Inaugural – Dissertation

zur

Erlangung der Doktorwürde

der

Gesamtfakultät für Mathematik, Ingenieur- und Naturwissenschaften

der

Ruprecht-Karls-Universität Heidelberg

vorgelegt von

Nico Albert Disch, Ms. Sc.

aus Heidelberg

Tag der mündlichen Prüfung: _____

Modeling of Sparse and Irregular Medical Image Time Series

Supervisor: Prof. Dr. Klaus Maier-Hein

Da steh' ich nun, ich armer Tor,
Und bin so klug als wie zuvor.

Für alle, die meinen Weg geprägt und mich unterstützt haben.

Abstract

This thesis presents several contributions toward modeling disease progression and temporal dynamics in medical imaging. We begin by revisiting key methods for temporal prediction, specifically adapting Neural Processes and Neural Ordinary Differential Equations to longitudinal medical data. We then analyze the limitations of these approaches in handling sparse and irregular observations. Building on these insights, we propose a framework for longitudinal augmentation and data generation using biologically-informed deformations. The core of this thesis is the development of Temporal Flow Matching (TFM), a flow-based generative model. This model learns continuous velocity fields that describe how anatomical structures evolve over time. TFM scales to 3D and 4D data, generalizes across datasets, and supports inference at arbitrary temporal resolutions through a continuous-time extension. To further constrain and interpret temporal dynamics, we propose two variants of TFM: one deformation-based and one using a Schrödinger Bridge formulation. These variants link flow-based modeling to physical and probabilistic motion representations. Extensive evaluations across synthetic and clinical, which contain cardiac MRI, perfusion CT, and longitudinal brain tumors, datasets show these methods outperform existing baselines, excelling in both predictive accuracy and temporal consistency. Together, these contributions establish a principled framework for learning continuous, anatomically meaningful trajectories from sparse longitudinal medical imaging data.

Zusammenfassung

Diese Arbeit präsentiert mehrere Beiträge zur Modellierung des Krankheitsverlaufs und der zeitlichen Dynamik in der medizinischen Bildgebung. Zunächst werden wichtige Methoden zur zeitlichen Vorhersage betrachtet, insbesondere die Anpassung von Neural Processes und Neural Ordinary Differential Equations an longitudinale medizinische Daten. Anschließend werden die Einschränkungen dieser Ansätze bei der Verarbeitung von wenigen und unregelmäßigen Beobachtungen analysiert. Aufbauend auf diesen Erkenntnissen schlagen wir einen Rahmen für longitudinale Augmentation und Datengenerierung unter Verwendung biologisch informierter Deformationen vor. Das Kernstück dieser Arbeit ist die Entwicklung von Temporal Flow Matching (TFM), eines flow-basierten generativen Modells. Dieses Modell lernt kontinuierliche Geschwindigkeitsfelder, die beschreiben, wie sich anatomische Strukturen im Laufe der Zeit verändern. TFM skaliert auf 3D- und 4D-Daten, generalisiert über Datensätze hinweg und unterstützt Inferenz bei beliebigen zeitlichen Auflösungen durch eine kontinuierliche Zeiterweiterung. Um die zeitliche Dynamik weiter einzuschränken und interpretierbar zu machen, schlagen wir zwei Varianten von TFM vor: eine deformationsbasierte und eine, die auf Schrödinger-Brücken aufbaut. Diese Varianten verbinden flow-basierte Modellierung mit physikalischen und probabilistischen Bewegungsrepräsentationen. Umfangreiche Auswertungen an synthetischen und klinischen Datensätzen zeigen, dass diese Methoden bestehende Baselines übertreffen und sowohl in der Vorhersagegenauigkeit als auch in der zeitlichen Konsistenz überzeugen. Zusammen begründen diese Beiträge einen prinzipiellen Rahmen für das Lernen kontinuierlicher, anatomisch sinnvoller Trajektorien aus spärlichen longitudinalen medizinischen Bildgebungsdaten.

Contents

List of Figures

List of Tables

Introduction 1

1.1 Problem and Motivation

Artificial Intelligence (AI) has achieved remarkable breakthroughs in recent years across a spectrum of fields, including computer vision [1, 134], natural language processing [78], robotics [128], and drug design [30]. The precise definition of "intelligence" in AI is debated in many fields [38, 77]. Yet its potential as a tool across science, medicine, and technology is now clear. Building on these achievements, since the public release of ChatGPT, AI has captured unprecedented attention from both the general public and the research community (see e. g. [39]), accelerating investment and innovation across sectors.

