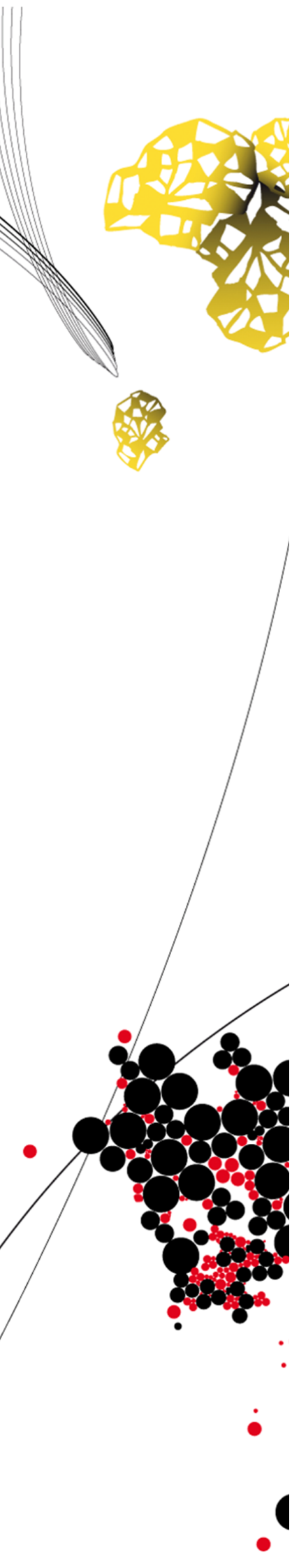


UNIVERSITY OF TWENTE.

Faculty of Electrical Engineering, Mathematics & Computer Science

Optimal transport regularization for implicit neural representations of cell shape sequences

Ties Martens

MSc. Thesis - Applied Mathematics
July 2025

Supervisors:
prof. dr. Christoph Brune
Sven Dummer MSc.

Graduation Committee:
prof. dr. Christoph Brune
Sven Dummer MSc.
dr. Antonios Antoniadis

Specialization:
Mathematics of Data Science (MDS)

Chair:
Mathematics of Imaging and AI (MIA)

Abstract

The study of living-cell sequences at the microscopic level is critical for advancing our understanding of cell dynamics and interactions. This knowledge is particularly valuable in fields such as disease monitoring and drug development, where it can provide insights into phenomena like cancer progression and the behavior of newly developed treatments. Despite its importance, high-quality 4D living-cell data remains sparse. Implicit neural representations (INRs) offer a promising solution by enhancing the temporal resolution of the cell shape sequences. However, they often suffer from overfitting to training data. This thesis investigates the application of optimal transport (OT)-based regularization to enhance 3D time-lapse INRs. We propose two regularization methods: one minimizes the Sinkhorn distance between consecutive time points, and the other guides the INR using OT-based interpolations at intermediate time points. Experiments on synthetic datasets demonstrate that both methods improve temporal interpolation, reduce output variability, and enhance visual coherence, particularly in modeling bifurcating structures. Nevertheless, there remains room for improvement, especially if the data contains abrupt deformations and high-frequency motion. Additionally, in terms of latent space generalization, the proposed methods did not yield significant benefits. Overall, this work highlights the potential of OT-based methods to improve temporal consistency in 3D time-lapse INRs, especially in structured or moderately complex settings.

Keywords: implicit neural representations; optimal transport; cell shape modeling; regularization; latent space

Acknowledgements

The completion of this thesis has been both challenging and rewarding, and I am grateful to all who have contributed to it. In particular, I would like to thank my supervisors Prof. Dr. Christoph Brune and Sven Dummer, MSc for providing me with their continuous support throughout the project. I have developed both academically and personally thanks to their guidance.
Additionally, I want to thank Dr. Jelmer Wolterink for mentoring me throughout the MSc in Applied Mathematics. His optimism and enthusiasm were truly inspiring. I also thank Dr. Gautam Pai for his guidance on optimal transport and Dr. Antonios Antoniadis for kindly agreeing to be a member of my graduation committee. I am grateful to all of my friends for the laughs, moral support and being the best distraction when I needed one. A particular thank you to Wei-Ting and Mei with whom I have often collaborated and who have inspired me to keep learning and improving.
A heartfelt thanks goes to my girlfriend Thea, for her unconditional support for over 6 years now. I am sure she will make a great doctor someday.
Finally, I want to thank my family—and especially my parents—for their unwavering emotional and practical support. Throughout my studies, they have always encouraged me to keep going, and without them, I would not have been able to achieve what I have today.

Contents

Chapter 1

Introduction

Living-cell dynamics and interactions have been an important part of medical research for a long time. Understanding cellular behavior is particularly valuable in fields such as disease monitoring and drug development, where it can provide insights into phenomena like cancer progression or the effects of newly developed treatments [1, 2, 3]. This understanding largely depends on 3D microscopy images, which are able to capture time-dependent changes in the shapes and interactions of living cells [4, 5].

However, acquiring high-quality 3D time-lapse imaging data of living cells presents substantial challenges, as high-intensity illumination can damage biological samples or reduce image quality. These challenges are two-fold. First, the quantity of available cell data is limited, as preparing suitable biological samples and acquiring high-resolution images is both time-consuming and resource-intensive. Second, for the datasets that do exist, the temporal resolution is often insufficient, potentially missing critical dynamic processes in cellular behavior.

To help mitigate the latter issue, temporal interpolation techniques can be used. More precisely, by constructing new data at intermediate time points, temporal density can be improved without the need for additional imaging. Consequently, these methods reduce the need for excessive imaging and therefore have the potential to enhance overall data quality [6, 7, 8].

Multiple approaches for temporal interpolation of cell data exist. More conventional approaches perform interpolation directly in the image domain, this is called video frame interpolation (VFI). This is a subfield of computer vision that aims to generate intermediate frames between existing ones. For cellular data, the primary approaches usually involve the use of convolutional neural networks (CNNs) [9, 10, 11, 12, 13]. However, these methods often produce inaccurate predictions or introduce artifacts, particularly when applied to datasets with complex motion or occlusions [14].

Alternatively, temporal interpolation can be done using shapes instead of images. This relies on shape representation methods. Traditionally, shape representation methods use discrete forms, such as 3D point clouds [15]. A shape is then described by a collection of 3D points in space, meaning that the topology of a shape can not be derived. Another approach is to train 2D manifolds [16], but this often results in shapes that are not closed and requires further techniques like a spherical parameterization [17] to solve this problem. Furthermore, both approaches require a large amount of memory and computation power for high resolution representations.

More recently, implicit neural representations (INRs) have been used to represent shapes in a resolution-independent way, by fitting a continuous function that describes the shape. Examples include, signed distance functions (SDF) [18], where the value of the function is equal to the shortest distance to the shape's surface and occupancy functions [19] where the value of the function is simply an indicator of whether a point is inside or outside of the shape. By training a neural network to approximate such a function, you get a shape representation that can be evaluated at any point in space, and uses little storage (just the network weights) compared to for example mesh-based methods where a single shape is stored by a large number of small surfaces.
Furthermore, an INR can represent multiple shapes at once by conditioning the neural network on an initially randomized vector called a "latent code". Each shape in the ground truth dataset is then rep-

resented by a distinct latent code and the neural network can be interpreted to describe a distribution of shapes. Taking this idea even further, instead of having a function describe a single shape, you can make the function time-dependent. In that case each latent code describes a sequence of shapes over time, examples include [20, 21].

Using an INR to represent a cell-shape sequence, temporal interpolation can be done at arbitrary resolution by just evaluating the learned function at the chosen point in time [22]. Therefore, accurate INRs of cell shapes have the potential to artificially increase the temporal resolution of microscopy data without the need for additional imaging. However, due to the limited number of time frames that cell shape sequence datasets usually have, the trained INR is prone to exhibit bias towards these time frames. This is problematic when we keep in mind that the goal is to create an accurate model of the cell at all points in time.

The usual approach to deal with bias in machine learning and deep learning is by introducing regularization to the model. Temporal regularization methods for INRs feature temporal total variation loss [23] or penalizing the temporal derivatives [24, 25]. However, these approaches lack structural depth. They treat time as a flat axis and ignore the underlying geometry of the data, often penalizing meaningful or structured transformations. As a result, they offer limited capacity regularize temporal behavior in a principled way.

To address this, more structured methods have been proposed. A popular approach involves modeling the temporal evolution of shapes through learned 3D deformation fields [21, 26, 27]. These methods represent shape changes over time as continuous spatial transformations, enabling smooth and coherent motion modeling. However, they often require careful regularization to avoid implausible deformations, and may struggle with shape sequences involving topological changes or occlusions.

Another principled approach is to use optimal transport (OT) as a reference for temporal interpolation. OT provides a mathematically grounded way to define smooth transitions between complex shapes by treating them as distributions and computing minimal-cost flows between them. This perspective has proven effective in a variety of shape analysis and interpolation contexts, offering geodesic paths through shape space that are both structurally consistent and temporally smooth [28, 29, 30, 31]. These properties make OT a promising tool for temporal regularization in neural representations of 3D shapes, particularly when available time frames are limited. By encouraging smooth, energy-efficient evolution between observed shapes, OT-based regularization could help mitigate overfitting to discrete time points and improve the continuity of the learned temporal dynamics.

1.1 Contributions

In this thesis, we investigate how optimal transport based regularization can be applied to implicit neural representations that represent living-cell sequences. The contributions of this thesis are as follows:

- We introduce two novel regularization methods based on OT theory to improve INRs for modeling 3D time-lapse living-cell sequences. The first method minimizes the Sinkhorn distance between consecutive time steps, while the second uses OT-based predictions at intermediate points to promote temporal consistency.
- Temporal interpolation tasks reveal that the proposed regularization methods improve performance at unseen time steps and reduce output variability on the synthetic datasets. In particular, the model offers more sensible interpolation in bifurcating structures.
- We analyze the impact of OT-based regularization on generalization of the latent space by testing the model's ability to reconstruct unseen shape sequences via latent code optimization. In this case the OT-regularization did not significantly improve performance.
- This thesis demonstrates the potential of OT-based regularization methods to enforce temporal coherence and produce more natural and biologically plausible visualizations in 3D time-lapse INRs. While limitations remain in complex real-world data, our findings provide a principled basis for future work on regularization strategies for dynamic implicit representations.

To our knowledge, this thesis presents the first optimal transport based temporal regularization of implicit neural representations.

1.2 Thesis Outline

This thesis is organized as follows: Chapter 2 covers relevant background theory of shape representation methods and optimal transport, Chapter 3 describes how we implement the optimal transport regularization in practice, Chapter 4 displays experiments performed to compare the used methods to the unregularized case, Chapter 5 concludes the findings of this research and finally Chapter 6 discusses limitations of the methods and offer recommendations for further research.

Chapter 2

Background

This chapter provides theoretical background information on implicit neural representations of shapes and optimal transport. A solid understanding of implicit neural representations (INRs) is important for following the technical details in this thesis, as they form the core of the methods discussed. In contrast, the concepts from optimal transport are presented at a higher level, as the detailed mathematics are less critical to the primary contributions.

