Aligned Diffusion Schrödinger Bridges

Vignesh Ram Somnath [*1,2] **Matteo Pariset** [*1,3] **Ya-Ping Hsieh** [1] **Maria Rodriguez Martinez** [2] **Andreas Krause** [1] **Charlotte Bunne** [1]

Abstract

Diffusion Schrödinger bridges (DSB) have recently emerged as a powerful framework for recovering stochastic dynamics via their marginal observations at different time points. Despite numerous successful applications, existing algorithms for solving DSBs have so far failed to utilize the structure of *aligned* data, which naturally arises in many biological phenomena. In this paper, we propose a novel algorithmic framework that, for the first time, solves DSBs while respecting the data alignment. Our approach hinges on a combination of two decades-old ideas: The classical Schrödinger bridge theory and Doob's h-*transform*. Compared to prior methods, our approach leads to a simpler training procedure with lower variance, which we further augment with principled regularization schemes. This ultimately leads to sizeable improvements across experiments on synthetic and real data, including the tasks of predicting conformational changes in proteins and temporal evolution of cellular differentiation processes.

1 Introduction

The task of transforming a given distribution into another lies at the heart of many modern machine learning applications such as single-cell genomics (Tong et al., 2020; Schiebinger et al., 2019; Bunne et al., 2022a), meteorology (Fisher et al., 2009), and robotics (Chen et al., 2021a). To this end, diffusion Schrödinger bridges (De Bortoli et al., 2021; Chen et al., 2022a; Vargas et al., 2021; Liu et al., 2022b) have recently emerged as a powerful paradigm due to their ability to generalize prior deep diffusion-based models, notably score matching with Langevin dynamics (Song & Ermon, 2019; Song et al., 2021) and denoising diffusion probabilistic models (Ho et al., 2020), which have achieved the state-of-the-art on many generative modeling problems.

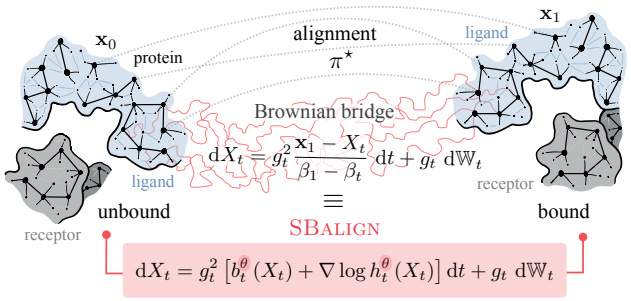

Figure 1: Overview of SBALIGN: In biological tasks such as modelling conformational changes in protein docking, one is naturally provided with *aligned* data in the form of unbound and bound structures of participating proteins. Our goal is to therefore recover a stochastic trajectory from the unbound ($\mathbf{x}_0$) to the bound ($\mathbf{x}_1$) structure. To achieve this, we connect the characterization of an SDE conditioned on $\mathbf{x}_0$ and $\mathbf{x}_1$ (utilizing the Doob's h-*transform*) with that of a Brownian bridge between $\mathbf{x}_0$ and $\mathbf{x}_1$ (classical Schrödinger bridge theory). We show that this leads to a simpler training procedure with lower variance and strong empirical results.

Despite the success of DSBs solvers, a significant limitation of existing frameworks is that they fail to capture the *alignment* of data: If $\hat{\mathbb{P}}_0, \hat{\mathbb{P}}_1$ are two (empirical) distributions between which we wish to interpolate, then a tacit assumption in the literature is that the dependence of $\hat{\mathbb{P}}_0$ and $\hat{\mathbb{P}}_1$ is unknown and somehow has to be recovered. Such an assumption, however, ignores important scenarios where the data is *aligned*, meaning that the samples from $\hat{\mathbb{P}}_0$ and $\hat{\mathbb{P}}_1$ naturally come in pairs $(\mathbf{x}_0^i, \mathbf{x}_1^i)_i^N$, which is common in many biological phenomena. Proteins, for instance, undergo conformational changes upon interactions with other biomolecules (protein docking, see Fig. 1). The goal is to model conformational changes by recovering a (stochastic) trajectory $\mathbf{x}_t$ based on the positions observed at two-time points $(\mathbf{x}_0, \mathbf{x}_1)$. Failing to incorporate this alignment would mean that we completely ignore information on the correspondence between the initial and final points of the molecules, resulting in a much harder problem than necessary.

[*]Equal contribution [1]Department of Computer Science, ETH Zürich, Zürich, Switzerland [2]IBM Research, Zürich, Switzerland [3]School of Computer Science, EPFL, Lausanne, Switzerland. Correspondence to: Vignesh Ram Somnath <vsomnath@ethz.ch>.

Workshop on New Frontiers in Learning, Control, and Dynamical Systems at the International Conference on Machine Learning (ICML), Honolulu, Hawaii, USA, 2023.

Beyond, the recent use of SBs has been motivated by an important task in molecular biology: Cells change their molecular profile throughout developmental processes (Schiebinger et al., 2019; Bunne et al., 2022b) or in response to perturbations such as cancer drugs (Lotfollahi et al., 2019; Bunne et al., 2021). As most measurement technologies are destructive assays, i.e., the same cell cannot be observed twice nor fully profiled over time, these methods aim at reconstructing cell dynamics from *unpaired* snapshots. Recent developments in molecular biology, however, aim at overcoming this technological limitation. For example, Chen et al. (2022b) propose a transcriptome profiling approach that preserves cell viability. Weinreb et al. (2020) capture cell differentiation processes by clonally connecting cells and their progenitors through barcodes (see illustrative Figure in Supplement).

Motivated by these observations, the goal of this paper is to propose a novel algorithmic framework for solving DSBs with (partially) *aligned* data. Our approach is in stark contrast to existing works which, due to the lack of data alignment, all rely on some variants of *iterative proportional fitting* (IPF) (Fortet, 1940; Kullback, 1968) and are thus prone to numerical instability. On the other hand, via a combination of the original theory of Schrödinger bridges (Schrödinger, 1931; Léonard, 2013) and the key notion of Doob's h-*transform* (Doob, 1984; Rogers & Williams, 2000), we design a novel loss function that completely bypasses the IPF procedure and can be trained with much lower variance.

To summarize, we make the following contributions:

- To our best knowledge, we consider, for the first time, the problem of interpolation with *aligned* data. We rigorously formulate the problem in the DSB framework.
- Based on the theory of Schrödinger bridges and h-transform, we derive a new loss function that, unlike prior work on DSBs, does not require an IPF-like procedure to train. We also propose principled regularization schemes to further stabilize training.
- We describe how interpolating aligned data can provide better reference processes for use in classical DSBs, paving the way to hybrid aligned/non-aligned Schrödinger bridges (SBs).
- We evaluate our proposed framework on both synthetic and real data. For experiments utilizing real data, we consider two tasks where such aligned data is naturally available. The first is the task of developmental processes in single-cell biology, and the second involves a subproblem associated with the protein docking problem - conformational changes modelling. In the second task, the goal is to predict the 3D structure of the bound protein, given the unbound 3D structure. Our method demonstrates a considerable improvement over prior methods across various metrics, thereby substantiating the importance of taking the data alignment into account.

Related work. Solving DSBs is a subject of significant interest in recent years and has flourished in a number of different algorithms (De Bortoli et al., 2021; Chen et al., 2022a; Vargas et al., 2021; Bunne et al., 2023; Liu et al., 2022a). However, all these previous approaches focus on *unaligned* data, and therefore the methodologies all rely on IPF and are hence drastically different from ours. In the experiments, we will demonstrate the importance of taking the alignment of data into consideration by comparing our method to these baselines.

An important ingredient in our theory is Doob's h-transform, which has recently also been utilized by (Heng et al., 2021; Liu et al., 2023) to solve the problem of conditional diffusion. However, their fundamental motivation is different from ours. Heng et al. (2021) focus on learning the h-transform for a given drift function by approximating two score functions, while Liu et al. (2023) focus on learning the drift of the diffusion model and the h-transform *together*. In contrast, our goal is to read off the drift *from* the h-transform with the help of *aligned data*, and use the learnt drift for unconditional simulation.

To the best of our knowledge, the concurrent work of Tong et al. (2023) is the only existing framework that can tackle aligned data, which, however, is not their original motivation. In the context of solving DSBs, their algorithm can be seen as learning a vector field that generates the correct *marginal* probability (cf. Tong et al., 2023, Proposition 4.3). Importantly, this is different from our aim of finding the *pathwise* optimal solution of DSBs: If $(\mathbf{x}^i_{0,\text{test}})_{i=1}^m$ is a test data set for which we wish to predict their destinations, then the framework of Tong et al. (2023) can only ensure that the marginal distribution $(\mathbf{x}^i_{1,\text{test}})_{i=1}^m$ is correct, whereas ours is capable of predicting that $\mathbf{x}^i_{1,\text{test}}$ is precisely the destination of $\mathbf{x}^i_{0,\text{test}}$ for each i. This latter property is highly desirable in tasks like ML-accelerated protein docking.

To solve aligned SB problems, we rely on mixtures of diffusion processes. Like in Peluchetti (2023), we construct them from pairings and define an associated training objective inspired by score-based modeling. However, we represent the learned drift as a sum of the solution to an SB problem (b) and a pairing-related term ($\nabla \log h$). We parametrize the second part of the drift with neural networks, unlike Schauer et al. (2017) which use an auxiliary (simpler) process.