Within this wave of progress, AI is increasingly integrated into drug discovery workflows in the pharmaceutical domain. For example, *DeepMind's AlphaFold* (first released in [4, 58], now in Version 3) revolutionized protein structure prediction. The system achieves near-experimental accuracy and enables rapid biological insights. In its wake, companies and researchers have leveraged generative models, reinforcement learning, and multi-omics integration. These methods help design novel molecules, accelerating discovery pipelines and advancing AI-designed drugs into human clinical trials, as discussed by [69]. In computer vision, the modern wave of success began largely with *AlexNet* [71], which demonstrated the power of deep convolutional networks when paired with large-scale datasets and GPUs, ushering in the deep learning era for visual recognition. Transformer-based architectures, adapted from natural language processing, have since made substantial inroads into vision tasks [54]. Applications now range from low-level tasks like denoising images and super-resolution, to high-level tasks like object detection, image classification, and segmentation [134]. While fully autonomous driving remains an open challenge, AI-powered perception systems have made substantial progress towards this goal. This is despite the complexity and safety demands of real-world environments [159].

Progress in generative modeling marks a shift from discriminative learning. These new approaches directly reconstruct, interpolate, and synthesize complex data distributions [152]. *Generative Adversarial Networks* (GANs) [70] introduced adversarial training for high-fidelity image synthesis. This enabled photorealistic generation and creative applications (see e. g. Figure 1.1). A prominent example is `https://this-person-does-not-exist.com`, which generates photorealistic images of people that do not exist. More recently, *diffusion models* [21, 47, 109] have

Figure 1.1: What ChatGPT thinks of the word forecasting.

surpassed GANs in image quality, as exemplified by improved FID scores on ImageNet. *Flow Matching* [79] provided a continuous and unified formulation of Flow models, which also includes Diffusion models. This modeling paradigm demonstrated competitive, more efficient results in image and video generation.
Beyond visual realism, these methods enable learning complex data manifolds [152]. Such methods have increasingly been adopted in scientific and medical imaging [160], where high-fidelity image generation is used for a range of tasks, including data augmentation, improving diagnostic accuracy, and preserving data privacy. While concerns around data ownership, privacy [90], deepfakes [110], and misinformation [130] remain valid, this thesis specifically investigates methodological approaches for generating realistic and high-fidelity medical image time series. In the following chapters, we build on these generative approaches and, ultimately, adapt Flow Matching to model longitudinal medical image data.

Furthermore, AI has the capability to forecast future events. In meteorology, it complements traditional physical models with data-driven approaches to improve the accuracy of weather prediction [104]. In finance, AI is applied in key areas such as exchange rate forecasting, financial modeling, risk management [136]. It is also used in areas such as demand forecasting [96] and modeling supply chains [3].

The desire to forecast the future, however, predates AI by millennia. From the oracle bones of Shang-dynasty China to the Oracle of Delphi in ancient Greece, humanity has long sought glimpses of what lies ahead. [1] While our tools have evolved

[1] At Delphi, prophecies were delivered by the priestess known as the *Pythia* or "pythonesse," a name etymologically linked to the slain serpent Python, a linguistic coincidence with the modern programming language Python, which now underpins much of AI research, including

from reading bones to training high-dimensional neural networks, the underlying motivation remains unchanged: to gain foresight, reduce uncertainty, and improve decision-making.
Against this backdrop, in this work, we focus on forecasting future observations in medical imaging. We aim to develop a model that leverages the intrinsic structure of longitudinal medical data to reconstruct the expected appearance of any image at arbitrary timepoints. Such a capability holds strong potential for personalized medicine, enabling earlier detection of abnormal disease trajectories and offering deeper insight into the dynamics of disease progression.

1.2 Research Gap

Problem Despite substantial advances in medical image analysis, the study of time series in medical imaging remains underdeveloped. In clinical practice, patients are repeatedly scanned over time, capturing anatomical and pathological changes, such as disease progression or treatment response. Moving beyond cross-sectional (i. e., only using single timepoints) analysis toward prediction is a natural and clinically meaningful next step. Forecasting how a patient's anatomy might evolve at future or intermediate time points builds on this foundation. This task is inherently challenging. Medical images are high-dimensional, acquisition intervals are irregular, and available temporal sequences are often sparse. Consequently, models must learn meaningful temporal dynamics from limited, unevenly spaced observations.

Methodological Gap Temporal metadata, such as acquisition intervals, are often available in clinical datasets but are rarely exploited in current modeling approaches. Incorporating this information could enable principled interpolation between scans and more reliable extrapolation into the future. Moreover, much of the existing literature remains confined to single-modality, single-timepoint analysis and does not address the challenges posed by sparse, irregularly sampled, or longitudinal data. When temporal information is considered, models typically aim for simplified objectives, such as classification, regression, or image-to-image prediction, rather than learning a continuous representation of disease evolution. Methodologically, many approaches are restricted to 2D data due to computational constraints, and reproducibility remains limited, as code and datasets are often unavailable for public evaluation.

this thesis.

