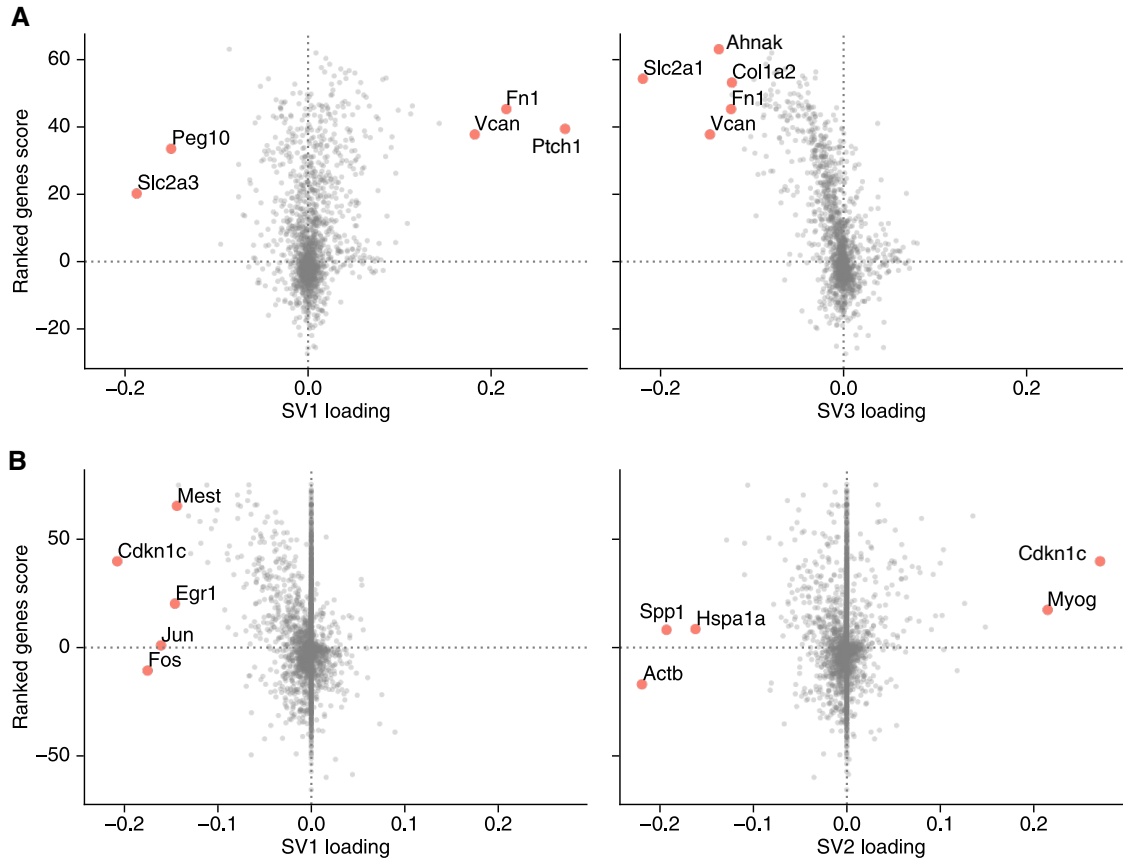
Supplementary Figure S8. Correlation between gene expression level and divergence in cluster 18. Each gene is displayed with a color scale indicating expression levels. The correlation coefficient for each gene is annotated as “[gene name], cor. = [value]”. The white circle marks the location of the point of interest (POI).



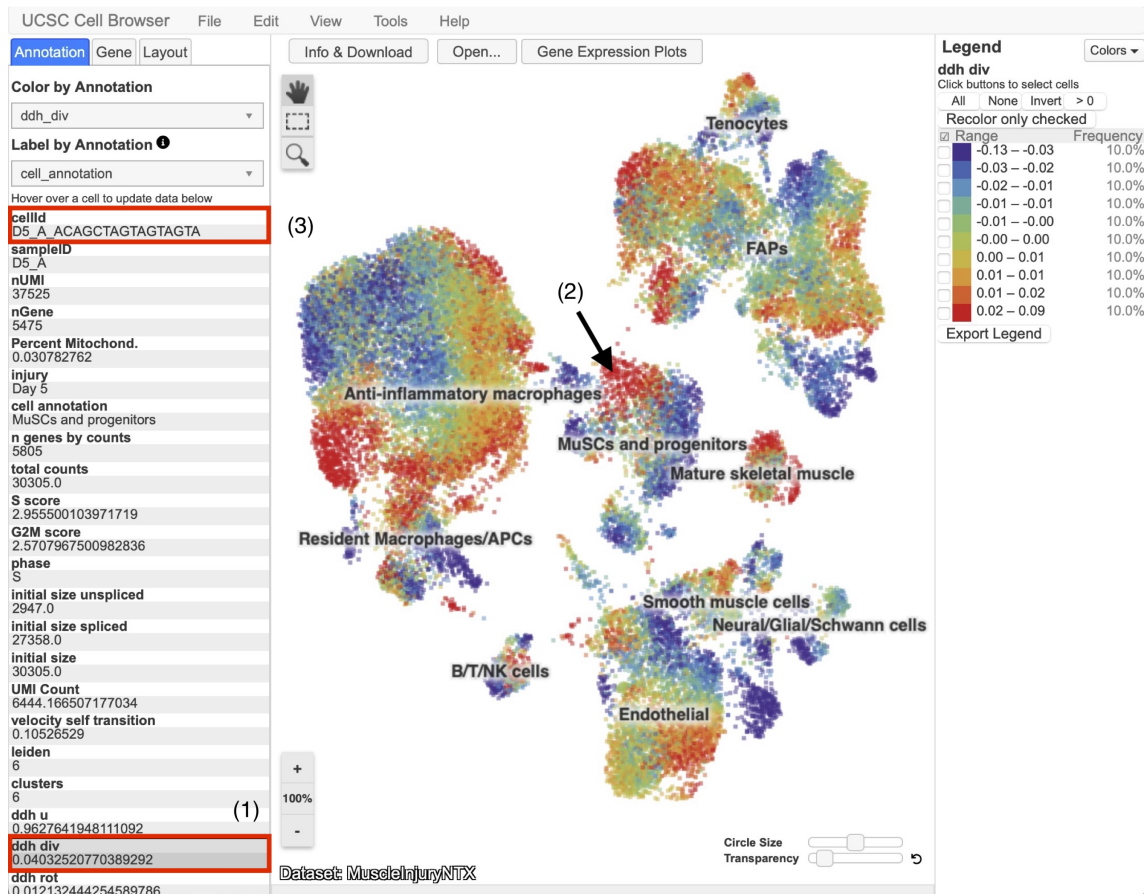
Supplementary Figure S9. Loadings of Schur vectors of the POI in cluster 18. Arrow direction indicates the axis of infinitesimal change represented by each Schur vector, and arrow length encodes magnitude. The vertical axis shows gene expression levels at the POI. The top 25 genes with positive and negative weights are listed.