2 Background

Problem formulation. Suppose that we are given access to i.i.d. *aligned* data $(\mathbf{x}^i_0, \mathbf{x}^i_1)_{i=1}^N$, where the marginal distri-

bution of $\mathbf{x}_0^i$'s is $\hat{\mathbb{P}}_0$ and of $\mathbf{x}_1^i$'s is $\hat{\mathbb{P}}_1$. Typically, we view $\hat{\mathbb{P}}_0$ as the empirical marginal distribution of a stochastic process observed at time $t = 0$, and likewise $\hat{\mathbb{P}}_1$ the empirical marginal observed at $t = 1$. The goal is to reconstruct the stochastic process $\mathbb{P}_t$ based on $(\mathbf{x}_0^i, \mathbf{x}_1^i)_{i=1}^N$, i.e., to *interpolate* between $\hat{\mathbb{P}}_0$ and $\hat{\mathbb{P}}_1$.

Such a task is ubiquitous in biological applications. For instance, understanding structural changes in biomolecules as part of molecular docking is of significant interest in biology and has become a topic of intense study in recent years (Tsaban et al., 2022; Corso et al., 2023). Here, $\mathbf{x}_0^i$ represents the 3D unbound structures of the participating biomolecules, while $\mathbf{x}_1^i$ represents the 3D structures in the bound version. Reconstructing a stochastic process that diffuses $\mathbf{x}_0^i$'s to $\mathbf{x}_1^i$'s is tantamount to recovering the associated underlying energy landscape. Similarly, in molecular dynamics simulations, we have access to trajectories $\left(\mathbf{x}_t^i\right)_{t\in[0,1]}$, where $\mathbf{x}_0^i$ and $\mathbf{x}_1^i$ represent the initial and final positions of the i-th molecule respectively. Any learning algorithm using these simulations should be able to respect the provided alignment.

Diffusion Schrödinger bridges. To solve the interpolation problem, in Section 3, we will invoke the framework of DSBs, which are designed to solve interpolation problems with *unaligned* data. More specifically, given two marginals $\hat{\mathbb{P}}_0$ and $\hat{\mathbb{P}}_1$, the DSB framework proceeds by first choosing a reference process $\mathbb{Q}_t$ using prior knowledge, for instance a simple Brownian motion, and then solve the entropy-minimization problem over all stochastic processes $\mathbb{P}_t$:

$$\min_{\mathbb{P}_0=\hat{\mathbb{P}}_0,\ \mathbb{P}_1=\hat{\mathbb{P}}_1} D_{\mathrm{KL}}(\mathbb{P}_t\|\mathbb{Q}_t). \tag{SB}$$

Despite the fact that many methods exist for solving (SB) (De Bortoli et al., 2021; Chen et al., 2022a; Vargas et al., 2021; Bunne et al., 2023), none of these approaches are capable of incorporating *alignment* of the data. This can be seen by inspecting the objective (SB), in which the coupling information $(\mathbf{x}_0^i, \mathbf{x}_1^i)$ is completely lost as only its individual marginals $\hat{\mathbb{P}}_0, \hat{\mathbb{P}}_1$ play a role therein. Unfortunately, it is well-known that tackling the marginals separately necessitates a forward-backward learning process known as the *iterative proportional fitting* (IPF) procedure (Fortet, 1940; Kullback, 1968), which constitutes the primary reason of high variance training, thereby confronting DSBs with numerical and scalability issues. Our major contribution, detailed in the next section, is therefore to devise the first algorithmic framework that solves the interpolation problem with aligned data *without* resorting to IPF.

3 Aligned Diffusion Schrödinger Bridges

In this section, we derive a novel loss function for DSBs with aligned data by combining two classical notions: The theory of Schrödinger bridges (Schrödinger, 1931; Léonard, 2013; Chen et al., 2021b) and Doob's h-transform (Doob, 1984; Rogers & Williams, 2000). We then describe how solutions to DSBs with aligned data can be leveraged in the context of classical DSBs.

3.1 Learning aligned diffusion Schrödinger bridges

Static SB and aligned data. Our starting point is the simple and classical observation that (SB) is the continuous-time analogue of the *entropic optimal transport*, also known as the *static* Schrödinger bridge problem (Léonard, 2013; Chen et al., 2021b; Peyré & Cuturi, 2019):

$$\pi^\star := \operatorname*{argmin}_{\mathbb{P}_0=\hat{\mathbb{P}}_0,\ \mathbb{P}_1=\hat{\mathbb{P}}_1} D_{\mathrm{KL}}(\mathbb{P}_{0,1}\|\mathbb{Q}_{0,1}) \tag{1}$$

where the minimization is over all *couplings* of $\hat{\mathbb{P}}_0$ and $\hat{\mathbb{P}}_1$, and $\mathbb{Q}_{0,1}$ is simply the joint distribution of $\mathbb{Q}_t$ at $t = 0, 1$. In other words, if we denote by $\mathbb{P}_t^\star$ the stochastic process that minimizes (SB), then the joint distribution $\mathbb{P}_{0,1}^\star$ necessarily coincides with the $\pi^\star$ in (1). Moreover, since in DSBs, the data is always assumed to arise from $\mathbb{P}_t^\star$, we see that:

> The *aligned* data $(\mathbf{x}_0^i, \mathbf{x}_1^i)_{i=1}^N$ constitutes samples of $\pi^\star$.

This simple but crucial observation lies at the heart of all derivations to come.

Our central idea is to represent $\mathbb{P}_t^\star$ via two different, but equivalent, characterizations, both of which involve $\pi^\star$: That of a *mixture* of reference processes with pinned end points, and that of conditional *stochastic differential equations* (SDEs).

$\mathbb{P}_t^\star$ from $\pi^\star$: $\mathbb{Q}_t$ with pinned end points. For illustration purposes, from now on, we will assume that the reference process $\mathbb{Q}_t$ is a Brownian motion with diffusion coefficient g_t:[1]

$$\mathrm{d}\mathbb{Q}_t = g_t\, \mathrm{d}\mathbb{W}_t. \tag{2}$$

In this case, it is well-known that $\mathbb{Q}_t$ *conditioned* to start at $\mathbf{x}_0$ and end at $\mathbf{x}_1$ can be written in another SDE (Mansuy & Yor, 2008; Liu et al., 2023):

$$\mathrm{d}X_t = g_t^2 \frac{\mathbf{x}_1 - X_t}{\beta_1 - \beta_t}\, \mathrm{d}t + g_t\, \mathrm{d}\mathbb{W}_t \tag{3}$$

where $X_0 = \mathbf{x}_0$ and

$$\beta_t := \int_0^t g_s^2\, \mathrm{d}s. \tag{4}$$

[1] Extension to more involved reference processes is conceptually straightforward but notationally clumsy. Furthermore, reference processes of the form (2) are dominant in practical applications (Song et al., 2021; Bunne et al., 2023), so we omit the general case.

We call the processes in (3) the *scaled Brownian bridges* as they generalize the classical Brownian bridge, which corresponds to the case of $g_t \equiv 1$.

The first characterization of $\mathbb{P}_t^\star$ is then an immediate consequence of a classical result in Schrödinger bridge theory: Draw a sample $(\mathbf{x}_0, \mathbf{x}_1) \sim \pi^\star$ and connect them via (3). The resulting path is a sample from $\mathbb{P}_t^\star$ (Léonard, 2013; Chen et al., 2021b). In other words, $\mathbb{P}_t^\star$ is a *mixture* of scaled Brownian bridges, with mixing weights given by $\pi^\star$.

$\mathbb{P}_t^\star$ from $\pi^\star$: SDE representation. Another characterization of $\mathbb{P}_t^\star$ is that it is itself given by an SDE of the form (Léonard, 2013; Chen et al., 2021b)

$$\mathrm{d}X_t = g_t^2 b_t(X_t)\,\mathrm{d}t + g_t\,\mathrm{d}\mathbb{W}_t. \tag{5}$$

Here, $b_t : \mathbb{R}^d \to \mathbb{R}^d$ is a time-dependent drift function that we wish to learn.

Now, by Doob's h-transform, we know that the SDE (5) *conditioned* to start at $\mathbf{x}_0$ and end at $\mathbf{x}_1$ is given by another SDE (Doob, 1984; Rogers & Williams, 2000):

$$\mathrm{d}X_t = g_t^2[b_t(X_t) + \nabla \log h_t(X_t)]\,\mathrm{d}t + g_t\,\mathrm{d}\mathbb{W}_t \tag{6}$$

where $h_t(\mathbf{x}) := \mathbb{P}(X_1 = \mathbf{x}_1 | X_t = \mathbf{x})$ is the *Doob's h function*. Notice that we have suppressed the dependence of h_t on $\mathbf{x}_0$ and $\mathbf{x}_1$ for notational simplicity.

Loss function. Since both (3) and (6) represent $\mathbb{P}_t^\star$, the solution of the DSBs, the two SDEs must coincide. In the stochastic optimal control literature, learning b_t is often framed as an energy minimization procedure, where the energy is defined as the appropriate ℓ_2 norm of b_t over time.

