Clustering Dynamics on Graphs: From Spectral Clustering to Mean Shift Through Fokker–Planck Interpolation

Katy Craig, Nicolás García Trillos, and Dejan Slepčev

Abstract In this work, we build a unifying framework to interpolate between density-driven and geometry-based algorithms for data clustering and, specifically, to connect the mean shift algorithm with spectral clustering at discrete and continuum levels. We seek this connection through the introduction of Fokker–Planck equations on data graphs. Besides introducing new forms of mean shift algorithms on graphs, we provide new theoretical insights on the behavior of the family of diffusion maps in the large sample limit as well as provide new connections between diffusion maps and mean shift dynamics on a fixed graph. Several numerical examples illustrate our theoretical findings and highlight the benefits of interpolating density-driven and geometry-based clustering algorithms.

1 Introduction

In this work we establish new connections between two popular but seemingly unrelated families of methodologies used in unsupervised learning. The first family that we consider is density based and includes mode seeking clustering approaches such as the mean shift algorithm introduced in [15] and reviewed in [9], while the second family is based on spectral geometric ideas applied to graph settings and includes methodologies such as Laplacian eigenmaps [3] and spectral clustering [32]. After discussing the mean shift algorithm in Euclidean space and reviewing a family of spectral methods for clustering on graphs, we seek these connections

K. Craig
University of California, Santa Barbara, CA, USA
e-mail: kcraig@math.ucsb.edu

N. García Trillos
University of Wisconsin, Madison, WI, USA
e-mail: garciatrillo@wisc.edu

D. Slepčev (✉)
Carnegie Mellon University, Pittsburg, PA, USA
e-mail: slepcev@math.cmu.edu

N. Bellomo et al. (eds.), *Active Particles, Volume 3*, Modeling and Simulation in Science, Engineering and Technology, https://doi.org/10.1007/978-3-030-93302-9_4

in two different ways. First, motivated by some heuristics at the continuum level (i.e., infinite data setting level), we take a suitable dynamic perspective and introduce appropriate interpolating *Fokker–Planck* equations on data graphs. This construction is inspired by the variational formulation in the Wasserstein space of Fokker–Planck equations at the continuum level and utilizes recent geometric formulations of PDEs on graphs. Second, we revisit the diffusion maps from [12] (with an extended range for the parameter indexing the family) and in particular show that, when parameterized conveniently and in the large data limit, the family of diffusion maps is closely related to the same family of continuum dynamics motivating our Fokker–Planck interpolations on graphs. At the finite data level, we show that by taking an extreme value of the parameter indexing the diffusion maps we can retrieve a specific graph version of the mean shift algorithm introduced in [25]. Our new theoretical insights are accompanied by extensive numerical examples aimed at illustrating the benefits of interpolating density and geometry-driven clustering algorithms.

To begin our discussion, let us recall that unsupervised learning is one of the fundamental settings in machine learning where the goal is to find structure in a data set $\mathcal{X}$ without the aid of any labels associated with the data. For example, if the data set $\mathcal{X}$ consisted of images of animals, a standard task in unsupervised learning would be to recognize the structure of groups in $\mathcal{X}$ without using information of the actual classes that may be represented in the data set (e.g., {dog, olinguito, caterpillar, ...}); in the literature, this task is known as *data clustering* and will be our main focus in this chapter. Other unsupervised learning tasks include dimensionality reduction [41] and anomaly detection [22], among others.

When clusters are geometrically simple, for example, when they are dense sets of points separated in space, elementary clustering methods, like k-means or k-median clustering, are sufficient to identify the clusters. However, in practice, clusters are often geometrically complex due to the natural variations that objects belonging to the cluster may have and also due to invariances that some object classes posses. To handle such data sets, there is a large class of clustering algorithms, including the ones that will be explored in this chapter, which are described as two-step procedures consisting of an embedding step and an actual clustering step where a more standard, typically simple, clustering method is used on the embedded data. In mathematical terms, in the first step, the goal is to construct a map

$$\Psi : \mathcal{X} \to \mathcal{Y}$$

between the data points in $\mathcal{X}$ and a space $\mathcal{Y}$ (e.g., $\mathcal{Y} = \mathbb{R}^k$ for some small k, or in general a metric space) to "disentangle" the original data as much as possible, and in the second step, the actual clustering is obtained by running a simple clustering algorithm such as K-means (if the set $\mathcal{Y}$ is the Euclidean space, for example). While both steps are important and need careful consideration, it is in the first step, and specifically in the choice of Ψ, that most clustering algorithms differ from each other. For example, as we will see in Sect. 1.1, in some version of mean shift, Ψ is induced by gradient ascent dynamics of a density estimator starting at the different

data points (when the data points are assumed to lie in Euclidean space), whereas in spectral methods for clustering in graph-based settings the map Ψ is typically built using the low-lying spectrum of a suitable graph Laplacian. At its heart, different choices of Ψ capture different heuristic interpretations of the loosely defined notion of "data cluster." Some constructions have a density-driven flavor (e.g., mean shift), while others are inspired by geometric considerations (e.g., spectral clustering). The notions of density-based and geometry-based algorithms, however, are not monolithic, and each individual algorithm has nuances and drawbacks that are important to take into account when deciding whether to use it or not in a given situation; in our numerical experiments, we will provide a series of examples that highlight some of the qualitative weaknesses of different clustering algorithms, density or geometry driven. Our main aim is to introduce a new mathematical framework to interpolate between these two seemingly unrelated families of clustering algorithms and to provide new insights for existing interpolations like *diffusion maps*.

In the rest of this introduction, we present some background material that is used in the remainder.

1.1 Mean Shift-Based Methods

Let $\mathcal{X} = \{x_1, \ldots, x_n\}$ be a data set in $\mathbb{R}^d$. One heuristic way to define data "clusters" is to describe them as regions in space of high concentration of points separated by areas of low density. One algorithm that uses this heuristic definition is the *mean shift algorithm*; see [9]. In essence, mean shift is a hill climbing scheme that seeks the local modes of a density estimator constructed from the observed data in an attempt to identify data clusters.

To make the discussion more precise, let us first describe the setting where the points $x_1, \ldots, x_n$ are obtained by sampling a probability distribution supported on the whole $\mathbb{R}^d$ with (unknown) smooth enough density $\rho : \mathbb{R}^d \to \mathbb{R}$. The first step in mean shift is to build an estimator for ρ of the form:

$$\hat{\rho}(x) := \frac{1}{n\delta^d} \sum_{i=1}^{n} \kappa\left(\frac{|x - x_i|}{\delta}\right),$$

where $\kappa : \mathbb{R}_+ \to \mathbb{R}_+$ is an appropriately normalized kernel, and $\delta > 0$ is a suitable bandwidth; for simplicity, we can take the standard Gaussian kernel $\kappa(s) = \frac{1}{\sqrt{2\pi}} \exp^{-s^2/2}$. Then, for every point x_i, one considers the iterations:

$$x_i(t+1) = x_i(t) + \nabla \log \hat{\rho}(x_i(t)) \tag{1.1}$$

starting at time $t = 0$ with $x_i(0) = x_i$. Each data point x_i is then mapped to its associated $x_i(T)$ for some user-specified T, implicitly defining in this way a map Ψ as described in the introduction. It is important to notice that mean shift is, in

the way introduced above, a *monotonic* scheme, i.e., $\hat{\rho}(x_i(t))$ is non-decreasing as a function of $t \in \mathbb{N}$. In [8], this is shown to be a consequence of a deeper property that in particular relates mean shift with the expectation-maximization algorithm applied to a closely related problem. The name mean shift originates from the fact that iterate $x_i(t+1)$ in (1.1) coincides with the mean of some distribution that is centered around the iterate $x_i(t)$, and thus (1.1) can be described as a "mean shifting" scheme.

We now introduce the continuum analogue of the mean shift algorithm (1.1). Namely, (1.1) can be seen as the iterates of the Euler scheme, for time step $h = 1$ of the following ODE system:

$$\begin{cases} \dot{x}_i(t) = \nabla \log \hat{\rho}(x_i(t)), \quad t > 0 \\ x_i(0) = x_i. \end{cases} \tag{1.2}$$

In the remainder, we will abuse the terminology slightly and refer to the above continuous time dynamics as *mean shift*. We note that the $\hat{\rho}$ is monotonically increasing along the trajectories of (1.2).

To utilize the mean shift dynamics for clustering, for some prespecified time $T > 0$ (that at least theoretically can be taken to be infinity under mild conditions on ρ), we consider the embedding map:

$$\Psi_{MS}(x_i) := x_i(T), \quad x_i \in \mathcal{X}.$$

When the number of data points n in $\mathcal{X}$ is large, and the bandwidth h is small enough (but not too small), one can heuristically expect that the gradient lines of the density estimator $\hat{\rho}$ resemble those of the true density ρ; see for example [2]. In particular, with an appropriate tuning of bandwidth δ as a function of n, and with a large value of T defining the time horizon for the dynamics, one can expect Ψ_{MS} to send the original data points to a set of points that are close to the local modes of the density ρ. In short, mean shift is expected to cluster the original data set by assigning points to the same cluster if they belong to the same basin of attraction of the gradient ascent dynamics for the density ρ.

If the density ρ is supported on a manifold, $\mathcal{M}$, embedded in $\mathbb{R}^d$, and that information is available, one can consider mean shift dynamics restricted to the manifold. Indeed, to define manifold mean shift, we just need to consider the flow ODE (1.2), where ∇ is replaced by the gradient on $\mathcal{M}$, which for manifolds in $\mathbb{R}^d$ is just the projection of ∇ to the tangent space $T_x\mathcal{M}$. We notice that this extends to manifolds with boundary where at boundary point $x \in \partial\mathcal{M}$ one is projecting to the interior half-space $T_x^{in}\mathcal{M}$. We denote the projection to the tangent space (interior tangent space at the boundary) by $P_{T\mathcal{M}}$ and write the resulting ODE as

$$\begin{cases} \dot{x}_i(t) = P_{T\mathcal{M}} \nabla \log(\hat{\rho})(x_i(t)), \quad t > 0 \\ x_i(0) = x_i \in \mathcal{M}. \end{cases} \tag{1.3}$$

1.1.1 Lifting the Dynamics to the Wasserstein Space

Looking forward to our discussion in subsequent sections where we introduce mean shift algorithms on graphs, it is convenient to rewrite (1.2) in an alternative way using dynamics in the *Wasserstein* space $\mathcal{P}_2(\mathbb{R}^d)$. As it turns out, the ODE (1.2) is closely related to an ODE in $\mathcal{P}_2(\mathbb{R}^d)$ (i.e., the space of Borel probability measures over $\mathbb{R}^d$ with finite second moments). Precisely, we consider

$$\partial_t \mu_t + \operatorname{div}(\nabla \log(\hat{\rho})\mu_t) = 0, \quad t > 0, \tag{1.4}$$

with initial datum $\mu_0 = \delta_{x_i}$; equation (1.4) must be interpreted in the weak sense (see Chapter 8.1. in [1]). Indeed, when $\mu_0 = \delta_{x_i}$, it is straightforward to see that the solution to (1.4) is given by $\mu_t = \delta_{x_i(t)}$, where $x_i(\cdot)$ solves the ODE (1.2) in the base space $\mathbb{R}^d$. What is more, in the same way that (1.2) can be understood as the gradient descent dynamics for $-\log(\hat{\rho})$ in the base space $\mathbb{R}^d$, it is possible to interpret (1.4) directly as the gradient flow of the potential energy:

$$\mathcal{E}(\mu) := -\int_{\mathbb{R}^d} \log(\hat{\rho}(x)) d\mu(x), \quad \mu \in \mathcal{P}_2(\mathbb{R}^d) \tag{1.5}$$

with respect to the Wasserstein metric d_W, which in dynamic form reads

$$d_W^2(\nu, \tilde{\nu}) = \inf_{t\in[0,1]\mapsto(\nu_t, \vec{V}_t)} \int_0^1 \int_{\mathbb{R}^d} |\vec{V}_t|^2 \, d\nu_t dt, \tag{1.6}$$

where the infimum is taken over all solutions $(\nu_t, \vec{V}_t)$ to the continuity equation

$$\partial_t \nu_t + \operatorname{div}(\nu_t \vec{V}_t) = 0,$$

with $\nu_0 = \nu$ and $\nu_1 = \tilde{\nu}$.

The previous discussion suggests the following alternative definition for the embedding map associated with mean shift:

$$\Psi_{MS}(x_i) := \mu_{i,T} \in \mathcal{P}_2(\mathbb{R}^d),$$

where in order to obtain $\mu_{i,T}$, we consider the gradient flow dynamics $\mathcal{E}$ in the Wasserstein space initialized at the point $\mu_0 = \delta_{x_i}$ (i.e., Eq. (1.4)). While this new interpretation may seem superfluous at first sight given that $\mu_{i,T} = \delta_{x_i(T)}$, we will later see that working in the space of probability measures is convenient, as this alternative representation motivates new versions of mean shift algorithms for data clustering on structures such as weighted graphs; see Sect. 2.2.

1.2 Spectral Methods

Let us now discuss another family of algorithms used in unsupervised learning that are based on ideas from spectral geometry. The input in these algorithms is a collection of edge weights w describing the similarities between data points in $\mathcal{X}$; we let $n = |\mathcal{X}|$. For simplicity, we assume that the weight function $w : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is symmetric and that all its entries are non-negative. We further assume that the weighted graph $\mathcal{G} = (\mathcal{X}, w)$ is connected in the sense that for every $x, x' \in \mathcal{X}$ there exists a path $x_0, \ldots, x_m \in \mathcal{X}$ with $x_0 = x$, $x_m = x'$ and $w(x_l, x_{l+1}) > 0$ for every $l = 0, \ldots, m-1$. At this stage, we do not assume any specific geometric structure in the data set $\mathcal{X}$ or on the weight function w (in Sect. 4, however, we focus our discussion on proximity graphs).

Let us now give the definition of well-known graph analogues of gradient, divergence, and Laplacian operators. To a function $\phi : \mathcal{X} \to \mathbb{R}$, we associate a *discrete gradient*, a function of the form $\nabla_{\mathcal{G}}\phi : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ defined by

$$\nabla_{\mathcal{G}}\phi(x, x') := \phi(x') - \phi(x).$$

Given a function $U : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ (i.e., a discrete vector field), we define its *discrete divergence* as the function $\mathrm{div}_{\mathcal{G}} U : \mathcal{X} \to \mathbb{R}$ given by

$$\mathrm{div}_{\mathcal{G}}\, U(x) := \frac{1}{2} \sum_{x'} (U(x', x) - U(x, x'))w(x, x').$$

With these definitions, we can now introduce the *unnormalized Laplacian* associated with the graph $\mathcal{G}$ as the operator $\Delta_{\mathcal{G}} : L^2(\mathcal{X}) \to L^2(\mathcal{X})$ defined according to

$$\Delta_{\mathcal{G}} := \mathrm{div}_{\mathcal{G}} \circ \nabla_{\mathcal{G}}$$

or more explicitly as

$$\Delta_{\mathcal{G}} u(x_i) = \sum_j (u(x_i) - u(x_j))w(x_i, x_j), \quad x_i \in \mathcal{X}, \quad u \in L^2(\mathcal{X}). \tag{1.7}$$

From the representation $\Delta_{\mathcal{G}} = \mathrm{div}_{\mathcal{G}} \circ \nabla_{\mathcal{G}}$, it is straightforward to verify that $\Delta_{\mathcal{G}}$ is a self-adjoint and positive semi-definite operator with respect to the $L^2(\mathcal{X})$ inner product (i.e., the Euclidean inner product in $\mathbb{R}^n$ after identifying real-valued functions on $\mathcal{X}$ with $\mathbb{R}^n$). It can also be shown that $\Delta_{\mathcal{G}}$ has zero as an eigenvalue with multiplicity equal to the number of connected components of $\mathcal{G}$ (in this case 1 by assumption); see [45]. Moreover, even when the multiplicity of the zero eigenvalue is uninformative about the group structure of the data set, the low-lying spectrum of $\Delta_{\mathcal{G}}$ still carries important geometric information for clustering. In particular, $\Delta_{\mathcal{G}}$'s small eigenvalues and their corresponding eigenvectors contain information

on *bottlenecks* in $\mathcal{G}$ and on the corresponding regions that are separated by them; the connection between the spectrum of $\Delta_{\mathcal{G}}$ and the bottlenecks in $\mathcal{G}$ is expressed more precisely with the relationship between Cheeger constants and Fiedler eigenvalues; see [45]. With this motivation in mind, [3] introduced a nonlinear transformation of the data points known as *Laplacian eigenmap*:

$$x_i \in \mathcal{X} \longmapsto \begin{pmatrix} \phi_1(x_i) \\ \vdots \\ \phi_k(x_i) \end{pmatrix} \in \mathbb{R}^k,$$

where $\phi_1, \ldots, \phi_k$ are the eigenvectors corresponding to the first k eigenvalues of $\Delta_{\mathcal{G}}$. The above Laplacian eigenmap and other similar transformations serve as the embedding map in the first step in most spectral methods for partitioning and data clustering. Said clustering algorithms have a rich history, and related ideas have been present in the literature for decades, see [32, 35, 45] and the references within. For example, some versions of spectral clustering consider a conformal transformation of the Laplacian eigenmap in which coordinates in the embedding space are rescaled differently according to corresponding eigenvalues and the choice of a timescale parameter. More precisely, one may consider

$$\hat{\Psi}_{SC}(x_i) := \begin{pmatrix} e^{-T\lambda_1}\phi_1(x_i) \\ \vdots \\ e^{-T\lambda_k}\phi_k(x_i) \end{pmatrix} \in \mathbb{R}^k, \quad x_i \in \mathcal{X},$$

for some $T > 0$, where in the above λ_l represents the eigenvalue corresponding to the eigenvector ϕ_l. In Sect. 2, we provide a dynamic interpretation of the map $\hat{\Psi}_{SC}$.