2.1 Implicit Neural Representations for 3D Shape Learning

Shapes can be implicitly represented using signed distance functions (SDFs), continuous functions that measure the distance of any point in space to the surface of the shape it represents, with a sign indicating whether the point is inside or outside of the shape. The actual shape is represented by the zero level set of such a function. As an example, you can think of a circle with unit radius. The SDF of this shape is given in Figure 2.1.

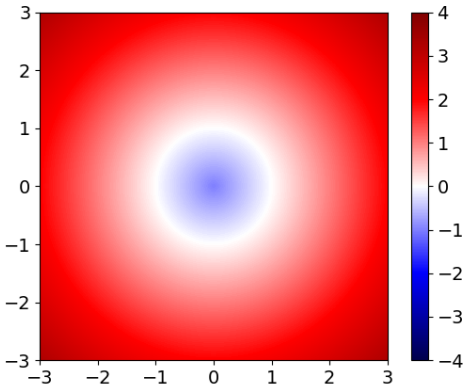


Figure 2.1: The zero level set of the function $\mathrm{SDF}(x, y) = \sqrt{x^2 + y^2} - 1$ is a unit circle with its center at the origin.

In this example, the value of the SDF depends only on the location in 2D space. This concept can easily be extended to 3D shapes, which are in turn represented by the zero-level set of a function that takes three input coordinates, for example $\mathrm{SDF}(x, y, z) = \sqrt{x^2 + y^2 + z^2} - 1$, for which the zero level set is a unit sphere.
Taking this idea another step further, we can introduce a fourth input to the SDF, representing time. Consequently, we have a single function that represents a 3D object, which changes in shape over time. As a straightforward example, imagine a sphere that grows from unit radius at $t = 0$ to having a radius of 2 at $t = 1$ (see Figure 2.2). The SDF representing this object could look something like this:

$$\mathrm{SDF}(x, y, z, t) = \sqrt{x^2 + y^2 + z^2} - (1 + t), \quad 0 \leq t \leq 1. \tag{2.1}$$

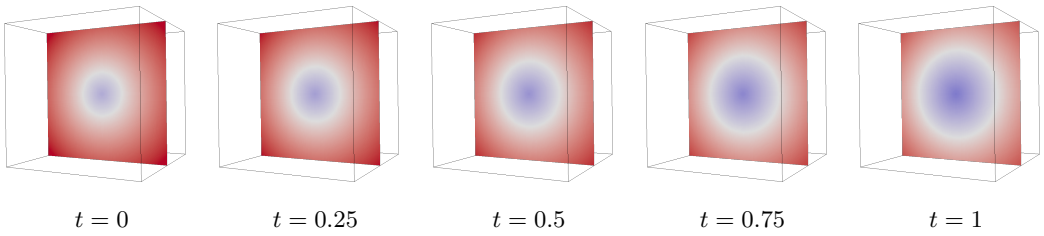


Figure 2.2: Cross-sectional slices of a 3D time-lapse SDF, described by equation 2.1. It represents a sphere growing in radius over time.

Thus, we can use an SDF with four input variables to represent one shape, evolving in time. Now we extend the function once more, such that we can use a single function to represent a whole family of shape sequences. We do this by associating each shape sequence S_i with a latent vector $\mathbf{z_i}$. The latent vector implicitly describes details about the sequence. Let us look at an example once more. Suppose we have two sequences:

- S_1 is a sphere centered at $(0, 1, 0)$. At $t = 0$ its radius is 1 and over time it expands, such that it has a radius of $\frac{3}{2}$ at $t = 1$.
- S_2 is a sphere centered at $(0, -1, 0)$. At $t = 0$ its radius is 1 and over time it shrinks, such that it has a radius of $\frac{1}{2}$ at $t = 1$.

These two spheres can be represented by letting the first element of $\mathbf{z}$, denoted by $z^{(1)}$, control the y-coordinate of the center of the sphere. The growth (or shrinkage) of the radius over time is controlled by $z^{(2)}$. When we include the latent code we then get the following function:

$$\text{SDF}(\mathbf{x}, t, \mathbf{z}) = \sqrt{(x - z^{(1)})^2 + y^2 + z^2} - (1 + z^{(2)}t), \quad 0 \leq t \leq 1, \tag{2.2}$$

where shape sequences S_1 and S_2 are represented by $\mathbf{z}_1 = [1 \quad \frac{1}{2}]^T$ and $\mathbf{z}_2 = [-1 \quad -\frac{1}{2}]^T$ respectively.

Equation 2.2 is capable of representing more than just S_1 and S_2, any other latent code represents another SDF that can be described in this way. Together, all of these latent vectors form the latent space (see Figure 2.3). This latent space allows for meaningful interpolation between any two shape sequences by interpolating their corresponding latent vectors and then applying the resulting vector in equation 2.2.

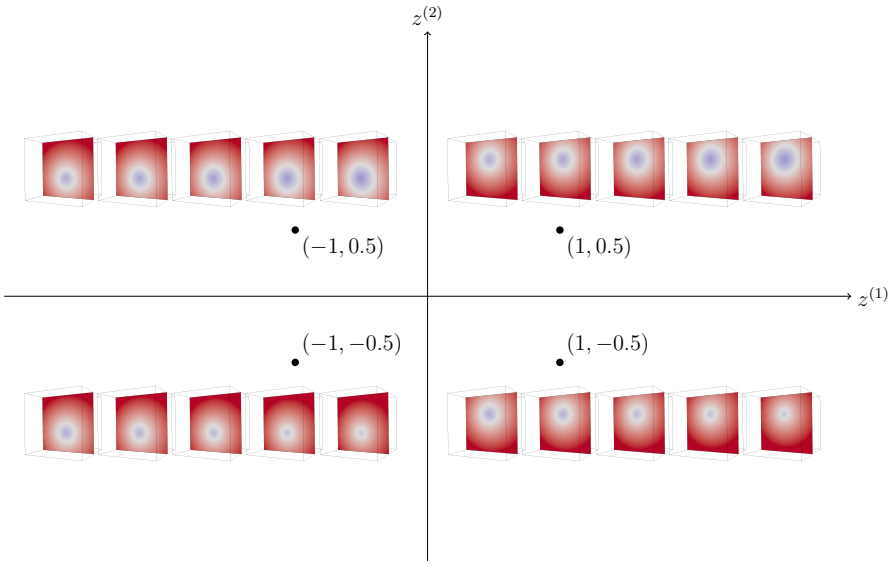


Figure 2.3: Latent space associated with equation 2.2, four points are highlighted and the associated SDF sequence is shown.

Although we have shown that a family of shape sequences can be represented by a single function, in general, finding such a function is infeasible. However, deep neural networks can be used to approximate such a function by learning from ground truth data. A neural network that implicitly represents a shape or scene is called an implicit neural representation (INR).

The earliest implementations of INR were done by Mescheder et al. [19], using occupancy functions and Park et al. [18] using SDFs. On their own, these methods often struggle to capture high frequency details of shapes. In order to deal with this low-frequency bias, usually one of two approaches is taken. With positional encoding, the input coordinates are mapped to a higher dimensional space using sinusoidal functions [32]. With SIREN, the typical ReLU activation functions are replaced by periodic ones [33]. The exact architecture that we use in this thesis is explained in section 3.1.

As mentioned before, instead of using SDFs for shape representation, one could use occupancy functions. These are functions that simply take the value 1 whenever a coordinate is inside the shape, and 0 whenever it is outside. This may seem like a more straightforward idea, but there are a couple of advantages to using SDFs:

- SDFs have more regular gradients, the spatial gradient of an SDF is 1 everywhere in the domain, whereas the gradients of an occupancy function are very small at points far away from the shapes' boundary and very large at points close to the boundary due to the sharp change in the function value. Regular gradients are particularly desirable when training a neural network to approximate this function.
- SDFs capture richer geometric information, improving the network's ability to learn complex, high-frequency shape details.
- SDFs predict an exact surface boundary, wherever the function is equal to zero, whereas occupancy functions have to use some threshold value that is to be determined heuristically.

Thus, SDFs offer clear advantages, even though occupancy functions are more intuitive representations considering ground truth segmentation masks of cell sequences are often in the occupancy function format.

2.2 Optimal Transport

This section provides a brief overview of optimal transport theory, focusing on the concepts most relevant to the developments in this thesis. The material is adapted in part from [34], which also offers a more thorough discussion of the subject.

Optimal transport is a mathematical framework that describes how two distinct distributions in space can be optimally transformed into one another, based on a certain cost function. As an example, we can imagine a pile of sand and a hole with the same volume as the pile. We want to move the sand such that the hole is perfectly filled. In this case, optimal transport describes the most efficient way to accomplish this. In particular, it describes how every individual grain of sand should be moved such that the total effort of moving all sand is minimized. The minimal total effort required to do this task is called the Wasserstein distance.

We mathematically describe the Wasserstein distance using the Kantorovich formulation. Suppose we have two distributions μ and ν in the same space X, where X is the space of probability distributions over domain Ω. A transport plan γ is a distribution on $\Omega \times \Omega$ that describes how mass is moved from μ to ν. Let us denote the set of all valid transport plans by T, which ensure that all mass from μ is moved to ν:

$$\begin{aligned} &\gamma(A \times \Omega) = \mu(A) \\ &\gamma(\Omega \times B) = \nu(B) \\ &\text{for all} \quad A, B \subseteq X \end{aligned}$$

Now suppose we have a cost function $c(\mathbf{x}, \mathbf{y})$ which represents the cost of moving a unit of mass from $\mathbf{x}$ to $\mathbf{y}$. Then, an optimal transport map would be one where the cost of moving all mass from μ to ν is minimal. The total cost of this optimal transport map between μ and ν is called the Wasserstein distance and is mathematically defined by:

$$\mathcal{W}(\mu, \nu) = \min_{\gamma \in T} \int_{\Omega \times \Omega} c(\mathbf{x}, \mathbf{y}) d\gamma(\mathbf{x}, \mathbf{y}). \tag{2.3}$$

The cost function $c(x, y)$ is often chosen to be the p-norm. In this case, we get:

$$\mathcal{W}_p(\mu, \nu) = \min_{\gamma \in T} \int_{\Omega \times \Omega} ||\mathbf{x} - \mathbf{y}||^p d\gamma(\mathbf{x}, \mathbf{y}). \tag{2.4}$$

The Wasserstein distance provides a measure of the similarity between two distributions. However, for arbitrary continuous distributions, a closed-form solution for this distance typically does not exist. Instead, continuous distributions can be approximated by discrete distributions. This is done by creating a regular grid that spans over the domain, evaluating the continuous distribution at each point on the grid and placing a point mass at each grid point proportional to the value. Then, the values are normalized such that the total mass is equal to the total mass of the continuous distribution. Doing this for both distributions still yields a meaningful measure and transforms the problem into a linear program that can be solved efficiently.