In other words, suppose we parametrize b_t as b_t^θ, then, by matching terms in (3) and (6), we can learn the optimal parameter $\theta^\star$ via minimization the following ℓ_2 norm,

$$L(\theta) := \mathbb{E}\left[\int_0^1 \left\| \frac{\mathbf{x}_1 - X_t}{\beta_1 - \beta_t} - \nabla \log h_t^\theta(X_t) \right\|^2 \mathrm{d}t\right] \tag{7}$$

where h_t^θ depends on b_t^θ as well as the drawn samples $(\mathbf{x}_0, \mathbf{x}_1)$. This is the case since h_t is defined as an expectation using trajectories sampled under b_t^θ with given endpoints. Therefore, assuming that, for each θ, we can compute h_t^θ *based only on* b_t^θ, we can then backprop through (7) and optimize it using any off-the-shelf algorithm.

A slightly modified (7). Even with infinite data and a neural network with sufficient capacity, the loss function defined in (7) does not converge to 0. To satisfy these properties, we instead propose to modify (7) to:

$$L(\theta) := \mathbb{E}\left[\int_0^1 \left\| \frac{\mathbf{x}_1 - X_t}{\beta_1 - \beta_t} - \left(b_t^\theta + \nabla \log h_t^\theta(X_t)\right) \right\|^2 \mathrm{d}t\right] \tag{8}$$

Algorithm 1 SBALIGN

Input: Aligned data $(\mathbf{x}_0^i, \mathbf{x}_1^i)_{i=1}^N$, learning rates $\gamma_\theta, \gamma_\phi$, number of iterations K

Initialize $\theta \leftarrow \theta_0$, $\phi \leftarrow \phi_0$.
for $k = 1$ **to** K **do**
  Draw a mini-batch of samples from $(\mathbf{x}_0^i, \mathbf{x}_1^i)_{i=1}^N$
  Compute empirical average of (12) with mini-batch.
  Update $\phi \leftarrow \phi - \gamma_\phi \nabla L(\theta, \phi)$
  Update $\theta \leftarrow \theta - \gamma_\theta \nabla L(\theta, \phi)$
end for

for which b_t is a minimizer. Notice that (8) bears a similar form as the popular score-matching objective employed in previous works (Song & Ermon, 2019; Song et al., 2021):

$$L(\theta) := \mathbb{E}\left[\int_0^1 \left\| \nabla \log p(\mathbf{x}_t|\mathbf{x}_0) - s^\theta(X_t, t) \right\|^2 \mathrm{d}t\right], \tag{9}$$

where the term $\frac{\mathbf{x}_1 - X_t}{\beta_1 - \beta_t}$ is akin to $\nabla \log p(\mathbf{x}_t|\mathbf{x}_0)$, while $\left(b_t^\theta + \nabla \log h_t^\theta(X_t)\right)$ corresponds to $s^\theta(X_t, t)$.

Computing h_t^θ. Inspecting h_t in (6), we see that, given $(\mathbf{x}_0, \mathbf{x}_1)$, it can be written as the conditional expectation of an indicator function:

$$h_t(\mathbf{x}) = \mathbb{P}(X_1 = \mathbf{x}_1 | X_t = \mathbf{x}) = \mathbb{E}\left[\mathbb{1}_{\{\mathbf{x}_1\}} | X_t = \mathbf{x}\right] \tag{10}$$

where the expectation is over (5). Functions of the form (10) lend itself well to computation since it solves simulating the *unconditioned* paths. Furthermore, in order to avoid overfitting on the given samples, it is customary to replace the "hard" constraint $\mathbb{1}_{\{\mathbf{x}_1\}}$ by its *smoothed* version (Zhang & Chen, 2022; Holdijk et al., 2022):

$$h_{t,\tau}(\mathbf{x}) := \mathbb{E}\left[\exp\left(-\frac{1}{2\tau}\|X_1 - \mathbf{x}_1\|^2\right) | X_t = \mathbf{x}\right]. \tag{11}$$

Here, τ is a regularization parameter that controls how much we "soften" the constraint, and we have $\lim_{\tau \to 0} h_{t,\tau} = h_t$.

Although the computation of (11) can be done via a standard application of the Feynman–Kac formula (Rogers & Williams, 2000), an altogether easier approach is to parametrize $h_{t,\tau}$ by a second neural network m^ϕ and perform alternating minimization steps on b_t^θ and m^ϕ. This choice reduces the variance in training, since it avoids the sampling of unconditional paths described by (5) (see §A.1 for a detailed explanation).

Regularization. Since it is well-known that $\nabla \log h_t$ typically explodes when $t \to 1$ (Liu et al., 2023), it is important to regularize the behavior of m^ϕ for numerical stability, especially when $t \to 1$. Moreover, in practice, it is desirable to learn a drift b_t^θ that respects the data alignment *in expectation*: If $(\mathbf{x}_0, \mathbf{x}_1)$ is an input pair, then multiple runs of

the SDE (5) starting from $\mathbf{x}_0$ should, on average, produce samples that are in the proximity of $\mathbf{x}_1$. This observation implies that we should search for drifts whose corresponding h-transforms are diminishing.

A simple way to simultaneously achieve the above two requirements is to add an ℓ^2-regularization term, resulting in the loss function:

$$L(\theta, \phi) := \mathbb{E}\left[\int_0^1 \left\| \frac{\mathbf{x}_1 - X_t}{\beta_1 - \beta_t} - \left(b_t^\theta + m^\phi(X_t)\right)\right\|^2 + \lambda_t \|m^\phi(\mathbf{x}_t)\|^2 \, \mathrm{d}t\right] \tag{12}$$

where λ_t can either be constant or vary with time. The overall algorithm is depicted in Algorithm 1.

3.2 Paired Schrödinger bridges as prior processes

Our algorithm finds solutions to SBs on aligned data by relying on samples drawn from the (optimal) coupling $\pi^\star$. This is what differentiates it from classical SBs –which instead only consider samples from $\hat{\mathbb{P}}_0$ and $\hat{\mathbb{P}}_1$– and plays a critical role in avoiding IPF-like iterates. However, SBALIGN reliance on samples from $\pi^\star$ may become a limitation, when the available information on alignments is insufficient.

If the number of pairings is limited, it is unrealistic to hope for an accurate solution to the aligned SB problem. However, the interpolation between $\hat{\mathbb{P}}_0$ and $\hat{\mathbb{P}}_1$ learned by SBALIGN can potentially be leveraged as a starting point to obtain a better reference process, which can then be used when solving a classical SB on the same marginals. In other words, the drift $b_t^{\text{aligned}}(X_t)$ learned through SBALIGN can be used *as is* to construct a data-informed alternative $\tilde{\mathbb{Q}}$ to the standard Brownian motion, defined by paths:

$$\tilde{X}_t = b_t^{\text{aligned}}(\tilde{X}_t)dt + g_t dW_t$$

Intuitively, solving a standard SB problem with $\tilde{\mathbb{Q}}$ as reference is beneficial because the (imperfect) coupling of marginals learned by SBALIGN ($\tilde{\mathbb{Q}}_{01}$) is, in general, closer to the truth than $\mathbb{Q}_{01}$.

Improving reference processes through pre-training or data-dependent initialization has been previously considered in the literature. For instance, both De Bortoli et al. (2021) and Chen et al. (2022a) use a pre-trained reference process for challenging image interpolation tasks. This approach, however, relies on DSBs trained using the classical score-based generative modeling objective between a Gaussian and the data distribution. It, therefore, pre-trains the reference process on a related –but different– process, i.e., the one mapping Gaussian noise to data rather than $\hat{\mathbb{P}}_0$ to $\hat{\mathbb{P}}_1$. An alternative, proposed by Bunne et al. (2023) draws on the closed-form solution of SBs between two Gaussian distributions, which are chosen to approximate $\hat{\mathbb{P}}_0$ and $\hat{\mathbb{P}}_1$, respectively. Unlike our method, these alternatives construct prior drifts by falling back to simpler and related tasks, or approximations of the original problem. We instead propose to shape a coarse-grained description of the drift based on alignments sampled directly from $\pi^\star_{01}$.

4 Experiments

In this section, we evaluate SBALIGN in different settings involving 2-dimensional synthetic datasets, the task of reconstructing cellular differentiation processes, as well as predicting the conformation of a protein structure and its ligand formalized as rigid protein docking problem.

4.1 Synthetic Experiments

We run our algorithm on two synthetic datasets (Figures in § B), and compare the results with classic diffusion Schrödinger bridge models, i.e., the forward-backward SB formulation proposed by (Chen et al., 2022a), herein referred to as FBSB. We equip the baseline with prior knowledge, as elaborated below, to further challenge SBALIGN.

Moon dataset. The first synthetic dataset (Fig. 2a-c) consists of two distributions, each supported on two semi-circles ($\hat{\mathbb{P}}_0$ drawn in *blue* and $\hat{\mathbb{P}}_1$ in *red*). $\hat{\mathbb{P}}_1$ was obtained from $\hat{\mathbb{P}}_0$ by applying a clockwise rotation around the center, i.e., by making points in the upper blue arm correspond to those in the right red one. This transformation is clearly not the most likely one under the assumption of Brownian motion of particles and should therefore not be found as the solution of a classical SB problem. This is confirmed by FBSB trajectories (Fig. 2a), which tend to map points to their closest neighbor in $\hat{\mathbb{P}}_1$ (e.g., some points in the upper arm of $\hat{\mathbb{P}}_0$ are brought towards the left rather than towards the right). While being a minimizer of (SB), such a solution completely disregards our prior knowledge on the alignment of particles, which is instead reliably reproduced by the dynamics learned by SBALIGN (Fig. 2c).