Supplementary Figure S10. ddHodge analysis of muscle satellite-cell (MuSC) regeneration dynamics. (A) UMAP visualization of mouse muscle regeneration data colored by cell type, days after injury, Grassmann distance (top), and ddHodge components: potential, divergence, curl (bottom). (B) Identification of a point of interest (POI) in the “MuSCs and progenitors” cluster showing high divergence. (C) Schur-decomposition analysis of the POI, displaying dynamics in the SV1–SV2 plane with cells colored by divergence values and velocity vectors projected onto the plane.



Supplementary Figure S11. Comparison of ddHodge Schur vector (SV) loadings with Wilcoxon ranking scores for marker gene detection. (A) Scatter plots comparing ranked gene scores with SV1 loadings (left) and SV3 loadings (right) for the VASA-seq dataset, with specific genes labeled. (B) Scatter plots comparing ranked gene scores with SV1 loadings (left) and SV2 loadings (right) for the muscle regeneration dataset, with specific genes labeled.



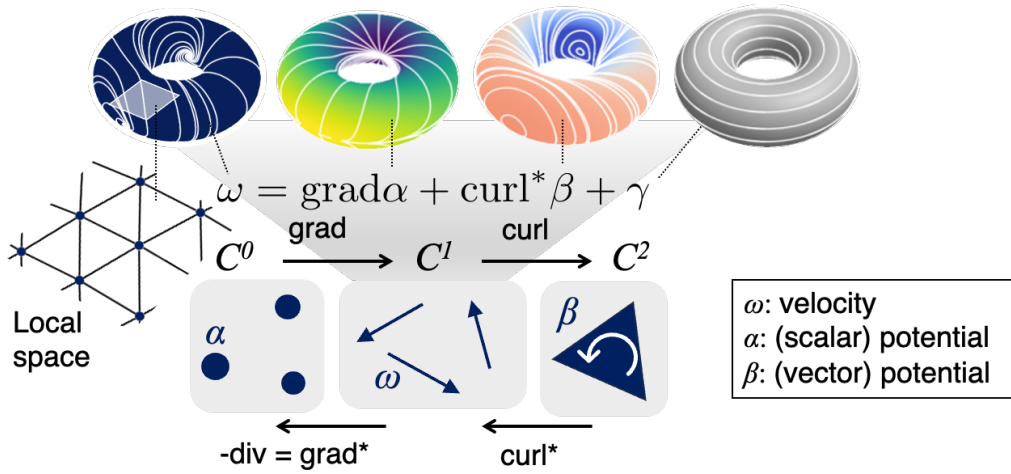
Supplementary Figure S12. Point of Interest (POI) selection workflow using UCSC Cell Browser. The browser displays mouse muscle regeneration data analyzed with ddHodge. The left panel shows cell annotations and ddHodge-computed metrics (prefixed with *ddh_**). UMAP visualization is colored by divergence values (red: high divergence, blue: low divergence). Users can visually identify high-divergence cells (red regions) and click to select them as POI, then copy cell IDs for further detailed analysis such as Schur plane visualization.

2 Supplemental Notes

Details and graphical explanations of the ddHodge method are provided in the following notes.

2.1 Hodge decomposition

Hodge decomposition is a mathematical theorem that states that a vector field ω on a manifold $\mathcal{M}$ can be decomposed into three orthogonal components: a gradient $\text{grad } \alpha$, a divergence-free rotational component $\text{curl}^* \beta$, and a residual, known as the harmonic component γ . Although the Hodge theory was built abstractly in terms of differential forms, the framework can be made computer friendly by discretizing the differential forms as a collection of the numerical values of the elements of a mesh or graph (Fig. S13).



Supplementary Figure S13. Illustration of *discrete* Hodge decomposition on a torus. A continuous vector field on the surface of a torus, a manifold without boundary, is discretized onto a triangular mesh, where vertex-, edge-, and face-associated values correspond to computational representations of 0-, 1-, and 2-forms (α , ω , and β). The vector field ω is decomposed into gradient, curl, and harmonic components. The harmonic component γ emerges as a flow circulating around the holes in the space, reflecting the global topology encoded by the discretized surface.

In particular, the estimation of the potential $\mathbf{u} \in \mathbb{R}^N$ corresponding to the scalar field α that gives the gradient component of the vector field requires only the construction of an $M \times N$ matrix $\delta_0 = \text{grad}$ called the gradient operator from a given directed graph $\mathcal{G} = (V, E)$. Once we have δ_0 and the data of the edge weights $\mathbf{w} \in \mathbb{R}^M$, we can obtain $\mathbf{u}$ via the least-squares method [13-14]:

$$\min_{\mathbf{u}} \|\delta_0 \mathbf{u} - \mathbf{w}\|^2$$

where N is the number of vertices and M is the number of edges in graph $\mathcal{G}$. This method, known as HodgeRank, was proposed for ranking problems and can also be viewed as a least squares solution to the discretized Poisson equation $\Delta f = \text{div} g$.

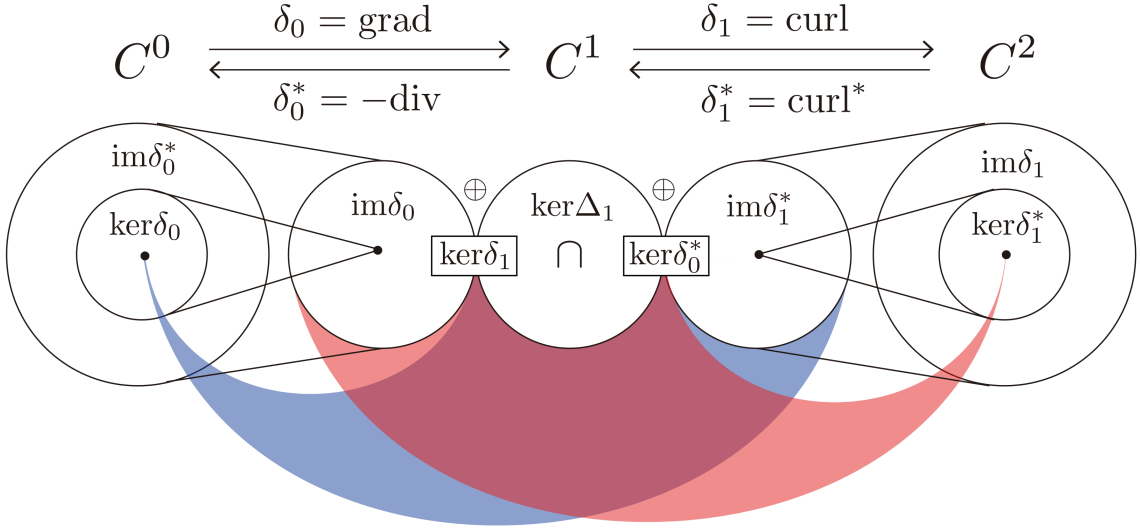
In *combinatorial* Hodge decomposition, cochains C^0, C^1, C^2 are defined as spaces of functions used to assign real values to vertices (0-simplices), edges (1-simplices), and triangles (2-simplices), respectively.