The space X, equipped with $\mathcal{W}_p$ is a metric space. Additionally, this metric space is proven to be a geodesic space, which has useful geometrical properties. In a geodesic space, we can interpolate between two points by minimizing the average metric distance from each point to a certain midpoint. When the space consists of distributions, this midpoint distribution provides a meaningful interpolation between the two distributions. More precisely, given distributions μ_0 and μ_1, their midpoint interpolation in the Wasserstein space is given by:

$$\mu_{bar} = \arg\min_{\mu \in X} \left(\mathcal{W}_p(\mu, \mu_0) + \mathcal{W}_p(\mu, \mu_1) \right). \tag{2.5}$$

In other words, we interpolate by taking the midpoint of a geodesic curve connecting distribution μ_0 and μ_1. Instead of finding just the midpoint, we can find other points along the curve as well by adding weights for each distribution:

$$\mu_{bar} = \arg\min_{\mu \in X} \left(\lambda \mathcal{W}_p(\mu, \mu_0) + (1 - \lambda) \mathcal{W}_p(\mu, \mu_1) \right). \tag{2.6}$$

In a similar way, we can find the barycenter between n distributions, weighing each distribution accordingly:

$$\mu_{bar} = \arg\min_{\mu \in X} \sum_{i=1}^{n} \lambda_i \mathcal{W}_p(\mu, \mu_i). \tag{2.7}$$

We demonstrate that this interpolation in the Wasserstein space provides a more meaningful interpolation in the context of shapes than an interpolation in the Euclidean space. Given a circular distribution in the left of the domain and a circular distribution on the right of the domain, sensible interpolations would show the distribution gradually traveling from left to right, not disappearing and reappearing on the other side (see Figure 2.4).

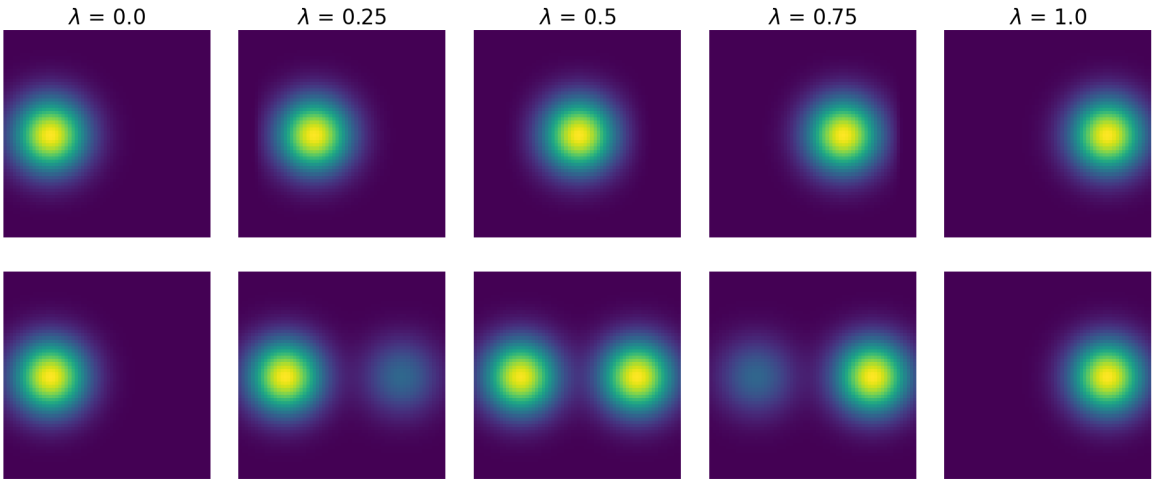


Figure 2.4: Comparison between interpolations of two circular probability distributions in the Wasserstein space (top row) and the Euclidean space (bottom row).

In practice, computing the Wasserstein distance in high-dimensional settings can be computationally expensive. To address this, the Sinkhorn algorithm has become a widely adopted approximation technique. Originally introduced by Sinkhorn and Knopp [35] in the context of matrix scaling and adapted to optimal transport by Cuturi [36], the method introduces an entropic regularization term controlled by a parameter ϵ, often referred to as the blur parameter. This parameter smooths the transport problem and enables efficient iterative updates of a transport matrix. Smaller values of ϵ lead to solutions closer to the true Wasserstein distance but require more iterations, while larger values result in faster convergence at the cost of increased smoothing. This approach provides a fast and scalable approximation to the Wasserstein distance. In this thesis, we use the GeomLoss library [37], which offers highly optimized and GPU-accelerated implementations of the Sinkhorn algorithm.

Chapter 3

Methods

This chapter first introduces the baseline model on which this research is based. Then, it presents two ways of incorporating regularization optimal transport theory.

3.1 Implicit Neural Representations for 3D Shape Sequences

The ultimate goal is to create a model that accurately describes deforming shapes over time. Given N shapes, this can be represented as approaching a family of signed distance functions $\mathcal{S}_t^i : \Omega \to \mathbb{R}$, with spatial domain $\Omega \subset \mathbb{R}^3$ and temporal domain $\tau = [0, 1]$ for $i = 1, ..., N$. This can be expressed as the following minimization problem:

$$\min_{\{f_i(\mathbf{x},t)\}_{i=1,...,N}} \sum_{i=1}^{N} \int_\tau \int_\Omega |\mathcal{S}_t^i(\mathbf{x}) - f_i(\mathbf{x}, t)| d\mathbf{x}dt, \tag{3.1}$$

A practical implementation of this is presented by Wiesner et al. [22]. In particular, $\mathcal{S}_t^i$ is approximated using a multilayer perceptron (MLP) $f_\theta(\mathbf{x}, t, \mathbf{z}_i)$, conditioned on a latent code $\mathbf{z}_i$ which encodes shape characteristics. In general, data is only available at some discrete time points $\mathcal{T} = \{t_1, ..., t_M\} \subset \tau$. Additionally, the integral over the spatial domain is approximated by taking a sample of points $\mathcal{X} = \{\mathbf{x}_1, ..., \mathbf{x}_K\} \subset \Omega$. The loss function associated with reconstruction of the original shape at a certain location $\mathbf{x}$ and time t is then given by:

$$\mathcal{L}_{recon}(\mathcal{S}_t^i(\mathbf{x}), f_\theta(\mathbf{x}, t, \mathbf{z}_i)) = |\mathcal{S}_t^i(\mathbf{x}) - f_\theta(\mathbf{x}, t, \mathbf{z}_i)|. \tag{3.2}$$

To encourage a smooth latent space, regularization of the latent code is included, with parameter σ:

$$\mathcal{L}_{code}(\mathbf{z}_i, \sigma) = \frac{1}{\sigma^2} ||\mathbf{z}_i||_2^2. \tag{3.3}$$

Combining the reconstruction loss and the code regularization loss, the baseline model is trained using the following loss function:

$$\mathcal{L}_{simple} = \sum_{i=1}^{N} \left(\mathcal{L}_{code}(\mathbf{z}_i, \sigma) + \sum_{t \in \mathcal{T}} \sum_{\mathbf{x} \in \mathcal{X}} \mathcal{L}_{recon}(\mathcal{S}_t^i(\mathbf{x}), f_\theta(\mathbf{x}, t, \mathbf{z}_i)) \right). \tag{3.4}$$

To investigate whether the theory from optimal transport can be applied to regularize shape sequence INRs, we introduce an additional loss term that describes how well the model conforms to optimal transport principles. There are several ways to implement this but we limit our focus on two methods. The first method uses the Sinkhorn distance between any two sequential shapes explicitly. The second method uses an OT-based estimation of what the shape should look like at intermediate time points and then measures the difference between this estimation and the model prediction.

3.2 Sinkhorn-based Regularization

The intuition behind this approach is that a cell's shape typically evolves gradually over time, so its form at one moment should be similar to its form shortly afterward. To capture this temporal continuity, we require a meaningful way to quantify differences between 3D shapes. The Wasserstein distance is particularly suitable in this context, as it measures the minimal amount of "work" needed to transform one shape into another. This makes it sensitive not only to where mass is located, but also to how far it must move, reflecting gradual deformations in shape. By using this distance to define a loss function, we can encourage temporal consistency by penalizing large shape changes over short time intervals.

However, to use the Wasserstein distance, the shapes need to be represented as probability distributions. To make the SDF compatible with the Wasserstein distance, we need to transform it into a valid probability distribution. This can be done in three steps:

1. For all points with $f_\theta > 0$, we set the value to zero. These points are outside the shape and should not contribute to the distribution.
2. For points with $f_\theta \leq 0$, we assign a constant positive value. This creates a uniform "mass" over the shape's interior.
3. We then scale the function so that the total integral sums to one, resulting in a valid probability distribution.

This transformation can be implemented using a reversed step function centered at $f_\theta = 0$. Specifically, the function outputs a positive constant when $f_\theta \leq 0$ and zero otherwise.

However, to incorporate the Wasserstein distance into a loss function, it must be computed in a differentiable manner. Since, step functions are not differentiable, a smooth approximation of the reversed step function is required. Several differentiable alternatives exist, among which the reversed sigmoid function

$$\sigma_\beta^-(f_\theta) = \frac{1}{1 + e^{\beta f_\theta}} \tag{3.5}$$

stands out as the most computationally efficient (see Figure 3.1). Using this function and normalizing afterwards, we can map SDFs to probability distributions.

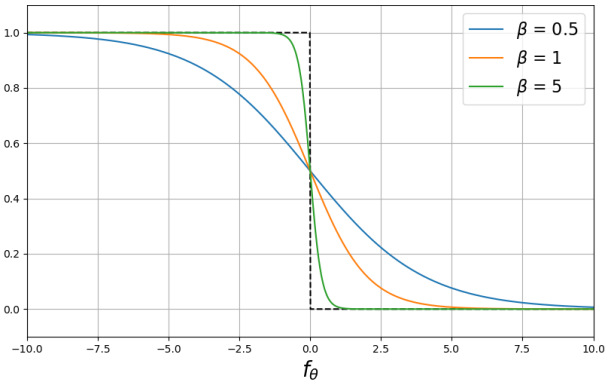


Figure 3.1: The reversed sigmoid function that we use as a differentiable approximation of the reversed step function. The parameter β controls the steepness of the function around $f_\theta = 0$

To encourage efficient mass transport across the entire shape sequence, we evaluate the model at regular time intervals and calculate the Wasserstein distance between any pair of consecutive shapes. Assuming we sample M points in time for each shape, we then get the following loss term:

$$\mathcal{L}_{Wass}(f_\theta(\mathbf{x}, t, \mathbf{z})) = \frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{M-1} \mathcal{W}_p \left(\frac{\sigma_\beta^-(f_\theta(\mathbf{x}, t_j, \mathbf{z}_i))}{\int_\Omega \sigma_\beta^-(f_\theta(\mathbf{x}, t_j, \mathbf{z}_i))d\mathbf{x}}, \frac{\sigma_\beta^-(f_\theta(\mathbf{x}, t_{j+1}, \mathbf{z}_i))}{\int_\Omega \sigma_\beta^-(f_\theta(\mathbf{x}, t_{j+1}, \mathbf{z}_i))d\mathbf{x}} \right), \tag{3.6}$$

where $\mathcal{W}_p(\cdot,\cdot)$ is the Wasserstein distance, equipped with the p-norm distance measure.
While in theory we have access to f_θ at an unlimited spatial resolution, we calculate the Wasserstein distance on a discrete approximation for computational reasons. For this, we approximate the spatially continuous function $f_\theta(\mathbf{x}, t, \mathbf{z})$ by evaluating it on a regular 3D grid, resulting in a discrete function denoted by $\hat{f}_\theta(\mathbf{x}, t, \mathbf{z})$. Even then, calculating the Wasserstein distance is often computationally infeasible and thus we approximate it using the Sinkhorn algorithm. Hence, the loss term that we use is finally given by:

$$\mathcal{L}_{Sink}(\hat{f}_\theta(\mathbf{x}, t, \mathbf{z})) = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{M-1}\mathcal{S}_{p,\epsilon}\left(\frac{\sigma_\beta^-(\hat{f}_\theta(\mathbf{x}, t_j, \mathbf{z}_i))}{\sum_\Omega \sigma_\beta^-(\hat{f}_\theta(\mathbf{x}, t_j, \mathbf{z}_i))}, \frac{\sigma_\beta^-(\hat{f}_\theta(\mathbf{x}, t_{j+1}, \mathbf{z}_i))}{\sum_\Omega \sigma_\beta^-(\hat{f}_\theta(\mathbf{x}, t_{j+1}, \mathbf{z}_i))}\right), \tag{3.7}$$

where $\mathcal{S}_{p,\epsilon}(\cdot,\cdot)$ is the Sinkhorn distance, equipped with the p-norm distance measure and blur parameter ϵ.