Remark 1.1 There are several other ways in the literature to construct the embedding maps Ψ from graph Laplacian eigenvectors. In [32], for example, an extra normalization step across eigenvectors is considered for each data point. By introducing this extra normalization step, one effectively maps the data points into the unit sphere in Euclidean space. The work [34] argues in favor of this type of normalization and proposes the use of an angular version of k-means clustering on the embedded data set. The work [17] also analyzes the geometric structure of spectral embeddings, both at the data level and at the continuum population level. The normalization step in [32] can also be motivated from a robustness to outliers perspective if one insists on running k-means with the ℓ^2 metric and not with for example the ℓ^1 metric. As discussed in the introduction, constructing a data embedding is only part of the full clustering problem. What metric and what clustering method should be used on the embedded data are important practical and theoretical questions. In subsequent sections, our embedded data points will have the form of probability vectors. Several metrics could then be used to cluster the embedded data points (TV, L^2, W_2, etc), each with its own advantages and disadvantages (theoretical or computational). The emphasis of our discussion in the

rest of the chapter, however, will be on the embedding maps themselves. We leave the analysis of the effect of different metrics used to cluster the embedded data for future work.

1.2.1 Normalized Versions of the Graph Laplacian

Different versions of graph Laplacians can be constructed to include additional information about vertex degrees as well as to normalize the size of eigenvalues. We distinguish between two ways to normalize the graph Laplacian $\Delta_{\mathcal{G}}$. One is based on reweighing operators and the other on renormalizing edge weights.

Operator-Based Renormalizations To start, we first write the graph Laplacian $\Delta_{\mathcal{G}}$ in matrix form. For that purpose, let $W = [w(x_i, x_j)]_{i,j}$ be the matrix of weights, and let

$$d(x_i) = \sum_{x_j \neq x_i} w(x_i, x_j) \tag{1.8}$$

be the weighted degrees; in the remainder, we may also use the notation $d_i = d(x_i)$ whenever no confusion arises from doing so. Let $D = \operatorname{diag}(d_1, \ldots, d_n)$ be the diagonal matrix of degrees. The Laplacian $\Delta_{\mathcal{G}}$ can then be written in matrix form as

$$\Delta_{\mathcal{G}} = D - W. \tag{1.9}$$

In terms of this matrix representation, the *normalized* graph Laplacian, as introduced in [45], can be written as

$$L = D^{-\frac{1}{2}} \Delta_{\mathcal{G}} D^{-\frac{1}{2}} = I - D^{-\frac{1}{2}} W D^{-\frac{1}{2}}. \tag{1.10}$$

Notice that the matrix L is symmetric and positive semi-definite as it follows directly from the same properties for $\Delta_{\mathcal{G}}$. The *random-walk* graph Laplacian, on the other hand, is given by

$$L^{rw} = D^{-1} \Delta_{\mathcal{G}} = I - D^{-1} W. \tag{1.11}$$

Remark 1.2 It is straightforward to show that the matrix L^{rw} is similar to the matrix L, and thus the random-walk Laplacian has the same eigenvalues as the normalized graph Laplacian. Moreover, if we explicitly use the representation $L^{rw^T} = D^{1/2} L D^{-1/2}$, where L^{rw^T} is the transpose of L^{rw}, we can see that if $\tilde{u}$ is an eigenvector of L with eigenvalue λ, then $\phi = D^{1/2}\tilde{\phi}$ is an eigenvector of L^{rw^T} with eigenvalue λ. In particular, since L is symmetric, we can find a collection of vectors $\phi_1, \ldots, \phi_n \in \mathbb{R}^n$ that form an orthonormal basis (with respect to the inner product $\langle D^{-1}\cdot, \cdot \rangle$) for $\mathbb{R}^n$ and where each of the ϕ_l is an eigenvector for L^{rw^T}.

We use the above observation in Sect. 1.2.2 and later at the beginning of Sect. 2.

Edge Weight Renormalizations Here, the idea is to adjust the weights of the graph $\mathcal{G} = (\mathcal{X}, w)$ and use one of the Laplacian normalizations introduced before on the new graph. One of the most popular families of graph Laplacian normalizations based on edge reweighing was introduced in [12] and includes the generators for the so-called *diffusion maps* which we now discuss.

For a given choice of parameter $\alpha \in (-\infty, 1]$, we construct new edge weights, $w_\alpha(x, y)$, as follows:

$$w_\alpha(x, y) := \frac{w(x, y)}{d(x)^\alpha d(y)^\alpha},$$

where recall d is the weighted vertex degree. On the new graph $(\mathcal{X}, w_\alpha)$, one can consider all forms of graph Laplacian discussed earlier. In the sequel, however, we follow [12] and restrict our attention to the reweighed random-walk Laplacian which in matrix form can be written as

$$L_\alpha^{rw} = I - D_\alpha^{-1} W_\alpha,$$

where W_α is the matrix of edge weights of the new graph and D_α its associated degree matrix, i.e., $d_\alpha(x) = \sum_{y \neq x} w_\alpha(x, y)$. We note that the weight matrix W_α is still symmetric.

Let Q_α^{rw} be the *weighted diffusion rate* matrix

$$Q_\alpha^{rw} := -C_\alpha L_\alpha^{rw},$$

where C_α is a positive constant that we introduce for modeling purposes. In particular, in Sect. 4, we will see that, in the context of proximity graphs on data sampled from a distribution on a manifold $\mathcal{M}$, by choosing the constant C_α appropriately, we can ensure a desirable behavior of Q_α^{rw} as the number of data points in $\mathcal{X}$ grows. Notice that we may alternatively write Q_α^{rw} as a function:

$$Q_\alpha^{rw}(x, y) := C_\alpha \begin{cases} \frac{w_\alpha(x,y)}{\sum_{z \neq x} w_\alpha(x,z)} & \text{if } x \neq y, \\ -1 & \text{if } x = y. \end{cases} \tag{1.12}$$

Remark 1.3 In this chapter, we consider the range $\alpha \in (-\infty, 1]$ and not $[0, 1]$ as usually done in the literature. One important point that we stress throughout the chapter is that by considering the interval $(-\infty, 1]$, we obtain an actual interpolation between density-based (in the form of some version of mean shift) and geometry-driven clustering algorithms.

1.2.2 More General Spectral Embeddings

Following Remark 1.2, for a given $\alpha \in (-\infty, 1]$, we consider an orthonormal basis (relative to the inner product $\langle D_\alpha^{-1}\cdot, \cdot\rangle$) $\phi_1, \ldots, \phi_n$ of eigenvectors of $L_\alpha^{rw^T}$ with corresponding eigenvalues $\lambda_1 \leqslant \lambda_2 \leqslant \cdots \leqslant \lambda_n$ and define

$$\hat{\Psi}'_\alpha(x_i) := \begin{pmatrix} e^{-T\lambda_1}\tilde{\phi}_1(x_i) \\ \vdots \\ e^{-T\lambda_k}\tilde{\phi}_k(x_i) \end{pmatrix} \in \mathbb{R}^k, \quad x_i \in \mathcal{X}, \tag{1.13}$$

where, in the above, $\tilde{\phi}_l = D_\alpha^{-1/2}\phi_l$. We can see that the map $\hat{\Psi}'_\alpha$ has the same form as the map $\hat{\Psi}_{SC}$ at the beginning of Sect. 1.2. In Sect. 2, we provide a more dynamic interpretation of the map $\hat{\Psi}'_\alpha$.

1.3 Outline

Having discussed the mean shift algorithm in Euclidean setting (or in general on a submanifold $\mathcal{M}$ of $\mathbb{R}^d$), as well as some spectral methods for clustering in the graph setting, in what follows we attempt to build bridges between geometry-based and density-driven clustering algorithms. Our first step is to introduce general data embedding maps Ψ associated with the dynamics induced by arbitrary rate matrices on $\mathcal{X}$. We will then define data graph analogues of mean shift dynamics. We do this in Sect. 2 where we define a new version of mean shift on graphs inspired by the discussion in Sect. 1.1.1 and review other versions of mean shift on graphs such as Quickshift [44] and KNF [25]. In Sect. 3.1, we discuss two versions of Fokker–Planck equations on graphs which serve as interpolating dynamics between geometry- and density-driven dynamics for clustering on data graphs. One version is based on a direct interpolation between diffusion and mean shift (the latter one as defined in Sect. 2.2.1) and is inspired by Fokker–Planck equations at the continuum level. The second version is an extended version of the diffusion maps from [12] obtained by appropriate reweighing and normalization of the data graph. In Sect. 3.2, we show that the KNF mean shift dynamics can be seen as a particular case of the family of diffusion maps when the parameter indexing this family is sent to negative infinity. This result is our first concrete connection between mean shift algorithms and spectral methods for clustering. In Sect. 4, we study the continuum limits of the Fokker–Planck equations introduced in Sect. 3 when the graph of interest is a proximity graph. This analysis will allow us to provide further insights into diffusion maps, mean shift, and spectral clustering. In Sect. 5, we present a series of numerical experiments aimed at illustrating some of our theoretical insights.

2 Mean Shift and Fokker–Planck Dynamics on Graphs

Consider a weighted graph $\mathcal{G} = (\mathcal{X}, w)$ as in Sect. (1.2). Let $\mathcal{P}(\mathcal{X})$ denote the set of probability measures on $\mathcal{X}$ which we identify with n-dimensional vectors. All of the dynamics we consider can be written as (continuous time) Markov chains on graphs.

Definition 2.1 A *continuous time Markov chain* $u : [0, T] \to \mathcal{P}(\mathcal{X})$ is a solution to the ordinary differential equation

$$\begin{cases} \partial_t u_t(y) = \sum_{x \in \mathcal{X}} u_t(x) Q(x, y), \quad t > 0 \\ u_0 = u^0, \end{cases} \tag{2.1}$$

where $u^0 \in \mathcal{P}(\mathcal{X})$ and $Q : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a *transition rate matrix* Q; that is, Q satisfies $Q(x, y) \geqslant 0$ for $y \neq x$ and $Q(x, x) = -\sum_{y \neq x} Q(x, y)$.

Notice that in terms of matrix exponentials, the solution to (2.1) can be written as

$$u_t(y) = \sum_{x \in \mathcal{X}} u^0(x)(e^{tQ})(x, y).$$

Remark 2.2 The operation on the right-hand side of the first equation in (2.1) can be interpreted as a matrix multiplication of the form $u_t Q$, where u_t is interpreted as a row vector. Alternatively, we can use the transpose of Q and write $Q^T u_t$ if we interpret u_t as a column vector.

Remark 2.3 (Conservation of Mass and Positivity) For any transition rate matrix, we have

$$\sum_{x \in \mathcal{X}} Q(x, y) = 0, \quad \forall y \in \mathcal{X},$$

which ensures that $\sum_x u_t(x) = \sum_x u_0(x) = 1$ for all $t \geqslant 0$. Note that, in practice, this is often accomplished by specifying the *off-diagonal* entries of the transition rate matrix, $Q(x, y)$ for $x \neq y$, and then setting the diagonal entries to equal opposite the associated diagonal degree matrix, $d(x, x) = \sum_{y \neq x} Q(x, y)$. Likewise, for any transition rate matrix, the fact that $Q(x, y) \geqslant 0$ for $y \neq x$ ensures that if $u_0(x) \geqslant 0$, then $u_t(x) \geqslant 0$ for all $t \geqslant 0$.

Remark 2.4 Notice that $Q = -\Delta_{\mathcal{G}}$, where we recall $\Delta_{\mathcal{G}}$ is defined in (1.7), is indeed a rate matrix and thus induces evolution equations in $\mathcal{P}(\mathcal{X})$. Likewise, the weighted diffusion rate matrix Q_α^{rw} from (1.12) is a rate matrix as introduced in Definition 2.1.

For a general rate matrix Q, we define the data embedding map:

$$\hat{\Psi}_Q(x_i) := u_{i,T,Q} \in \mathcal{P}(X), \quad x_i \in \mathcal{X}, \tag{2.2}$$

where $u_{i,T,Q}$ represents the solution of (2.1) when the initial condition $u^0 \in \mathcal{P}(\mathcal{X})$ is defined by $u^0(x) = 1$ for $x = x_i$ and $u^0(x) = 0$ otherwise. In the sequel, and specifically in our numerics section, we will use the $L^2(\mathcal{X})$ metric between the points $\hat{\Psi}_Q(x_i)$ in order to build clusters through K-means regardless of the rate matrix Q that we use to construct the embedding $\hat{\Psi}_Q$. Remember that by $L^2(\mathcal{X})$ distance we mean the quantity:

$$\sum_{x'} (u_{i,T,Q}(x') - u_{j,T,Q}(x'))^2.$$

In the next subsections, we discuss some specifics of the choice $Q = Q_\alpha^{rw}$ and then introduce two classes of rate matrices Q that give meaning to the idea of mean shift on graphs.

2.1 Dynamic Interpretation of Spectral Embeddings

When we take $Q = Q_\alpha^{rw}$, we abuse notation slightly and write $\hat{\Psi}_\alpha$ instead of $\hat{\Psi}_{Q_\alpha^{rw}}$ and $u_{i,T,\alpha}$ instead of $u_{i,T,Q_\alpha^{rw}}$. In the next proposition, we make the connection between the embedding map $\hat{\Psi}_\alpha$ and the spectral embedding $\hat{\Psi}'_\alpha$ from (1.13) explicit.

Proposition 2.5 *For every $\alpha \in (-\infty, 1]$, we can write*

$$u_{i,T,\alpha}(x) = \sum_{l=1}^{n} e^{-T\lambda_l} \frac{\phi_l(x_i)}{(d_\alpha(x_i))^{1/2}} \phi_l(x), \quad \forall x \in \mathcal{X},$$

where the $\phi_1, \ldots, \phi_n$ form an orthonormal basis for $\mathbb{R}^n$ (with respect to the inner product $\langle D_\alpha^{-1} \cdot, \cdot \rangle$) and each ϕ_l is an eigenvector of $L_\alpha^{rw^T}$ with eigenvalue λ_l. In other words, the coordinates of the vector $\hat{\Psi}'_\alpha(x_i)$ correspond to the representation of $\hat{\Psi}_\alpha(x_i)$ in the basis $\phi_1, \ldots, \phi_n$.

Proof This follows from a simple application of the spectral theorem using Remark 1.2. □

Remark 2.6 An alternative way to compare the points $\hat{\Psi}_\alpha(x_i)$ and $\hat{\Psi}_\alpha(x_j)$ is to compute their weighted distance:

$$\sum_{x'} \left(\frac{u_{i,T,Q}(x')}{\sqrt{d_\alpha(x')}} - \frac{u_{j,T,Q}(x')}{\sqrt{d_\alpha(x')}} \right)^2.$$

This construction is introduced in [12] and is referred to as *diffusion distance*. It is worth mentioning that, as pointed out in [12], the diffusion distance between $\hat{\Psi}_\alpha(x_i)$ and $\hat{\Psi}_\alpha(x_j)$ coincides with the Euclidean distance between $\hat{\Psi}'_\alpha(x_i)$ and $\hat{\Psi}'_\alpha(x_j)$ when $k = n$. Notice that the diffusion metric is conformal to the $L^2(\mathcal{X})$ metric.

Remark 2.7 We remark that the embedding maps $\hat{\Psi}_{SC}$ and $\hat{\Psi}_Q$ for $Q = -\Delta_\mathcal{G}$ are connected in a similar way as the maps $\hat{\Psi}'_\alpha$ and $\hat{\Psi}_\alpha$ are. Indeed, if we let $\phi_1, \ldots, \phi_n$ be an orthonormal basis for $L^2(\mathcal{X})$ consisting of eigenvectors of $\Delta_\mathcal{G}$ (remember that $\Delta_\mathcal{G}$ is positive semi-definite with respect to $L^2(\mathcal{X})$), then the coordinates of $\hat{\Psi}_{SC}(x_i)$ are precisely the coordinates of the representation of $u_{i,T,Q}$ in the basis $\phi_1, \ldots, \phi_n$.

2.2 The Mean Shift Algorithm on Graphs

In the next two subsections, we discuss two different ways to introduce mean shift on $\mathcal{G} = (\mathcal{X}, w)$. In both cases, we define an associated rate matrix Q.

2.2.1 Mean Shift on Graphs as Inspired by Wasserstein Gradient Flows

The discussion in Sect. 1.1.1 shows that the mean shift dynamics can be viewed as a gradient flow in the spaces of probability measures endowed with Wasserstein metric. Recent works [14, 29] provide a way to consider Wasserstein type gradient flows which are restricted to graphs. This allows one to take advantage of the information about the geometry of data that their initial distribution provides. More importantly, for our considerations, it allows one to combine the mean shift and spectral methods.