One way of encoding this additional information on the nature of the process is to modify $\mathbb{Q}_t$ by introducing a clockwise radial drift, which describes the prior tangential velocity of particles moving circularly around the center. Solving the classical SB with this updated reference process indeed generates trajectories that respect most alignments (Fig. 2b), but requires a hand-crafted expression of the drift that is only possible in very simple cases.

T dataset. In most real-world applications, it is very difficult to define an appropriate reference process $\mathbb{Q}_t$, which respects the known alignment without excessively distorting the trajectories from a solution to (SB). This is already visible in simple examples like (Fig. 2d-f), in which the value of good candidate prior drifts at a specific location

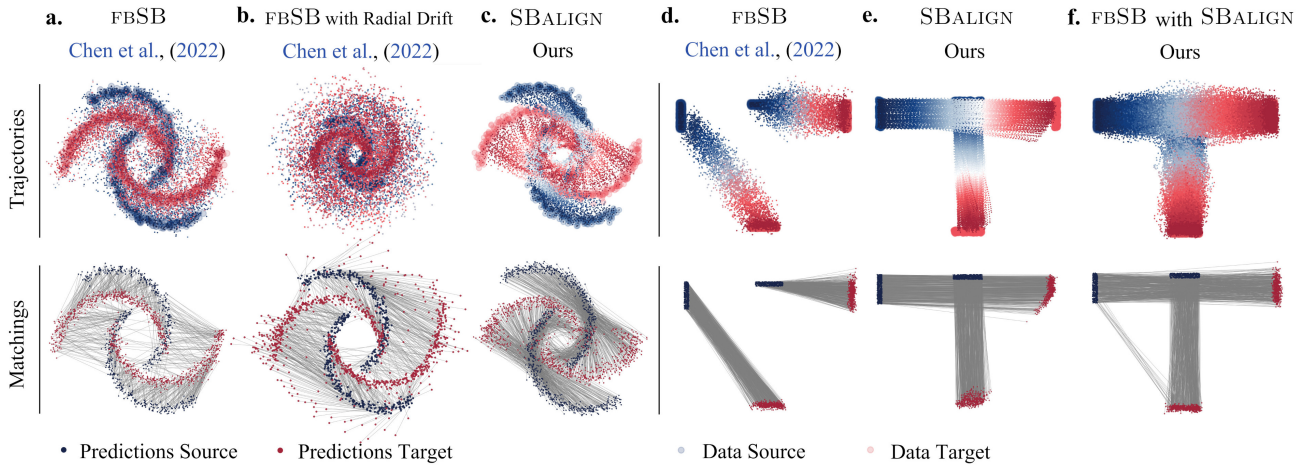

Figure 2: Experimental results on the Moon dataset (**a-c**) and T-dataset (**d-f**). The top row shows the trajectory sampled using the learned drift, and the bottom row shows the matching based on the learnt drift. Compared to other baselines, SBALIGN is able to learn an appropriate drift respecting the true alignment. (**f**) further showcases the utility of SBALIGN's learnt drift as a suitable reference process to improve other training methods.

needs to vary wildly in time. In this dataset, $\hat{\mathbb{P}}_0$ and $\hat{\mathbb{P}}_1$ are both bi-modal distributions, each supported on two of the four extremes of an imaginary T-shaped area. We target alignments that connect the two arms of the T as well as the top cloud with the bottom one. We succeed in learning them with SBALIGN (Fig. 2e) but unsurprisingly fail when using the baseline FBSB (Fig. 2d) with a Brownian motion prior.

In this case, however, attempts at designing a better reference drift for FBSB must take into account the additional constraint that the horizontal and vertical particle trajectories intersect (see Fig. 2e), i.e., they cross the same area at times t_h and t_v (with $t_h > t_v$). This implies that the drift b_t, which initially points downwards (when $t < t_v$), should swiftly turn rightwards (for $t > t_h$). Setting imprecise values for one of t_h and t_v when defining custom reference drifts for classical SBs would hence not lead to the desired result and, worse, would actively disturb the flow of the other particle group.

As described in § 3.2, in presence of hard-to-capture requirements on the reference drift, the use of SBALIGN offers a remarkably easy and efficient way of learning a parameterization of it. For instance, when using the drift obtained by SBALIGN as reference drift for the computation of the SB baseline (FBSB), we find the desired alignments (Fig. 2f).

4.2 Cell Differentiation

Biological processes are determined through heterogeneous responses of single cells to external stimuli, i.e., developmental factors or drugs. Understanding and predicting the dynamics of single cells subject to a stimulus is thus crucial to enhance our understanding of health and disease and the focus of this task. Most single-cell high-throughput technologies are destructive assays —i.e., they destroy cells upon measurement— allowing us to only measure *unaligned* snapshots of the evolving cell population. Recent methods address this limitation by proposing (lower-throughput) technologies that keep cells alive after transcriptome profiling (Chen et al., 2022b) or that genetically tag cells to obtain a clonal trace upon cell division (Weinreb et al., 2020).

Dataset. To showcase SBALIGN's ability to make use of such (partial) alignments when inferring cell differentiation processes, we take advantage of the genetic barcoding system developed by Weinreb et al. (2020). With a focus on fate determination in hematopoiesis, Weinreb et al. (2020) use expressed DNA barcodes to clonally trace single-cell transcriptomes over time. The dataset consists of two snapshots: the first, recorded on day 2, when most cells are still undifferentiated (see Fig. 3a), and a second, on day 4, comprising many different mature cell types (see Fig. 3b). Using SBALIGN as well as the baseline FSSB, we attempt to reconstruct cell evolution between day 2 and day 4, all while capturing the heterogeneity of emerging cell types. For details on the dataset, see § B.

Baselines. We benchmark SBALIGN against previous DSBs such as (Chen et al., 2022a, FBSB). Beyond, we compare SBALIGN in the setting of learning a prior reference process. Naturally, cell division processes and subsequently the propagation of the barcodes are very noisy. While this genetic annotation provides some form of assignment, it does not capture the full developmental process. We thus test SBALIGN in a setting where it learns a prior from such partial alignments and, plugged into FBSB, is fine-tuned on the full dataset.

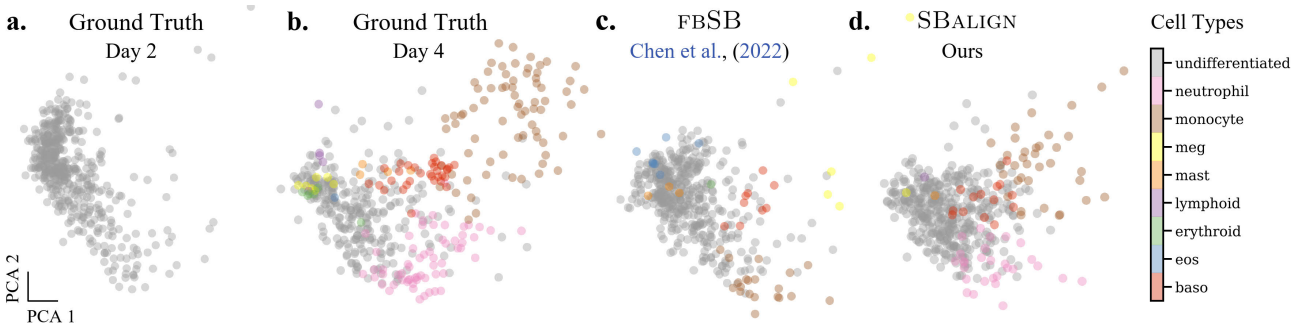

Figure 3: Cell type prediction on the differentiation dataset. All distributions are plotted on the first two principal components. **a-b:** Ground truth cell types on day 2 and day 4 respectively. **c-d:** FBSB and SBALIGN cell type predictions on day 4. SBALIGN is able to better model the underlying differentiation processes and capture the diversity in cell types.

Table 1: **Cell differentiation prediction results.** Means and standard deviations (in parentheses) of distributional metrics ((MMD), W_ϵ), alignment-based metrics (ℓ_2, RMSD), and cell type classification accuracy.

	Cell Differentiation				
Methods	MMD ↓	W_ε ↓	ℓ_2(PS) ↓	RMSD ↓	Class. Acc. ↑
FBSB	1.55e-2 (0.03e-2)	12.50 (0.04)	4.08 (0.04)	9.64e-1 (0.02e-1)	56.2% (0.7%)
FBSB with SBALIGN	5.31e-3 (0.25e-3)	10.54 (0.08)	0.99 (0.12)	9.85e-1 (0.07e-1)	47.0% (1.5%)
SBALIGN	1.07e-2 (0.01e-2)	11.11 (0.02)	1.24 (0.02)	9.21e-1 (0.01e-1)	56.3% (0.7%)

Evaluation metrics. To assess the performance of SBALIGN and the baselines, we monitor several metrics, which include distributional distances, i.e., MMD (Gretton et al., 2012) and W_ϵ (Cuturi, 2013), as well as average (perturbation scores), i.e., ℓ_2(PS) (Bunne et al., 2022a) and RMSD. Moreover, we also train a simple neural network-based classifier to annotate the cell type on day 4 and we report the accuracy of the predicted vs. actual cell type for all the models. See § C.1 for further details.