During training, $\mathcal{L}_{Sink}$ is calculated each batch and added to the loss function with hyperparameter α to tune the importance given to the optimal transport loss. The 'Sinkhorn regularized' loss function is then given by:

$$\mathcal{L}_{SR} = \mathcal{L}_{simple} + \alpha\mathcal{L}_{Sink}. \tag{3.8}$$

In practice, applying the Sinkhorn regularized loss from the start of training is inefficient. Because the autodecoder is initialized with identical values across all spatiotemporal locations, the Sinkhorn distance between consecutive time steps is initially low. As the model begins to fit the ground truth data, the distance increases—causing the Sinkhorn loss to rise and potentially slowing convergence. This phenomenon essentially slows down the convergence of the model towards the correct solution. Therefore, instead of using Sinkhorn regularization from the start of training, we set $\alpha = 0$ for the first 200 training epochs. Then we use the Sinkhorn loss from the 200th epoch onward to guide the model towards a solution that minimizes transport.

3.3 Barycentric Regularization

This method is based on the idea that, given a shape at two consecutive time points, the intermediate shape should lie close to their corresponding barycenter. This method aims to improve the capability of the model to make sensible predictions at intermediate time points, even when the ground truth data includes two consecutive time points where the shape suddenly changes.

3.3.1 Computing Barycenters

We compute barycenters using a multiscale convolutional Sinkhorn algorithm, where an exponentially decaying blur parameter is used to promote convergence. Since the algorithm operates on nonnegative functions, the SDFs must first be converted to the occupancy function domain. This is achieved using a reversed step function, which maps all nonnegative values to zero (outside the shape) and negative values to one (inside the shape).

The barycenters are then computed in this binary occupancy representation. To convert the result back to the SDF domain, we compute the difference between the distance to the nearest point outside the shape and the distance to the nearest point inside. The result is an SDF that represents the barycenter in the original domain.

Importantly, because these barycenters depend solely on the ground truth data, they can be precomputed and stored prior to training, eliminating runtime computation overhead. In the next section we denote the stored barycenters of shape i by $f_{bary}(\mathbf{x}, t, \mathbf{z}_i)$.

3.3.2 Regularization via Barycenters

To ensure that the model predictions are similar to the barycenters over the entire time domain, we compute barycenters between any two consecutive ground truth shapes and compare to the model's prediction at the corresponding points in time (see Figure 3.2). The comparison is made by evaluating the model on a regular 3D grid and comparing the predicted values to the barycenter values using an L1 loss. If you compute K barycenters between every two consecutive ground truth points, the loss term is as follows:

$$\mathcal{L}_{bary}(\hat{f}_\theta(\mathbf{x},t,\mathbf{z})) = \sum_{i=1}^{N}\sum_{j=1}^{M-1}\sum_{k=1}^{K} ||\hat{f}_\theta(\mathbf{x}, t_j + \frac{k}{K+1}(t_{j+1}-t_j), \mathbf{z}_i) - \hat{f}_{bary}(\mathbf{x}, t_j + \frac{k}{K+1}(t_{j+1}-t_j), \mathbf{z}_i)||_1, \tag{3.9}$$

where $\hat{f}_{bary}$ is the barycenter SDF evaluated on the 3D grid.

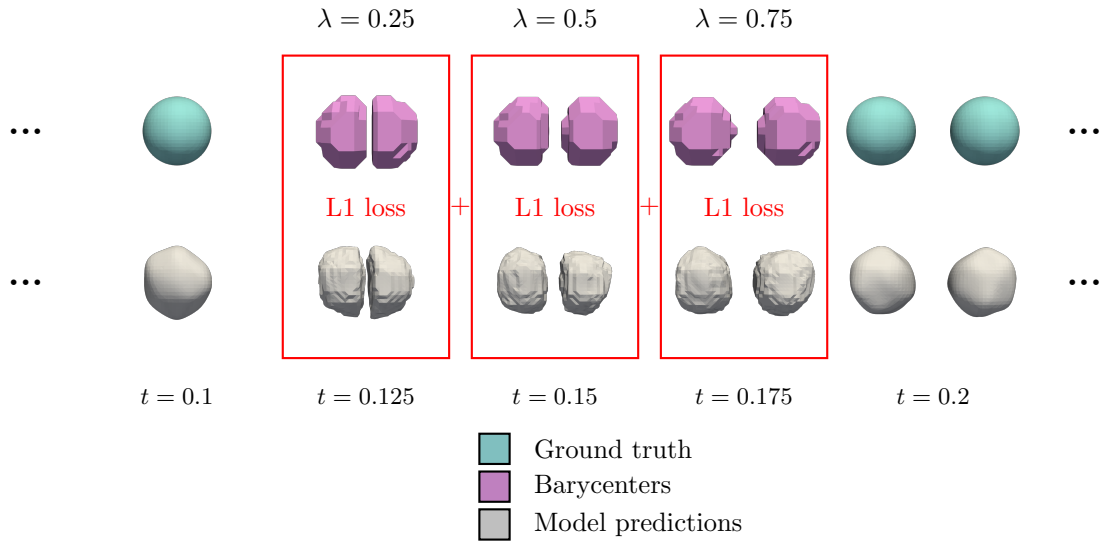


Figure 3.2: Visualization of the barycenter regularization method. Between any two ground truth frames, a number of barycenters is generated before training (in this case $K = 3$). During training, $\mathcal{L}_{bary}$ is calculated by summing the L1 loss between each barycenter and the model's prediction at the corresponding point in time.

Once more, we introduce a hyperparameter γ to regulate how much influence the barycenter loss should have on the total training loss. The barycenter regularized loss function is then given by:

$$\mathcal{L}_{BR} = \mathcal{L}_{simple} + \gamma\mathcal{L}_{bary}. \tag{3.10}$$

Chapter 4

Experiments

In this chapter, we evaluate the effectiveness of the regularization methods introduced previously in guiding the neural network's learning process. The goal of this research is to reduce the bias of the baseline model towards the provided dataset by introducing optimal transport regularization. We begin by evaluating temporal interpolation exploration through an academic problem, with a known solution. Then we investigate the effect of the regularization on modeling bifurcations, and finally we perform experiments on real-world cellular data. Then we investigate the effect of the regularization on the latent space as a whole. All experiments were conducted over 1000 training epochs using a Tesla L40 GPU.

For the Sinkhorn regularization, we evaluated the decoder on an 11x11x11 spatial grid at 30 time points. These values were used to calculate the Sinkhorn loss in equation 3.7. We used $p = 2$ such that the distance measure is the L2 norm, and we used a blur parameter of $\epsilon = 0.05$. The parameter controlling the steepness of the reversed sigmoid function was set to $\beta = 8$, such that the function was sufficiently steep, without resulting in exploding gradients. The factor controlling the amount the Sinkhorn loss contributes to the total loss was set to $\alpha = 0.1$.

For the barycenter regularization, we subsampled each barycenter by a factor 2, in order to reduce memory requirements. Unless stated otherwise, we generated one barycenter between each set of consecutive frames in the ground truth dataset. The factor controlling the amount the barycenter loss contributes to the total loss was set to $\gamma = 0.5$.

4.1 Temporal Interpolation

4.1.1 Expanding and Shrinking Spheres

To assess whether the proposed regularization methods effectively reduce the model's bias toward the training dataset, we conduct a temporal interpolation experiment using a synthetic academic example with SDFs. This controlled setting allows for precise, quantitative evaluation.

The ground truth SDF used in this experiment is introduced in Section 2.1 and defined by Equation 2.2. Specifically, we generate four sequences:

- A sphere centered at $(0, 1, 0)$, starting at radius 1 at $t = 0$ and expanding to a radius of $\frac{3}{2}$ at $t = 1$.
- A sphere centered at $(0, 1, 0)$, starting at radius 1 at $t = 0$ and shrinking to a radius of $\frac{1}{2}$ at $t = 1$.
- A sphere centered at $(0, -1, 0)$, starting at radius 1 at $t = 0$ and expanding to a radius of $\frac{3}{2}$ at $t = 1$.
- A sphere centered at $(0, -1, 0)$, starting at radius 1 at $t = 0$ and shrinking to a radius of $\frac{1}{2}$ at $t = 1$.

We train the model with and without optimal transport-based regularization, using ground truth data sampled from the known SDFs. The model is then evaluated at intermediate time points not included in the training set. Its predictions are compared to the ground truth SDF at those interpolated times, using the Hausdorff distance, Chamfer distance, and Dice score as evaluation metrics. The Chamfer and

Hausdorff distances are calculated on a surface point cloud, which we extract using the Marching Cubes algorithm [38]. The Dice score is calculated using binary voxel volumes, formed by combining all voxels with a negative SDF value. While the Chamfer and Hausdorff distance should be as low as possible, the Dice score should be as close to 1 as possible.

We conduct the experiment under two different training conditions to evaluate the influence of temporal supervision:

- Dense Supervision: The model is trained using five time points per sequence, including the start and end frames.
- Sparse Supervision: The model is trained using only the start and end points of each sequence.

This setup allows us to assess the model's ability to generalize and interpolate temporal dynamics under varying degrees of temporal guidance. The results are shown in Figure 4.1.