1.3 Overview and Outline

The chapters of this thesis are organized sequentially, with each chapter building on the foundations established by the previous one. Rather than containing a single method with results and discussion, the content is distributed across chapters according to focus: Earlier chapters introduce the theoretical and methodological foundations. Later ones present experimental results and their discussion. Chapter 2 introduces the necessary background, including medically relevant imaging modalities, neural network fundamentals, and key methodological concepts. This chapter also discusses the Last Context Image heuristic (simply using the last image in the time series), and time-continuous[2] models, such as Neural Ordinary Differential Equations (NODEs), Diffusion Models, and Flow Matching, which form the conceptual basis of our approach. We will talk about datasets in Chapter 3, and Chapter 4 presents the methods developed in this thesis. We begin with the ASP framework [102] and explore its extensions. To establish a robust comparison, we examine several natural-image baselines on the same prediction task and subsequently adapt the most effective methods to the medical domain, extending them to volumetric (3D) data. In addition, we introduce a spatio-temporal augmentation framework; Longitudinal data AUgmentation and data GENeration, termed **LAUGEN** [23]. Our principal contribution, TFM (Temporal Flow Matching), is then described as an efficient approach for temporal medical image prediction capable of modelling both 3D+t (sequences of volumetric acquisitions over time) and 4D (continuous spatio-temporal volumes such as perfusion imaging) irregularly sampled data. Finally, we extend TFM to a continuous-time formulation, enabling predictions at arbitrary temporal resolutions. Chapter 5 reports the experimental results obtained with these methods, followed by a discussion in Chapter 6 of their implications, limitations, and relation to existing approaches. Chapter 7 concludes the thesis by summarizing the main contributions and outlining potential directions for future work.

1.4 Contributions

Neural ODEs for Medical Imaging This thesis first extends the medical imaging baselines using NODEs. While these extensions yield partial performance improvements, the original baseline is computationally intensive and lacks scalability. We therefore propose lighter architectural variants that replace expensive attention mechanisms with more efficient alternatives. Although these adaptations reduce computational overhead, this backbone reveals failure modes, highlighting fundamental limitations of such methods in modeling spatio-temporal data.

[2]The term "time-continuous" is used here in its general methodological sense, following prior work, and will be specified in more detail in later chapters.

Augmentation and Data Generation for Medical Imaging We introduce a novel augmentation and data generation framework for synthesizing longitudinal medical imaging from single images [23]. This method applies biologically-informed deformations along predefined longitudinal trajectories in latent space, producing plausible temporal sequences that mimic anatomical change. Our biologically-informed longitudinal augmentation approach is computationally efficient and provides a practical baseline for addressing data scarcity in longitudinal imaging, where acquiring temporal data remains challenging.

TFM for Temporal Medical Image Time Series We present Temporal Flow Matching (TFM [25]), a flow-based generative framework designed to model image evolution in sparse and irregularly sampled longitudinal data. TFM learns sequence velocity fields that describe how anatomical structures change between observations, enabling predictions across clinically relevant timescales. In contrast to baseline approaches, TFM is designed to scale to volumetric, spatio-temporal, and longitudinal data, covering both 3D+t sequences and 4D data, to forecast future images.
We further derive a continuous-time variant of TFM (under submission) by replacing flow steps with real-time vector embeddings, enabling inference at arbitrary time points without altering the architecture. To better capture cases where the change is primarily due to motion or deformation, we couple TFM with a deformation-based parametrization: First, through a displacement-field variant and by relating it to concepts from unbalanced optimal transport. Finally, we introduce a Schrödinger Bridge extension that describes trajectories governed by learned regularizations. Together, these components establish TFM as a unified framework for robust, scalable modeling of longitudinal medical imaging analysis.

Background 2

In this chapter, we introduce the background needed to follow the remainder of this thesis. Section 2.1 provides a brief overview of the relevant medical context and imaging modalities. Section 2.2 outlines the neural network fundamentals that form the basis of the methods used throughout this work. We then formally define the problem in Section 2.3, with an example which highlights the complexity of this task. Finally, in Section 2.5, we review related work across natural imaging, show how works in medical imaging are restricted, and continuous-time modeling, which we will use to build the methods for this work.

2.1 Medical Imaging Background

This section provides a brief overview of medical imaging modalities relevant to this thesis, with a focus on their spatio-temporal appearance. Medical imaging encompasses various techniques for visualizing internal anatomy, including X-ray, Computed Tomography, Magnetic Resonance Imaging, ultrasound, Positron Emission Tomography, endoscopy, and OCT. While each imaging modality has specific clinical uses, MRI and CT are emphasized here due to their widespread roles in longitudinal studies of disease progression. Key references are [10, 105].