Results. SBALIGN finds matching between cell states on days 2 and 4 (Fig. 4c, bottom) which resemble the observed ones (Fig. 4a) but also reconstructs the entire evolution path of transcriptomic profiles (Fig. 4c, top). It outperforms the baseline FBSB (Table 1) in all metrics: Remarkably, our method exceeds the performances of the baseline also on distributional metrics and not uniquely on alignment-based ones. We also leverage SBALIGN predictions to recover the type of cells at the end of the differentiation process (Fig. 3d). We do that by training a classifier on differentiated cells observed on day 4, and subsequently classify our predictions. While capturing the overall differentiation trend, SBALIGN (as well as FBSB) struggles to isolate rare cell types. Lastly, we employ SBALIGN to learn a prior process from noisy alignments based on genetic barcode annotations. When using this reference process within FBSB, we learn an SB which compensates for inaccuracies stemming from the stochastic nature of cell division and barcode redistribution and achieves better scores on distributional metrics (see Tab. 1). Further results can be found in § A.

4.3 Conformational Changes in Proteins

Proteins are dynamic, flexible biomolecules that undergo conformational changes upon interactions with other biomolecules (e.g. in docking). Understanding and predicting these conformational changes is a crucial step for the tasks of protein engineering and drug design, and is the focus of this task. More formally, given the 3D structure of the protein in the unbound state, we wish to predict the 3D structure of the protein in the bound state. While it is possible to frame this problem as a (*conditional*) point cloud translation, an approach using Schrödinger bridges is more natural since it leverages the dynamic and flexible nature of proteins, and accounts for the underlying stochasticity in the conformational change process.

Dataset. The task of modeling conformational changes starting from a given protein structure is largely unexplored, mainly due to the lack of high-quality large datasets. Here we utilize the recently proposed D3PM dataset (Peng et al., 2022) that provides protein structures before (*apo*) and after (*holo*) binding, covering various types of protein motions. The dataset was generated by filtering examples from the Protein Data Bank (PDB) corresponding to the same protein but bound to different biomolecules, with additional quality control criteria. For the scope of this work, we only focus on protein pairs where the provided Root Mean Square Deviation (RMSD) of the Cα carbon atoms between unbound and bound 3D structures is > 3.0Å, resulting in 2370 examples.

For each pair of structures, we first identify common residues, and compute the RMSD between Cα carbon atoms of the common residues after superimposing them using the

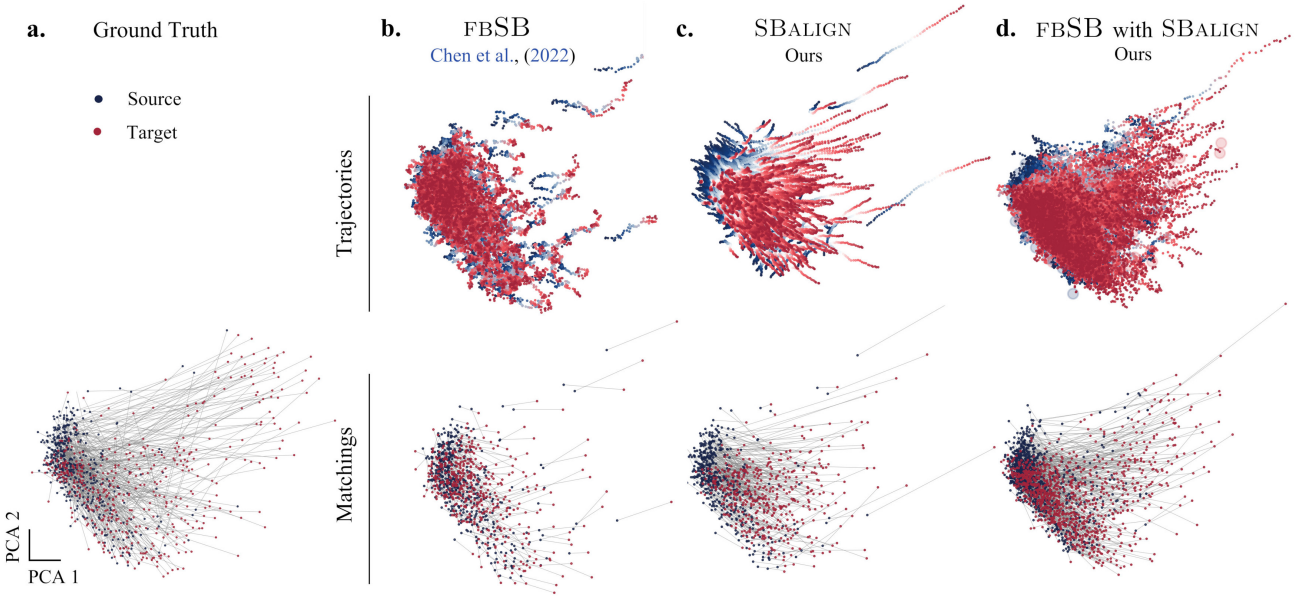

Figure 4: Cell differentiation trajectories based on (**a**) the ground truth and (**b-d**) learned drifts. SBALIGN is able to learn an appropriate drift underlying the true differentiation process while respecting the alignment. (**d**) Using the learned drift from SBALIGN as a reference process helps improve the drift learned by other training methods.

Kabsch (Kabsch, 1976) algorithm, and only accept the structure if the computed Cα RMSD is within a certain margin of the provided Cα RMSD. The rationale behind this step is to only retain examples where we can reconstruct the RMSD values provided with the dataset. The above preprocessing steps give us a dataset with 1591 examples, which is then divided into a train/valid/test split of 1291/150/150 examples respectively. More details in § B.3.

Baselines. Since the goal of the task is to predict 3D structures, our model must satisfy the relevant SE(3) symmetries of rotation and translations. To this end, we evaluate SBALIGN against the EGNN model (Satorras et al., 2021), which satisfies the SE(3) symmetries and is a popular architecture used in many point-cloud transformation tasks (Satorras et al., 2021; Hoogeboom et al., 2022).

Results. To evaluate our model, we report (Table 2) summary statistics of the RMSD between the Cα carbon atoms of the predicted structure and the ground truth, and the fraction of predictions with RMSD values $< 2.0, 5.0$ and 10.0Å. SBALIGN outperforms EGNN by a large margin and is able to predict almost 70% examples with an RMSD<5Å. One of the drawbacks attributed to diffusion models is their slow sampling speed, owing to multiple function calls to a neural network. Remarkably, our model is able to achieve impressive performance with just 10 steps of simulation. We leave it to future work to explore the tradeoff between sampling speed and quality of the predicted conformations.

Future outlook. In this section, we presented a proof of concept application of SBALIGN for modelling conformational changes in proteins. A combination of SBALIGN with more recent methods for rigid-protein docking (Ketata et al., 2023) can provide a complete solution for the protein docking problem, which we leave to future work.

Table 2: **Conformational changes results.** RMSD between predicted and true structures in the bound state. First term in the parentheses refers to number of poses sampled, and the second term refers to the simulation steps for the trajectory.

	D3PM Test Set					
	RMSD (Å)			% RMSD(Å)		
Methods	Median	Mean	Std	%<2	%<5	%<10
EGNN	19.99	21.37	8.21	1	1	3
SBALIGN (10, 10)	3.80	4.98	3.95	0	69	93
SBALIGN (10, 100)	3.81	5.02	3.96	0	70	93

5 Conclusion

In this paper, we propose a new framework to tackle the interpolation task with aligned data via diffusion Schrödinger bridges. Our central contribution is a novel algorithmic framework derived from Schrödinger bridge theory and Doob's h-transform. Via a combination of the two notions, we derive novel loss functions which, unlike prior methods, do not rely on the iterative proportional fitting procedure and are hence numerically stable. We verify our proposed algorithm on various synthetic and real-world tasks and demonstrate noticeable improvement over the previous state-of-the-art, thereby substantiating the claim that data alignment is a highly relevant feature warranting further research.

ACKNOWLEDGEMENTS

This publication was supported by the NCCR Catalysis (grant number 180544), a National Centre of Competence in Research funded by the Swiss National Science Foundation as well as the European Union's Horizon 2020 research and innovation programme 826121. We thank Caroline Uhler for introducing us to the dataset by Weinreb et al. (2020), which was instrumental in this research.

REFERENCES

Bradbury, J., Frostig, R., Hawkins, P., Johnson, M. J., Leary, C., Maclaurin, D., Necula, G., Paszke, A., VanderPlas, J., Wanderman-Milne, S., and Zhang, Q. JAX: composable transformations of Python+NumPy programs, 2018. URL http://github.com/google/jax.

Bunne, C., Stark, S. G., Gut, G., del Castillo, J. S., Lehmann, K.-V., Pelkmans, L., Krause, A., and Ratsch, G. Learning Single-Cell Perturbation Responses using Neural Optimal Transport. *bioRxiv*, 2021.

Bunne, C., Krause, A., and Cuturi, M. Supervised Training of Conditional Monge Maps. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022a.

Bunne, C., Meng-Papaxanthos, L., Krause, A., and Cuturi, M. Proximal Optimal Transport Modeling of Population Dynamics. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 25, 2022b.

Bunne, C., Hsieh, Y.-P., Cuturi, M., and Krause, A. The Schrödinger Bridge between Gaussian Measures has a Closed Form. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2023.

Chen, T., Liu, G.-H., and Theodorou, E. A. Likelihood Training of Schrödinger Bridge using Forward-Backward SDEs Theory. In *International Conference on Learning Representations (ICLR)*, 2022a.

Chen, W., Guillaume-Gentil, O., Rainer, P. Y., Gäbelein, C. G., Saelens, W., Gardeux, V., Klaeger, A., Dainese, R., Zachara, M., Zambelli, T., et al. Live-seq enables temporal transcriptomic recording of single cells. *Nature*, 608, 2022b.