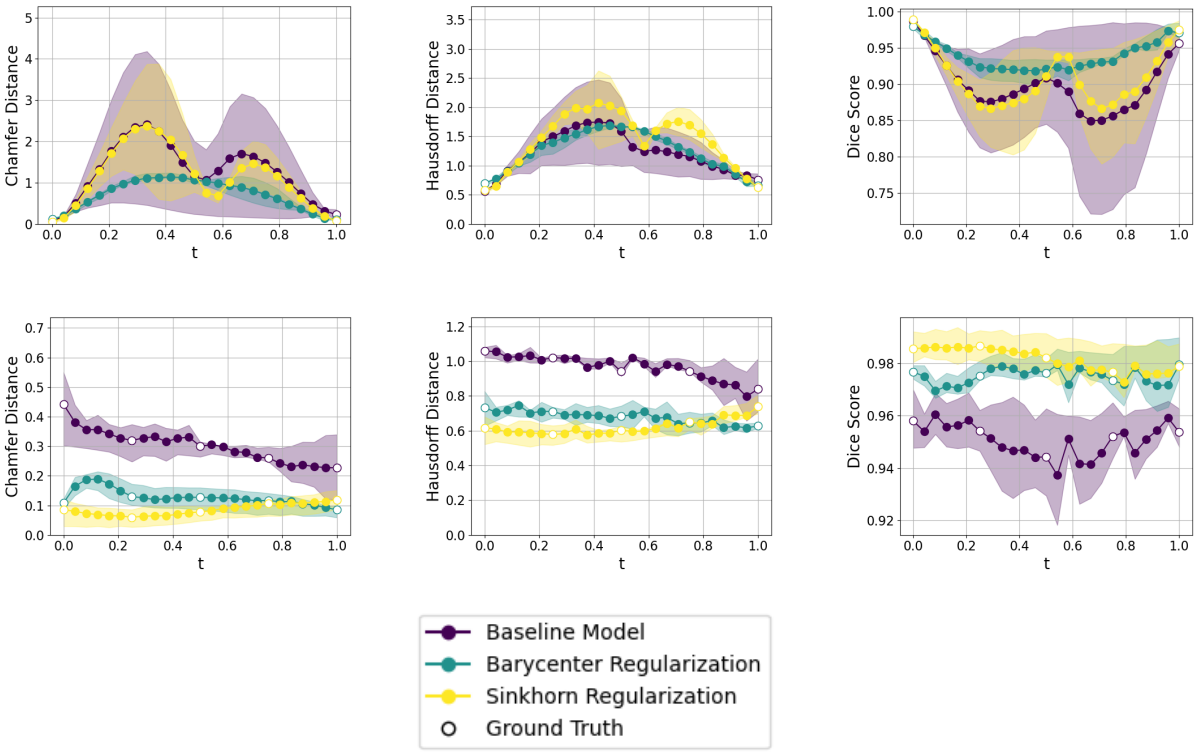


Figure 4.1: Quantitative evaluation of temporal interpolation on the expanding and shrinking spheres dataset. Results are shown for two training conditions: sparse supervision (top row) and dense supervision (bottom row). The lines represent the mean score over the four sequences, and the areas surrounding the line are their respective interquartile ranges (IQRs), provided as a measure of variability. The time points at which ground truth is available to the model are marked with white dots.

In the sparse supervision setting, the baseline model performs well at the first and last time steps, where data is available. However, at the intermediate time points, the performance worsens noticeably, especially as the model moves further away from the supervised steps. In addition, the interquartile range (IQR) is quite wide, showing that the quality of interpolation strongly depends on which training sequence the model is working with. These two issues indicate a bias in the model, which our regularization methods are designed to address.

With Sinkhorn regularization, the average performance is roughly similar to the unregularized model, and in terms of Hausdorff distance, results are slightly worse. However, the variability across all three metrics is clearly reduced, suggesting that this method helps to some extent in reducing the model's bias toward specific sequences.

Finally, barycenter regularization leads to clear improvements in both Chamfer distance and Dice score. While the model still performs best at the time steps where data was provided, the overall average results are better, and the variation in performance is much smaller compared to the unregularized case.

In the dense supervision setting, the bias towards the data is less present, even in the unregularized case. Still, the results of both regularized models is significantly better than the baseline, with the Sinkhorn regularization slightly outperforming the barycenter regularization in this case.

Overall, the regularization methods appear to both improve performance and reduce variability, suggesting that they help mitigate bias in the model. Sinkhorn regularization seems to be more effective in densely supervised settings, whereas barycenter regularization shows benefits in both sparse and dense supervision.

A possible explanation for this difference is as follows: Barycenter regularization introduces explicit structural targets, in the form of intermediate shapes, which provide the model with additional guidance, independent of how much supervision is available. In contrast, Sinkhorn regularization encourages smoother transitions between consecutive time steps by minimizing the optimal transport loss. This approach is more effective when the baseline model already captures the correct structure to some extent, as it refines rather than redefines the model's trajectory.

4.1.2 Bifurcation

The baseline model, trained without any form of regularization, performs poorly in scenarios involving shape bifurcation. To find out if regularization can mitigate this limitation, we constructed a benchmark problem in which a single sphere splits into two identical spheres. The signed distance function (SDF) for this setup is defined using the smooth union of two spheres, each translating at a constant speed of $\frac{3}{2}$ units per time step to the left and right, respectively. The SDFs of the two moving spheres are given by:

$$c_1 = \sqrt{(x - \tfrac{3}{2}t)^2 + y^2 + z^2} - 1,$$
$$c_2 = \sqrt{(x + \tfrac{3}{2}t)^2 + y^2 + z^2} - 1.$$

To prevent sharp edges and large gradients in the combined SDF, we apply exponential smoothing following the method described in [39]. The blended SDF is defined as:

$$SDF(x, y, z, t) = -\frac{1}{k} \ln(e^{-kc_1} + e^{-kc_2}),$$

where k is a blending parameter that controls the smoothness of the transition. In our experiments, we set $k = 5$ to obtain a natural-looking bifurcation. The corresponding surfaces of this synthetic dataset are shown in Figure 4.3 (top row).

Once again, we train the model with and without regularization under different levels of supervision. The evaluation metrics are given in Figure 4.2 and, for the sparse supervision case, surface representations of the predicted shapes are given for a number of time points in Figure 4.3.

Under sparse supervision, the baseline model again demonstrates a satisfactory fit at time points near the beginning and end of the sequence, where ground truth data are available. At intermediate time points, particularly around $t = 0.5$, performance on all three evaluation metrics declines markedly. This observation is supported by the surface representations: the prediction at $t = 0.5$ comprises three shapes of varying sizes, asymmetrically distributed across the spatial domain.

When employing Sinkhorn regularization, the most prominent improvement is observed in the Chamfer distance and Dice score around $t = 0.5$, while the Hausdorff distance remains largely unchanged. Examination of the surface representations reveals that the bifurcation is modeled in a more natural and symmetrical manner compared to the baseline.

With barycenter regularization, using three barycenters in this case to provide additional structure, the metric scores are similar to those obtained using Sinkhorn regularization. However, this approach performs less effectively at earlier time points and better at later ones. The corresponding surface representations also exhibit increased symmetry and a more natural morphology relative to the baseline model.

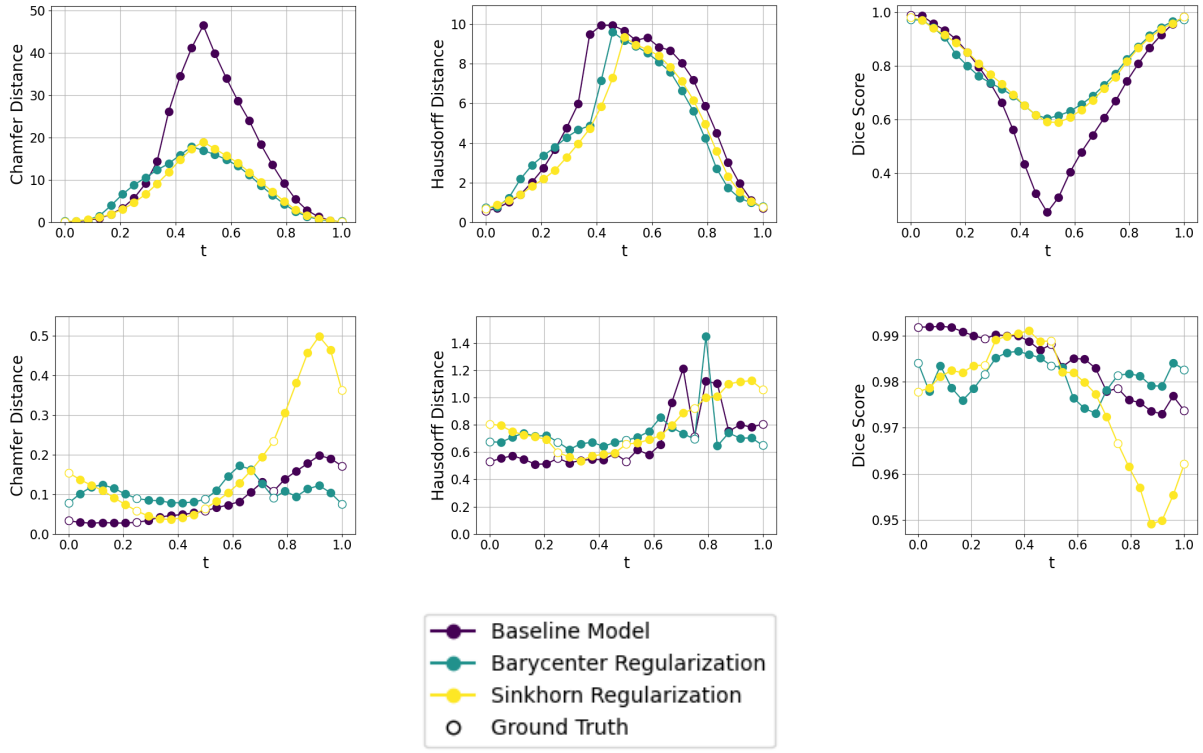


Figure 4.2: Quantitative evaluation of temporal interpolation of the bifurcation dataset. Results are shown for two training conditions: sparse supervision (top row) and dense supervision (bottom row). The time points at which ground truth is available to the model are marked with white dots.

Comparison of the predicted surface representations with the ground truth reveals that all three methods anticipate the bifurcation to occur earlier in time than it does in the reference data. Since the model is provided only with the initial and final frames, this discrepancy should not be interpreted as an objective error, but rather as an alternative modeling of the bifurcation process. Nonetheless, the visualizations suggest that both regularization methods yield predictions that are more symmetrical and natural, while still aligning with the available ground truth.

Under dense supervision, the baseline model achieves strong performance, and bias towards ground truth seems no longer present. In this setting, the regularization methods do not appear to confer additional benefits, and may even slightly degrade overall performance.

4.1.3 C. Elegans Cells

Having demonstrated that regularization methods can improve temporal interpolation performance in a controlled academic setting, we now investigate whether these benefits extend to real-world biological data.

For this experiment, we utilize two C. elegans cell sequences, each comprising 30 time points. At each time point, an SDF is provided at a resolution of 256×256×256 [40]. This particular dataset was selected because it is used as a training sequence in the baseline method's original paper and features a clear bifurcation approximately midway through the sequence, offering an opportunity to validate if the regularization methods improve the behavior around bifurcations for real-world data as well.

To test the interpolation ability of the model with and without regularization we use a subset of time frames from the original dataset for training and evaluate on all 30 time points. Since this dataset has more radical changes between consecutive frames, we choose to use a higher number of frames for training than for the previous two experiments. More precisely, under sparse supervision we use every fourth frame of each sequence, and under dense supervision, we retain every second frame for training.