The notion of Wasserstein metric on graphs introduced by Maas [29] provides the desired framework. Here, we will consider the upwind variant of the Wasserstein geometry on graphs introduced in [14] since it avoids the problems that the metric of [29] has when dealing with the continuity equations on graphs, see Remark 1.2 in [14].

In particular, we actually consider a *quasi-metric* on $\mathcal{P}(\mathcal{X})$ (and not a metric) defined by

$$\hat{d}^2_W(v, \tilde{v}) := \inf_{t\in[0,1]\mapsto(v_t,V_t)} \int_0^1 \sum_{x,y} |V_t(x,y)_+|^2 \, \overline{v}_t(x,y) w(x,y) dt,$$

where the infimum is taken over all solutions (v_t, V_t) to the discrete continuity equation:

$$\partial_t v_t + \mathrm{div}_\mathcal{G}(\overline{v}_t \cdot V_t) = 0,$$

with $v_0 = v$ and $v_1 = \tilde{v}$ and where V_t is anti-symmetric for all t. In the above, the constant $C_{ms} > 0$ is introduced for modeling purposes and will become relevant in Sect. 4. We use the upwinding interpolation [11, 14]:

$$\overline{v}_t(x, y) := \begin{cases} v_t(x) & \text{if } V_t(x, y) \geqslant 0, \\ v_t(y) & \text{if } V_t(x, x) < 0 \end{cases} \tag{2.3}$$

and interpret the discrete vector field $\overline{v}_t \cdot V_t$ as the elementwise product of $\overline{v}_t$ and V_t. Finally, we use a_+ to denote the positive part of the number a.

Next, we consider the general potential energy:

$$\hat{\mathcal{E}}(u) := -\sum_{x \in \mathcal{X}} B(x)u(x), \quad u \in \mathcal{P}(\mathcal{X}),$$

for some $B : \mathcal{X} \to \mathbb{R}$. This energy serves as an analogue of (1.5).

Following the analysis and geometric interpretation in [14], it is possible to show that the gradient flow of $\hat{\mathcal{E}}$ with respect to the quasi-metric $\hat{d}_W$ takes the form:

$$\partial_t u_t(y) = \sum_{x \in \mathcal{X}} u_t(x) Q^B(x, y), \tag{2.4}$$

where Q^B is the rate matrix defined by

$$Q^B(x, y) := \begin{cases} (B(y) - B(x))_+ w(x, y), & \text{for } x \neq y, \\ -\sum_{z \neq x} (B(z) - B(x))_+ w(x, z), & \text{for } x = y. \end{cases} \tag{2.5}$$

In Sect. 4.1, we explore the connection between the graph mean shift dynamics (2.4) and the mean shift dynamics on an m-dimensional submanifold of $\mathbb{R}^d$ as introduced in Sect. 1.4. This is done in the context of proximity graphs over a data set $\mathcal{X} = \{x_1, \ldots, x_n\}$ obtained by sampling a distribution with density ρ on $\mathcal{M}$. In particular, we formally show that the graph mean shift dynamics converges to the continuum one if

$$B(x) = -\frac{C_{ms}}{\rho(x)}, \quad x \in \mathcal{M}.$$

In practice, however, since ρ is in general unavailable, ρ above can be replaced with a density estimator $\hat{\rho}$. Given such an estimator $\hat{\rho}$ (which in principle can be considered on general graphs, not just ones embedded in $\mathbb{R}^d$), we define the transition kernel for the *graph mean shift* dynamics as

$$Q^{ms}(x, y) := C_{ms} \begin{cases} \left(-\frac{1}{\hat{\rho}(y)} + \frac{1}{\hat{\rho}(x)}\right)_+ w(x, y), & \text{for } x \neq y, \\ -\sum_{z \neq x} \left(-\frac{1}{\hat{\rho}(z)} + \frac{1}{\hat{\rho}(x)}\right)_+ w(x, z), & \text{for } x = y, \end{cases} \tag{2.6}$$

where the constant $C_{ms} > 0$ just sets the timescale. It will be specified in Sect. 4.1 so that the equation has the desired limit as $n \to \infty$. For data in $\mathbb{R}^d$, it is natural to consider a kernel density estimator $\hat{\rho}$. In particular, in all of our experiments, we consider the following kernel density estimator:

$$\hat{\rho}_\delta(x) := \frac{1}{n} \sum_{y \in \mathcal{X}} \psi_\delta(x - y), \quad \psi_\delta(x) = \frac{1}{(2\pi)^{m/2}\delta^m} e^{-|x|^2/(2\delta^2)}. \tag{2.7}$$

For an abstract graph $\mathcal{G}$ (i.e., $\mathcal{X}$ is not necessarily a subset of Euclidean space), one can consider the degree of the graph at each $x \in \mathcal{X}$ as a substitute for $\hat{\rho}$ in the above expressions.

From the previous discussion and in direct analogy with the discussion in Sect. 1.1.1, we introduce the data embedding map:

$$\hat{\Psi}_{ms}(x_i) := u_{i,T,Q^{ms}} \in \mathcal{P}(\mathcal{X}). \tag{2.8}$$

Remark 2.8 The quasi-metric $\hat{d}_W$ and the geometry of the PDEs on graphs that it induces have been studied in [14]. One important point made in that paper (see Remark 1.2. in [14]) is that the support of the solution to the induced equations may change and move as time increases. For us, this property is essential as we initialize our graph mean shift dynamics at Diracs located at each of the data points.

We now provide a couple of illustrations comparing the mean shift dynamics (1.2) and the graph mean shift dynamics defined by (2.6). We consider data on manifold $\mathcal{M} = [0, 4] \times \{0, 0.7\}$. The measure ρ has uniform density on the two line segments. We consider 280 data points sampled from ρ, Fig. 1, and sampled from ρ with Gaussian noise of variance 0.1 in vertical direction, Fig. 3a.

We compare the dynamics on $\mathcal{M}$ for different bandwidths δ of the kernel density estimator. In particular, we consider a value of δ that is small enough for the strips to be seen as separate and a value of delta that is large enough for the strips to be considered together, Fig. 2. For large δ, we see rather different behavior of the two dynamics. The standard mean shift quickly mixes the data from the two lines and the information about the two clusters is lost. On the other hand, while the driving

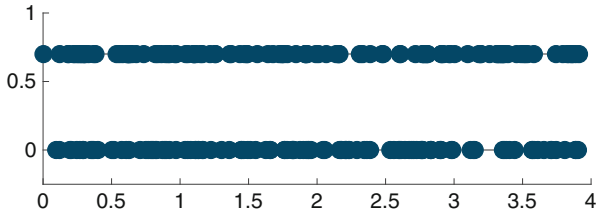
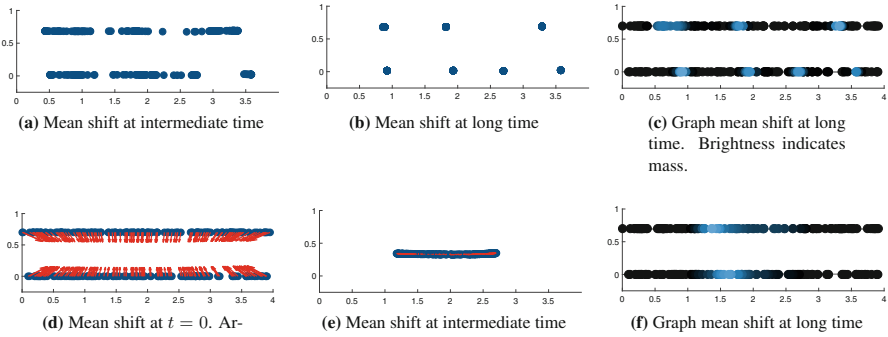


Fig. 1 Initial data for the experiments below. There are 280 points sampled from a uniform distribution on two line segments



(a) Mean shift at intermediate time

(b) Mean shift at long time

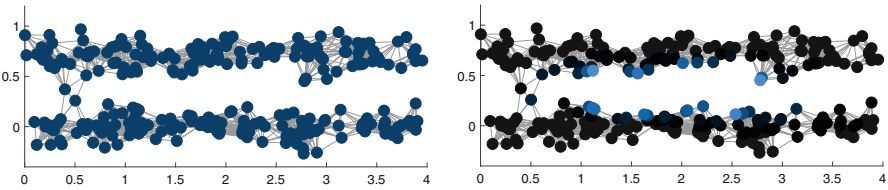
(c) Graph mean shift at long time. Brightness indicates mass.

(d) Mean shift at $t = 0$. Arrows represent velocity

(e) Mean shift at intermediate time

(f) Graph mean shift at long time

Fig. 2 We compare the dynamics for the mean shift (1.2) and graph mean shift (2.4)–(2.6). The top row shows the dynamics for $\delta = \frac{1}{4}$ bandwidth of the KDE. Both approaches give similar results. The stripes evolve independently and there are spurious local maxima due to randomness. The bottom row shows the dynamics for a larger $\delta = \frac{1}{\sqrt{2}}$. The KDE has a unique maximum. Mean shift quickly mixes the stripes into one, which then collapses to a point. On the other hand, since graph mean shift dynamics is constrained to the sample points the stripes do not mix and a single mode is identified in each stripe. (**a**) Mean shift at intermediate time . (**b**) Mean shift at long time. (**c**) Graph mean shift at long time. Brightness indicates mass. (**d**) Mean shift at $t = 0$. Arrows represent velocity. (**e**) Mean shift at intermediate time. (**f**) Graph mean shift at long time



(a) $t = 0$

(b) $t = \infty$

Fig. 3 Graph mean shift for $\delta = \frac{1}{\sqrt{2}}$. If noise is added to the data above, most of the dynamics behave as before. The exception is shown. The graph mean shift does not reach the modes as on Fig. 2f. Namely due to geometric roughness of the data the dynamics gets trapped at blue points. (**a**) $t = 0$. (**b**) $t = \infty$

force is the same, in the graph mean shift the dynamics is restricted to the data, thus preventing the mixing. In particular, separate modes are identified in each clump.

We note that this desirable behavior is somewhat fragile when noise is present, Fig. 3b. In particular, the roughness of the boundary prevents the mass to reach the mode. We will discuss later that this is mitigated by adding a bit of diffusion to the dynamics, see Sect. 5.2.6.

2.2.2 Quickshift and KNF

There are some alternative definitions of mean shift on graphs that are popular in the literature. One such algorithm is Quickshift [44], which is similar to an earlier algorithm by Koontz, Narendra, and Fukunaga [25]. Both algorithms can be described as hill climbing iterative algorithms for the maximization of a potential function $\hat{B}$.

Let $\hat{B} : \mathcal{X} \to \mathbb{R}$ be the potential for which we want to define "gradient ascent dynamics" along the graph $(\mathcal{X}, w)$. Let $\hat{D}(x, y) \geqslant 0$ be a notion of "distance" between points x and y which is typically defined through the weights w. Both the Quickshift and KNF algorithms have a Markov chain interpretation that we describe in a general form that allows for the (unlikely) existence of non-unique maximizers of $\hat{B} : \mathcal{X} \to \mathbb{R}$ around a given node $x \in \mathcal{X}$. To describe the associated rate matrices, let us define for every $x \in \mathcal{X}$ the sets

$$M_{QS,x} := \left\{ y \in \mathcal{X} \ : \ y \text{ maximizes: } \frac{1}{\hat{D}(x, y)} \mathbf{1}_{\hat{B}(y) > \hat{B}(x)} \right\}$$

and

$$M_{KNF,x} := \left\{ y \in \mathcal{X} \ : \ y \text{ maximizes: } \frac{(\hat{B}(y) - \hat{B}(x))_+}{\hat{D}(x, y)} \mathbf{1}_{\hat{D}(x,y) < r} \right\}.$$

The Quickshift and KNF algorithms are then the paths in the Markov chains with rate matrices:

$$Q_{QS}(x, y) = \begin{cases} \frac{1}{\sharp M_{QS,x}} & \text{if } y \in M_{QS,x}, \\ -1 & \text{if } y = x, \\ 0 & \text{otherwise,} \end{cases} \tag{2.9}$$

$$Q_{KNF}(x, y) = \begin{cases} \frac{1}{\sharp M_{KNF,x}} & \text{if } y \in M_{KNF,x}, \\ -1 & \text{if } y = x, \\ 0 & \text{otherwise,} \end{cases} \tag{2.10}$$

respectively. In Sect. 3.2, we establish a connection between the family of rate matrices Q_α^{rw} and Q_{KNF}.

3 Fokker–Planck Equations on Graphs

3.1 Fokker–Planck Equations on Graphs via Interpolation

The first type of interpolation between density- and geometry-driven clustering algorithms that we discuss in this chapter is based on a direct interpolation of the rate matrices Q^{ms} and Q_1^{rw}. Namely, for $\beta \in [0, 1]$, we consider

$$Q_\beta := \beta Q^{ms} + (1 - \beta) Q_1^{rw}. \tag{3.1}$$

It is straightforward to see that the resulting Q_β continues to be a rate matrix and as such it induces dynamics in the space $\mathcal{P}(\mathcal{X})$. We can then use the framework from Sect. 2 and abuse notation slightly to write $\hat{\Psi}^\beta$ instead of $\hat{\Psi}_{Q_\beta}$ as well as $u_{i,T,\beta}$ instead of u_{i,T,Q_β}.

The choice of rate matrix Q_β is motivated by the Fokker–Planck equation:

$$\partial_t f_t = \beta \mathrm{div}(\nabla \phi f_t) + (1 - \beta)\Delta f_t$$

on a submanifold $\mathcal{M}$ of $\mathbb{R}^d$, which in the context of Sect. 4 can be proved to be a formal continuum limit of the evolution induced by Q_β as the number of data points grows. On the other hand, we notice that when we take $\beta = 1$ in Q_β, we recover the mean shift dynamics from Sect. 2.2.1. If on the contrary we set $\beta = 0$, we obtain the dynamics induced by the rate matrix Q_1^{rw}, which, at least in the context of Sect. 4, can be shown to be connected in the large sample limit to the heat equation on a manifold $\mathcal{M}$ where the data density plays no role.

3.2 Fokker–Planck Equation on Graphs via Reweighing and Connections to Graph Mean Shift

Another interpolation between density-driven and geometry-based clustering dynamics is induced by the family of rate matrices $\{Q_\alpha^{rw}\}_{\alpha \in (-\infty, 1]}$. Indeed, in Sect. 4.2, we prove that in the proximity graph setting, the discrete dynamics associated with the rate matrices Q_α^{rw} are closely related, in the large data limit, to the same family of Fokker–Planck equations at the continuum level mentioned in Sect. 3.1. What is more, without taking a large sample limit, we see that the family $\{Q_\alpha^{rw}\}_{\alpha \in (-\infty, 1]}$ interpolates between Q_1^{rw} and a rate matrix inducing graph mean shift dynamics, only that this time the version of mean shift that is meaningful is a particular case of the KNF formulation from Sect. 2.2.2. We prove this in the next proposition.

Proposition 3.1 *Let $(\mathcal{X}, w)$ be an arbitrary weighted graph satisfying the conditions at the beginning of Sect. 1.2. Set $C_\alpha = 1$ for every $\alpha \in (-\infty, 1]$. Then,*

$$\lim_{\alpha\to-\infty} Q_\alpha^{rw} = Q_{-\infty}^{rw}, \tag{3.2}$$

where

$$Q_{-\infty}^{rw}(x,y) := -\mathbf{1}_{y=x} + \begin{cases} \frac{w(x,y)}{\sum_{z\in M_{KNF,x}} w(x,z)} & \text{if } y \in M_{KNF,x} \\ 0 & \text{otherwise,} \end{cases} \tag{3.3}$$

where in the definition of $M_{KNF,x}$ *we are using* $\hat{B}(z) = d(z)$, $\hat{D}(x,y) = 1$ *if* $w(x,y) > 0$ *and* $\hat{D}(x,y) = \infty$ *if* $w(x,y) = 0$, *and* $r > 1$.

We notice that this is essentially the KNF rate matrix defined in (2.10) with only a difference in the way ties are broken when the maximum of d around a point is not unique. This distinction is mostly irrelevant since generically we may expect no ties. On the other hand, if for some reason there are ties but the non-zero weights in the graph are equal, then the two tie-breaking rules coincide.