Chen, Y., Georgiou, T. T., and Pavon, M. Optimal Transport in Systems and Control. *Annual Review of Control, Robotics, and Autonomous Systems*, 4, 2021a.

Chen, Y., Georgiou, T. T., and Pavon, M. Stochastic Control Liaisons: Richard Sinkhorn Meets Gaspard Monge on a Schrödinger Bridge. *SIAM Review*, 63(2), 2021b.

Corso, G., Stärk, H., Jing, B., Barzilay, R., and Jaakkola, T. Diffusion Steps, Twists, and Turns for Molecular Docking. In *International Conference on Learning Representations (ICLR)*, 2023.

Cuturi, M. Sinkhorn Distances: Lightspeed Computation of Optimal Transport. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 26, 2013.

Cuturi, M., Meng-Papaxanthos, L., Tian, Y., Bunne, C., Davis, G., and Teboul, O. Optimal Transport Tools (OTT): A JAX Toolbox for all things Wasserstein. *arXiv Preprint arXiv:2201.12324*, 2022.

De Bortoli, V., Thornton, J., Heng, J., and Doucet, A. Diffusion Schrödinger Bridge with Applications to Score-Based Generative Modeling. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, 2021.

Doob, J. *Classical Potential Theory and Its Probabilistic Counterpart*, volume 549. Springer, 1984.

Fisher, M., Nocedal, J., Trémolet, Y., and Wright, S. J. Data assimilation in weather forecasting: a case study in PDE-constrained optimization. *Optimization and Engineering*, 10(3), 2009.

Fortet, R. Résolution d'un systeme d'équations de M. Schrödinger. *J. Math. Pure Appl. IX*, 1, 1940.

Geiger, M. and Smidt, T. e3nn: Euclidean neural networks. *arXiv preprint arXiv:2207.09453*, 2022.

Gretton, A., Borgwardt, K. M., Rasch, M. J., Schölkopf, B., and Smola, A. A Kernel Two-Sample Test. *Journal of Machine Learning Research*, 13, 2012.

Heng, J., De Bortoli, V., Doucet, A., and Thornton, J. Simulating diffusion bridges with score matching. *arXiv preprint arXiv:2111.07243*, 2021.

Ho, J., Jain, A., and Abbeel, P. Denoising Diffusion Probabilistic Models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Holdijk, L., Du, Y., Hooft, F., Jaini, P., Ensing, B., and Welling, M. Path Integral Stochastic Optimal Control for Sampling Transition Paths. *arXiv preprint arXiv:2207.02149*, 2022.

Hoogeboom, E., Satorras, V. G., Vignac, C., and Welling, M. Equivariant diffusion for molecule generation in 3d. In *International Conference on Machine Learning*, pp. 8867–8887. PMLR, 2022.

Kabsch, W. A solution for the best rotation to relate two sets of vectors. *Acta Crystallographica Section A: Crystal Physics, Diffraction, Theoretical and General Crystallography*, 32(5), 1976.

Ketata, M. A., Laue, C., Mammadov, R., Stärk, H., Wu, M., Corso, G., Marquet, C., Barzilay, R., and Jaakkola, T. S. Diffdock-pp: Rigid protein-protein docking with diffusion models. *arXiv preprint arXiv:2304.03889*, 2023.

Kullback, S. Probability densities with given marginals. *The Annals of Mathematical Statistics*, 39(4):1236–1243, 1968.

Léonard, C. A survey of the Schrödinger problem and some of its connections with optimal transport. *arXiv preprint arXiv:1308.0215*, 2013.

Liaw, R., Liang, E., Nishihara, R., Moritz, P., Gonzalez, J. E., and Stoica, I. Tune: A Research Platform for Distributed Model Selection and Training. *arXiv preprint arXiv:1807.05118*, 2018.

Liu, G.-H., Chen, T., So, O., and Theodorou, E. Deep Generalized Schrödinger Bridge. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022a.

Liu, G.-H., Chen, T., So, O., and Theodorou, E. A. Deep Generalized Schrödinger Bridge. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022b.

Liu, X., Wu, L., Ye, M., and Liu, q. Learning Diffusion Bridges on Constrained Domains. *International Conference on Learning Representations (ICLR)*, 2023.

Lotfollahi, M., Wolf, F. A., and Theis, F. J. scGen predicts single-cell perturbation responses. *Nature Methods*, 16 (8), 2019.

Mansuy, R. and Yor, M. *Aspects of Brownian motion*. Springer Science & Business Media, 2008.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Peluchetti, S. Diffusion bridge mixture transports, schrödinger bridge problems and generative modeling, 2023.

Peng, C., Zhang, X., Xu, Z., Chen, Z., Yang, Y., Cai, T., and Zhu, W. D3pm: a comprehensive database for protein motions ranging from residue to domain. *BMC bioinformatics*, 23(1):1–11, 2022.

Peyré, G. and Cuturi, M. Computational Optimal Transport. *Foundations and Trends in Machine Learning*, 11(5-6), 2019.

Rogers, L. C. G. and Williams, D. *Diffusions, Markov Processes and Martingales: Volume 2, Itô Calculus*, volume 2. Cambridge University Press, 2000.

Satorras, V. G., Hoogeboom, E., and Welling, M. E(n)-equivariant graph neural networks. *arXiv preprint arXiv:2102.09844*, 2021.

Schauer, M., van der Meulen, F., and van Zanten, H. Guided proposals for simulating multi-dimensional diffusion bridges. *Bernoulli*, 23(4A), 2017.

Schiebinger, G., Shu, J., Tabaka, M., Cleary, B., Subramanian, V., Solomon, A., Gould, J., Liu, S., Lin, S., Berube, P., et al. Optimal-Transport Analysis of Single-Cell Gene Expression Identifies Developmental Trajectories in Reprogramming. *Cell*, 176(4), 2019.

Schrödinger, E. *Über die Umkehrung der Naturgesetze*. Verlag der Akademie der Wissenschaften in Kommission bei Walter De Gruyter u. Company, 1931.

Song, Y. and Ermon, S. Generative Modeling by Estimating Gradients of the Data Distribution. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. Score-Based Generative Modeling through Stochastic Differential Equations. In *International Conference on Learning Representations (ICLR)*, volume 9, 2021.

Stathias, V., Jermakowicz, A. M., Maloof, M. E., Forlin, M., Walters, W., Suter, R. K., Durante, M. A., Williams, S. L., Harbour, J. W., Volmar, C.-H., et al. Drug and disease signature integration identifies synergistic combinations in glioblastoma. *Nature Communications*, 9(1), 2018.

Thomas, N., Smidt, T., Kearnes, S., Yang, L., Li, L., Kohlhoff, K., and Riley, P. Tensor field networks: Rotation-and translation-equivariant neural networks for 3d point clouds. *arXiv preprint arXiv:1802.08219*, 2018.

Tong, A., Huang, J., Wolf, G., Van Dijk, D., and Krishnaswamy, S. TrajectoryNet: A Dynamic Optimal Transport Network for Modeling Cellular Dynamics. In *International Conference on Machine Learning (ICML)*, 2020.

Tong, A., Malkin, N., Huguet, G., Zhang, Y., Rector-Brooks, J., Fatras, K., Wolf, G., and Bengio, Y. Conditional Flow Matching: Simulation-Free Dynamic Optimal Transport. *arXiv preprint arXiv:2302.00482*, 2023.

Tsaban, T., Varga, J. K., Avraham, O., Ben-Aharon, Z., Khramushin, A., and Schueler-Furman, O. Harnessing protein folding neural networks for peptide-protein docking. *Nature Communications*, 13(1):176, 2022.

Vargas, F., Thodoroff, P., Lawrence, N. D., and Lamacraft, A. Solving Schrödinger Bridges via Maximum Likelihood. *Entropy*, 23(9), 2021.

Weinreb, C., Rodriguez-Fraticelli, A., Camargo, F. D., and Klein, A. M. Lineage tracing on transcriptional landscapes links state to fate during differentiation. *Science*, 367, 2020.

Wolf, F. A., Angerer, P., and Theis, F. J. SCANPY: large-scale single-cell gene expression data analysis. *Genome Biology*, 19(1), 2018.

Xu, M., Yu, L., Song, Y., Shi, C., Ermon, S., and Tang, J. Geodiff: A geometric diffusion model for molecular conformation generation. In *International Conference on Learning Representations*, 2022.

Zhang, Q. and Chen, Y. Path Integral Sampler: A Stochastic Control Approach For Sampling. In *International Conference on Learning Representations (ICLR)*, 2022.