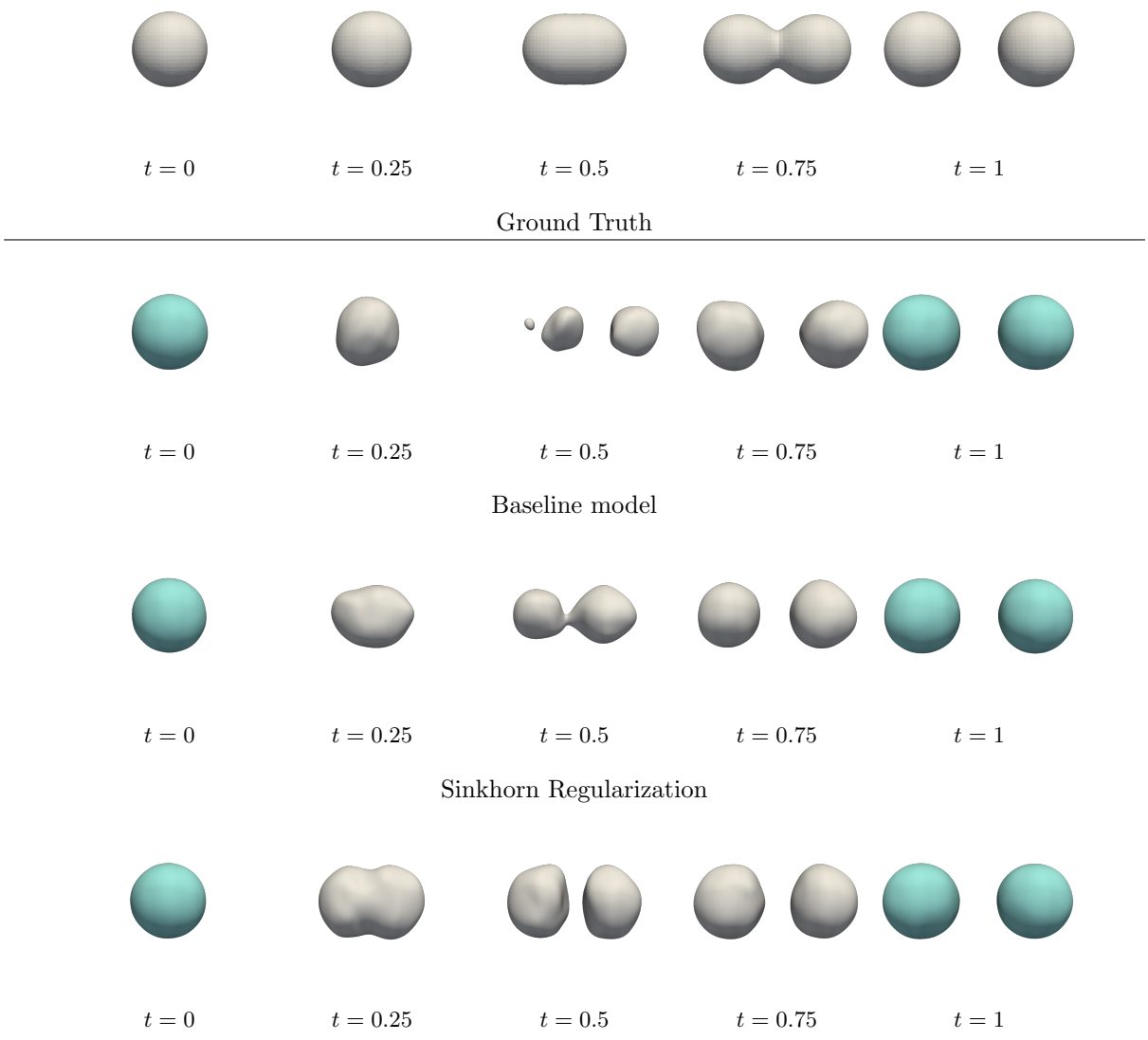


Figure 4.3: Surface representations of the results from training on the bifurcation dataset under sparse supervision with and without our regularization methods. The blue shapes indicate time points where ground truth data is available to the model.

We observe from the metric scores in Figure 4.4 that the model's performance drops significantly around the bifurcation point at $t = 14$, even under dense supervision. The proposed regularization methods do not lead to improvements in these scores. Among them, barycenter regularization produces visually smoother transitions and reduces the formation of artifacts, as shown in Figure 4.5. However, this comes at the cost of reduced fidelity to the ground truth, indicating a trade-off between visual smoothness and accuracy.

A likely explanation for this behavior lies in the nature of the dataset. For most of the sequence, consecutive frames show minimal shape variation. Then, between $t = 12$ and $t = 13$, the shape abruptly shrinks to nearly half its size. Following this, between $t = 14$ and $t = 15$, the cell suddenly bifurcates, without any gradual transition or intermediate signal. This means that the SDF values around the center of the domain are swapped from negative to positive, while the opposite happens at the location of the two new cell shapes. These sudden changes require the SDF to exhibit sharp gradients within just a small subset of its temporal domain, which poses a challenge for the regularization methods used.

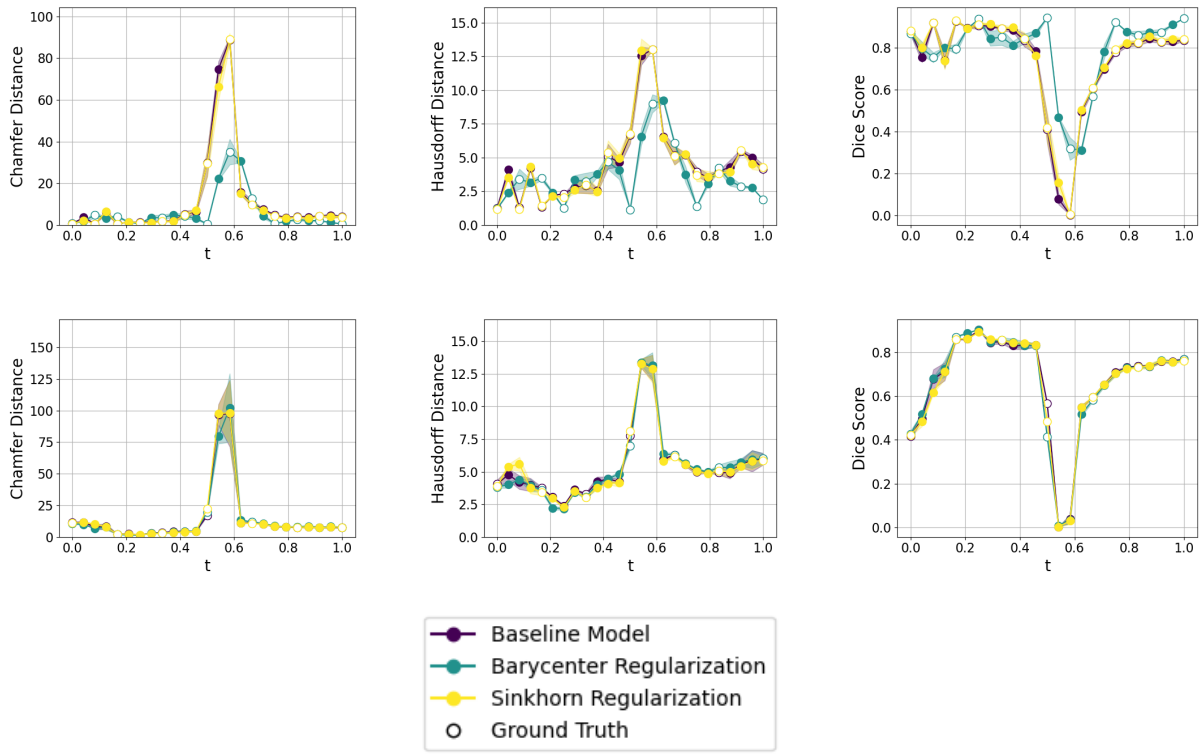
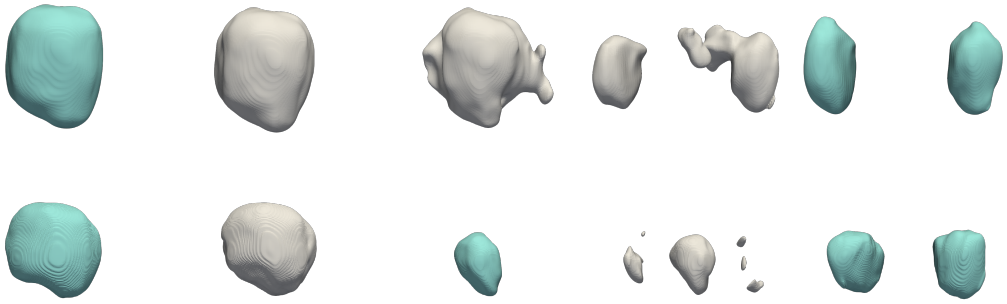


Figure 4.4: Quantitative evaluation of temporal interpolation of the C. Elegans dataset under sparse and dense supervision. The dataset features a bifurcation between time frames 14 and 15.

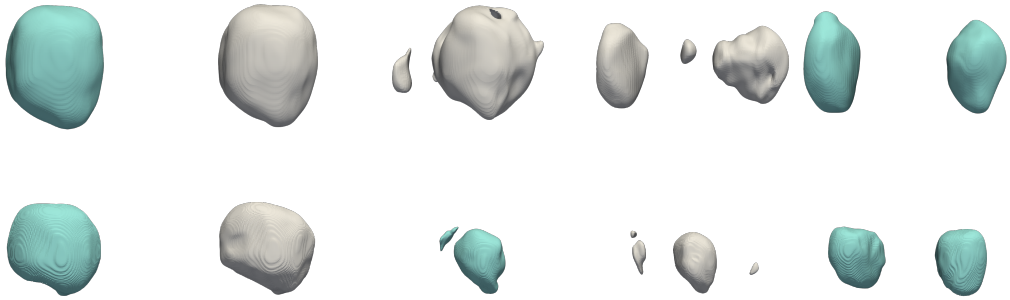
$t = 12$ $t = 13$ $t = 14$ $t = 15$ $t = 16$

Ground Truth



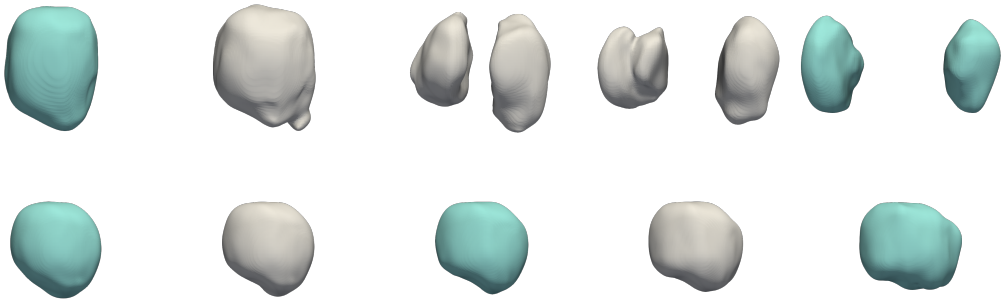
$t = 12$ $t = 13$ $t = 14$ $t = 15$ $t = 16$

Baseline Model



$t = 12$ $t = 13$ $t = 14$ $t = 15$ $t = 16$

Sinkhorn Regularization



$t = 12$ $t = 13$ $t = 14$ $t = 15$ $t = 16$

Barycenter Regularization

Figure 4.5: Surface representations of the results from training on the C. Elegans dataset with and without our regularization methods. This was done under both sparse and dense supervision (first and second row of each method), the blue shapes indicate time points where ground truth data is available to the model.