The gradient operator $\delta_0 : C^0 \rightarrow C^1$ is a linear coboundary operator that can be represented as a matrix $\delta_0 \in \mathbb{R}^{M \times N}$ by considering $C^0 = \mathbb{R}^N$ and $C^1 = \mathbb{R}^M$. Specifically,

for an edge $e = (i, j) \in E$, the matrix δ_0 is constructed using elements $\{-1, 0, 1\}$ such that $(\delta_0 \mathbf{u})_e = u_j - u_i$. Although $\mathcal{G}$ is assumed to be a directed graph, this approach can also be applied to an undirected graph by adopting a consistent rule for the edge orientation, such as by regarding (i, j) as a directed edge when $i < j$. Additionally, the alternativity $w_{(i,j)} = -w_{(j,i)}$ must be satisfied for the edge weights. Similarly, the curl operator $\text{curl} = \delta_1 : C^1 \rightarrow C^2$, which takes the sum of the weights of the edges in the counterclockwise direction, can be represented by a matrix $\delta_1 \in \mathbb{R}^{L \times M}$ with elements $\{-1, 0, 1\}$.

In addition to the coboundary operators, for $x \in C^{k-1}$ and $y \in C^k$ with $k \in \{1, 2\}$, the adjoint of the coboundary operators is defined such that the inner product $\langle \delta_k x, y \rangle = \langle x, \delta_k^* y \rangle$ is preserved, resulting in the generation of a mapping with the reverse direction: $\delta_k^* = C^k \rightarrow C^{k-1}$. In the framework of combinatorial Hodge decomposition, these adjoints can be simply constructed as a matrix transpose $\delta_k^* = \delta_k^\top$. The divergence $\text{div} = -\delta_0^\top$ is the sum of the weights of the edges incident to the vertex, analogous to the usual definition of the divergence as the total flux per unit volume. Note that even if $\mathcal{G}$ is constructed using points in Euclidean space, the volume in the original space is not considered in the combinatorial Hodge decomposition.

On the basis of these definitions of cochains, coboundary operators, and their adjoints, the property $\delta_k \delta_{k-1} = \delta_{k-1}^* \delta_k^* = 0$ always holds¹, allowing the vector space $C^1 = \mathbb{R}^M$ to be decomposed into the direct sum of the image of δ_0 (gradient component), the image of δ_1^* (curl component), and the kernel of Δ_1 (harmonic component) (Fig. S14), thus providing the same mathematical structure as Hodge decomposition for functions on manifold. Here, $\Delta_k = \delta_{k-1} \delta_{k-1}^* + \delta_k^* \delta_k$ is the abstract Hodge k -Laplacian².



Supplementary Figure S14. Schematic of combinatorial Hodge decomposition on a graph. The vector space of edge-associated values C^1 admits an orthogonal decomposition into three subspaces: the image of the gradient operator δ_0 (gradient component), the image of the adjoint of the curl operator δ_1^* (curl component), and the kernel of the Hodge 1-Laplacian Δ_1 (harmonic component).

The combinatorial Hodge decomposition approach, which provides a good estimation method of the potential, can be applied to vector fields defined by differential equations in continuous space. However, since the divergence and curl are the quantities per unit volume/area, they cannot be directly used as is. On the other hand, with numerical

¹Similar to $dd = \delta\delta = 0$ in exterior calculus

²When $k = 0$, $\Delta_0 = \delta_0^* \delta_0$ is the graph Laplacian.

methods, which are preferable approaches when dealing with differential equations, challenges in constructing meshes and evaluating volumes in high-dimensional spaces are often encountered. Therefore, establishing methods for estimating the divergence and curl of vector fields via Hodge decomposition on graphs while avoiding the difficulties of mesh construction and volume evaluation is a key challenge addressed by ddHodge.

2.2 Defining edge weights from a given velocity

We focus on estimating the potential function $U : \mathbb{R}^N \rightarrow \mathbb{R}$ from the given velocity $\mathbf{v}_i$ for each point $\mathbf{x}_i$ in a sample of N points in d -dimensional vector space.

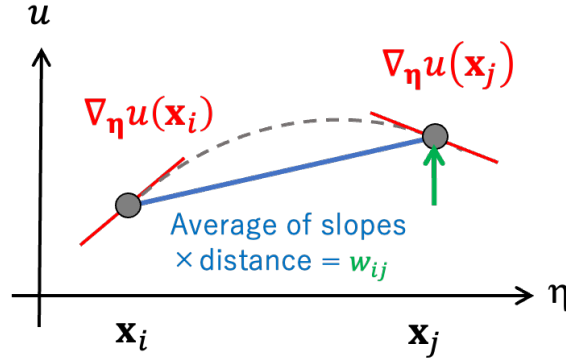
The edge weights $w_{(i,j)}$ are defined as follows:

$$w_{(i,j)} = \frac{1}{2} (\mathbf{v}_i + \mathbf{v}_j)^\top (\mathbf{x}_j - \mathbf{x}_i)$$

From the perspective of discretization, this definition approximates the integral of the flux along an edge from $\mathbf{x}_i$ to $\mathbf{x}_j$. Geometrically, this definition gives an approximation of the potential difference $U(\mathbf{x}_j) - U(\mathbf{x}_i)$.

Let $\boldsymbol{\eta} = \mathbf{x}_j - \mathbf{x}_i$ be the vector indicating the direction of the edge. The projection of $\mathbf{v}_i$ onto $\boldsymbol{\eta}$ is given by $\mathbf{v}_i^\top \boldsymbol{\eta} = \mathbf{v}_i^\top \frac{\boldsymbol{\eta}}{\|\boldsymbol{\eta}\|} \|\boldsymbol{\eta}\|$. If the velocity is determined solely by the gradient of U , i.e., $\mathbf{v}_i = \nabla U(\mathbf{x}_i)$, this projection gives the coefficient of the directional derivative $\nabla_{\boldsymbol{\eta}} U(\mathbf{x}_i) = \mathbf{v}_i^\top \boldsymbol{\eta} / \|\boldsymbol{\eta}\|$. Thus, $w_{(i,j)}$ represents the average slope of U along $\boldsymbol{\eta}$, which is then multiplied by the length $\|\boldsymbol{\eta}\|$.

In other words, $w_{(i,j)}$, as the average slope, determines the increment in the height of U along $\boldsymbol{\eta}$. The shape of the potential function, for which the second derivative is a constant, corresponds to a quadratic function (Fig. S15).



Supplementary Figure S15. Schematic illustrating how edge weights w_{ij} approximate the potential difference between two points x_i and x_j . When only the directional derivatives (slopes) $\nabla_{\boldsymbol{\eta}} u(x_i)$ and $\nabla_{\boldsymbol{\eta}} u(x_j)$ along the edge direction $\boldsymbol{\eta} = \mathbf{x}_j - \mathbf{x}_i$ are known, the problem of estimating the relative height $u(x_j) - u(x_i)$ reduces to fitting a quadratic function along this one-dimensional segment. The average of the two local slopes multiplied by the distance $\|\mathbf{x}_j - \mathbf{x}_i\|$ yields the edge weight w_{ij} , providing an approximation of the potential difference.