Physical Background MRI is governed by the Bloch equations:

$$\frac{d\mathbf{M}}{dt} = \gamma\,\mathbf{M} \times \mathbf{B} - \begin{pmatrix} M_x/T_2 \\ M_y/T_2 \\ (M_z - M_0)/T_1 \end{pmatrix}, \tag{2.1}$$

where $\mathbf{M} = (M_x, M_y, M_z)^\top$ is the magnetization vector, M_0 the equilibrium magnetization, γ, the gyromagnetic ratio, and $\mathbf{B}$ is the magnetic field, and T_1 , T_2 are the longitudinal and transverse relaxation times, respectively. For CT, the Lambert-Beer law governs the intensity I within the body:

$$I = I_0 \exp\left(-\int \mu(x)dx\right), \tag{2.2}$$

where μ is the linear attenuation coefficient, describing local X-ray absorption. I_0 represents the initial intensity.

Temporal and Longitudinal Imaging Most imaging modalities become *longitudinal* when acquisitions are repeated over time. Some imaging modalities are inherently temporal, such as ultrasound, which captures continuous real-time images; surgical videos, which record dynamic procedures; perfusion CT, which monitors contrast flow through tissues; and cine MRI, which acquires time-resolved MRI frames to study periodic motion. In those cases, these modalities capture dynamic processes such as motion or contrast propagation.

Cine MRI Cine MRI rapidly acquires time-resolved MRI frames to visualize periodic motion, such as heartbeats or organ tracking during breathing. To achieve adequate temporal resolution, one spatial dimension is sampled at a lower resolution to maintain anatomical detail.

perfusion CT Computed Tomography uses X-rays to generate cross-sectional images of the body, with or without contrast media to enhance visualization of anatomical structures. Perfusion CT captures the passages of a contrast agent through the vasculature over time, enabling quantitative assessment of tissue perfusion. It thus represents an inherent temporal extension of standard CT. While we are not primarily interested in the tasks typically associated with perfusion CT, we use the temporal aspect to evaluate the performance of our models.

2.2 Deep Learning and Neural Networks

In this section, we establish the terminology and conventions for deep learning used throughout this thesis. We also provide additional background information and, where relevant, historical context. Then, we outline the main architectural components used in the methodology of modern neural networks. These include convolutions, attention mechanisms, and state-space models, specifically the Mamba block. These elements are crucial, as they are the backbone of the models in the methods section. Since this thesis centers on image prediction, we devote special attention to this area. While it is discussed in particular in [89], we consider it important to clarify which losses we optimize by, and what our metrics actually measure. Finally, to complete the methodological picture, we provide a brief overview of optimization methods, anchoring them within the framework established above.

2.2.1 Definitions

Neural Networks

We define a neural network as a parametric mapping

$$f_\theta : \mathbb{R}^n \to \mathbb{R}^m, \tag{2.3}$$

where n is the dimension of the input space, m is the dimension of the output space, and θ denotes all learnable parameters of the network. A *fully-connected* feedforward neural network of depth $L \in \mathbb{N}$ is constructed as a composition of L layers. Let the pre-activation of the l-th layer be

$$z^{(l)} \in \mathbb{R}^{n_l}, \tag{2.4}$$

where n_l is the number of neurons in layer l, and let $z^{(0)}$ denote the input to the network. For $l = 1, \ldots, L$, the layer computation is

$$z^{(l)} = W^{(l)}\,\sigma\big(z^{(l-1)}\big) + b^{(l)}, \tag{2.5}$$

where $W^{(l)} \in \mathbb{R}^{n_l \times n_{l-1}}$ is the weight matrix, $b^{(l)} \in \mathbb{R}^{n_l}$ is the bias vector, and σ is a non-linear activation function. We call the layer in (2.5) fully connected if **the weight matrix W is dense**. A neural network is **fully connected if every layer is fully connected**. The network output is $f_\theta(x) = z^{(L)}$, with the complete parameter set

$$\theta = \big\{W^{(l)}, b^{(l)}\big\}_{l=1}^{L}. \tag{2.6}$$

Common choices for σ include the Rectified Linear Unit (ReLU), $\sigma(x) = \max(x, 0)$, the hyperbolic tangent, $\sigma(x) = \tanh(x)$, and the sigmoid, $\sigma(x) = \frac{1}{1+e^{-x}}$ [111], and with many more modern variants. [1] More generally, most modern architectures can be viewed as parametrized computational graphs. Multilayer feedforward neural networks are universal approximators Hornik et al. [50], yet this solely does not explain their utility [2]

Architecture and Neural Network We denote the *architecture* as the specific *wiring*. This includes the dimension of the weight matrices, or the composition of architectural blocks 2.2.2. A *neural network* then refers to an instance of this architecture with a specified set of (learned) parameters. While this distinction is conceptually important, the engineering and applied literature often uses the two terms interchangeably. We find that this differentiation can be beneficial in certain contexts.