Proof As the cases are analogous, let us consider only the case $y \neq x$. Note that

$$Q_\alpha^{rw}(x,y) = \frac{w_\alpha(x,y)}{\sum_{z\neq x} w_\alpha(x,z)} = \frac{w(x,y)d(x)^{-\alpha}d(y)^{-\alpha}}{\sum_{z\neq x} w(x,z)d(x)^{-\alpha}d(z)^{-\alpha}} = \frac{w(x,y)d(y)^{-\alpha}}{\sum_{z\neq x} w(x,z)d(z)^{-\alpha}}.$$

If $y \notin M_{KNF,x}$, consider $z \in M_{KNF,x}$. Then,

$$Q_\alpha^{rw}(x,y) \leqslant \frac{w(x,y)}{w(x,z)}\left(\frac{d(y)}{d(z)}\right)^{-\alpha} \to 0 \quad \text{as } \alpha \to -\infty.$$

If $y \in M_{KNF,x}$, then

$$Q_\alpha^{rw}(x,y) = \frac{w(x,y)}{\sum_{z\neq x} w(x,z)\left(\frac{d(z)}{d(y)}\right)^{-\alpha}} \to \frac{w(x,y)}{\sum_{z\in M_{KNF,x}} w(x,z)} \quad \text{as } \alpha \to -\infty.$$

□

4 Continuum Limits of Fokker–Planck Equations on Graphs and Implications

In this section, we further study the Fokker–Planck equations introduced in Sect. 3 and discuss their connection with Fokker–Planck equations at the continuum level. For such connection to be possible, we impose additional assumptions on the graph $\mathcal{G} = (\mathcal{X}, w)$. In particular, we assume that $\mathcal{G}$ is a *proximity graph* on $\mathcal{X} = \{x_1, \ldots, x_n\}$, where the x_i are assumed to be i.i.d. samples from a distribution on a smooth compact m-dimensional manifold without boundary $\mathcal{M}$ embedded in

$\mathbb{R}^d$ and having density $\rho : \mathcal{M} \to \mathbb{R}$ with respect to the volume form on $\mathcal{M}$. By proximity graph, we mean that the weights $w(x_i, x_j)$ are defined according to

$$w(x, y) := \eta_\varepsilon(|x - y|), \quad \eta_\varepsilon(r) = \frac{1}{\varepsilon^m}\eta\left(\frac{r}{\varepsilon}\right), \tag{4.1}$$

where $\varepsilon > 0$ is a bandwidth appropriately scaled with the number of samples n, η is a function $\eta : [0, \infty) \to [0, \infty)$ with compact support, and $|x - y|$ denotes the Euclidean distance between x and y.

4.1 Continuum Limit of Mean Shift Dynamics on Graphs

In order to formally derive the large sample limit of equation (2.4), we study the action of Q^{ms} on a smooth function $u : \mathcal{M} \to \mathbb{R}$. That is, we compute

$$\sum_{x\in\mathcal{X}} u(x)Q^{ms}(x, y) = -C_{ms} \sum_{x\in\mathcal{X}} \left[(B(x) - B(y))_+ u(y) - (B(x) - B(y))_- u(x)\right] w(x, y)$$

as $n \to \infty$ and $\varepsilon \to 0$ at a slow enough rate. Since our goal below is to deduce formal continuum limits, we will assume that $\mathcal{M}$ is flat. We note that when $\mathcal{M}$ is a smooth manifold, the deflection of the manifold from the tangent space is at most quadratic, and thus the error introduced is small when ε is small. In this way, we can avoid using the notation and constructions from differential geometry as well as some approximation arguments that obscure the reason why the limit holds. Providing a rigorous argument for the convergence of the dynamics remains an open problem.

In what follows, we use $\rho_n = \frac{1}{n}\sum_{x\in\mathcal{X}} \delta_x$ to denote the empirical distribution on the data points; here, we use the notation ρ_n to highlight the connection between the data points and the density function ρ. We also consider the constants

$$C_{ms} = \frac{1}{n\varepsilon^2\sigma_{\eta'}}, \quad \sigma_{\eta'} = \frac{1}{2m}\int_{\mathbb{R}^m} |z|^2\eta(|z|)dz,$$

and assume that the potential B is a $C^3(\mathcal{M})$ function. With the above definitions, we can explicitly write

$$-\sum_{x\in\mathcal{X}} u(x)Q^{ms}(x, y) = \frac{1}{n\varepsilon^{m+2}\sigma_{\eta'}} \sum_{x\in\mathcal{X}} \left[(B(x) - B(y))_+ u(y) - (B(x) - B(y))_- u(x)\right] \times \eta\left(\frac{|x - y|}{\varepsilon}\right) \tag{4.2}$$

and using the smoothness of u and B equate the above to

$$
\begin{aligned}
&= \frac{1}{\varepsilon^{m+2}\sigma_{\eta'}} \int_{\mathcal{M}} \left[(B(x) - B(y))_+ u(y) - (B(x) - B(y))_- u(x) \right] \eta \left(\frac{|x-y|}{\varepsilon} \right) \rho_n(x) dx \\
&= \frac{1}{\varepsilon^{m+2}\sigma_{\eta'}} \int_{\mathcal{M}} \Big[(B(x) - B(y))_+ (u(y) - u(x)) \\
&\quad + \big((B(x) - B(y))_+ - (B(x) - B(y))_- \big) u(x) \Big] \eta \left(\frac{|x-y|}{\varepsilon} \right) \rho_n(x) dx \\
&= \frac{1}{\varepsilon^{m+2}\sigma_{\eta'}} \int_{\mathcal{M}} \left[(B(x) - B(y))_+ (u(y) - u(x)) + (B(x) - B(y)) u(x) \right] \eta \\
&\quad \times \left(\frac{|x-y|}{\varepsilon} \right) \rho_n(x) dx \\
&\approx \frac{1}{\varepsilon^{m+2}\sigma_{\eta'}} \int_{\mathcal{M}} \left[(B(x) - B(y))_+ \langle \nabla u(y), y - x \rangle + \langle \nabla B(y), x - y \rangle u(x) \right] \eta \\
&\times \left(\frac{|x-y|}{\varepsilon} \right) \rho_n(x) dx \\
&\quad + \frac{1}{2\varepsilon^{m+2}\sigma_{\eta'}} \int_{\mathcal{M}} \Big[(B(x) - B(y))_+ \langle D^2 u(y)(x-y), y - x \rangle \\
&\quad + \langle D^2 B(y)(x-y), x - y \rangle u(x) \Big] \eta \left(\frac{|x-y|}{\varepsilon} \right) \rho_n(x) dx \\
&=: A_1 + A_2 + A_3 + A_4.
\end{aligned}
$$

Next, we analyze each of the terms A_1, A_2, A_3, and A_4. For A_1, we see that

$$
\begin{aligned}
A_1 &\approx \frac{1}{\varepsilon^{m+2}\sigma_{\eta'}} \int_{\mathcal{M}} (B(x) - B(y))_+ \langle \nabla u(y), y - x \rangle \eta \left(\frac{|x-y|}{\varepsilon} \right) \rho(x) dx \\
&\approx -\frac{1}{\varepsilon^{m+2}\sigma_{\eta'}} \int_{\langle x-y, \nabla B(y) \rangle \geqslant 0} \langle \nabla u(y), x - y \rangle \langle x - y, \nabla B(y) \rangle \eta \left(\frac{|x-y|}{\varepsilon} \right) \rho(x) dx \\
&\approx -\frac{\rho(y)}{\varepsilon^{m+2}\sigma_{\eta'}} \int_{\langle x-y, \nabla B(y) \rangle \geqslant 0} \langle \nabla u(y), x - y \rangle \langle x - y, \nabla B(y) \rangle \eta \left(\frac{|x-y|}{\varepsilon} \right) dx \\
&= -\frac{\rho(y)}{\sigma_{\eta'}} \int_{\langle z, v \rangle \geqslant 0} \langle v', z \rangle \langle z, v \rangle \eta(|z|) dz,
\end{aligned}
\tag{4.3}
$$

where $v = \nabla B(y)$ and $v' = \nabla u(y)$; notice that in the first line we have replaced the empirical measure ρ_n with the measure $\rho(x)dx$ (introducing some estimation error) and in the second line we have considered a Taylor expansion of B around y. On the other hand, notice that

$$
\int_{\langle z, v \rangle \geqslant 0} \langle v', z \rangle \langle z, v \rangle \eta(|z|) dz = \langle Sv, v' \rangle,
$$

where S is a rank one symmetric matrix which can be written as $S = a\zeta \otimes \zeta$ for some vector ζ and some scalar a. Now, $a\langle \zeta, v\rangle^2$ is equal to

$$\begin{aligned}\langle Sv, v\rangle &= \int_{\langle z,v\rangle \geqslant 0} \langle v, z\rangle^2 \eta(|z|)dz = |v|^2 \frac{1}{2m} \sum_{l=1}^{m} \int \langle e_l, z\rangle^2 \eta(|z|)dz \\ &= |v|^2 \frac{1}{2m} \int |z|^2 \eta(|z|)dz = \sigma_{\eta'} |v|^2.\end{aligned}$$

The above computation shows that ζ can be taken to be v, and $a = \frac{\sigma'_\eta}{|v|^2}$. Thus,

$$A_1 \approx -\rho(y) \langle \nabla u(y), \nabla B(y)\rangle . \tag{4.4}$$

Regarding A_2, we have

$$A_2 \approx \frac{1}{\varepsilon^{m+2}\sigma_{\eta'}} \int_{\mathcal{M}} \langle \nabla B(y), x - y\rangle u(x) \eta\left(\frac{|x-y|}{\varepsilon}\right) \rho(x)dx,$$

introducing an estimation error to replace the integration with respect to the empirical measure with integration with respect to the measure $\rho(x)dx$. We can further decompose the computation introducing an approximation error:

$$A_2 \approx A_{21} + A_{22} + A_{23},$$

where

$$\begin{aligned}A_{21} &:= \frac{1}{\varepsilon^{m+2}\sigma_{\eta'}} \int_{\mathcal{M}} \langle \nabla B(y), x - y\rangle \langle \nabla u(y), x - y\rangle \eta\left(\frac{|x-y|}{\varepsilon}\right) \rho(y)dx, \\ A_{22} &:= \frac{1}{\varepsilon^{m+2}\sigma_{\eta'}} \int_{\mathcal{M}} \langle \nabla B(y), x - y\rangle u(y) \eta\left(\frac{|x-y|}{\varepsilon}\right) \rho(y)dx, \\ A_{23} &:= \frac{1}{\varepsilon^{m+2}\sigma_{\eta'}} \int_{\mathcal{M}} \langle \nabla B(y), x - y\rangle \langle \nabla \rho(y), x - y\rangle u(y) \eta\left(\frac{|x-y|}{\varepsilon}\right) dx.\end{aligned}$$

By symmetry, the term A_{22} is seen to be equal to zero. On the other hand, the terms A_{21} and A_{23} are computed similarly to the second expression in (4.3) only that in this case there is no sign constraint in the integral. From a simple change of variables, we can see that for arbitrary vectors v and v', we have

$$\int_{\langle z,v\rangle \geqslant 0} \langle v', z\rangle \langle z, v\rangle \eta(|z|)dz = \int_{\langle z,v\rangle \leqslant 0} \langle v', z\rangle \langle z, v\rangle \eta(|z|)dz.$$

In particular,

$$\int \langle v', z\rangle\langle z, v\rangle\eta(|z|)dz = 2\int_{\langle z,v\rangle\geqslant 0} \langle v', z\rangle\langle z, v\rangle\eta(|z|)dz = 2\sigma'_\eta\langle v, v'\rangle,$$

and thus

$$A_{21} = 2\rho(y)\langle\nabla B(y), \nabla u(y)\rangle, \quad A_{23} = 2u(y)\langle\nabla B(y), \nabla\rho(y)\rangle.$$

In summary,

$$A_2 \approx 2\rho(y)\langle\nabla B(y), \nabla u(y)\rangle + 2u(y)\langle\nabla B(y), \nabla\rho(y)\rangle. \tag{4.5}$$

It is straightforward to see that $A_3 = O(\varepsilon)$ and so for our computation we can treat A_3 as zero:

$$A_3 \approx 0. \tag{4.6}$$

For the final term A_4, we start by introducing an estimation error to write

$$A_4 \approx \frac{1}{2\varepsilon^{m+2}\sigma_{\eta'}}\int_{\mathcal{M}} \langle D^2B(y)(y-x), x-y\rangle u(x)\eta\left(\frac{|x-y|}{\varepsilon}\right)\rho(x)dx.$$

We can further replace the term $u(x)$ with $u(y)$ (and $\rho(x)$ with $\rho(y)$) in the formula above. This replacement introduces an $O(\varepsilon)$ term that we can ignore. It follows that

$$\begin{aligned} A_4 &\approx u(y)\rho(y)\frac{1}{2\varepsilon^{m+2}\sigma_{\eta'}}\int_{\mathcal{M}} \langle D^2B(y)(x-y), x-y\rangle\eta\left(\frac{|x-y|}{\varepsilon}\right)dx \\ &= u(y)\rho(y)\frac{1}{2\sigma_{\eta'}}\int \langle D^2B(y)z, z\rangle\eta(|z|)dz \\ &= u(y)\rho(y)\Delta B(y). \end{aligned}$$

Combining the above estimate with (4.4), (4.5), and (4.6), we see that

$$\begin{aligned} \sum_{x\in\mathcal{X}} u(x)Q^{ms}(x, y) \approx &-(\rho(y)\,\langle\nabla u(y), \nabla B(y)\rangle + 2u(y)\langle\nabla B(y), \nabla\rho(y)\rangle \\ &+u(y)\rho(y)\Delta B(y)) \\ = &-\frac{1}{\rho}\mathrm{div}\left(u\rho^2\nabla B\right). \end{aligned}$$

Note that the graph dynamics takes place on the provided data points, that is, on $\mathcal{P}(\mathcal{X}) \subset \mathcal{P}(\mathcal{M})$. As $n \to \infty$, $\mathcal{P}(\mathcal{X})$ approximates $\mathcal{P}(\mathcal{M})$. This partly explains that had we carried out the argument above in the full manifold setting the resulting

dynamics would be restricted to the manifold and in particular both the divergence and the gradient above would take place on $\mathcal{M}$. That is, for data on a manifold,

$$\sum_{x\in\mathcal{X}} u(x)Q^{ms}(x,y) \approx -\frac{1}{\rho}\mathrm{div}_{\mathcal{M}}\left(u\rho^2\nabla_{\mathcal{M}}B\right).$$

Remark 4.1 Notice that with the choice $B(x) := \log(\rho(x))$, the above becomes

$$-\frac{1}{\rho}\mathrm{div}_{\mathcal{M}}\left(u\rho\nabla_{\mathcal{M}}\rho\right),$$

whereas with the choice $B(x) = -\frac{1}{\rho(x)}$, we get

$$-\frac{1}{\rho}\mathrm{div}_{\mathcal{M}}\left(u\rho\nabla_{\mathcal{M}}\log(\rho)\right).$$

The above analysis suggests that the formal continuum limit of the evolution (2.4) when $B = -\frac{1}{\rho}$ is the PDE:

$$\partial_t u_t = -\frac{1}{\rho}\mathrm{div}_{\mathcal{M}}(u_t\rho\nabla_{\mathcal{M}}\log(\rho)).$$

Notice, however, that the solution u_t of the above equation must be interpreted as a "density" with respect to the measure $\rho(x)dVol_{\mathcal{M}}$ ($\rho(x)dx$ in the flat case). Thus, in terms of "densities" with respect to $dVol_{\mathcal{M}}$, we obtain

$$\partial_t f_t = -\mathrm{div}_{\mathcal{M}}(f_t\nabla_{\mathcal{M}}\log(\rho)),$$

where $f_t := u_t\rho$. We recognize this latter equation as the PDE describing the mean shift dynamics (1.3).

4.2 Continuum Limits of Fokker–Planck Equations on Graphs

In this section, we formally derive the large sample limit of the two types of Fokker–Planck equations on $\mathcal{G}$ that we consider in this chapter, i.e., Eq. (2.1) when $Q = Q_\alpha^{rw}$ (for $\alpha \in (-\infty, 1]$) and when $Q = Q_\beta$ (for $\beta \in [0, 1]$).

We start our computations by pointing out that after appropriate scaling and under some regularity conditions on the density ρ, the diffusion operator L_α^{rw} converges toward the differential operator:

$$\mathcal{L}_\alpha v := -\frac{1}{\rho^{2(1-\alpha)}}\mathrm{div}_{\mathcal{M}}(\rho^{2(1-\alpha)}\nabla_{\mathcal{M}}v). \tag{4.7}$$

To be precise, if we set

$$C_\alpha = \frac{1}{\sigma_\eta \varepsilon^2}, \quad \sigma_\eta := \int_{\mathbb{R}^m} |z|^2 \eta(|z|) dz \Big/ \int_{\mathbb{R}^m} \eta(|z|) dz,$$

then, for all smooth $v : \mathcal{M} \to \mathbb{R}$, we have

$$-\sum_{y \in \mathcal{X}} Q_\alpha^{rw}(x, y) v(y) = C_\alpha \sum_y L_\alpha^{rw}(x, y) v(y) \to \mathcal{L}_\alpha v(x),$$

as $n \to \infty$ and $\varepsilon \to 0$ at a slow enough rate. This type of pointwise consistency result can be found in [37] and [12]. Furthermore, eigenvalues and eigenvectors of the graph Laplacians converge as $n \to \infty$ and $\varepsilon \to 0$ (with $\varepsilon \gg \left(\frac{\ln n}{n}\right)^{1/d}$) to eigenvalues and eigenfunctions of the corresponding Laplacian on $\mathcal{M}$, see [18]. If $\mathcal{M}$ is a manifold with boundary, then the continuum Laplacian is considered with no-flux boundary conditions. We note that the results from [18] are only stated for the case $\alpha = 0$, i.e., for the standard *random-walk Laplacian*, but the proof in [18] adapts to all $\alpha \in (-\infty, 1]$ assuming that the density ρ is smooth enough and is bounded away from zero and infinity.