Specifically, consider a quadratic function along the axis of $\boldsymbol{\eta}$ with height $u(x)$ at position x :

$$\begin{aligned} u(x) &= a(x - c)^2 + b \\ u'(x) &= 2a(x - c) \end{aligned}$$

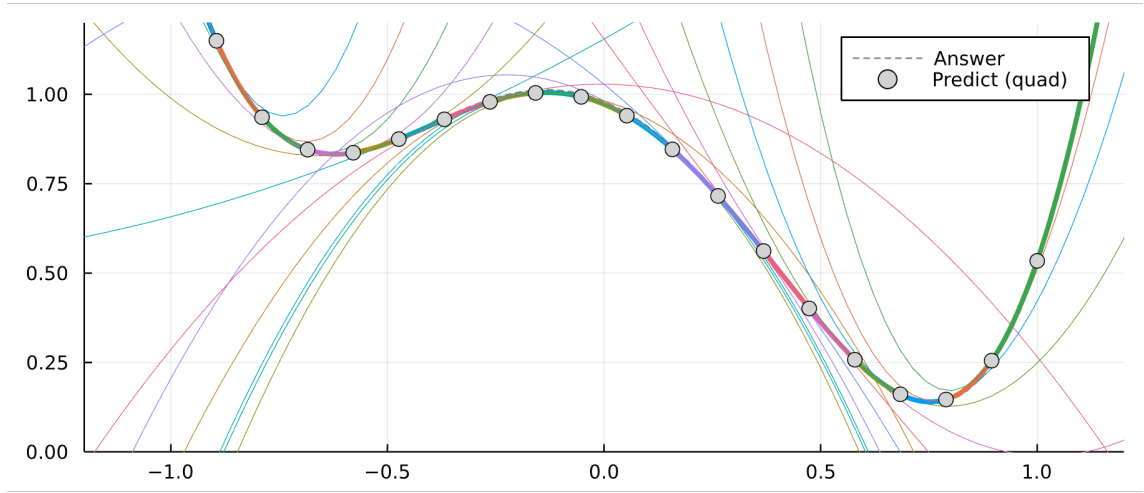
The increment in the height $u(x_2) - u(x_1)$ can be obtained using the differential coefficients at each point $d_1 = u'(x_1)$ and $d_2 = u'(x_2)$:

$$u(x_2) - u(x_1) = a(x_2 - c)^2 - a(x_1 - c)^2 = \frac{d_2^2 - d_1^2}{4a}$$

Given $d_2 - d_1 = 2a(x_2 - x_1)$, substituting $a = (d_2 - d_1) / 2(x_2 - x_1)$ yields the definition of the edge weight as follows:

$$u(x_2) - u(x_1) = \frac{1}{2} (d_1 + d_2) (x_2 - x_1)$$

The Figure S16 visualizes the shape of the potential function U (double-well potential), which is represented by piecewise quadratic functions fitted to the intervals of the sampled points and their slopes.



Supplementary Figure S16. Visualization of a double-well potential reconstructed by stitching together piecewise quadratic functions. Each colored curve represents a local quadratic fit determined from a pair of neighboring sampled points and their slopes, and the collection of these fits approximates the overall potential U .

2.3 Reconstruction of the Hessian matrix of U

Reconstruction of the Hessian matrix of U

The combinatorial divergence derived from the adjoint (transpose) of the gradient $\delta_0^\top$ differs from the usual divergence used in vector analysis. This is because the sum of the fluxes (weights) at the edges of the vertex does not match the definition of the divergence as the outgoing flux per unit volume. Leveraging the successful estimation of the potential using piecewise quadratic functions, we construct a div operator that can accurately restore the divergence in the original space.

From the previous results, the second-order derivative of the quadratic function can be easily derived as:

$$u''(x) = 2a = \frac{d_2 - d_1}{x_2 - x_1}$$

Since x_1, x_2, d_1, d_2 are all considered along the axis of $\boldsymbol{\eta}$, the second-order differential coefficient of the fitted quadratic function can be obtained by projecting the velocities onto $\boldsymbol{\eta}$ as follows:

$$U_{\eta\eta} = \frac{(\mathbf{x}_j - \mathbf{x}_i)^\top (\mathbf{v}_i - \mathbf{v}_j)}{\|\mathbf{x}_j - \mathbf{x}_i\|^2}$$

Note that the sign was flipped here: $d_2 - d_1 = \{(-\mathbf{v}_j) - (-\mathbf{v}_i)\}^\top \boldsymbol{\eta} / \|\boldsymbol{\eta}\|$.

Since the divergence is calculated from the trace of the Hessian matrix $U_{xx} + U_{yy} + \dots$, we next need to obtain the second-order partial derivatives U_{xx}, U_{yy} in the original coordinate system (x, y) .

For simplicity, consider a linear relationship between the two-dimensional coordinate systems (x, y) and (φ, ψ) . The relationship between the original Cartesian coordinates (x, y) and an oblique coordinate system (φ, ψ) spanned by two independent edges can be written as follows:

$$\begin{aligned} x &= a\varphi + b\psi \\ y &= c\varphi + d\psi \end{aligned}$$

Next, consider the partial derivatives of $U(x(\varphi, \psi), y(\varphi, \psi))$ with respect to φ and ψ . Using the chain rule, we obtain:

$$\begin{aligned} \frac{\partial U}{\partial \varphi} &= \frac{\partial U}{\partial x} \frac{\partial x}{\partial \varphi} + \frac{\partial U}{\partial y} \frac{\partial y}{\partial \varphi} \\ &= a \frac{\partial U}{\partial x} + c \frac{\partial U}{\partial y} \end{aligned}$$

Similarly, for the second-order partial derivatives, we obtain:

$$\begin{aligned} \frac{\partial^2 U}{\partial \varphi^2} &= \frac{\partial}{\partial \varphi} \left(a \frac{\partial U}{\partial x} + c \frac{\partial U}{\partial y} \right) \\ &= a \frac{\partial}{\partial x} \frac{\partial U}{\partial \varphi} + c \frac{\partial}{\partial y} \frac{\partial U}{\partial \varphi} \\ &= \left(a \frac{\partial}{\partial x} + c \frac{\partial}{\partial y} \right) \left(a \frac{\partial U}{\partial x} + c \frac{\partial U}{\partial y} \right) \\ &= a^2 \frac{\partial^2 U}{\partial x^2} + 2ac \frac{\partial^2 U}{\partial x \partial y} + c^2 \frac{\partial^2 U}{\partial y^2} \end{aligned}$$