[1] As (2.5) describes a linear transformation, the universal approximator needs a non-linear activation, because the composition of two linear functions is linear.

[2] *The universal approximation property does not a priori explain the success of deep learning.* The argument is often invoked and sometimes conflated as a justification for it. In reality, many other classes of universal approximators exist, such as Taylor series or Fourier expansions, the former of which are seldom used *directly* in machine learning, though the latter remains foundational in image processing. Deep learning's impact comes from hardware-optimized algorithms [71], the backpropagation algorithm [111], and the increasing availability of large annotated datasets such as [20]. These factors collectively enabled neural networks to outperform other universal function approximators in real-world applications.

Model While we are not aware of prior works that clearly separate the semantic notions of *neural network, model*, and *method*, we find it useful to define these terms explicitly for our purposes. We define the term *model* to denote the mapping defined by a network f_θ together with its specific input–output relationship. For instance, both Diffusion (see e. g. 2.5.2) and Flow Matching (see 2.5.3) models can use the same *architecture* f_θ. In diffusion, the model input is a noisy sample and predicts the *score or noise*; in Optimal Transport Flow Matching, the model receives a linear interpolation and predicts the *velocity* u_τ between the OT map of two distributions. Hence, the distinction between these two models lies in the object predicted, rather than in the dimensionality of the input or output spaces. We use the term *method* to refer to the complete procedure, comprising the optimization objective, data processing, and the model, which defines how a neural network is trained end-to-end. When referring to a specific trained network instance, we denote it by its learned parameters θ.

Latents We define **latents** (or **encodings**) as intermediate, most often lower-dimensional, representations of the input produced within the network. Colloquially, (2.5) are called latents if the dimension is smaller than the input dimension. Such representations are often called *embeddings*. However, the term has a specific meaning in differential geometry. In deep learning, embeddings typically refer to lower-dimensional representations of data, whereas in differential geometry, an embedding is an injective homeomorphism, and thus cannot be lower-dimensional. Thus, while the two usages are conceptually related, their directions are essentially opposite. To prevent ambiguity, we therefore refrain from using the term *embedding* in the deep-learning sense in this thesis. We will only use the term in the differential geometry sense if needed.

2.2.2 Architectural Blocks

In this section, we provide a concise overview of the fundamental architectural components commonly used in neural networks, with a focus on those most relevant for image processing. We do not cover auxiliary components such as activation functions or normalization layers in detail. The key building blocks discussed here are convolutional layers, attention mechanisms, and more recent sequence models such as Mamba. Convolutional layers were first introduced in the Neocognitron [33], later trained via backpropagation [76, 139], and popularized by AlexNet [71]. The attention mechanism, introduced by Vaswani et al. [137] for natural language processing, has since become the backbone of modern large language models. It has also been adapted for image processing, as in the Vision Transformer [27], and extended to video in the Video Vision Transformer [5]. More recently, Mamba [43] has been proposed as a linear-time sequence-to-sequence model. Like attention, it operates on sequences. However, it is more efficient due to linear time complexity.

Convolutional Layers The (discrete) convolution operation is defined as

$$(f * g)(x) = \sum_{a=-\infty}^{\infty} f(a)\, g(x-a), \tag{2.7}$$

and a convolutional layer applies this operation using a learnable kernel K to the input I, yielding $(I * K)$. This definition naturally extends to matrices and higher-dimensional tensors. It does this by performing element-wise multiplication, interpreting the arguments as positions of the matrices. A convolutional layer applies this operation by learning kernel K and operating across all input channels at once. For any layer $l-1$ we have

$$z_j^{(l)} = \sum_{i=1}^{C_{\text{in}}^{(l-1)}} \left(z_i^{(l-1)} * K_{j,i}^{(l)} \right) + b_j^{(l)}, \tag{2.8}$$

where $C_{\text{in}}^{(l-1)}$ denotes the number of input channels, $K_{j,i}^{(l)}$ the convolution connecting input channel i to output channel j. Typically, convolutional kernel is of size k^d with d being the inherent dimension, not to be confused with S (e. g. $d = 3$ for volumes).

Attention Mechanism The Attention [137] operation is given by

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}} \right) V, \tag{2.9}$$

where $Q = XW_Q$, $K = XW_K$, and $V = XW_V$ are linear layers of the input X (colloquially called projections, but projection has usually a different definition for matrices). The projection matrices $W_Q, W_K, W_V \in \mathbb{R}^{d_{\text{in}} \times d_k}$ are learnable parameters (d_{in} input dimension, and d_k the output dimension). The multi-head variant , several such operations are performed in parallel, then concatenated, and then projected via a linear layer.