Aligned Diffusion Schrödinger Bridges (Supplementary Material)

A Additional Results

A.1 Variance Reduction

In this paragraph, we elaborate on the need to parametrize also Doob's h_t function, along with the drift b_t. Introducing m^ϕ removes the need to evaluate (11) which is difficult to approximate in practice on high-dimensional spaces. This equation amounts, in fact, to a Gaussian Kernel Density Estimation of the conditional probability $\mathbb{P}(X_1 = \mathbf{x}_1 | X_t = \mathbf{x})$ along (unconditional) paths obtained from (5). Faithful approximations of (11) would, therefore, require:

- good-quality paths, which are scarce at the beginning of training when the drift b_t^θ has not yet been learned;
- exponentially many trajectories (in the dimension of the state space);
- that points x_1 (obtained from conditional trajectories, Eq. 6) be reasonably close to x_1 (obtained from unconditional trajectories, Eq. 5);

Even if all the above conditions were satisfied, the quantity $h_t(x) = \mathbb{P}(X_1 = \mathbf{x}_1 | X_t = \mathbf{x})$ would still be challenging to directly manipulate. It is, in fact, much smaller at earlier times t (see Table 3), since knowledge of the far past has a weaker influence on the location X_1 of particles at time $t = 1$. Precision errors at $t \approx 0$ would then be amplified when computing the score of h_t (i.e, $\nabla \log h_t$) –which appears in the loss (8)– and accumulate over timesteps, eventually leading trajectories astray. By directly parameterizing the score, we instead sidestep this problem. The magnitude of $m_t^\phi \approx \nabla \log h_t$ can, in fact, be more easily controlled and regularized.

	Time t						
	0	**0.15**	**0.30**	**0.45**	**0.60**	**0.75**	**0.90**
Mean $h_{t,\tau}$ value	2.92e-14	4.03e-13	2.54e-11	1.72e-09	1.47e-07	2.66e-05	8.53e-3

Table 3: Average $h_{t,\tau}$ values along paths, at different timesteps. $\mathbb{P}(X_1 = \mathbf{x}_1 | X_t = \mathbf{x})$ ranges over 11 orders of magnitude across the time interval and is smallest when $t \approx 0$.

B Datasets

B.1 Synthetic Datasets

In the following, we provide further insights and experimental results in order to access the performance of SBALIGN in comparison with different baselines and across tasks of various nature. For each dataset, we describe in detail its origin as well as preprocessing and featurization steps.

Moon dataset. The `moon` toy dataset (Fig. 5a) is generated by first sampling $\hat{\mathbb{P}}_1$ and then applying a clockwise rotation of 233° around the origin to obtain $\hat{\mathbb{P}}_0$. The points on the two semi-circumferences supporting $\hat{\mathbb{P}}_1$ are initially placed equally-spaced along each semi-circumference and then moved by applying additive Gaussian noise to both coordinates. While classic generative models will choose the shortest path and connect ends of both moons closest in Euclidean distance, only methods equipped with additional knowledge or insight on the intended alignment will be able to solve this task.

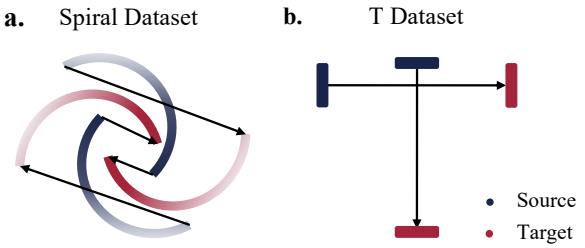

Figure 5: Initial (*blue*) and final (*red*) marginals for the two toy datasets **(a)** moon and **(b)** T, together with arrows indicating a few alignments

T dataset. This toy dataset (Fig. 5b) is generated by placing an equal amount of samples at each of the four extremes of a T-shaped area having ratio between x and y dimensions equal to 51/55. If run with a Brownian prior, classical SBs also fail on this dataset because they produce swapped pairings: i.e., they match the left (*resp.* top) point cloud with the bottom (*resp.* right) one. At the same time, though, this dataset prevents reference drifts with simple analytical forms (such as spatially-symmetric or time-constant functions) from fixing classical SBs runs. It therefore illustrates the need for general, plug-and-play methods capable of generating approximate reference drifts to use in the computation of classical SBs.

B.2 Cell Differentiation Datasets

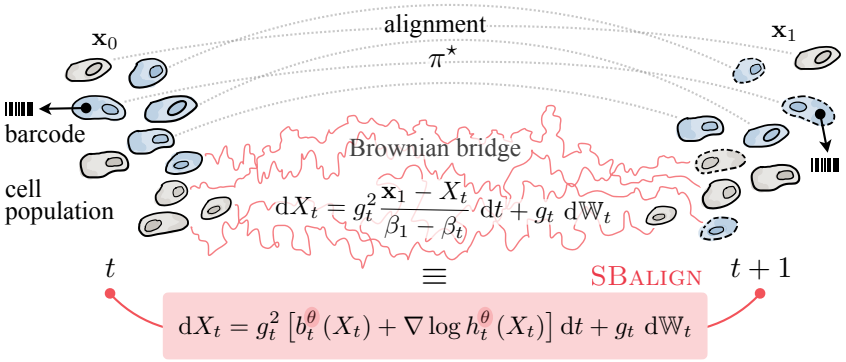

Figure 6: Overview of SBALIGN in the setting of cell differentiation with the goal of learning the evolutionary process that morphs a population from its stat at t to $t+1$. Through genetic tagging (i.e., barcodes) we are able to trace progenitor cells at time point t into their descendants $t+1$. This provides us with an alignment between populations at consecutive time steps. Our goal is then to recover a stochastic trajectory from $\mathbf{x}_0$ to $\mathbf{x}_1$. To achieve this, we connect the characterization of a SDE conditioned on $\mathbf{x}_0$ and $\mathbf{x}_1$ (utilizing the Doob's h-*transform*) with that of a Brownian bridge between $\mathbf{x}_0$ and $\mathbf{x}_1$ (classical Schrödinger bridge theory), leading to a simpler training procedure with lower variance and strong empirical results.

Dataset description. We obtain the datapoints used in our cell differentiation task from the dataset generated by (Weinreb et al., 2020), which contains 130861 observations/cells. We follow the preprocessing steps in Bunne et al. (2021) and use the Python package `scanpy` (Wolf et al., 2018). After processing, each observation records the level of expression of 1622 different highly-variable genes as well as the following meta information per cell:

- a `timestamp`, expressed in days and taking values in $\{2, 4, 6\}$;
- a `barcode`, which is a short DNA sequence that allows tracing the identity of cells and their lineage by means of single-cell sequencing readouts;
- an additional `annotation`, which describes the current differentiation fate of the cell.

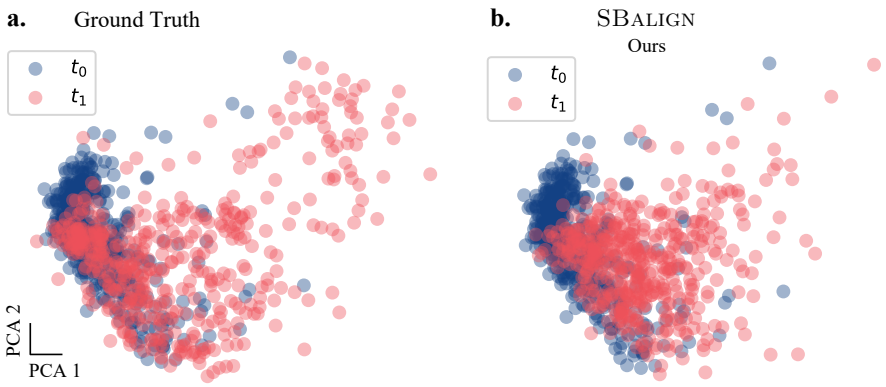

Figure 7: Distribution of the cell population (i.e., marginals) at time $t = t_0$ and $t = t_1$ for (**a**) the ground truth, and (**b**) SBALIGN, after projection along their first two principal components.

Dataset preparation. We only retain cells with barcodes that appear both on days 2 and 4, taking care of excluding cells that are already differentiated on day 2. We construct matchings by pairing cells measured at two different times but which share the barcode. Additionally, we filter cells to make sure that no one appears in more than one pair. To reduce the very high dimensionality of these datapoints, we perform a PCA projection down to 50 components.

We end up with a total of 4702 pairs of cells, which we partition into train, validation, and test sets according to the split 80%/10%/10%.

B.3 Conformational Changes

Dataset description. For the task of predicting protein conformational changes, we utilize the D3PM dataset. The dataset consists of both unbound and bound structures for 4330 proteins, under different types of protein motions. The PDB IDs were downloaded from `https://www.d3pharma.com/D3PM/`. For the PDB IDs making up the dataset, we download the corresponding (.cif) files from the Protein Data Bank.

Dataset preparation. For the scope of this work, we only focus on protein structure pairs, where the provided RMSD between the Cα carbon atoms is > 3Å, amounting to 2370 examples in the D3PM dataset. For each pair of structures, we first identify common residues, and compute the RMSD between Cα carbon atoms of the common residues after superimposing them using the Kabsch (Kabsch, 1976) algorithm, and only accept the structure if the computed Cα RMSD is within a certain margin of the provided Cα RMSD. The rationale behind this step was to only retain examples where we could reconstruct the RMSD values provided with the dataset. Common residues are identified through a combination of residue position and name. This step is however prone to experimental errors, and we leave it to future steps to improve the common residue identification step (using potentially, a combination of common subsequences and/or residue positions).

After applying the above preprocessing steps, we obtain a dataset with 1591 examples, which is then split into a train/valid/test split of 1291/150/150 examples respectively. The structures used in training and inference are the Kabsch superimposed versions, therefore ensuring that the Brownian bridges are sampled between the unbound and bound states of the proteins, and no artifacts are introduced by 3D rotations and translations, which do not contribute to conformational changes.