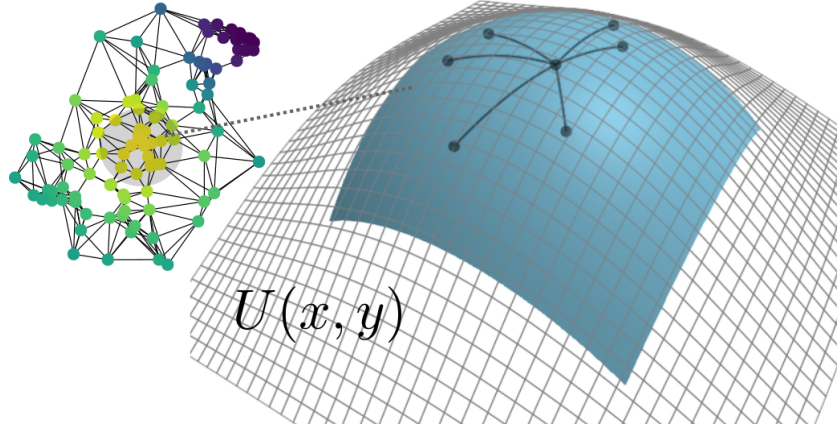
Focusing on the coefficients, the transformation of the second-order partial differential coefficients can be written in quadratic form. Thus, the relationship between the second-order directional derivative $U_{\eta\eta}$ with $\boldsymbol{\eta} = (a, c) = (\eta_1, \eta_2)$ and the Hessian matrix in the original coordinate system is given by:

$$\begin{aligned} U_{\eta\eta} &= (\eta_1 \ \eta_2) \begin{pmatrix} U_{xx} & U_{xy} \\ U_{xy} & U_{yy} \end{pmatrix} \begin{pmatrix} \eta_1 \\ \eta_2 \end{pmatrix} \\ &= \eta_1^2 U_{xx} + 2\eta_1 \eta_2 U_{xy} + \eta_2^2 U_{yy} \end{aligned}$$

By gathering the known second-order differential coefficients such as $U_{\eta\eta}$ in independent directions $\varphi, \psi, \dots$, a linear equation to estimate the elements of H can be obtained.

$$\begin{pmatrix} \varphi_1^2 & 2\varphi_1\varphi_2 & \varphi_2^2 \\ \psi_1^2 & 2\psi_1\psi_2 & \psi_2^2 \\ \eta_1^2 & 2\eta_1\eta_2 & \eta_2^2 \end{pmatrix} \begin{pmatrix} U_{xx} \\ U_{xy} \\ U_{yy} \end{pmatrix} = \begin{pmatrix} U_{\varphi\varphi} \\ U_{\psi\psi} \\ U_{\eta\eta} \end{pmatrix}$$

We denote the equation as $C\boldsymbol{\beta} = \mathbf{y}$. The divergence is obtained from $-trH$ after estimating H .



Supplementary Figure S17. Illustration of extending the one-dimensional piecewise quadratic fitting to higher dimensions. Around each sampled point, a local quadratic surface is estimated by combining the previously fitted quadratic curves along multiple neighboring edges, providing estimates of the Hessian components of the potential U . These locally fitted quadratic surfaces collectively approximate the shape of the underlying multi-dimensional potential landscape.

In practice, the problem is better solved as a least square problem $\min_{\beta} \|C\beta - \mathbf{y}\|^2$ to mitigate issues in real data analysis, such as noise and limited sample size.

In conclusion, we obtain a statistical estimation of the Hessian matrix H of U without directly addressing the mesh design, area/volume calculation, or boundary conditions, relying solely on the sampling of the second-order partial differential coefficients from quadratic functions fitted along incident edges (Fig. S17).

2.4 l_2 regularization on a graph

To enable the regularization of the estimated values (Hessian matrix elements) between neighboring vertices, we merged the individual Hessian matrix estimation problems into a single linear equation:

$$C\beta = \begin{pmatrix} C_1 & & \\ & \ddots & \\ & & C_N \end{pmatrix} \begin{pmatrix} \beta_1 \\ \vdots \\ \beta_N \end{pmatrix}$$

This allows us to introduce a penalty (Dirichlet energy on the graph) to the squares of the differences $\beta_i - \beta_j$ between adjacent vertices as:

$$P(\beta) = (D_0\beta)^\top (D_0\beta) = \beta^\top L_0\beta$$

Here, assume that the p parameters are arranged in a certain order as $\beta_i = (U_{xx}^i, U_{xy}^i, U_{yy}^i)$, such that the differences between each parameter can be individually evaluated using the extended gradient matrix $D_0 = \delta_0 \otimes I_p$ with the Kronecker product and $L_0 = D_0^\top D_0$.

The optimization problem with this regularization term is then solved as:

$$\min_{\beta} \|\mathbf{y} - C\beta\|_2^2 + \lambda P(\beta)$$

with some small constant $\lambda \in \mathbb{R}$.

The following solution can be obtained from the extrema with respect to the parameter β :

$$\hat{\beta} = \left(C^\top C + \lambda L_0 \right)^{-1} C^\top \mathbf{y}$$

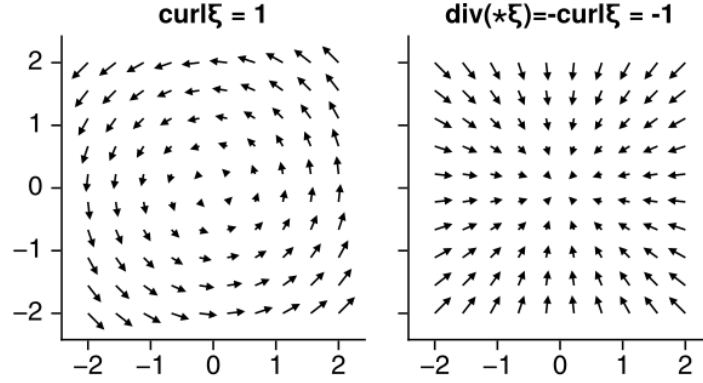
Given the large $Np \times Np$ size of the sparse matrix L_0 , efficient solvers such as the Krylov subspace method are recommended.

2.5 Reconstruction of the curl

For differential forms in the Euclidean space $\mathbb{R}^2$, the relationship between *div* and *curl* is given by:

$$\text{div}(\star \xi) = (\star d\star)(\star \xi) = (\star d)(\star \star)\xi = -\text{curl}\xi$$

where d is the exterior derivative and $\star$ is the Hodge star operator. Therefore, the curl can be approximated by the divergence of the *dual velocity field* $\star \xi$, which is obtained by rotating the velocity by $\pi/2$ in the PC1-2 plane (Fig. S18).



Supplementary Figure S18. Dual of velocity field. Taking the dual of a vector field in 2D corresponds to rotating each velocity vector by 90° .