Now, to understand the large sample limit of the dynamics (2.1) when $Q = Q_\alpha^{rw}$, we actually need to study the expression:

$$\sum_{x \in \mathcal{X}} u(x) Q_\alpha^{rw}(x, y), \quad y \in \mathcal{X}, \tag{4.8}$$

which in matrix form can be written as ${Q_\alpha^{rw}}^T u$ provided we view u as a column vector. For that purpose, we consider two smooth test functions g and h on $\mathcal{M}$. By definition of transpose,

$$\frac{1}{n} \sum_{i=1}^n h(x_i) ({Q_\alpha^{rw}}^T g)(x_i) = \frac{1}{n} \sum_{i=1}^n g(x_i) (Q_\alpha^{rw} h)(x_i). \tag{4.9}$$

At the continuum level, the definition of $\mathcal{L}_\alpha$ and integration by parts provide that

$$\begin{aligned} &\int_{\mathcal{M}} h(x) \frac{1}{\rho(x)} \mathrm{div}_{\mathcal{M}} \left(\rho^{2(1-\alpha)} \nabla_{\mathcal{M}} \left(\frac{g}{\rho^{1-2\alpha}} \right) \right) \rho(x) dVol_{\mathcal{M}}(x) \\ &\quad = -\int_{\mathcal{M}} g(x) \mathcal{L}_\alpha h(x) \rho(x) dVol_{\mathcal{M}}(x) \end{aligned}$$

By the convergence of Q_α^{rw} toward $-\mathcal{L}_\alpha$ as $n \to \infty$, we can conclude that right-hand sides converge, and thus the left-hand sides do too; notice that the $\rho(x)dx$ on both sides appear because in both sums in (4.9) the points x_i are distributed according to ρ. From this computation, we can identify the limit of (4.8) as

$$\frac{1}{\rho(y)}\mathrm{div}_{\mathcal{M}}\left(\rho^{2(1-\alpha)}\nabla_{\mathcal{M}}\left(\frac{u}{\rho^{1-2\alpha}}\right)\right).$$

In turn, we obtain the formal continuum limit of the dynamics (2.1) when $Q = Q_\alpha^{rw}$:

$$\partial_t u_t = \frac{1}{\rho}\mathrm{div}_{\mathcal{M}}\left(\rho^{2(1-\alpha)}\nabla_{\mathcal{M}}\left(\frac{u_t}{\rho^{1-2\alpha}}\right)\right),$$

where u_t represents the density with respect to $\rho(x)dx$. If we consider

$$f_t(x) := u_t(x)\rho(x), \tag{4.10}$$

that is, f_t is a probability density w.r.t. dx, we see it satisfies

$$\partial_t f_t = \mathrm{div}_{\mathcal{M}}\left(\rho^{2(1-\alpha)}\nabla_{\mathcal{M}}\left(\frac{f_t}{\rho^{2(1-\alpha)}}\right)\right) = \Delta_{\mathcal{M}} f_t - 2(1-\alpha)\mathrm{div}_{\mathcal{M}}(f_t\nabla_{\mathcal{M}}\log(\rho)), \tag{4.11}$$

where the last equality follows from an application of the product rule to the term $\nabla_{\mathcal{M}}\left(\frac{f}{\rho^{2(1-\alpha)}}\right)$. Notice that after considering a time change $t \leftarrow \frac{t}{3-2\alpha}$, we can rewrite Eq. (4.11) as

$$\partial_t f_t = (1-\beta_\alpha)\Delta_{\mathcal{M}} f_t - \beta_\alpha \mathrm{div}_{\mathcal{M}}(f_t\nabla_{\mathcal{M}}\log(\rho)), \tag{4.12}$$

where $\beta_\alpha = (2-2\alpha)/(3-2\alpha) \in [0,1]$.

Using the above analysis and Remark 4.1, we can also conclude that the (formal) large sample limit of equation (2.1) with $Q = Q_\beta$ and potential $B = -\frac{1}{\rho}$ is given by

$$\partial_t f_t = (1-\beta)\Delta_{\mathcal{M}} f_t - \beta \mathrm{div}_{\mathcal{M}}(f_t\nabla_{\mathcal{M}}\log(\rho)), \tag{4.13}$$

that is, the same continuum limit as for the Fokker–Planck equations constructed using the rate matrix Q_α for α such that $\beta = \beta_\alpha$.

Remark 4.2 Notice that when $\beta = 1$, Eq. (4.13) reduces to the heat equation on $\mathcal{M}$ where no role is played by ρ. In this case, clustering is determined completely by the geometric structure of $\mathcal{M}$. On the other hand, when $\beta = 0$, Eq. (4.13) reduces to mean shift dynamics on $\mathcal{M}$ as discussed in Sect. 1.1.

Remark 4.3 Several works in the literature have established precise connections between operators such as graph Laplacians built from random data and analogous differential operators defined at the continuum level on smooth compact manifolds without boundary. For pointwise consistency results, we refer the reader to [37, 21, 20, 4, 40, 19]. For *spectral convergence* results, we refer the reader to [46] where the regime $n \to \infty$ and ε constant has been studied. Works that have studied regimes

where ε is allowed to decay to zero (where one recovers differential operators and not integral operators) include [36, 5, 16, 28, 6, 13, 48]. Recent work [10] considers the spectral convergence of $\mathcal{L}_\alpha$ with self-tuned bandwidths and includes the $\alpha < 0$ range. The work [7] provides regularity estimates of graph Laplacian eigenvectors.

The case of manifolds *with* boundary has been studied in papers like [43, 39, 28, 18]. It is important to highlight that the specific computations presented in our Sect. 4.1 would have to be modified to take into account the effect of the boundary, in particular on the kernel density estimate. However, we remark that the tools and analysis from the papers mentioned above can be used to generalize these computations.

Remark 4.4 A connection between Fokker–Planck equations at the continuum level and the graph dynamics induced by Q_α^{rw} when $\alpha = 1/2$ was explicitly mentioned in [31]. To establish an explicit link between mean shift and spectral clustering, however, we need to consider the range $(-\infty, 1]$ for α. In the diffusion maps literature, the interval $[0, 1]$ is considered as natural range for α, but the analysis presented in this section explains why $(-\infty, 1]$ is in fact a more natural choice.

Remark 4.5 Besides the Fokker–Planck interpolations considered in Sect. 3.1, another family of data embeddings that are used to interpolate geometry-based and density-driven clustering algorithms is based on the path-based metrics studied in [26, 27].

4.3 The Witten Laplacian and Some Implications for Data Clustering

In the previous section, we presented a (formal) connection between Fokker–Planck equations on proximity graphs and Fokker–Planck equations on manifolds. In this section, we use this connection to illustrate why the Fokker–Planck interpolation is expected to produce better clusters in settings like the blue sky problem discussed in our numerical experiments in Sect. 5.2.6 where both pure mean shift and pure spectral clustering perform poorly. For simplicity, we only consider the Euclidean setting.

We start by noticing that Eq. (4.13) can be rewritten as

$$\partial_t \tilde{f}_t = -\Delta_\varrho \tilde{f}_t, \tag{4.14}$$

after considering the transformation:

$$f = \exp\left(-\frac{1-\beta}{2\beta}\varrho\right)\tilde{f}, \quad \varrho := -\log(\rho).$$

In the above, the operator Δ_ϱ is the *Witten Laplacian* (see [47] and [30]) associated with the potential $\frac{1-\beta}{2}\varrho$ which is defined as

$$\Delta_\varrho v := -\beta^2 \Delta v + \frac{(1-\beta)^2}{4}|\nabla \varrho|^2 v - \frac{\beta(1-\beta)}{2}(\Delta \varrho)v. \tag{4.15}$$

From the above, we conclude that the Fokker–Planck dynamics (4.13) can be analyzed by studying the dynamics (4.14). In turn, some special properties of the Witten Laplacian Δ_ϱ that we review next allow us to use tools from spectral theory to study equation (4.14) and in turn also the type of data embedding induced by our Fokker–Planck equations on graphs.

To begin, notice that

$$\Delta_\varrho = \left(-\beta \mathrm{div} + \frac{(1-\beta)}{2}\nabla \varrho\right)\left(\beta \nabla + \frac{(1-\beta)}{2}\nabla \varrho\right). \tag{4.16}$$

From the above, we see that $\langle \Delta_\varrho f, g\rangle_{L^2(\mathcal{M})}$ can be written as

$$\langle \Delta_\varrho g, h\rangle_{L^2(\mathcal{M})} = \int_{\mathcal{M}} \left\langle \beta \nabla g + g\frac{(1-\beta)}{2}\nabla \varrho, \beta \nabla h + h\frac{(1-\beta)}{2}\nabla \varrho \right\rangle dx,$$

from where we conclude that $\langle \Delta_\varrho f, g\rangle_{L^2(\mathcal{M})}$ is a quadratic form with associated Dirichlet energy:

$$D(f) := \int_{\mathcal{M}} \left| \nabla f + f\frac{(1-\beta)}{2}\nabla \varrho \right|^2 dx. \tag{4.17}$$

When $\mathcal{M}$ is compact, it is straightforward to show that there exists an orthonormal basis $\{\varrho_k\}_{k\in\mathbb{N}}$ for $L^2(\mathcal{M})$ consisting of eigenfunctions of Δ_ϱ with corresponding eigenvalues $0 = \lambda_1 < \lambda_2 \leqslant \lambda_2 \leqslant \dots$ that can be characterized using the Courant–Fisher minmax principle. Using the spectral theorem, we can then represent a solution to (4.14) as

$$\tilde{f}_t = \sum_{k=1}^{\infty} e^{-t\lambda_k}\langle \tilde{f}_0, \varrho_k\rangle_{L^2(\mathcal{M})}\varrho_k$$

and conclude that the dynamics (4.14) are strongly influenced by the eigenfunctions with smallest eigenvalues.

We now explain the implication of the above discussion on data clustering. Suppose that we consider a data distribution in $\mathbb{R}^2$ as the one considered in Sect. 5.2.6 modeling the blue sky problem, so that in particular it has product structure, i.e., $\rho(x, y) = \rho_1(x)\rho_2(y)$. In this case, we can use the additive structure of the potential $\varrho = -\log(\rho(x, y)) = -\log(\rho_1(x)) - \log(\rho_2(y)) =: \varrho_1(x) + \varrho_2(y)$ to conclude that the set of eigenvalues of Δ_ϱ and a corresponding orthonormal basis of eigenfunctions can be obtained from

$$\lambda_{1,i} + \lambda_{2,j}, \quad \varrho_{1,i}(x)\varrho_{2,j}(y),$$

where $(\lambda_{1,i}, \varrho_{1,i})$ are the eigenpairs for the 1D Witten Laplacian Δ_{ϱ_1} and $(\lambda_{2,j}, \varrho_{2,j})$ are the eigenpairs for Δ_{ϱ_2}. In particular, the first non-trivial eigenvalue of Δ_ϱ and its corresponding eigenfunction (which will be the effective discriminators of the two desired clusters if λ_3 is considerably larger than λ_2) are either $\lambda_{1,2}$ and $\varrho_{1,2}(x)\varrho_{2,1}(y)$ or $\lambda_{2,2}$ and $\varrho_{1,1}(x)\varrho_{2,2}(y)$. This discussion captures the competition between a horizontal and a vertical partitioning of the data in the context of the blue sky problem from Sect. 5.2.6. While we are not able to retrieve the desired horizontal partitioning by setting $\beta = 0$ or $\beta = 1$, we can identify the correct clusters by setting β strictly between zero and one (closer to one than to zero). We notice that the results from [30] can be used to obtain precise quantitative information on the small eigenvalues of the 1D Witten Laplacians Δ_{ϱ_1} and Δ_{ϱ_2} when β is close to one (i.e., the diffusion term is small), which we can use to determine whether $\lambda_{2,2} < \lambda_{1,2}$ or vice versa.

5 Numerical Examples

We now turn to the details of our numerical method and examples illustrating its properties. We begin, in Sect. 5.1, by describing the details of our numerical approach. We provide Algorithm 5.1 for its practical implementation.

In Sect. 5.2, we consider several numerical examples, beginning with examples in one spatial dimension. In Fig. 4, we illustrate how the graph dynamics for the transition rate matrices Q_β and Q_α^{rw} can be visualized as the evolution of a continuum density, and in Fig. 5, we illustrate the good agreement between the graph dynamics and the dynamics of the corresponding continuum Fokker–Planck equation. In Fig. 6, we show how the clustering performance of our method depends on the balance between drift and diffusion (β), the time of clustering (t), and the number of clusters (k); we also illustrate the benefits and limitations of using the energy of the k-means clustering to identify the number of clusters. In Fig. 7, we consider the role of the kernel density estimate in clustering dynamics, showing how adding diffusion to mean shift dynamics can help the dynamics overcome spurious local minimizers in the kernel density estimate, leading to better clustering performance. In Fig. 8, we illustrate the interplay between the underlying data distribution and the balance between drift and diffusion (β) .

Next, we consider several examples in two dimensions. In Figs. 10 and 11, we consider a model of the *blue sky problem*, in which data points are distributed over two elongated clusters that are separated by a narrow low-density region. We illustrate how diffusion dominant dynamics prefer to cluster based on the geometry of the data, leading to poor performance. Similarly, pure mean shift dynamics can exhibit poor clustering due to local maxima in the kernel density estimate. By interpolating between the two extremes, we observe robust clustering performance, for a wide range of graph connectivity (ε). Finally, in Figs. 12 and 13, we consider an

example in which three blobs are connected by two bridges: one wide, low-density bridge and another narrow, high-density bridge. This example is constructed so that there is no correct clustering into two clusters. Instead, a geometry-based clustering method would prefer to cut the thin bridge, and a density-based clustering method would prefer to cut the wide bridge. We show how varying the balance between drift and diffusion in our method (β) allows our method to cut either bridge.

5.1 Numerical Method

For our numerical experiments, we consider a domain $\Omega \subseteq \mathbb{R}^d$ and a density $\rho : \Omega \to [0, +\infty)$ normalized so that $\int_\Omega \rho = 1$. All PDEs on Ω will be considered with no-flux boundary conditions, as the solutions of the graph-based equations converge to the solutions of PDE with no-flux boundary conditions (observed in [12] and rigorously proved in [18] for Laplacians).

We draw n samples $\{x_i\}_{i=1}^n$ from ρ on Ω. These samples are the nodes of our weighted graph, and for all simulations, the weights on the graph are given by a Gaussian weight function

$$w(x_i, x_j) = \varphi_\varepsilon(|x_i - y_j|), \quad \varphi_\varepsilon(a) = \frac{e^{-a^2/2\varepsilon^2}}{(2\pi\varepsilon^2)^{d/2}}, \quad a \in \mathbb{R}. \tag{5.1}$$

In our one-dimensional simulations, we take the graph bandwidth parameter ε to be

$$\varepsilon = \sqrt{2} \max_i \min_{j:j\neq i} |x_i - x_j|; \tag{5.2}$$

that is, ε equals the maximum distance to the closest node. We note that even in higher dimensions, the ε above scales as $(\ln n/n)^{1/d}$ with the number of nodes n. This has been identified as the threshold, in terms of n, at which the graph Laplacian is spectrally consistent with the manifold Laplacian [18]. In Fig. 11, we illustrate how the choice of ε impacts dynamics and, ultimately, clustering performance.

With this graphical structure, we now recall the weighted diffusion transition rate matrix Q_α^{rw}, for $\alpha \in (-\infty, 1]$, as in Eq. (1.12), with the constant $C_\alpha = ((3 - 2\alpha)\varepsilon^2)^{-1}$,

$$Q_\alpha^{rw}(x, y) := \frac{1}{(3-2\alpha)\varepsilon^2} \begin{cases} \frac{w_\alpha(x,y)}{\sum_{z\neq x} w_\alpha(x,z)} & \text{if } x \neq y, \\ -1 & \text{if } x = y, \end{cases} \tag{5.3}$$

$$w_\alpha(x, y) := \frac{w(x, y)}{d(x)^\alpha d(y)^\alpha}, \qquad d(x_i) = \sum_{x_j \neq x_i} w(x_i, x_j). \tag{5.4}$$

Similarly, we recall the transition rate matrix Q_β, for $\beta \in [0, 1]$, as in Eq. (3.1), with the constant $C_{ms} = (\varepsilon^2 n)^{-1}$,

$$Q_\beta := \beta Q^{ms} + (1-\beta) Q_1^{rw}, \tag{5.5}$$

$$Q^{ms}(x, y) := \frac{1}{\varepsilon^2 n} \begin{cases} \left(-\frac{1}{\hat{\rho}_\delta(y)} + \frac{1}{\hat{\rho}_\delta(x)}\right)_+ w(x, y), & \text{for } x \neq y, \\ -\sum_{z \neq x} \left(-\frac{1}{\hat{\rho}_\delta(z)} + \frac{1}{\hat{\rho}_\delta(x)}\right)_+ w(x, z), & \text{for } x = y, \end{cases} \tag{5.6}$$

$$\hat{\rho}_\delta(x) := \frac{1}{n} \sum_{y \in \mathcal{X}} \varphi_\delta(x - y). \tag{5.7}$$

Unless otherwise specified, we take the bandwidth δ in our kernel density estimate for our one-dimensional examples to be

$$\delta = \sqrt{2} \left(\frac{|\Omega|}{n}\right)^{0.5}. \tag{5.8}$$

With these transition rate matrices in hand, we may now consider solutions u_t of (2.1) when $Q = Q_\alpha^{rw}$ or when $Q = Q_\beta$. We solve the ordinary differential equations describing the graph dynamics by directly computing the matrix exponential e^{tQ} in each case; see Definition 2.1. Following the discussion in Sect. 4.2, we know that for each of these dynamics, as $n \to +\infty$ and $\varepsilon, \delta \to 0$ (at an n dependent rate that is not too fast), the measures $\sum_{j=1}^n u_t(x_j)\delta_{x_j}$ are expected to converge to solutions f_t of the following Fokker–Planck equation:

$$\partial_t f_t = (1-\beta)\Delta f_t - \beta \mathrm{div}(f_t \nabla \log(\rho)), \tag{5.9}$$

where for the Q_α^{rw} dynamics, we take

$$\beta = \beta_\alpha = (2 - 2\alpha)/(3 - 2\alpha) \tag{5.10}$$

The steady state of the equation is the corresponding Maxwellian distribution

$$c_{\rho,\beta}\ \rho^{\beta/(1-\beta)}(x), \tag{5.11}$$

where $c_{\rho,\beta} > 0$ is a normalizing constant chosen so that the distribution integrates to one over Ω. Note that, if $d(x_i)$ represents the degrees of the graph vertices, as in Eq. (1.8), then the function $u_t(x_i)d(x_i)$ likewise converges to $f_t(x)$ as the number of nodes in our sample $n \to +\infty$. Consequently, when comparing our graph dynamics to the PDE dynamics, we will often plot $u_t(x_i)d(x_i)$ and $f_t(x)$.