4.2 Latent Space Exploration

Regularization methods are generally introduced to improve a model's ability to generalize beyond the training data. In this work, however, the proposed methods are specifically designed to promote generalization in the temporal domain, rather than in the latent space. To explore how the model behaves outside of its intended scope, we conduct an additional experiment that investigates generalization in the latent space. While not the primary target of our regularization strategies, this analysis offers insight into the extent and nature of the model's capacity to adapt to unseen latent codes.

This experiment revisits the setup introduced in Section 4.1.1, involving expanding and shrinking spheres. Here, each training sequence is generated using a ground-truth latent code $\mathbf{z}_{\text{gen}} \in \mathbb{R}^2$, which parameterizes Equation 2.2. Four such sequences, each with a different $\mathbf{z}_{\text{gen}}$, are used to train the autodecoder.

The autodecoder itself assigns a learned latent code $\mathbf{z}_{\text{model}} \in \mathbb{R}^d$ to each input sequence, where d is the dimension of the model's internal latent space (in this case $d = 64$). These $\mathbf{z}_{\text{model}}$ codes are optimized during training to minimize reconstruction loss for their respective sequences.

To test generalization, we generate new sequences using unseen values of $\mathbf{z}_{\text{gen}}$ drawn from the same distribution. For each such sequence we determine whether a corresponding $\mathbf{z}_{\text{model}}$ exists, that allows the trained model to accurately reconstruct the sequence, without retraining the network. If such a $\mathbf{z}_{\text{model}}$ can be found for any new $\mathbf{z}_{\text{gen}}$, this would suggest that the latent space learned by the model effectively captures the entire family of sequences, indicating that the model has good generalization ability.

The experiment is setup as follows: we construct a grid of 25 ground truth latent codes (see Figure 4.6) and generate their corresponding sequences using equation 2.2. For each sequence, we optimize the corresponding latent code using the Adam optimizer [41] with learning rate 0.005 for 400 epochs. The loss function that we use is a simple L1 loss between the ground truth SDF and the SDF generated by the model when using the current latent code over 10 different points in time. Each latent code has 64 elements and is initialized according to a Normal distribution $\mathcal{N}(0, 0.01^2)$.

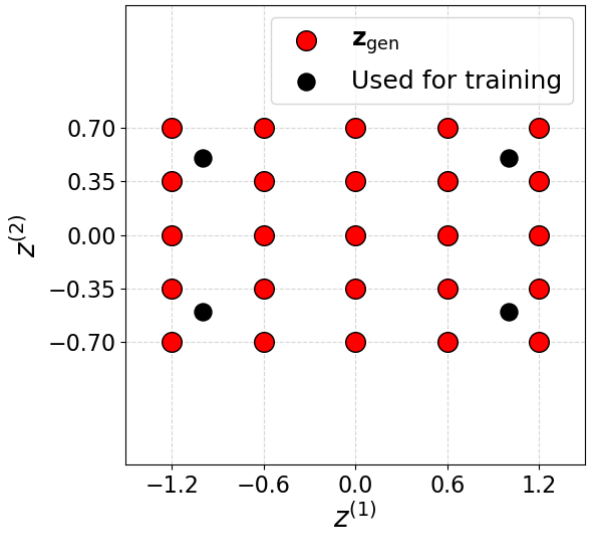


Figure 4.6: The 25 latent codes used for generating sequences are sampled on a grid. The grid is created in such a way that the spheres remain completely inside the autodecoder's spatial domain $[-3, 3]^3$ at any point in time.

We assess the generalizability of the model by computing the same evaluation metrics used in the temporal interpolation experiment, allowing for a direct comparison of performance. A model with good generalizability should exhibit consistent performance across all $\mathbf{z}_{\text{gen}}$ within the domain. The corresponding results are presented in Figure 4.7.

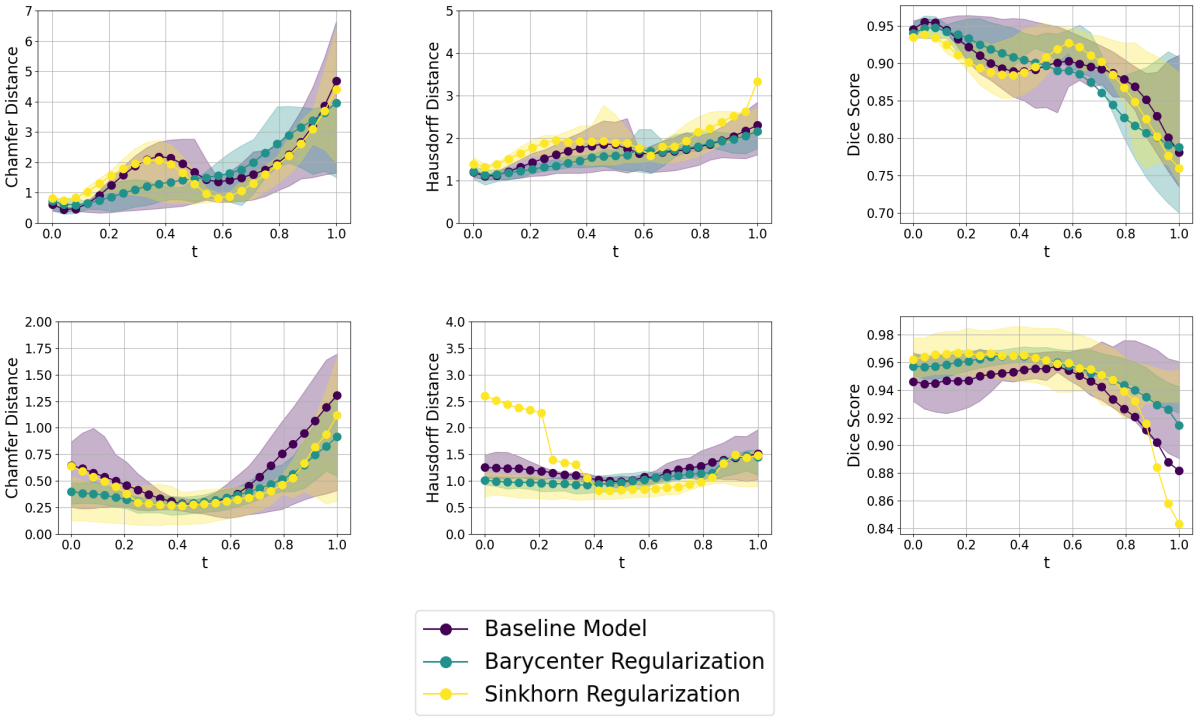


Figure 4.7: Qualitative evaluation of reconstructions for 25 sequences excluded from the training set. The model was trained under sparse supervision (top row) and dense supervision (bottom row). The solid lines denote the mean metric values across the sequences, while the shaded regions indicate the IQRs, providing a measure of variability.

Across most time steps, there is a modest decrease in performance relative to the metrics obtained on the training data, regardless of the presence of regularization. Notably, performance declines more substantially toward the end of the sequences. This trend can be attributed to the greater variability in shape at later time points, where sequences typically reach their maximum or minimum size.

Although the metric scores with barycenter regularization under dense supervision are marginally higher than those of the baseline, this improvement mirrors the pattern observed on the ground truth data. This suggests that while barycenter regularization yields a slightly better-performing model, it does not necessarily enhance the model's generalizability with regard to the latent space.

These results underscore the fact that the regularization methods proposed in this work are not designed to encourage generalization across the latent space, but rather to improve temporal coherence and interpolation. As such, the observed limitations in latent space generalization are not unexpected. Enhancing the model's capacity to generalize to unseen latent codes would likely require alternative regularization strategies that explicitly promote structure or continuity in the latent representation.

Chapter 5

Conclusions

In this research, we explored the potential of optimal transport (OT)-based regularization to enhance 3D time-lapse implicit neural representations (INRs), particularly by addressing their tendency to overfit known data. We proposed two OT-inspired regularization strategies. The first involves directly minimizing the Sinkhorn distance between consecutive time steps, while the second leverages OT-based predictions at intermediate time points to guide the model toward more stable and generalized solutions.

To evaluate the effectiveness of these methods, we conducted temporal interpolation experiments on both synthetic and real-world datasets. On synthetic data, both regularization strategies demonstrated clear benefits, improving performance at intermediate time steps and reducing variability in model outputs. This suggests a mitigation of bias towards the training data under both sparse and dense supervision settings. In particular, the model exhibited more visually coherent results when modeling bifurcating structures, producing more symmetrical and artifact-free representations.

However, performance on real-world data offered no significant improvement. In particular, modeling abrupt deformations and high-frequency changes is challenging, indicating that further refinement of the regularization methods or a complementary approach may be necessary in this case.

We also examined the generalizability of the models by optimizing latent codes to reconstruct previously unseen shape sequences. On the synthetic dataset used, both the baseline model and the models trained with OT-based regularization achieved similar levels of performance in this setting. This suggests that the proposed regularization strategies, while effective for improving temporal generalization, do not have a substantial impact on generalization in the latent space. Improving performance in this regard would likely require different approaches specifically aimed at shaping the latent representation.

Overall, our findings highlight the promise of OT-based regularization in improving the temporal coherence and visual fidelity of 3D time-lapse INRs, particularly in controlled or moderately complex settings. However, challenges persist, especially when dealing with highly dynamic real-world data. This work provides a foundation for future research into principled regularization methods aimed at enhancing spatiotemporal neural representations.

Chapter 6

Future work

This research demonstrates that OT regularization can improve the performance of temporal INRs. However, challenges persist when modeling datasets with high-frequency temporal variations.

One potential direction for addressing this issue is to incorporate positional encoding, as introduced in NeRF by Mildenhall et al. [32]. In that work, input coordinates are mapped to a higher-dimensional space using sinusoidal functions, enabling the network to better capture high-frequency details. Although the current method already employs periodic activation functions to handle high-frequency signals, positional encoding may offer an additional advantage by allowing independent control over the frequency content in the spatial and temporal dimensions. This could enable more flexible and expressive representations, particularly for dynamic scenes with localized high-frequency motion.

Another promising direction is to extend the proposed regularization methods to capture temporal relationships over longer time scales. Currently, the regularization operates locally, between consecutive frames, which may no sufficiently guide the model in scenarios involving high-frequency or long-range temporal dependencies. One possibility is to compute the Sinkhorn loss at multiple temporal resolutions, allowing the model to capture both short-term and long-term mass transport. For the barycenter method, barycenters could be calculated between frames separated by more than just one time step. These adaptations could help the model better capture coherent temporal dynamics, reduce overfitting to local fluctuations and improve robustness in more complex or rapidly changing sequences.

We also observed that the generalizability of the model, particularly with respect to its latent space, was not significantly improved by the regularization methods introduced in this work.