2.6 Reconstruction of the Jacobian matrix

According to the result of Hodge decomposition, a vector field f can be expressed as the sum of a gradient part $-\nabla U$ and a divergence-free residual part r :

$$f = -\nabla U + r$$

Here, let D be the operator that gives the Jacobian matrix of f . Since D is linear and $D\nabla U$ is the Hessian matrix H of U , $D(f)$ has the following decomposition:

$$D(f) = D(-\nabla U + r) = -H + D(r)$$

If r is in 2D Euclidean space and consists of only rotational components (i.e., there are no holes in the space), $D(r)$ can be expressed as:

$$D(r) = \frac{\theta}{2} Q$$

where Q is the infinitesimal rotation matrix:

$$Q = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix},$$

$$\exp(\theta Q) = \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix}$$

Here, $\theta \in R$ is the magnitude of the rotation, which is calculated from the previously derived *curl*. The factor $1/2$ in $D(r)$ ensures that when $\theta = 1$, $\text{curl}(r) = (Q_{21} - Q_{12})/2 = 1$.

At this time, we do not perform a complete reconstruction of the rotational components in dimensions higher than 3. Instead, we approximate the rotation only in the PC1-2 plane of each tangent space. To obtain the $m \times m$ matrix of $D(r)$, we use the basis transformation

$$Q' = (W^\top W)^{-1} W^\top S Q (S^\top S)^{-1} S^\top W$$

This transformation maps the rotation $Q \in SO(2)$ on the PC1-2 plane to the rotation $Q' \in SO(m)$ on the same plane in the m -dimensional tangent space. Here, S is the matrix composed of the basis of the PC1-2 plane, and W is the matrix composed of the basis of the m -dimensional tangent space.

In conclusion, the Jacobian matrix of f in the m -dimensional tangent space can be approximated as follows:

$$D(f) \approx -H + \frac{\theta}{2} Q'$$

2.7 Simulating tangent-space dynamics in parameter space

We present the procedure with the unit sphere S^2 . Let θ and ϕ denote longitude and latitude in geographic coordinates, respectively, and define the map $f : (\theta, \phi) \mapsto (x, y, z)$ by

$$f(\theta, \phi) = (x, y, z) = (\cos \theta \cos \phi, \sin \theta \cos \phi, \sin \phi).$$

The Jacobian matrix of f is

$$J_f = \begin{pmatrix} \partial x / \partial \theta & \partial x / \partial \phi \\ \partial y / \partial \theta & \partial y / \partial \phi \\ \partial z / \partial \theta & \partial z / \partial \phi \end{pmatrix} = \begin{pmatrix} -\sin \theta \cos \phi & -\cos \theta \sin \phi \\ \cos \theta \cos \phi & -\sin \theta \sin \phi \\ 0 & \cos \phi \end{pmatrix} = (\mathbf{e}_\theta \quad \mathbf{e}_\phi).$$

Using this, the gradient on the tangent space can be evaluated in parameter space. The induced metric is

$$G = J_f^\top J_f = \begin{pmatrix} \cos^2 \phi & 0 \\ 0 & 1 \end{pmatrix},$$

showing that displacements in the θ -direction shrink by a factor $|\cos \phi| \leq 1$. Because the unit change of θ becomes non-uniform change on the sphere, the naïve parameter-space gradient $\nabla_{(\theta, \phi)} U = (\partial U / \partial \theta, \partial U / \partial \phi)$ is not appropriate one on the surface.

To compensate for this non-uniform scaling, we simulate dynamics in parameter space using $G^{-1} \nabla_{(\theta, \phi)} U$. After mapping via f , the scaled gradient in the tangent-space becomes $G^{1/2} G^{-1} \nabla_{(\theta, \phi)} U = G^{-1/2} \nabla_{(\theta, \phi)} U = \left(\frac{1}{|\cos \phi|} \partial U / \partial \theta, \partial U / \partial \phi \right)$, cancelling the $|\cos \phi|$ factor.

2.8 Setting gradient and rotational components on the sphere

We choose the potential as

$$U(\theta, \phi) = \boldsymbol{\mu}^\top f(\theta, \phi),$$

where $\boldsymbol{\mu} = f(-\pi/2, 0)$ (west end). This corresponds to a von Mises–Fisher-type distribution $p(x) \propto e^{-U(x)}$ that uses the cosine of the angle between $\boldsymbol{\mu}$ and $f(\theta, \phi)$ as potential energy. The gradient component is $\mathbf{g}(\theta, \phi) = -\nabla_{(\theta, \phi)} U$. For the rotational component we set $\mathbf{r}(\theta, \phi) = (\cos^2 \phi, 0)$. When the parameter-space velocity field is defined as $G^{-1}\mathbf{r}(\theta, \phi)$, mapping through f yields $J_f G^{-1}\mathbf{r} = \mathbf{e}_\theta$, i.e. constant tangent velocity along longitudinal direction.

The combined dynamics in parameter space are therefore

$$\left(\frac{d\theta}{dt}, \frac{d\phi}{dt} \right) = G^{-1} \{ \mathbf{g}(\theta, \phi) + \mathbf{r}(\theta, \phi) \}.$$

2.9 Designing the parallel coordinate system

To perform l_2 regularization on the data manifold (a set of local PCA spaces), it is necessary to establish local coordinate systems that provide correspondence between adjacent tangent vector spaces. In other words, in a curved space, determining the direction in which smoothness is required is not straightforward. This issue is addressed by the concept of *parallel transport* in differential geometry.

In this section, we introduce the idea of near parallel coordinate design in the n -dimensional space by using the subspace alignment method employed with vector diffusion maps [20] and the approach for designing discrete connection Laplacians via sheaves on graphs [26–27], which is an extension of combinatorial Hodge decomposition. The procedure consists of two steps: (1) subspace alignment and (2) minimization of the Rayleigh quotient of the discrete connection Laplacian.

Step 1: Subspace alignment

For two vertices i and j in a graph, let O_i and O_j be the $d \times r$ matrices composed of the orthonormal basis of the r -dimensional tangent vector spaces $T_i\mathcal{M}$ and $T_j\mathcal{M}$, respectively. The vector spaces spanned by the columns of O_i and O_j are denoted as $\text{Col}(O_i) = T_i\mathcal{M}$ and $\text{Col}(O_j) = T_j\mathcal{M}$.

If the two spaces have a flat connection, i.e., $\text{Col}(O_i) = \text{Col}(O_j)$, then there exists an orthogonal matrix O_{ij} that maps O_i into O_j via the linear combination of the columns of O_i :

$$\begin{aligned} O_i O_{ij} &= O_j \\ O_{ij} &= O_i^\top O_j \end{aligned}$$

Thus, for any vector $\mathbf{z}_i \in \text{Col}(O_i)$, O_{ij} , called the *transfer matrix*, provides the alignment of the axes in different spaces $\mathbf{z}_j = O_{ij}^\top \mathbf{z}_i \in \text{Col}(O_j)$ such that $O_i \mathbf{z}_i = O_j \mathbf{z}_j$.