Mamba Sequence-to-Sequence The Mamba block [43] is a state space model which performs a sequence-to-sequence transformation with linear-time complexity parametrized by (Δ, A, B, C). In its simplified discrete form, the recurrence is

$$h_t = \bar{A} h_{t-1} + \bar{B} x_t, \tag{2.10}$$

$$y_t = C h_t, \tag{2.11}$$

which can also be expressed as a long convolution:

$$\mathbf{K} = (C\bar{B},\ C\bar{A}\bar{B},\ \ldots,\ C\bar{A}^k\bar{B}, \ldots), \tag{2.12}$$

$$y = x * \mathbf{K}. \tag{2.13}$$

There, x is the input, and y the corresponding output of the Mamba block. This formulation expresses the sequence transformation as a linear convolution over the input, where the kernel $\mathbf{K}$ implicitly encodes temporal dependencies through the recurrent dynamics. While full implementations involve additional steps, the core idea remains the same. Despite the rise of attention-based and Mamba-style architectures, convolutional models remain highly competitive for medical imaging [97]. These building blocks, together with fully connected layers, form the core components used throughout this thesis.

2.2.3 Metrics and Evaluations

In this section, we summarize the metrics[3] used to evaluate image prediction quality. The measures are grouped into pixel-level metrics and perceptual metrics, each measuring different aspects of similarity between predicted and reference image. We emphasize that it is essential to understand what each metric actually measures [89]. While evaluation is more straightforward for other tasks, it is less so for forecasting full images, where changes are happening on a sparse region. Most metrics quantify the validity or similarity of the predicted image rather than its predictive fidelity with respect to the underlying temporal process. Proposing new or task-specific metrics is beyond the scope of this work; we follow prior studies that rely on established image-based measures for fair comparison.

Pixel-level Metrics

directly compare images in their raw intensity space. They are more sensitive to small spatial misalignments but provide straightforward quantitative measures of reconstruction or prediction quality

Dice Score For segmentation, the Dice score is defined as

$$DSC(x, y) = \frac{2|x \cap y|}{|x| + |y|}. \tag{2.14}$$

Mean Squared Error (MSE): The Mean Squared Error (MSE) is defined as

$$\mathrm{MSE}(x, \hat{x}) = \frac{1}{N} \sum_{i=1}^{N} (x_i - \hat{x}_i)^2, \tag{2.15}$$

[3] Colloquially, these quantities are referred to as *metrics*, but not all of them are metrics in the formal mathematical sense. Many are better described as similarity or divergence measures. We nevertheless retain the term metric, as it is ubiquitous in the literature. Additionally, the term similarity measure is used in its conventional sense to denote a function quantifying resemblance, not a measure in the mathematical sense.

where N typically equals the number of spatial elements (pixels or voxels) in the image ($N = S$), x_i and $\hat{x}_i$ are the pixel values of the target image and the predicted image, respectively. For most cases, we use MSE as the loss function, as it is differentiable and easy to optimize. A related measure is Normed Root Mean Squared Error (NRMSE), which is the square root of MSE divided by a normalization factor (e. g.image dynamic range, mean or standard deviation). Without an explicit specification of the normalization scheme, Normed Root Mean Squared Error (NRMSE) is not directly comparable across different images or datasets. Therefore, we will keep the normalization factor as 1 throughout, i. e. the root of MSE.

Peak Signal-to-Noise Ratio (PSNR): The PSNR is defined as

$$\mathrm{PSNR}(x, \hat{x}) = 10 \cdot \log_{10} \left(\frac{(\mathrm{MAX}_x)^2}{\mathrm{MSE(x, \hat{x})}} \right), \tag{2.16}$$

where MAX_x is the maximum possible pixel (or voxel) intensity of the image x.

Structural Similarity Index Measure (SSIM): The SSIM is defined as

$$\mathrm{SSIM}(I, \hat{I}) = \frac{(2\mu_I \mu_{\hat{I}} + C_1)(2\sigma_{I\hat{I}} + C_2)}{(\mu_I^2 + \mu_{\hat{I}}^2 + C_1)(\sigma_I^2 + \sigma_{\hat{I}}^2 + C_2)}, \tag{2.17}$$

where μ_I, σ_I^2, and $\sigma_{I\hat{I}}$ (and the corresponding values for $\hat{I}$) being the local means, variances, and cross-covariance where C_1 and C_2 are small constants introduced to avoid division by zero.