Finally, we use the embedding maps $\hat{\Psi}_\alpha$ and $\hat{\Psi}^\beta$ from Sects. 2.1 and 3.1 to cluster the nodes. In particular, we apply k-means to the vectors $\{\hat{\Psi}_\alpha(x_i)\}_{i=1}^n$ and $\{\hat{\Psi}^\beta(x_i)\}_{i=1}^n$, obtaining in this way a series of maps from nodes $\{x_i\}_{i=1}^n$ to

cluster centers $\{l_m\}_{m=1}^k$. Nodes mapped to the same cluster center are identified as belonging to the same cluster. While we will not discuss at any depth the methods to select the best number of clusters, we note that a number of methods to do so (in particular the elbow method and the gap statistics [38]) rely on the value of the k-means energy,

$$E_k = \frac{1}{n} \sum_{i=1}^{n} \min_{m=1,\dots k} |\Psi(x_i) - l_j|^2, \tag{5.12}$$

for each relevant Ψ. Note that E_k always decreases with k. While a large decrease in the energy as k increases is indicative of the improved approximation of data by cluster centers, the size of the jumps is truly telling only if we compare it with the relevant model for the data considered, see [38] and discussion in Sect. 5.2.3. For ease of visualization, in our numerical examples, we will plot the *normalized* k-means energy, which is rescaled so that energy of a single cluster equals one,

$$E_k^{\text{norm}} = E_k / E_1. \tag{5.13}$$

All of our simulations are conducted in Python, using the Numpy, SciPy, Sci kit-learn, and MatPlotLib libraries [42, 24, 23, 33]. In particular, we use the Sci kit-learn implementation of k-means to cluster the embedding maps.

Algorithm 1 Dynamic clustering algorithm for Q_β or Q_α^{rw}

Input: $\{x_i\}_{i=1}^n$, ε, δ, t, k
$Q = Q_\beta$ or $Q = Q_\alpha^{rw}$
$\hat{\Psi}_Q(x_i) = (e^{tQ})_{(i,j=1,\dots n)}$ for $i = 1, \dots, n$
$l_m = \text{Kmeans.fit}(\hat{\Psi}_Q(x_1), \dots, \hat{\Psi}_Q(x_n))$ with $\text{n}_{\text{clusters}} = k$

5.2 *Simulations*

We now turn to simulations of the graph dynamics, PDE dynamics, and clustering.

5.2.1 Graph Dynamics as Density Dynamics

In Fig. 4, we illustrate how the dynamics on a graph can be visualized as the evolution of a density on the underlying domain $\Omega = [-1.5, 1.5]$. The right column of Fig. 4 illustrates two choices of data density (blue line),

$$\rho_{\text{two bump}}(x) = 4c\varphi_{0.5}(x + 0.5) + c\varphi_{0.25}(x - 1.25)$$

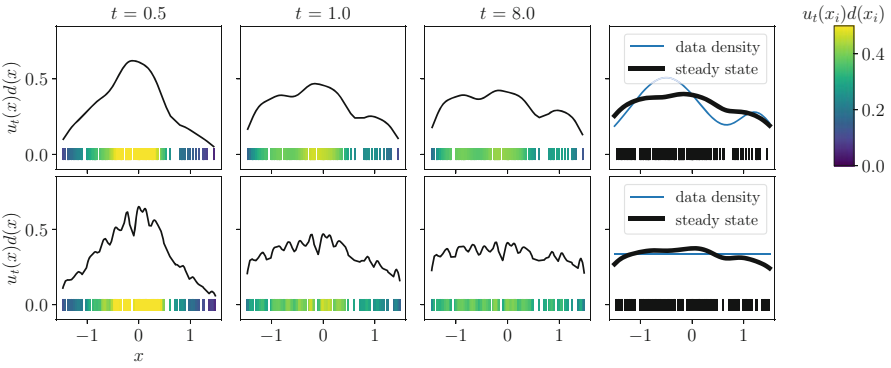


Fig. 4 Illustration of the graph dynamics u_t for Q_β, $\beta = 0.25$, from initial condition δ_{x_i}, $x_i = -0.1$, for two choices of data density: $\rho_{\text{two bump}}$ (top) and ρ_{uniform} (bottom). The first three columns show the evolution of $u_t(x)d(x)$ at times $t = 0.1, 0.5$, and 8.0, with the color of the markers representing the value of $u_t(x_i)d(x_i)$ at each node. The last column depicts the data density (blue line) from which the nodes of the graph $\{x_i\}_{i=1}^n$ (black markers) are sampled, as well as the steady state of the dynamics (thick black line)

$$\text{and} \quad \rho_{\text{uniform}}(x) = \frac{1}{3}, \quad x \in \mathbb{R}. \tag{5.14}$$

The constant $c > 0$ is chosen so that the integral of both densities over the domain equals one. We sample the nodes of the graph $\{x_i\}_{i=1}^n$ (black markers) from each density, with $n = 147$ nodes sampled for $\rho_{\text{two bump}}$ and $n = 140$ nodes sampled for ρ_{uniform}. The first three columns show the evolution of the graph dynamics $u_t(x)d(x)$ from Eq. (2.1) for $Q = Q_\beta$ with $\beta = 0.25$ and initial condition δ_{x_i}, $x_i = -0.1$, where the top row corresponds to the graph arising from $\rho_{\text{two bump}}$ and the bottom row corresponds to the graph arising from ρ_{uniform}. The color of the markers represents the value of $u_t(x_i)d(x_i)$ at each node. We observe in both rows that $u_t(x)d(x)$ approaches the steady state of the corresponding continuum PDE (5.11), depicted in a thick black line in the fourth column.

The fact that $d(x)u_t(x)$ appears more jagged in the bottom row compared to the top row is due to the smaller value of ε in the graph weight matrix: see Eqs. (5.1–5.2). Since our sample of the data density in the top row has an isolated node at $x_i = 1.44$, this leads to a significantly larger value of ε in the simulations on the top row ($\varepsilon = 0.13$), compared to the bottom row ($\varepsilon = 0.03$).

5.2.2 Comparison of Graph Dynamics and PDE Dynamics

In Fig. 5, we compare the graph dynamics to the corresponding Fokker–Planck equation (5.9). We consider the data density given by $\rho_{\text{two bump}}$ and initial condition δ_{x_i}, for $x_i = -0.1$. The graphs are built from $n = 625$ samples of the data density, and solutions are plotted at times $t = 0.5, 1.0, 8.0$.

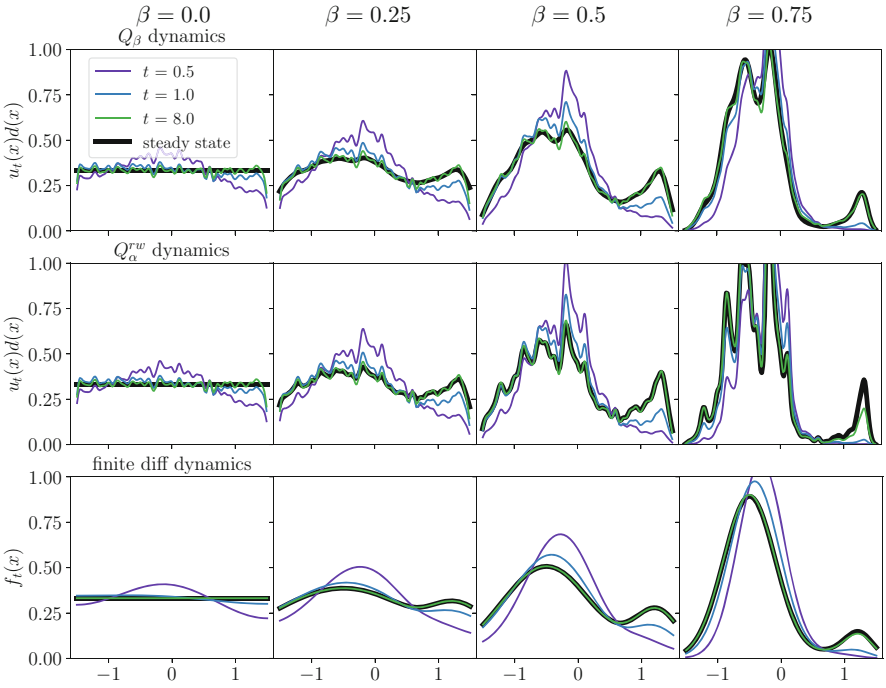


Fig. 5 Comparison of the graph dynamics for Q_β (top) and Q_α (middle) with the PDE dynamics (bottom). The data density is $\rho_{\text{two bump}}$, and the initial data is δ_{x_i} for $x_i = -0.1$. The graphs are built from $n = 625$ samples of the data density. The steady states are obtained from equation (5.15) for the graph dynamics and equation (5.11) for the finite difference dynamics

The top row illustrates the graph dynamics $u_t(x)d(x)$ arising from the transition rate matrix Q_β, for $\beta = 0, 0.25, 0.5, 0.75$. The middle row illustrates $u_t(x)d(x)$ arising from Q_α^{rw} for $\alpha = 1.0, 0.83, 0.5, -0.5$. (The values of α are chosen to give the same balance between drift and diffusion as in the top row; see equation (5.10).) The last row shows a finite difference approximation of the Fokker–Planck equation (5.9). We compute solutions of the PDEs using a semi-discrete, upwinding finite difference scheme on a one-dimensional grid, with 200 spatial gridpoints and continuous time. This reduces the PDEs to a system of ODEs, which we then solve using the SciPy odeint method.

The steady states we plot for the graph dynamics are given by the following equation:

$$c_{n,\delta,\beta}(\hat{\rho}_\gamma(x))^{\beta/(1-\beta)}, \quad \hat{\rho}_\gamma(x) = \frac{1}{n}\sum_{y\in\mathcal{X}} \psi_\gamma(x-y), \quad \psi_\gamma(x) = \frac{1}{(2\pi)^{1/2}\gamma} e^{-|x|^2/(2\gamma^2)}, \tag{5.15}$$

where $c_{n,\delta,\beta}$ is a normalizing constant chosen, so the steady state integrates to one over Ω. For the Q_β dynamics, we choose the standard deviation $\gamma = \delta$, and for the Q_α^{rw} dynamics, we choose $\gamma = \varepsilon$. Recall that $\hat{\rho}_\delta$ is the kernel density estimator used in the construction of the transition matrix Q^β; see Eq. (2.7). The steady states for the PDE dynamics are given by Eq. (5.11).

Interestingly, even though there is no explicit kernel density estimate of the data in the construction of the transition rate matrix Q_α^{rw}, the above simulations demonstrate better agreement of these dynamics as $t \to +\infty$ with the steady state arising from a kernel density estimate (5.15) than with the steady state arising directly from the data density (5.11). This can be seen by observing the good agreement at time $t = 8.0$ with the solid black line shown in the middle row, rather than the solid black line shown in the bottom row. This suggests that the Q_α^{rw} operator effectively takes a KDE of the data density with bandwidth $\varepsilon > 0$, corresponding to the scaling of the weight matrix on the graph.

5.2.3 Clustering Dynamics

In Fig. 6, we illustrate how the graph dynamics u_t of the transition rate matrix Q_β can be used for clustering. The underlying data density is $\rho_{\text{two bump}}$, from which we choose $n = 204$ samples. We consider $\beta = 0.25, 0.9$, and 1.0, corresponding to the three columns of the figure. The top portion of the figure shows the results of the k-means clustering algorithm for $k = 2, 3, 4$. Each plot depicts the data samples at times $t = 10^{-1}, 1, 10$, coloring the samples according to which cluster they belong. The top right panel on the figure shows the data distribution and the kernel density estimate of the data distribution, which is used to construct the transition rate matrix Q_β. The bottom of the figure shows the value of the k-means energy E_k (5.13) for each clustering normalized so that $k = 1$ clustering (all nodes in a single cluster) has energy $E_1 = 1$.

For all β and k, there is poor clustering behavior early in time, $t = 0.1$, suggesting that the Fokker–Planck dynamics have not had time to effectively mix within clusters. This can be seen by comparing the colors of the nodes to the data distribution displayed on the right: a correct clustering should identify one cluster for the large bump and another cluster for the small bump. This can also be seen by considering the k-means energy, which is largest at $t = 0.1$, and shows little variation for different choices of k.

On the other hand, we observe the best clustering performance for $\beta = 0.9$ and time $t = 10$. Examining the colors of the nodes for $k = 2$ reveals that the correct clusters are found. Furthermore, this clustering remains fairly stable as k is increased. This can also be seen in the k-means energy, which shows a substantial decrease from $k = 1$ to $k = 2$, but remains stable for $k = 3, 4$, suggesting that two clusters are the correct number of clusters.

While $\beta = 0.25$ and $\beta = 1$ do not offer good clustering performance, they do shed light on key properties of our method, once time is sufficiently large to have allowed the dynamics to effectively mix, $t = 10$. For example, when $\beta =$

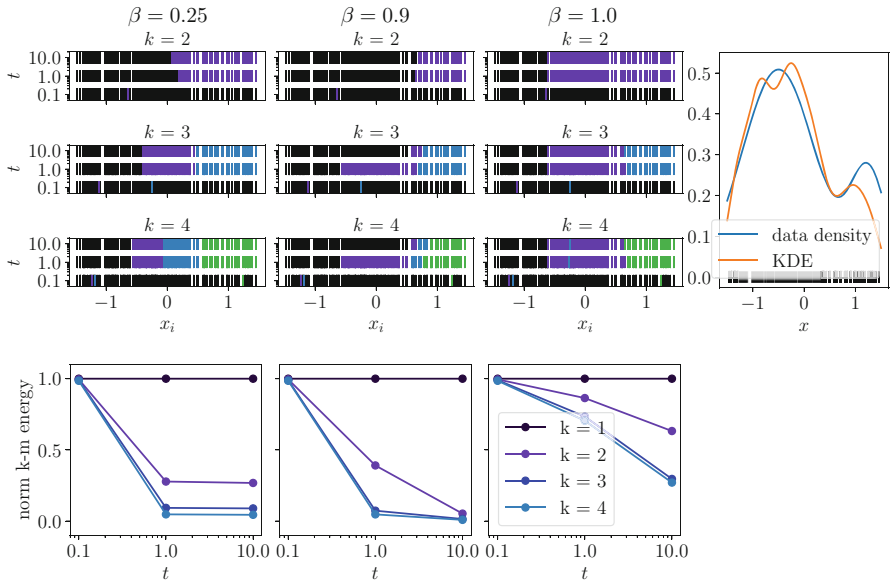


Fig. 6 Clustering performance of the graph dynamics for the transition rate matrix Q_β. The top portion of the figure shows the results of the k-means clustering algorithm for $k = 2, 3, 4$ (rows) and $\beta = 0.25, 0.9, 1.0$ (columns). The color of a node indicates the cluster to which it belongs. On the top right, we show the data density from which $n = 204$ nodes are sampled and the KDE of the data density used to construct Q_β. The bottom of the figure shows the value of the normalized k-means energy

0.25, diffusion dominates the dynamics, so that the density of the data distribution does not play a strong role in clustering. In fact, we see that the clusters are almost entirely driven by the geometry of the data distribution, which is fairly uniform on the domain: when $k = 2$, the clusters are essentially even halves of the domain; when $k = 3$, they are even thirds; and when $k = 4$, they are even quarters. The lack of awareness of density when $\beta = 0.25$ inhibits correct cluster identification.

We observe the opposite problem when $\beta = 1$. In this case, the dynamics are driven entirely by density, with no diffusion. However, the density driving the dynamics is not the exact data density, but the kernel density estimate. Due to noise in the KDE, an artificial local minimum appears near $x = -0.75$, causing $k = 2$ to cluster the nodes to the left and right of this local minimum and causing $k = 3$ to cluster the nodes into three even groups, separated by the two local minima of the KDE. Unlike in the case $\beta = 0.9$, when $\beta = 1.0$, there is no diffusion to help the dynamics overcome spurious local minima in the KDE, leading to inferior clustering performance.