One interesting strategy for addressing this limitation is to adapt the barycenter-based regularization approach. In its current form, barycenters are computed between consecutive frames within a sequence. We propose extending this to compute barycenters across frames from different sequences in the training set. These cross-sequence barycenters could serve as anchors in the latent space, encouraging interpolations produced by the autodecoder to remain close to meaningful intermediate representations. This may promote a smoother and more coherent latent space structure, thereby enhancing generalization across varying shape sequences.

Bibliography

[1] Michael Boutros, Florian Heigwer, and Christina Laufer. "Microscopy-Based High-Content Screening". In: *Cell* 163.6 (Dec. 2015), pp. 1314–1325. ISSN: 10974172. DOI: 10.1016/j.cell.2015.11.007.

[2] Martin Hailstone et al. "CytoCensus, mapping cell identity and division in tissues and organs using machine learning". In: *eLife* 9 (May 2020), pp. 1–31. ISSN: 2050084X. DOI: 10.7554/ELIFE.51085.

[3] Z. Li, M. E. Cvijic, and L. Zhang. "Cellular Imaging in Drug Discovery: Imaging and Informatics for Complex Cell Biology". In: *Comprehensive Medicinal Chemistry III* 2-8 (Jan. 2017), pp. 362–387. DOI: 10.1016/B978-0-12-409547-2.12328-5.

[4] Abolfazl Zargari et al. "DeepSea: An efficient deep learning model for single-cell segmentation and tracking of time-lapse microscopy images". In: *bioRxiv* (Mar. 2022). DOI: 10.1101/2021.03.10.434806.

[5] Carsen Stringer et al. "Cellpose: a generalist algorithm for cellular segmentation". In: *Nature Methods* 18.1 (Jan. 2021), pp. 100–106. ISSN: 15487105. DOI: 10.1038/s41592-020-01018-x.

[6] David Svoboda and Vladimir Ulman. "MitoGen: A framework for generating 3d synthetic time-lapse sequences of cell populations in fluorescence microscopy". In: *IEEE Transactions on Medical Imaging* 36.1 (Jan. 2017), pp. 310–321. ISSN: 1558254X. DOI: 10.1109/TMI.2016.2606545.

[7] Dmitry V. Sorokin et al. "FiloGen: A Model-Based Generator of Synthetic 3-D Time-Lapse Sequences of Single Motile Cells with Growing and Branching Filopodia". In: *IEEE Transactions on Medical Imaging* 37.12 (Dec. 2018), pp. 2630–2641. ISSN: 1558254X. DOI: 10.1109/TMI.2018.2845884.

[8] Joanna W. Pylvänäinen et al. "Live-cell imaging in the deep learning era". In: *Current Opinion in Cell Biology* 85 (Dec. 2023), p. 102271. ISSN: 0955-0674. DOI: 10.1016/J.CEB.2023.102271.

[9] Wenbo Bao et al. "Depth-Aware Video Frame Interpolation". In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* 2019-June (Apr. 2019), pp. 3698–3707. ISSN: 10636919. DOI: 10.1109/CVPR.2019.00382.

[10] Xiaoyu Xiang et al. "Zooming Slow-Mo: Fast and Accurate One-Stage Space-Time Video Super-Resolution". In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (Feb. 2020), pp. 3367–3376. ISSN: 10636919. DOI: 10.1109/CVPR42600.2020.00343.

[11] Nazeli Ter-Petrosyan, Davit Gyulnazaryan, and Varduhi Yeghiazaryan. "Temporal resolution enhancement for cell tracking in microscopy image sequences". In: *Medical Imaging* (Apr. 2025). Ed. by John E. Tomaszewski and Aaron D. Ward, p. 5. DOI: 10.1117/12.3047198.

[12] Rohit Saha et al. "W-Cell-Net: Multi-frame Interpolation of Cellular Microscopy Videos". In: *arXiv preprint arXiv:2005.06684* (2020). DOI: 10.48550/arXiv.2005.06684.

[13] Zejin Wang et al. "DAN: A Deformation-Aware Network for Consecutive Biomedical Image Interpolation". In: *arXiv preprint arXiv:2004.11076* (Apr. 2020). DOI: 10.48550/arXiv.2004.11076.

[14] Martin Priessner et al. "Content-aware frame interpolation (CAFI): deep learning-based temporal super-resolution for fast bioimaging". In: *Nature Methods* 21.2 (Feb. 2024), pp. 322–330. ISSN: 15487105. DOI: 10.1038/s41592-023-02138-w.

[15] Charles R. Qi et al. "PointNet: Deep learning on point sets for 3D classification and segmentation". In: *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017* 2017-January (Nov. 2017), pp. 77–85. DOI: 10.1109/CVPR.2017.16.

[16] Jan Bednarik et al. “Shape Reconstruction by Learning Differentiable Surface Representations”. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (2020), pp. 4715–4724. ISSN: 10636919. DOI: 10.1109/CVPR42600.2020.00477.

[17] Thibault Groueix et al. “A Papier-Mache Approach to Learning 3D Surface Generation”. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (Dec. 2018), pp. 216–224. ISSN: 10636919. DOI: 10.1109/CVPR.2018.00030.

[18] Jeong Joon Park et al. “Deepsdf: Learning continuous signed distance functions for shape representation”. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* 2019-June (June 2019), pp. 165–174. ISSN: 10636919. DOI: 10.1109/CVPR.2019.00025.

[19] Lars Mescheder et al. “Occupancy Networks: Learning 3D Reconstruction in Function Space”. In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* 2019-June (June 2019), pp. 4455–4465. ISSN: 10636919. DOI: 10.1109/CVPR.2019.00459.

[20] Albert W. Reed et al. “Dynamic CT Reconstruction from Limited Views with Implicit Neural Representations and Parametric Motion Fields”. In: *IEEE International Conference on Computer Vision* (2021), pp. 2238–2248. ISSN: 15505499. DOI: 10.1109/ICCV48922.2021.00226.

[21] Zhengqi Li et al. “Neural Scene Flow Fields for Space-Time View Synthesis of Dynamic Scenes”. In: *Computer Vision and Pattern Recognition* (2020), pp. 6494–6504. ISSN: 10636919. DOI: 10.1109/CVPR46437.2021.00643.

[22] David Wiesner et al. “Generative modeling of living cells with SO(3)-equivariant implicit neural representations”. In: *Medical Image Analysis* 91 (Jan. 2024). ISSN: 13618423. DOI: 10.1016/j.media.2023.102991.

[23] Jie Feng et al. “Spatiotemporal implicit neural representation for unsupervised dynamic MRI reconstruction”. In: *IEEE Transactions on Medical Imaging* 44.5 (Jan. 2025), pp. 2143–2156. ISSN: 1558254X. DOI: 10.1109/TMI.2025.3526452.

[24] Aisha L. Shuaibu et al. “Capturing Longitudinal Changes in Brain Morphology Using Temporally Parameterized Neural Displacement Fields”. In: *Proceedings of Machine Learning Research-Under Review* (Apr. 2025), pp. 1–20. DOI: 10.48550/arXiv.2504.09514.

[25] Xiaoyu Xie, Saviz Mowlavi, and Mouhacine Benosman. “Smooth and Sparse Latent Dynamics in Operator Learning with Jerk Regularization”. In: *arXiv preprint arXiv:2402.15636* (Feb. 2024). DOI: 10.48550/arXiv.2402.15636.

[26] Keunhong Park et al. “Nerfies: Deformable Neural Radiance Fields”. In: *IEEE International Conference on Computer Vision* (Feb. 2020), pp. 5845–5854. ISSN: 15505499. DOI: 10.1109/ICCV48922.2021.00581.

[27] Jelmer M Wolterink, Jesse C Zwienenberg, and Christoph Brune. “Implicit Neural Representations for Deformable Image Registration”. In: *Proceedings of Machine Learning Research*. Vol. 172. PMLR, July 2022, pp. 1349–1359. URL: https://proceedings.mlr.press/v172/wolterink22a.html.

[28] Benedikt Wirth et al. “A continuum mechanical approach to geodesics in shape space”. In: *International Journal of Computer Vision* 93.3 (July 2011), pp. 293–318. ISSN: 09205691. DOI: 10.1007/s11263-010-0416-9.

[29] Arthur Ecoffet et al. “Application of transport-based metric for continuous interpolation between cryo-EM density maps”. In: *AIMS Mathematics* 7.1 (2021), pp. 986–999. ISSN: 24736988. DOI: 10.3934/MATH.2022059.

[30] Chih-Jung Tsai, Cheng Sun, and Hwann-Tzong Chen. “Multiview Regenerative Morphing with Dual Flows”. In: *Computer Vision – ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XVI*. Tel Aviv, Israel: Springer-Verlag, 2022, pp. 492–509. ISBN: 978-3-031-19786-4. DOI: 10.1007/978-3-031-19787-1_28.

[31] Tim Golla et al. “Temporal Upsampling of Point Cloud Sequences by Optimal Transport for Plant Growth Visualization”. In: *Computer Graphics Forum* 39.6 (Sept. 2020), pp. 167–179. ISSN: 14678659. DOI: 10.1111/CGF.14009.

[32] Ben Mildenhall et al. “NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis”. In: *Communications of the ACM* 65.1 (Dec. 2021), pp. 99–106. ISSN: 15577317. DOI: 10.1145/3503250.

[33] Vincent Sitzmann et al. "Implicit Neural Representations with Periodic Activation Functions". In: *Advances in Neural Information Processing Systems* 2020-December (June 2020). ISSN: 10495258. DOI: 10.48550/arXiv.2006.09661.

[34] Matthew F Thorpe. *Introduction to Optimal Transport*. Tech. rep. University of Cambridge, Mar. 2018. URL: https://www.damtp.cam.ac.uk/research/cia/files/teaching/Optimal_Transport_Notes.pdf.

[35] Paul Knopp and Richard Sinkhorn. "Concerning nonnegative matrices and doubly stochastic matrices." In: *Pacific Journal of Mathematics* 21.2 (Jan. 1967), pp. 343–348. ISSN: 0030-8730. DOI: 10.2140/pjm.1967.21.343.

[36] Marco Cuturi. "Sinkhorn Distances: Lightspeed Computation of Optimal Transport". In: *Advances in Neural Information Processing Systems* 26 (2013). DOI: 10.48550/arXiv.1306.0895.

[37] Jean Feydy et al. "Interpolating between Optimal Transport and MMD using Sinkhorn Divergences". In: *AISTATS 2019 - 22nd International Conference on Artificial Intelligence and Statistics* (Oct. 2018). DOI: 10.48550/arXiv.1810.08278.

[38] William E. Lorensen and Harvey E. Cline. "Marching cubes: A high resolution 3D surface construction algorithm". In: *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1987* 21.4 (Aug. 1987), pp. 163–169. DOI: 10.1145/37401.37422.

[39] Inigo Quilez. *Smooth Minimum and Maximum Functions*. 2013. URL: https://iquilezles.org/articles/smin/.

[40] Martin Maška et al. "The cell tracking challenge: 10 years of objective benchmarking". In: *Nature Methods* 20 (7 2023), pp. 1010–1020. DOI: 10.1038/s41592-023-01879-y.

[41] Diederik P. Kingma and Jimmy Lei Ba. "Adam: A Method for Stochastic Optimization". In: *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings* (Dec. 2014). DOI: 10.48550/arXiv.1412.6980.