Perceptual Metrics

Learned Perceptual Image Patch Similarity (LPIPS): The Learned Perceptual Image Patch Similarity (LPIPS) [157] is defined as

$$\mathrm{LPIPS}(x, \hat{x}) = \sum_l \frac{1}{S_l} \sum_s \| w_l \odot (\phi_l(x)_s - \phi_l(\hat{x})_s) \|_2^2 \tag{2.18}$$

where $\phi_l(\cdot)$ are deep features at layer l, w_l are learned weights, and S_l are spatial dims of each layer, and $\odot$ the elementwise (Hadamard) product.

Fréchet Inception Distance (FID): The FID is defined as

$$\mathrm{FID}(X_{\mathrm{real}}, X_{\mathrm{gen}}) = |\mu_r - \mu_g|_2^2 + \mathrm{Tr}\left(\Sigma_r + \Sigma_g - 2\left(\Sigma_r \Sigma_g\right)^{1/2} \right), \tag{2.19}$$

where (μ_r, Σ_r) and (μ_g, Σ_g) denote the empirical means and covariances of the Inception feature embeddings computed from the sets of real and generated images,

respectively and Tr denotes the trace operation on matrices. These features are taken over deep features, either via Inception network, see e. g.LPIPS, or via a pre-trained VGG network. Unlike image-wise metrics, such as MSE or LPIPS, Fréchet Inception Distance (FID) operates on distributions of features across a dataset, rather than individual image pairs.

2.2.4 Optimization

In order to efficiently compute the gradient of the loss function $\mathcal{L}$ with respect to all parameters, neural networks are trained with backpropagation (see [111, 146]). For each layer l, the derivative $\partial\mathcal{L}/\partial\theta_l$ is with the use of the chain rule

$$\frac{\partial\mathcal{L}}{\partial\theta_l} = \frac{\partial\mathcal{L}}{\partial a_l}\frac{\partial a_l}{\partial z_l}\frac{\partial z_l}{\partial\theta_l}. \tag{2.20}$$

Since all the layers are linear, this computation can be calculated easily for each layer, and thus making this backpropagation efficient.

Gradient based optimization: the simplest optimizer is stochastic gradient descent (SGD), which updates parameters with learning rate η as

$$\theta_{i+1} = \theta_i - \eta\nabla_\theta\mathcal{L}(\theta_i). \tag{2.21}$$

A common optimizer is the Adam [65] optimizer, which we use for the experiments. The motivation is the momentum, which is the physical motivation that objects stay in motion and avoid possible troughs. For shorthand, we define $g_{i-i} \coloneqq \nabla_\theta\mathcal{L}(\theta_{i-1})$, the momentum is

$$m_i \coloneqq \beta_1 m_{i-1} + (1-\beta_1)g_{i-1}, \tag{2.22}$$

and the velocity

$$v_i = \beta_2 v_{i-1} + (1-\beta_2)g_{i-1}^2. \tag{2.23}$$

Then $\hat{m}_i = \frac{m_i}{1-\beta_1^i}$ and $\hat{v}_i = \frac{v_i}{1-\beta_2^i}$ are the bias corrected terms. With set parameters β_1 and β_2. Finally, the parameters θ are updated via

$$\theta_i = \theta_{i-1} - \eta\frac{\hat{m}_i}{\sqrt{v_i}+\epsilon}, \tag{2.24}$$

where ϵ prevents divisions by zero.

2.3 Spatio Temporal Learning Foundations

This section establishes the formal problem setting for spatio-temporal forecasting, introduces a simple heuristic, and highlights conceptual challenges.

2.3.1 Formal Problem Definition

Let the dataset consist of $p \sim \Pi$ spatio-temporal image sequences, each corresponding to one patient. For each patient, we assume T **context** images $\mathcal{I} = \{I_1, \ldots, I_T\}$ with $I_i \in \mathbb{R}^{H \times D \times W}$. Each image is acquired at ordered time points $\mathcal{T} = \{t_1, \ldots, t_T\}$. We further denote a **target** image I_{target} at a time t_{target}. For shorthand, let $S := D \times H \times W$ denote the spatial size. The term *sparse* is used informally to indicate that only a few context images are available.

We distinguish between different types of temporal sampling:

- **Regular** time series: The time points t_i are evenly spaced, i.e. $t_i - t_{i-1} = \delta t$ for all i.
- **Irregular** time series: The time points t_i are not evenly spaced, i.e. $t_i = \delta t * k$ for some $k \in \mathbb{N}$.
- **Continuous** time series: The time points t_i are spaced on a continuum, i.e. $t_i - t_{i-1} \in \mathbb{R}_+$ for all i, j.

Both the regular and irregular settings operate in discrete time, since the acquisition intervals are discrete. Irregular sampling typically arises when acquisitions in an otherwise regular schedule are missing, for example, due to corrupted data or patient-specific deviations. The continuous case, in contrast, refers to acquisition schedules determined solely by clinical necessity, such as follow-up imaging for acute glioblastoma, rather than fixed-interval protocols used in controlled studies.