We close by considering the role of the k-means energies in identifying the correct number of clusters. First, consider the case of a uniform data distribution. In this case, the k-means energy for $k = 2$ would be $\frac{1}{4}$ and for $k = 3$ would be $\frac{1}{9}$.

Consequently, while the correct number of clusters for the uniform data distribution is one, the k-means energy still drops significantly as k is increased. For this reason, we caution that looking for the largest drop of the energy alone is not a good criterion for determining the correct number of clusters. Determining the correct number of clusters remains an active area of research, including, for example, the study of gap statistics [38], in which the energy is compared to the energy one would have if the data were uniform.

5.2.4 Effect of the Kernel Density Estimate on Clustering

Figure 7 illustrates the effect that the bandwidth δ of the kernel density estimate of the data distribution has on clustering, see Eq. (2.7). The data distribution is given by a piecewise constant function, shown in the rightmost column. The number of samples chosen is $n = 680$, and the clustering is performed at time $t = 30$. The graph connectivity parameter ε is chosen as in Eq. (5.2), equaling 0.015. The top two rows show clustering performance for Q_β for $\beta = 0.25, 0.9, 1$, and the bottom

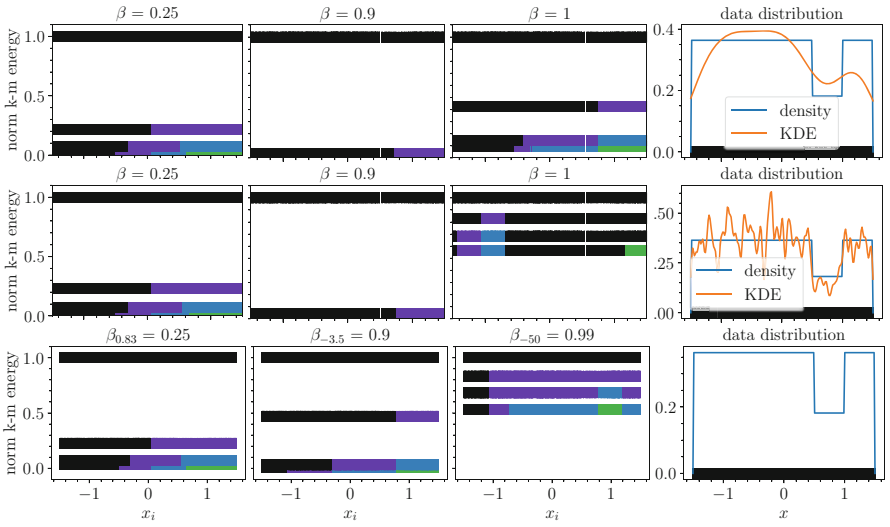


Fig. 7 Effect of the bandwidth of the kernel density estimator on clustering, for $n = 676$ samples clustered at time $t = 30$. First two rows show the clustering with Q_β, (5.5) for KDE bandwidths $\delta = 0.2$ and $\delta = 0.015$, while the third shows the dynamics of Q_α^{rw}, (5.3). The first three columns show clustering performance for different balances of drift and diffusion, and the fourth column shows the data distribution and kernel density estimate. Note that, since no explicit kernel density estimate is used in the construction of Q_α^{rw}, none is shown in the third row. The colors of the samples indicate the clusters to which they belong, and the height of the samples in each frame indicates the value of the normalized k-means energy (5.12). The top row of markers in each frame corresponds to a single cluster ($k = 1$), the next one represents two clusters ($k = 2$), then three and four clusters

row shows clustering performance for Q_α^{rw} for $\alpha = 0.83, -3.5, -50$, where the values of α are chosen to give a comparable balance between drift and diffusion at the level of the continuum PDE; see Eqs. (5.9–5.10).

The first three columns show the clustering results for $\beta = 0.25, 0.90, 1.00$. The color of a marker indicates the cluster to which it belongs, and the height of the marker in the frame represents the value of the normalized k-means energy (5.13). Since the normalized k-means energy is decreasing in k, the top row of markers in each frame corresponds to a single cluster ($k = 1$), the next one represents two clusters ($k = 2$), then three and four clusters.

In the top row, the bandwidth of the kernel density estimate used to construct Q_β is $\delta = 0.20$, and in the middle row, $\delta = 0.015$. The effect of the bandwidth on the kernel density estimate can be seen in the rightmost column: the larger value of δ in the top row leads to a more accurate estimator of the data density than the smaller value of δ in the middle row. As no explicit kernel density estimate is used to construct the transition rate matrix Q_α^{rw}, no estimator is shown in the rightmost column of the bottom row. However, our previous numerical simulations in Fig. 5 suggest that the dynamics of Q_α^{rw} most closely match the continuum Fokker–Planck equation with a steady state induced by a kernel density estimate with bandwidth $\delta = \varepsilon$ (5.15). This is the motivation behind our choice of $\delta = 0.015 = \varepsilon$ in the second row, since it provides the closest comparison between the clustering dynamics of Q^β and Q_α^{rw}. Finally, we note that the choice of δ we suggest in Eq. (5.8) would lead to the choice $\delta = 0.07$, which is between the values considered in the top and middle rows of the figure, and leads to very similar performance for $\beta < 1$.

In the top row, when the bandwidth in the KDE is large, we observe good clustering performance for $\beta = 0.9$ and 1.0 and $k = 2$. On the other hand, $\beta = 0.25$ performs poorly, since the large amount of diffusion causes the dynamics to ignore the changes in relative density and cluster based on the fairly uniform geometry of the sampling. In the middle row, when the bandwidth in the KDE is small, we still observe good performance for $\beta = 0.9$, though $\beta = 1.0$ clusters poorly: without diffusion, the dynamics cluster based on spurious local maxima. As before, $\beta = 0.25$ identifies incorrect clusters, since it lacks information about density. Finally, as we expected, the clustering performance in the bottom row is similar to the middle row, due to the fact that the bandwidth in the middle row was chosen to match the bandwidth of the implicit kernel density estimate which appears to drive the dynamics of Q_α^{rw}. Note that, for the bottom row, the only way to increase the bandwidth of the implicit kernel density estimate would be to increase the graph connectivity parameter ε, which, for compactly supported graph weights, would lead to a more densely connected graph and thus higher computational cost.

5.2.5 Effect of Data Distribution on Clustering

Figure 8 illustrates the effect that different choices in data distribution have on the clustering method based on Q_β, for $n = 160$ nodes and at time $t = 30$. Each row considers a different data distribution: ρ_{twobump} (see Eq. (5.14)) and

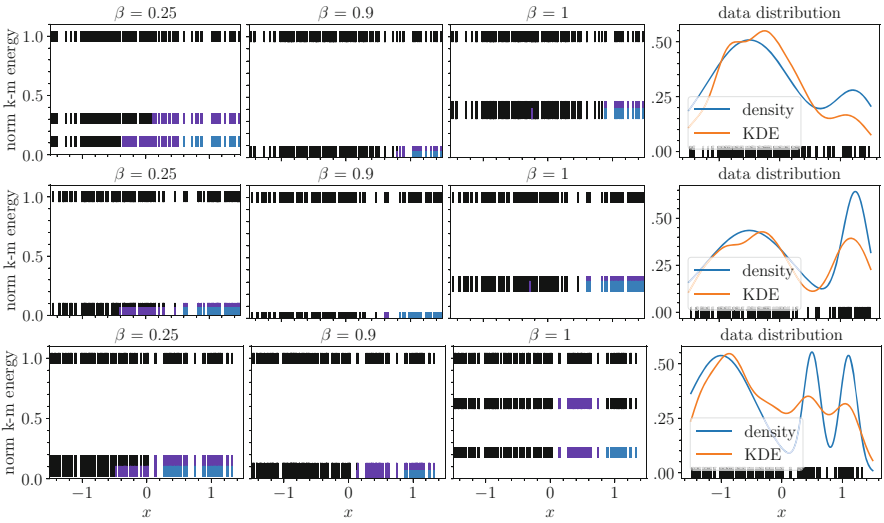


Fig. 8 Illustration of the effect that different choices of data distribution have on the clustering method based on Q_β for $n = 160$ samples at time $t = 30$. The first three columns illustrate different choices of β, where the color of a marker indicates the cluster it belongs to for $k = 1, 2, 3$, and the height of the marker in the frame represents the value of the normalized k-means energy

$$\rho_{\text{deep valley}}(x) = 7c_0\varphi_{0.5}(x + 0.5) + 3c_0\varphi_{0.15}(x - 1.25),$$

$$\rho_{\text{three bump}}(x) = c_1\varphi_{0.1}(x - 0.5) + c_1\varphi_{0.1}(x - 1.1) + 4c_1\varphi_{0.4}(x + 1),$$

where $c_0, c_1 > 0$ are normalizing constants chosen, so the densities integrate to one on $\Omega = [-1.5, 1.5]$. The right column shows the data density, the sample of $n = 160$ nodes, and the kernel density estimate used to construct the transition rate matrix Q_β. The first three columns show the clustering results for $\beta = 0.25, 0.90, 1.00$. The color of a marker indicates the cluster to which it belongs for $k = 1, 2, 3$, and the height of the marker in the frame represents the value of the normalized k-means energy (5.13).

In the top row, for $\rho_{\text{two bump}}$, we observe good clustering performance for all $\beta \geqslant 0.9$ and poor performance for $\beta = 0.25$: due to the good behavior of the KDE for this data distribution, problems do not arise as $\beta \to 1$, and as usual, $\beta = 0.25$ suffers due to the dominance of diffusion. In the middle row, for $\rho_{\text{deep valley}}$, we again observe good performance for all $\beta \geqslant 0.9$, and we even observe good performance for $\beta = 0.25$ when $k = 2$. This is due to the sparse sampling at the deep valley, which leads to a change in the geometry of the nodes: a gap that even diffusion dominant dynamics can detect. Finally, in the bottom row, for $\rho_{\text{three mountains}}$, we observe good performance for $\beta \geqslant 0.9$. Again, $\beta = 0.25$ is

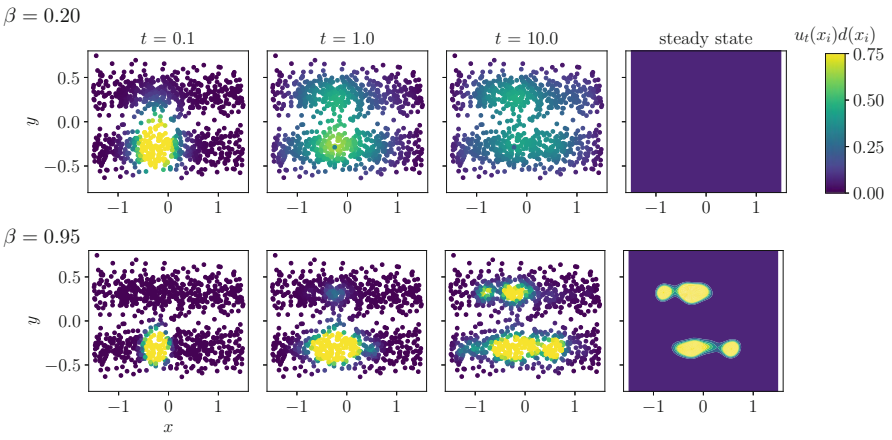


Fig. 9 Graph dynamics of Q_β on $\rho_{\text{blue sky}}$ for $n = 965$ samples for $\beta = 0.20$ (top) and $\beta = 0.95$ (bottom). The initial condition is δ_{x_i,y_i} for $(x_i, y_i) = (-0.26, -0.29)$. In the first three columns, the dots represent the locations of the samples, and the colors of the markers represent the value of $u_t(x_i)d(x_i)$. In the fourth column, we plot the steady state of the corresponding continuum PDE (5.15)

able to capture some correct information when $k = 2$, due to the sparsity of the data near the left valley, but it fails at the most relevant $k = 3$.

5.2.6 Blue Sky Problem

In Fig. 9, we consider the graph dynamics of Q_β on a two-dimensional data distribution inspired by the blue sky problem from image analysis, for $n = 965$ samples on the domain $\Omega = [-1.5, 1.5] \times [-1, 1]$. We choose $\varepsilon = 0.04$ and $\delta = 0.10$, in order to optimize agreement between the discrete dynamics and the steady state of the continuum PDE (5.15).

In simple terms, the blue sky problem can be described as a setting in which data points are distributed over two elongated clusters that are separated by a narrow low-density region. For concreteness, we model this with a density of the form:

$$\rho_{\text{blue sky}}(x, y) = \varphi_{1.0}(x) * (\varphi_{0.09}(y - 0.32) + \varphi_{0.09}(y + 0.32)).$$

In the top row, we choose $\beta = 0.20$, and in the bottom row, we choose $\beta = 0.95$. In both cases, we choose the initial condition for the dynamics to be δ_{x_i,y_i} for $(x_i, y_i) = (-0.26, -0.29)$. As in Fig. 4, the markers in the first four columns represent the locations of the samples, which form the nodes of our graph, and the colors of the markers represent the value of $u_t(x_i)d(x_i)$ at each node. In the rightmost column, we plot the steady state of the continuum PDE (5.15). We observe good agreement between the graph dynamics and the steady state by time $t = 10.0$. In the case

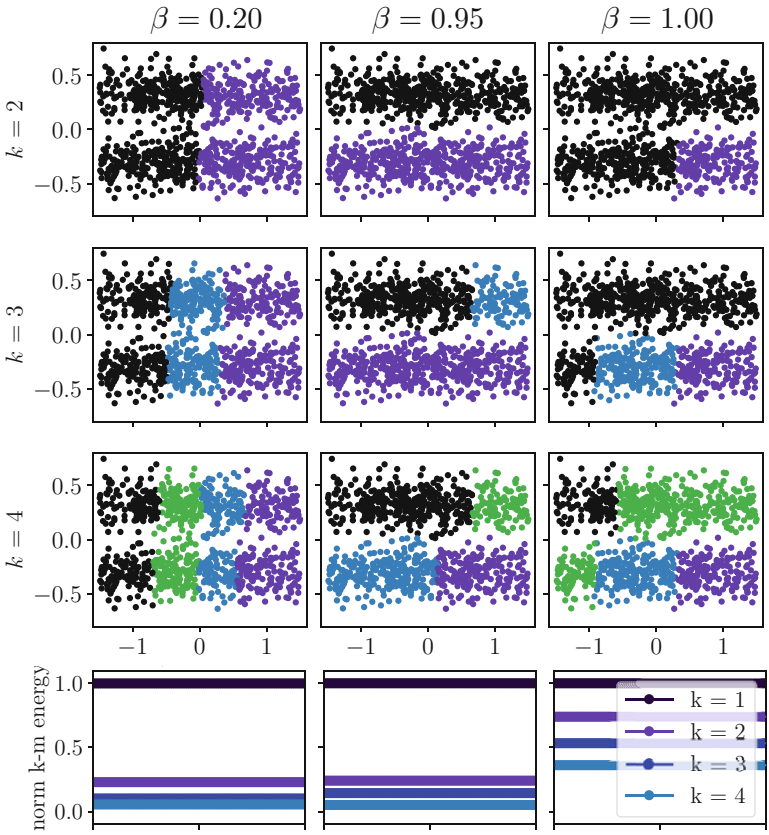


Fig. 10 Illustration of the clustering behavior of Q_β on $\rho_{\text{blue sky}}$ for $n = 965$ samples at time $t = 10$. The first three rows show the clustering behavior for $k = 2, 3, 4$, with each node colored according to which cluster it belongs. The fourth row shows the normalized k-means energy for each number of clusters k

$\beta = 0.95$, the diffuse profile of the steady state illustrates that there is significant diffusion, in spite of the fact that β is close to one.

In Fig. 10, we show the clustering behavior of Q_β on $\rho_{\text{blue sky}}$ for $n = 965$ samples at time $t = 10$. The columns correspond to $\beta = 0.2, 0.95, 1.0$. The first three rows show the clustering behavior for $k = 2, 3, 4$, with each node colored according to which cluster it belongs. In the fourth row, we show the normalized k-means energy for each choice of β.