However, in general, $\text{Col}(O_i) \neq \text{Col}(O_j)$ when the space has a nonzero curvature, and a linear combination cannot provide such alignment. Therefore, an approximate version of the transfer matrix is obtained as follows:

$$\tilde{O}_{ij} = \underset{O \in O(d)}{\operatorname{argmin}} \left\| O - O_i^\top O_j \right\|_F^2$$

The solution is obtained via singular value decomposition (SVD):

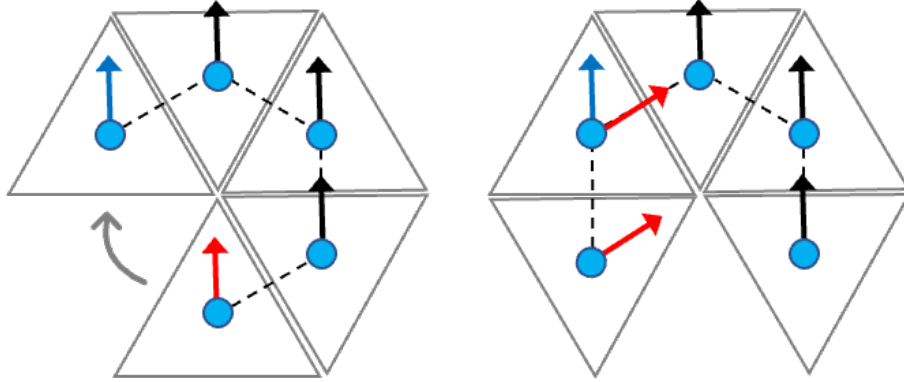
$$\begin{aligned} O_i^\top O_j &= U \Sigma V^\top \\ \tilde{O}_{ij} &= UV^\top = \tilde{O}_i^\top \tilde{O}_j \end{aligned}$$

where $\tilde{O}_i = U^\top$ and $\tilde{O}_j = V^\top$.

The columns of U and V corresponding to the largest singular value σ provide the pair $\mathbf{u}$ and $\mathbf{v}$ that maximizes $\cos \theta = \langle O_i \mathbf{u}, O_j \mathbf{v} \rangle = \sigma \leq 1$, where θ is the angle between $O_i \mathbf{u}$ and $O_j \mathbf{v}$. This is equivalent to canonical correlation analysis (CCA), which is used to find the closest axis pair in the column spaces of the given matrices X and Y . Additionally, the angles $\theta_i = \arccos(\sigma_i)$ corresponding to the singular values σ_i are called canonical angles, and the magnitude $\|\boldsymbol{\theta}\| = \|(\theta_1, \dots, \theta_m)\|$ can be used as the **Grassmann distance** to measure the distance between different subspaces.

Step 2: Minimizing the Rayleigh quotient of the connection Laplacian

The above method provides parallel transport between a pair of tangent spaces. However, since the solution depends on the selection of the pair (cf. holonomy; Fig. S19), it cannot be used as a consistent local coordinate system.



Supplementary Figure S19. Illustration of angular defect and holonomy. A vector is parallel transported along a closed path on the surface of a cone, represented here by unfolding the surface into five tangent triangle planes. The transported vector (red) differs by 60° after traversing the loop because of the angular defect (the discrete analogue of curvature).

Instead, we consider obtaining pairs of $\mathbf{z}_i \in \text{Col}(O_i)$ and $\mathbf{z}_j \in \text{Col}(O_j)$ for all edges $(i, j) \in E$ that are as close to parallel transport as possible. Let the *deviation from parallel transport* be $\boldsymbol{\varepsilon}_{(i,j)} = \tilde{O}_i \mathbf{z}_i - \tilde{O}_j \mathbf{z}_j$.

We aim to minimize:

$$\begin{aligned} \|\boldsymbol{\varepsilon}_{(i,j)}\|^2 &= \left(\tilde{O}_i \mathbf{z}_i - \tilde{O}_j \mathbf{z}_j \right)^\top \left(\tilde{O}_i \mathbf{z}_i - \tilde{O}_j \mathbf{z}_j \right) \\ &= 2 \left(1 - \mathbf{z}_i^\top \tilde{O}_i^\top \tilde{O}_j \mathbf{z}_j \right) \end{aligned}$$

where $\|\mathbf{z}_i\|^2 = \|\mathbf{z}_j\|^2 = 1$. This becomes the problem of maximizing $\mathbf{z}_i^\top \tilde{O}_i^\top \tilde{O}_j \mathbf{z}_j$. According to the previous results, if we select $\mathbf{z}_i = \mathbf{u}$ and $\mathbf{z}_j = \mathbf{v}$ via SVD, where $\mathbf{z}_i^\top \tilde{O}_i^\top \tilde{O}_j \mathbf{z}_j = 1$, we can still check that perfect parallel transport is possible with $\|\boldsymbol{\varepsilon}_{(i,j)}\|^2 = 0$.

If there is only one pair of tangent spaces, we can select $\mathbf{u}$ and $\mathbf{v}$ as described. However, if vertex j also has an edge (k, j) , different solutions $\mathbf{v}' \neq \mathbf{v}$ are given by the SVD of $O_k^\top O_j$.

Keeping $\mathbf{z}_j = \mathbf{v} \neq \mathbf{v}'$ may introduce a large deviation $\|\boldsymbol{\varepsilon}(j, k)\|^2$. To prevent this, we need to determine $\mathbf{z}_i, \mathbf{z}_j, \mathbf{z}_k, \dots \in \mathbb{R}^d$ that minimize the total deviation $\sum_{e \in E} \|\boldsymbol{\varepsilon}_e\|^2$ over all edges. This type of optimization problem on a graph with vector-valued functions $\mathbf{z}_i$ on vertices can be addressed using the framework of sheaves on graphs.

Similar to Hodge decomposition, ε_e can be evaluated on the basis of the sheaf version of the gradient δ_s of the selected coordinates $\mathbf{z}_i$ as follows:

$$\delta_s \mathbf{z} = \begin{pmatrix} * & \cdots & & & \\ O & \mathcal{F}_{i \leq e} & O & -\mathcal{F}_{j \leq e} & O \\ & & \cdots & & * \end{pmatrix} \begin{pmatrix} * \\ \mathbf{z}_i \\ * \\ \mathbf{z}_j \\ * \end{pmatrix} = \begin{pmatrix} & * \\ \mathcal{F}_{i \leq e} \mathbf{z}_i - \mathcal{F}_{j \leq e} \mathbf{z}_j \\ * \end{pmatrix}$$

In combinatorial Hodge decomposition, the gradient δ_0 represents the difference $\mathbf{z}_j - \mathbf{z}_i \in \mathbb{R}$ between vertices i and j for edge (i, j) . In contrast, this gradient δ_s generalizes δ_0 to a block matrix form, allowing the evaluation of differences $\mathcal{F}_{i \leq e} \mathbf{z}_i - \mathcal{F}_{j \leq e} \mathbf{z}_j$ with vector-valued functions $\mathbf{z}_i, \mathbf{z}_j \in \mathbb{R}^d$ at the vertices. These linear transformations $\mathcal{F}_{i \leq e}$ and $\mathcal{F}_{j \leq e}$ are called restriction maps.