Featurization. Following standard practice and for memory and computational efficiency, we only use the Cα coordinates of the residues to represent our protein structures instead of the full-atom structures. For each amino acid residue, we compute the following features: a one hot encoding of the amino acid identity f_e of size 23, hydrophobicity $f_h \in [-4.5, 4.5]$, volume $f_v \in [60.1, 227.8]$, the charge $f_c \in \{-1, 0, 1\}$, polarity $f_p \in \{0, 1\}$, and whether the amino acid residue is a hydrogen bond donor $f_d \in \{0, 1\}$ or acceptor $f_a \in \{0, 1\}$. The hydropathy and volume features are expanded into a radial basis with interval sizes 0.1 and 10 respectively. To equip the model with a notion of time, we use a sinusoidal embedding of time $\phi(t)$ of embedding dimensionality 32. These are concatenated to the amino acid features to form our input features for the amino acid residues. The edge features consist of a radial basis expansion of the distances between the residues. We also compute the spherical harmonics of the edge vectors between the residues, which is used in the tensor product message passing layers.

Position at t. For any time t, we sample the positions of the Cα atoms using the Brownian Bridge - given the coordinates $\mathbf{x}_0$ at $t = 0$ and the coordinates $\mathbf{x}_1$ at $t = 1$ with a Brownian bridge between $\mathbf{x}_0$ and $\mathbf{x}_1$, we know that $x_t \sim \mathcal{N}(x_t; (1-t)\mathbf{x}_0 + t\mathbf{x}_1, t(1-t))$.

C Experimental Details

In the following, we provide further experimental details on the chosen evaluation metrics, network architectures, and hyperparameters.

C.1 Evaluation Metrics

C.1.1 Cell Differentiation

For fairness of comparison between our method and the baseline (FBSB) —which only works at the level of distribution of cells— we also consider three evaluation metrics (i.e., W_ε, MMD and ℓ_2) that capture the similarity between the end marginal $\hat{\mathbb{P}}_1$ and our prediction $\pi_1^\star$, irrespective of matchings.

In what follows, we denote with $\hat{\nu}$ the predicted end marginal $\pi_1^\star$ —i.e., the predicted status of cells at day 4— and with ν the distribution of observed transcriptomes.

Wasserstein-2 distance. We measure accuracy of the predicted target population $\hat{\nu}$ to the observed target population ν using the entropy-regularized Wasserstein distance (Cuturi, 2013) provided in the `OTT` library (Bradbury et al., 2018; Cuturi et al., 2022) defined as

(C.1)

where $H(\mathbf{P}) := -\sum_{ij} \mathbf{P}_{ij}(\log \mathbf{P}_{ij} - 1)$ and the polytope $U(\hat{\nu}, \nu)$ is the set of $n \times m$ matrices $\{\mathbf{P} \in \mathbb{R}_+^{n\times m}, \mathbf{P}\mathbf{1}_m = \hat{\nu}, \mathbf{P}^\top \mathbf{1}_n = \nu\}$.

Maximum mean discrepancy. Kernel maximum mean discrepancy (Gretton et al., 2012) is another metric to measure distances between distributions, i.e., in our case between predicted population $\hat{\nu}$ and observed one ν. Given two random variables x and y with distributions $\hat{\nu}$ and ν, and a kernel function ω, Gretton et al. (2012) define the squared MMD as:

$$\text{MMD}(\hat{\nu}, \nu; \omega) = \mathbb{E}_{x,x'}[\omega(x, x')] + \mathbb{E}_{y,y'}[\omega(y, y')] - 2\mathbb{E}_{x,y}[\omega(x, y)].$$

We report an unbiased estimate of $\text{MMD}(\hat{\nu}, \nu)$, in which the expectations are evaluated by averages over the population particles in each set. We utilize the RBF kernel, and as is usually done, report the MMD as an average over the length scales: $2, 1, 0.5, 0.1, 0.01, 0.005$.

Perturbation signature ℓ_2. A common method to quantify the effect of a perturbation on a population is to compute its perturbation signature (Stathias et al., 2018, (PS)), computed via the difference in means between the distribution of perturbed states and control states of each feature, e.g., here individual genes. ℓ_2(PS) then refers to the ℓ_2-distance between the perturbation signatures computed on the observed and predicted distributions, ν and $\hat{\nu}$. The ℓ_2(PS) is defined as

$$\text{PS}(\nu, \mu) = \frac{1}{m}\sum_{y_i \in \nu} y_i - \frac{1}{n}\sum_{x_i \in \mu} x_i,$$

where n is the size of the unperturbed and m of the perturbed population. We report the ℓ_2 distance between the observed signature $\text{PS}(\nu, \mu)$ and the predicted signature $\text{PS}(\hat{\nu}, \mu)$, which is equivalent to simply computing the difference in the means between the observed and predicted distributions.

RMSD. To measure the quality of matchings sampled from SBALIGN $(\hat{x}_0^i, \hat{x}_1^i)$ —compared to the observed ones (x_0^i, x_1^i)— we compute:

$$\text{RMSD}(\{x_1^i\}^n, \{\hat{x}_1^i\}^n) = \sqrt{\frac{1}{n}\sum_{i=1}^n \|x_1^i - \hat{x}_1^i\|^2} \tag{C.2}$$

which, when squared, represents the mean of the square norm of the differences between predicted and observed statuses of the cells on day 4.

Cell type classification accuracy. We assess the quality of SBALIGN trajectories by trying to predict the differentiation fate of cells, starting from (our compressed representation of) their transcriptome. For this, we train a simple MLP-based classifier on observed cells and use it on the last time-frame of trajectories sampled from SBALIGN to infer the differentiation of cells on day 4. We use the classifier `MLPClassifier` offered by the library `scikit-learn` with the following parameters:

- 2 hidden layers, each with a hidden dimension of 50,
- the logistic function as non-linearity
- ℓ_2 norm, regularization with coefficient 0.1.

We report the subset accuracy of the predictions on the *test* set, measured as the number of labels (i.e., cell types) coinciding with the ground truth.

C.2 Network Architectures

C.2.1 Cell Differentiation and Synthetic Datasets

We parameterize both $b^\theta(t, X_t)$ and $m^\phi(t, b_t, X_t)$ using a model composed of:

1. **`x_enc`**: 3-layer MLP performing the expansion of spatial coordinates (or drift) into hidden states (of dimension 64 to 256);
2. **`t_enc`**: sinusoidal embedding of time (on 64 to 256 dimensions), followed by a two layer MLP;
3. **`mlp`**: 3-layer MLP which maps the concatenation of embedded spatial and temporal information (output of modules 1 and 2 above) to drift magnitude values along each dimension.

After every linear layer (except the last one), we apply a non-linearity and dropout (level 0.1). In all the experiments, we set the diffusivity function $g(t)$ in (5) to a constant g, which is optimized (see § C.3).

C.2.2 Protein Conformational Changes

As our architecture $b_t^\theta(X_t)$ suitable for approximating the true drift b_t, we construct a graph neural networks with tensor-product message passing layers using `e3nn` (Thomas et al., 2018; Geiger & Smidt, 2022). To build the graph, we consider a maximum of 40 neighbors –located within a radius of 40Å for each residue. The model is SE(3) equivariant and receives node and edge features capturing relevant residue properties, and distance embeddings.

For the baseline EGNN model, we consider the variant proposed in (Xu et al., 2022), owing to its strong performance on the molecule conformer generation task.

C.3 Hyperparameters

In the following, we will provide an overview of the selected hyperparameters as well as chosen training procedures.

C.3.1 Synthetic Tasks

We perform hyper-parameter optimization using the Python package `ray.tune` (Liaw et al., 2018) on:

- **activation**, chosen among `leaky_relu`, `relu`, `selu` and `silu` as implemented in the Python library `PyTorch` (Paszke et al., 2019). We find `selu` to achieve marginally better performance on toy datasets.
- **g**, the value of the diffusivity constant, chosen among $\{1, 2, 5, 10\}$. We find $g = 1$ to yield optimal results.

C.3.2 Cell Differentiation

We perform hyper-parameter optimization using the Python package `ray.tune` (Liaw et al., 2018) on:

- **activation**, chosen among `leaky_relu`, `relu`, `selu`, and `silu` as implemented in the Python library `PyTorch` (Paszke et al., 2019). We observe that `silu` brings noticeable performance improvements on the cell differentiation dataset.
- **g**, the value of the diffusivity constant, chosen among $\{0.01, 0.1, 0.8, 1, 1.2, 2, 5\}$. We find $g = 1$ to yield optimal results.

C.3.3 Protein Conformational Changes

We use `AdamW` as our optimizer with a initial learning rate of 0.001, and training batch size of 2. For each protein pair, we sample 10 timepoints in every epoch, so the model sees realizations from different timepoints of the corresponding Brownian Bridge. This was done to improve the training speed. We use a regularization strength of 1.0 for m^{ϕ} for all t. Inference on the validation set using training is carried out using the exponential moving average of parameters, and the moving average is updated every optimization step with a decay rate of 0.9. The model training is set to a maximum of 1000 epochs but training is typically stopped after 200 epochs beyond which no improvements in the validation metrics are observed.

Our model has 0.54M parameters and is trained for 200 epochs. After every epoch, we simulate trajectories on the validation set using our model and compute the mean RMSD. The best model selected using this procedure is used for inference on the test set. The baseline EGNN model has 0.76M parameters and is trained for 1000 epochs.

D Reproducibility

Code utilized in this publication can be found at `https://github.com/vsomnath/aligned_diffusion_bridges`, with a mirror at `https://github.com/IBM/aligned_diffusion_bridges`.