We observe the best clustering performance for $\beta = 0.95$ and $k = 2$. Furthermore, in this case, the plot of the normalized k-means energy indicates that higher values of k do not lead to significant decreases of the energy, providing further evidence that $k = 2$ is the correct number of clusters. The clustering performance deteriorates for both $\beta = 0.2$ and $\beta = 1.0$. In the case of $\beta = 0.2$, diffusion dominates and the clustering is based on the geometry of the sample

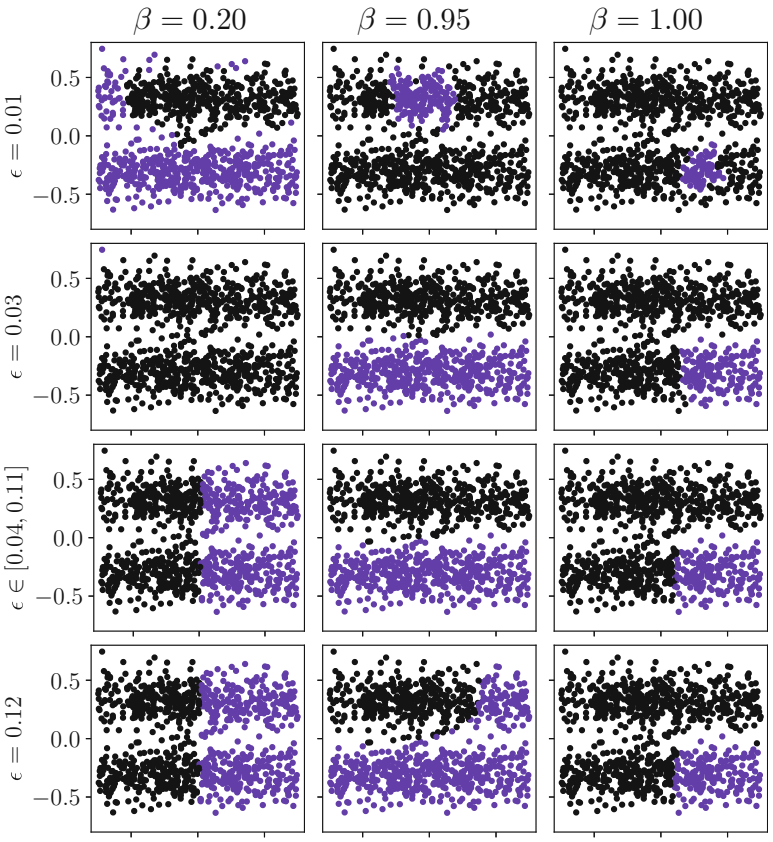


Fig. 11 For the same data distribution as in Fig. 10, we investigate how the clustering behavior of Q_β depends on the graph connectivity length scale ε for $\beta = 0.20, 0.95, 1.00$

points, preferring to cluster by slicing the sample points evenly in two pieces via the shortest cut through the data set. In the case of $\beta = 1.0$, we expect that inaccuracies in the kernel density estimation lead to spurious local minima, and in the absence of diffusion to help overcome these local minima, incorrect clusters are found. Note that simply increasing the bandwidth δ of kernel density estimate in this case would not necessarily improve performance for $\beta = 1.0$, since for a large enough bandwidth, the two lines would merge into one line.

Finally, in Fig. 11, we investigate how the clustering behavior of Q_β for $\rho_{\text{blue sky}}$ depends on the graph connectivity ε; see Eq. (5.1). The columns correspond to $\beta = 0.20, 0.95, 1.00$ and the rows correspond to $\varepsilon = 0.01, 0.03, [0.04, 0.11]$, and 0.12. We note that for a wide range of ε, the diffusion-dominated regime $\beta = 0.2$ prefers to make shorter cuts even over parts of the domain where data are dense, which is undesirable for the data considered. On the other hand, the pure mean shift suffers, as in other examples, from the tendency to

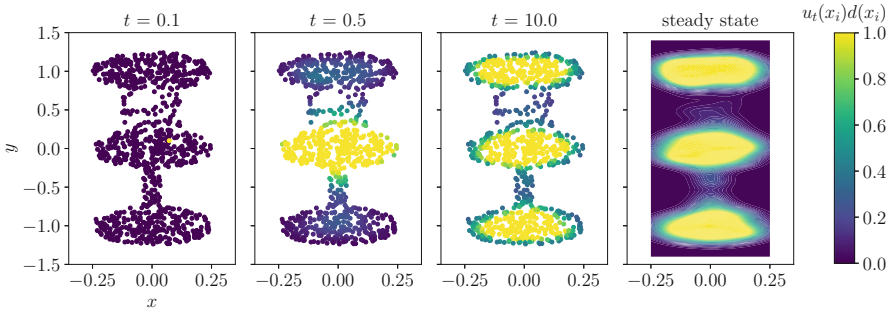


Fig. 12 Illustration of graph dynamics of Q_β on $\rho_{\text{three blobs}}$ for $n = 966$ samples, with $\beta = 0.7$ and initial condition δ_{x_i,y_i} for $(x_i, y_i) = (0.07, 0.10)$. The markers in the first three columns represent the locations of the samples, and the colors of the markers represent the value of $u_t(x_i)d(x_i)$. In the right column, we plot the steady state of the corresponding continuum PDE (5.15)

identify spurious local maxima of KDE as clusters. We observe the best clustering performance over the wide range of ε for $\beta = 0.95$. Considered together with other experiments, this suggests that adding even a small amount of diffusion goes a long way toward correct clustering.

5.2.7 Density vs. Geometry

In Fig. 12, we consider the graph dynamics of Q_β on a two-dimensional data distribution chosen to illustrate how the competing effects of density and geometry depend on the parameter β. We choose $n = 966$ samples, $\varepsilon = 0.07$, and $\delta = 0.05$, in order to optimize agreement between the discrete dynamics and the continuum steady state.

The data density, which we refer to as $\rho_{\text{three blobs}}$, is given by a piecewise constant function that is equal to height one on the three circles of radius 0.25, as well as on the wide rectangle $[0.25, 0.75] \times [-0.125, 0.125]$ on the top. On the narrow rectangle $[-0.75, 0.25] \times [-0.04, 0.04]$ on the bottom, the piecewise constant function has height four. Finally, the data density is multiplied by a normalizing constant so that it integrates to one over the domain $\Omega = [-1.5, 1.5] \times [-1, 1]$.

In Fig. 12, we choose $\beta = 0.7$ and initial condition for the dynamics to be δ_{x_i,y_i} for $(x_i, y_i) = (0.07, 0.10)$. The locations of the markers represent the samples from the data distribution, and the colors of the markers represent the value of $u_t(x_i)d(x_i)$ at each node. In the right column, we plot the steady state of the corresponding continuum PDE (5.15). We observe good agreement with the graph dynamics and the steady state by time $t = 10$.

In Fig. 13, we show the clustering behavior of the Q_β on $\rho_{\text{three blobs}}$ for $n = 966$ samples at time $t = 10$. The two columns correspond to $\beta = 0.7$ and 0.75. The

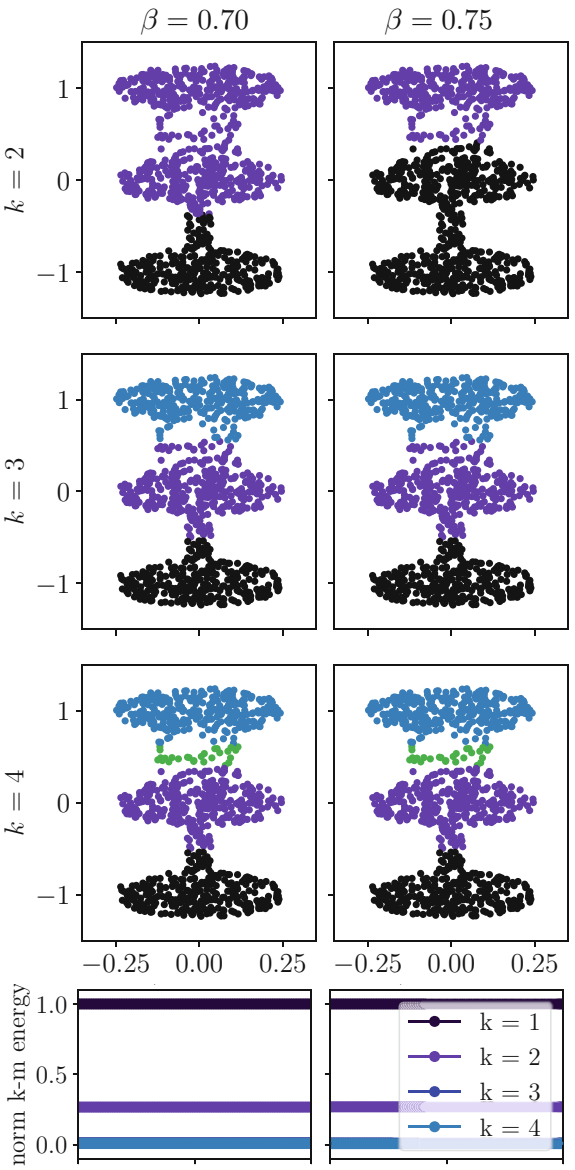


Fig. 13 Illustration of the clustering behavior of the Q_β on $\rho_{\text{three blobs}}$ for $n = 966$ samples at time $t = 10$. The first three rows show the clustering behavior for $k = 2, 3, 4$, with each node colored according to which cluster it belongs. In the fourth row, we show the normalized k-means energy for each choice of k.

first three rows show the clustering behavior for $k = 2, 3, 4$, with each node colored according to which cluster it belongs. In the fourth row, we show the normalized k-means energy for each choice of k.

This simulation provides an example of a data distribution where there is no single "correct" choice of clustering for $k = 2$: a "good" clustering algorithm might seek to cut either the thin, high density rectangle on the bottom or the wide, low

density rectangle on the top. For small values of $\beta \leqslant 0.7$, diffusion dominates, and the clusters are chosen based on the geometry of the data, preferring to cut the thin, high density rectangle. For large values of $\beta \geqslant 0.75$, density dominates, and the clustering prefers to cut the wide, low density rectangle. For intermediate values of β, there is a phase transition for which the clustering becomes unstable.

Acknowledgments The authors would like to thank Eric Carlen, Paul Atzberger, Anna Little, James Murphy, and Daniel McKenzie for helpful discussions. KC was supported by NSF DMS grant 1811012 and a Hellman Faculty Fellowship. NGT was supported by NSF-DMS grant 2005797. DS was supported by NSF grant DMS 1814991. NGT would also like to thank the IFDS at UW-Madison and NSF through TRIPODS grant 2023239 for their support. The authors gratefully acknowledge the support from the Simons Center for Theory of Computing, at which this work was completed.

References

1. L. Ambrosio, N. Gigli, and G. Savaré. *Gradient flows in metric spaces and in the space of probability measures*. Lectures in Mathematics ETH Zürich. Birkhäuser Verlag, Basel, second edition, 2008.
2. E. Arias-Castro, D. Mason, and B. Pelletier. On the estimation of the gradient lines of a density and the consistency of the mean-shift algorithm. *J. Mach. Learn. Res.*, 17:Paper No. 43, 28, 2016.
3. M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural computation*, 15(6):1373–1396, 2003.
4. M. Belkin and P. Niyogi. Towards a theoretical foundation for Laplacian-based manifold methods. In *International Conference on Computational Learning Theory*, pages 486–500. Springer, 2005.
5. D. Burago, S. Ivanov, and Y. Kurylev. A graph discretization of the Laplace-Beltrami operator. *J. Spectr. Theory*, 4(4):675–714, 2014.
6. J. Calder and N. García Trillos. Improved spectral convergence rates for graph Laplacians on epsilon-graphs and k-nn graphs. *arXiv preprint arXiv:1910.13476*, 2019.
7. J. Calder, N. G. Trillos, and M. Lewicka. Lipschitz regularity of graph Laplacians on random data clouds, 2020.
8. M. A. Carreira-Perpiñán. Gaussian mean-shift is an EM algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(5):767–776, 2007.
9. M. A. Carreira-Perpiñán. Clustering methods based on kernel density estimators: mean-shift algorithms. In *Handbook of cluster analysis*, Chapman & Hall/CRC Handb. Mod. Stat. Methods, pages 383–417. CRC Press, Boca Raton, FL, 2016.
10. X. Cheng and H.-T. Wu. Convergence of graph Laplacian with KNN self-tuned kernels. *arXiv preprint arXiv:2011.01479*, 2020.
11. S.-N. Chow, L. Dieci, W. Li, and H. Zhou. Entropy dissipation semi-discretization schemes for Fokker-Planck equations. *J. Dynam. Differential Equations*, 31(2):765–792, 2019.
12. R. R. Coifman and S. Lafon. Diffusion maps. *Appl. Comput. Harmon. Anal.*, 21(1):5–30, 2006.
13. D. B. Dunson, H.-T. Wu, and N. Wu. Spectral convergence of graph Laplacian and heat kernel reconstruction in l^∞ from random samples, 2019.
14. A. Esposito, F. S. Patacchini, A. Schlichting, and D. Slepcev. Nonlocal-interaction equation on graphs: gradient flow structure and continuum limit. *Arch. Ration. Mech. Anal.*, 240(2):699–760, 2021.

15. K. Fukunaga and L. Hostetler. The estimation of the gradient of a density function, with applications in pattern recognition. *IEEE Transactions on Information Theory*, 21(1):32–40, 1975.
16. N. García Trillos, M. Gerlach, M. Hein, and D. Slepčev. Error estimates for spectral convergence of the graph Laplacian on random geometric graphs toward the Laplace–Beltrami operator. *Foundations of Computational Mathematics*, pages 1–61, 2019.
17. N. García Trillos, F. Hoffmann, and B. Hosseini. Geometric structure of graph Laplacian embeddings. *Journal of Machine Learning Research*, 22(63):1–55, 2021.
18. N. García Trillos and D. Slepčev. A variational approach to the consistency of spectral clustering. *Applied and Computational Harmonic Analysis*, 45(2):239–281, 2018.
19. E. Giné and V. Koltchinskii. Empirical graph Laplacian approximation of Laplace-Beltrami operators: large sample results. In *High dimensional probability*, volume 51 of *IMS Lecture Notes Monogr. Ser.*, pages 238–259. Inst. Math. Statist., Beachwood, OH, 2006.
20. M. Hein, J.-Y. Audibert, and U. v. Luxburg. Graph Laplacians and their convergence on random neighborhood graphs. *Journal of Machine Learning Research*, 8(6), 2007.
21. M. Hein, J.-Y. Audibert, and U. Von Luxburg. From graphs to manifolds–weak and strong pointwise consistency of graph laplacians. In *International Conference on Computational Learning Theory*, pages 470–485. Springer, 2005.
22. V. J. Hodge and J. Austin. A survey of outlier detection methodologies. *Artificial Intelligence Review*, 22(2):85–126, Oct 2004.
23. J. D. Hunter. Matplotlib: a 2d graphics environment. *Comput. Sci. Eng.*, 9(3):90–95, 2007.
24. E. Jones, T. Oliphant, P. Peterson, et al. *SciPy: Open source scientific tools for Python*, 2001–. Available at http://www.scipy.org/.
25. W. L. G. Koontz, P. M. Narendra, and K. Fukunaga. A graph-theoretic approach to nonparametric cluster analysis. *IEEE Transactions on Computers*, (9):936–944, 1976.
26. A. Little, M. Maggioni, and J. M. Murphy. Path-based spectral clustering: Guarantees, robustness to outliers, and fast algorithms. *Journal of Machine Learning Research*, 21(6):1–66, 2020.
27. A. Little, D. McKenzie, and J. Murphy. Balancing geometry and density: Path distances on high-dimensional data, 2020.
28. J. Lu. Graph approximations to the Laplacian spectra. *arXiv: Differential Geometry*, 2019.
29. J. Maas. Gradient flows of the entropy for finite Markov chains. *Journal of Functional Analysis*, 8(261):2250–2292, 2011.
30. L. Michel. About small eigenvalues of the Witten Laplacian. *Pure and Applied Analysis*, 1(2):149–206, 2019.
31. B. Nadler, S. Lafon, R. R. Coifman, and I. G. Kevrekidis. Diffusion maps, spectral clustering and reaction coordinates of dynamical systems. *Applied and Computational Harmonic Analysis*, 21(1):113–127, 2006. Special Issue: Diffusion Maps and Wavelets.
32. A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems (NIPS)*, pages 849–856. MIT Press, 2001.
33. F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
34. G. Schiebinger, M. J. Wainwright, and B. Yu. The geometry of kernelized spectral clustering. *The Annals of Statistics*, 43(2):819–846, 2015.
35. J. Shi and J. Malik. Normalized cuts and image segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(8):888–905, 2000.
36. Z. Shi. Convergence of Laplacian spectra from random samples. arXiv preprint arXiv:1507.00151, 2015.
37. A. Singer. From graph to manifold Laplacian: The convergence rate. *Applied and Computational Harmonic Analysis*, 21(1):128–134, 2006.

38. R. Tibshirani, G. Walther, and T. Hastie. Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(2):411–423, 2001.
39. H. tieng Wu and N. Wu. When locally linear embedding hits boundary, 2018.
40. D. Ting, L. Huang, and M. Jordan. An analysis of the convergence of graph laplacians. *arXiv preprint arXiv:1101.5435*, 2011.
41. L. van der Maaten, E. Postma, and H. van den Herik. Dimensionality reduction: A comparative review. Tilburg University Technical Report, TiCC-TR2009-005. 2009.
42. S. van der Walt, C. Colbert, and G. Varoquaux. The numpy array: a structure for efficient numerical computation. *Comput. Sci. Eng.*, 13(2):22–30, 2011.
43. R. Vaughn, T. Berry, and H. Antil. Diffusion maps for embedded manifolds with boundary with applications to PDEs, 2019.
44. A. Vedaldi and S. Soatto. Quick shift and kernel methods for mode seeking. In *European conference on computer vision*, pages 705–718. Springer, 2008.
45. U. von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, Dec 2007.
46. U. von Luxburg, M. Belkin, and O. Bousquet. Consistency of spectral clustering. *Ann. Statist.*, 36(2):555–586, 2008.
47. E. Witten. Supersymmetry and Morse theory. *Journal of Differential Geometry*, 17(4):661–692, 1982.
48. C. L. Wormell and S. Reich. Spectral convergence of diffusion maps: improved error bounds and an alternative normalisation, 2020.