Thus, by setting $\mathcal{F}_{i \leq e} = \tilde{O}_i$ and $\mathcal{F}_{j \leq e} = \tilde{O}_j$ for each row of δ_s , we can concisely describe the total deviation over the graph as:

$$\text{loss}(\mathbf{z}) = \sum_{(i,j) \in E} \|\tilde{O}_i \mathbf{z}_i - \tilde{O}_j \mathbf{z}_j\|^2 = \mathbf{z}^\top \Delta_s \mathbf{z}$$

where $\Delta_s = \delta_s^\top \delta_s$ is called the sheaf Laplacian.

The minimization of $\text{loss}(\mathbf{z})$ while keeping the magnitude of $\mathbf{x}$ constant is known as Rayleigh quotient $(\mathbf{z}^\top \Delta_s \mathbf{z}) / (\mathbf{z}^\top \mathbf{z})$ minimization. By definition, Δ_s is positive semidefinite, so it always has a nonnegative minimum eigenvalue, and the minimum value of the Rayleigh quotient is the smallest eigenvalue of Δ_s .

In conclusion, we adopted the eigenvectors $\mathbf{z}_i$ corresponding to the smallest eigenvalues of Δ_s as the local coordinate axes, which are nearly parallel and consistent across all tangent spaces.

2.10 Using near-parallel coordinates

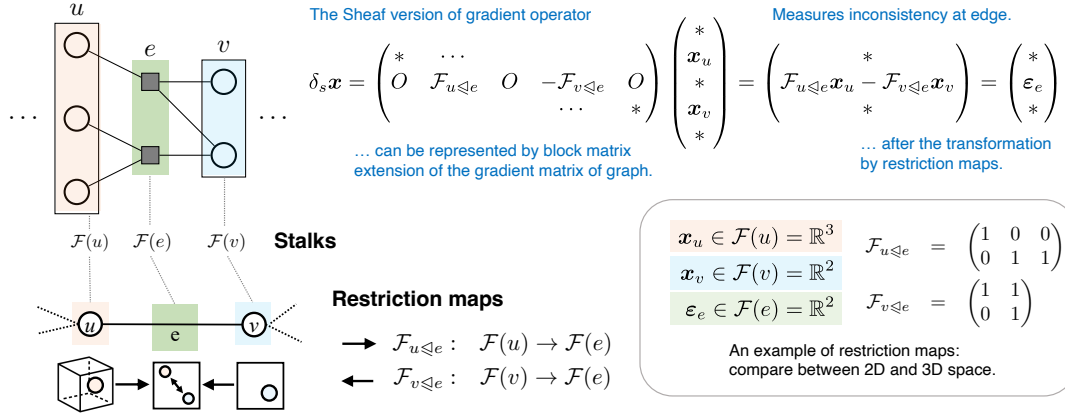
Starting from the smallest eigenvalues of the sheaf Laplacian, we select eigenvectors $\mathbf{z}_i$ up to the target reduced dimension r to construct the *aligned* local coordinate systems at each vertex. The obtained directions $O_i \mathbf{z}_i$ and $O_j \mathbf{z}_j$ serve as local bases for vertices i and j , and the vectors $O_i \mathbf{z}_i$ were also used when visualizing the direction fields in Figure 3.

When building the coefficient matrix C in $C\boldsymbol{\beta} = \mathbf{y}$ (Section 2.3), the edge direction $\boldsymbol{\eta} = \mathbf{x}_j - \mathbf{x}_i$ in the *unaligned* coordinates by PCA is replaced by a new representation $\boldsymbol{\xi}$ via projection onto $O_i \mathbf{z}_i$. Concretely, we compute $\boldsymbol{\xi} = (\xi_1, \dots, \xi_k, \dots, \xi_r)$ where $\xi_k = \boldsymbol{\eta}^\top O_i \mathbf{z}_i / \|\mathbf{z}_i\|$ in ascending order of eigenvalues. Subsequent Hessian reconstruction (Sections 2.3–2.4) proceeds as in the flat-space case. The resulting Hessian H_i and Jacobian J_i at vertex i are expressed in the new coordinate system rather than the original (unaligned) space. Eigenvalues of H_i , J_i , divergence, and curl are invariant under this change of basis. However, some applications that require specific coordinate representations—such as interaction analyses between genes—can employ the `basischange` function provided in the ddHodge Julia package to transform H and J back to a desired basis as explained in the Section 2.6.

2.11 Basic idea of sheaves on graphs

Sheaves on graphs, introduced by Hansen and Ghrist, extend the combinatorial Hodge decomposition reviewed earlier in these Notes. Although sheaf theory involves abstract mathematics, the constructions used here are expected to provide an intuitive understanding when viewed as data-analysis tools on graphs.

Each vertex u is assigned a local data space (a *stalk*) $\mathcal{F}(u)$, typically a vector space such as $\mathbb{R}^d$. Edges connect these local data spaces through *restriction maps* $\mathcal{F}_{u \leq e}$ and $\mathcal{F}_{v \leq e}$, which specify how to compare data at adjacent vertices. In our setting, the stalks store direction vectors in tangent spaces, and the restriction maps are linear transportation operators. The combinatorial Hodge decomposition with gradient operator δ_0 in 2.1 corresponds to the special case of a *constant sheaf* of fixed stalks $\mathbb{R}$ on $\mathcal{G}$ where each restriction map is ± 1 . The sheaf Laplacian $\Delta_s = \delta_s^\top \delta_s$, analogous to the ordinary graph Laplacian, updates the local data $\mathbf{x}_u, \mathbf{x}_v, \dots$ to minimize discrepancies $\boldsymbol{\varepsilon}_e$ across edges, thereby maintaining global consistency (harmonization).



Supplementary Figure S20. Illustration of how discrepancies are evaluated in sheaves on graphs when data at adjacent vertices have different dimensions. A three-dimensional vector at u and a two-dimensional vector at v are compared through the restriction maps shown in the square frame, producing a two-dimensional residual vector in the edge stalk $\mathcal{F}(e)$.

Figure S20 illustrates a toy example in which $\mathcal{F}(u) = \mathbb{R}^3$ and $\mathcal{F}(v) = \mathbb{R}^2$. To compare quantities in different dimensions, the stalk $\mathcal{F}(e) = \mathbb{R}^2$ at edge e stores the residuals: (i) the first component of $\mathbf{x}_u$ minus the sum of the first and second components of $\mathbf{x}_v$, and (ii) the sum of the second and third components of $\mathbf{x}_u$ minus the third component of $\mathbf{x}_v$. These comparisons correspond to the restriction maps shown in the square frame. In this paper, every stalk has the same fixed dimension r , and the restriction maps are fixed as calculated transportation operators; therefore, the optimization concerns the vertex data $\mathbf{x}_u$ and $\mathbf{x}_v$. More general designs—for example, stalks of differing dimensions or restriction maps treated as optimization targets—are also possible within this framework.