Central Objective The central objective of this work can be formalized as learning a function

$$f_\theta(\mathcal{I}, \mathcal{T}, t_{\text{target}}) = \hat{I}_{\text{target}} \tag{2.25}$$

where f_θ is a parametric function that, given context images and times $\mathcal{I}, \mathcal{T}$, approximates the unknown distribution process producing I_{target} at a time t_{target}. This formulation defines the spatio-temporal prediction problem, which underlines the image-based forecasting part of this work.

2.3.2 LCI: Heuristic Baseline

As discussed in the metrics section, quantitative metrics alone may not directly capture predictive performance. In general, these metrics evaluate either similarity (e. g. Peak Signal-to-Noise Ratio (PSNR) or Structural Similarity Index Measure (SSIM)) or distance (e. g. MSE, LPIPS) To contextualize the values of these metrics, we introduce the Last Context Image (LCI), a simple heuristic baseline that calculates the metrics between the last image available in the series and the prediction image. In the following paragraph, we formally define this heuristic and discuss its

relevance, both as a lower-complexity reference and as a means to interpret metric-based results in longitudinal image prediction. We define the LCI:

$$I_{\text{LIB}} \coloneqq I_T, \tag{2.26}$$

that is, the heuristic is simply the last available image in the sequence. Given an

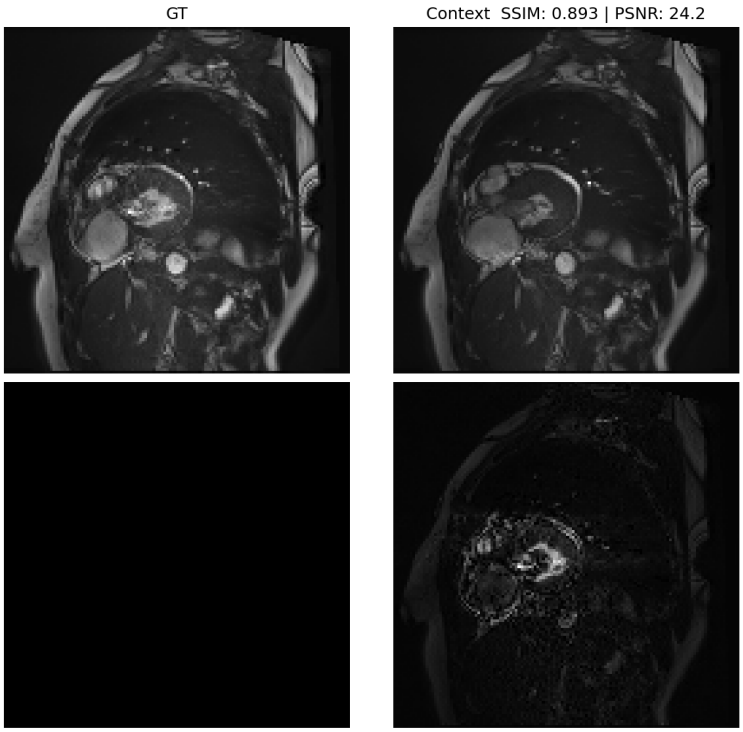


Figure 2.1: Example of a longitudinal image sequence, with the LCI on the right and the target (ground truth, GT) on the left. Data is from the Automated Cardiac Diagnosis Challenge (ACDC) dataset [7]. The lower row shows the pixel-level difference between the LCI and the GT.

evaluation metric $\mathcal{M}$, the corresponding value is

$$\mathcal{M}_{\text{LCI}} \coloneqq \mathcal{M}(I_T, \text{target}). \tag{2.27}$$

This heuristic is naive but often competitive in medical imaging. The rationale for this is discussed further in section 3.2.6, where we explain why the heuristic can still score well under previously discussed image-based metrics in this domain. Although the same heuristic and some alterations appear in related work, they are rarely explicitly defined or discussed in detail. Here, we define it explicitly to provide a reference for interpreting metric values in temporal image modeling. For

completeness, we also define an oracle variant that selects the context images most similar to the target:

$$\mathcal{M}_{\text{MIB}} \coloneqq \min_{i \in \{1,\dots,n-1\}} \mathcal{M}(I_i, I_n). \tag{2.28}$$

The optimality of LCI vs MIB holds when the time series is monotonic with respect to the evaluation metric. However, since the MIB requires knowledge of the future observation, it represents an unrealistic, strict heuristic. In clinical settings, only the LCI is accessible, as the optimal context image is unknown at inference time (even though it can be identified during experimental evaluation). Therefore, the LCI will serve as the reference baseline throughout this work's experiments.

Performance of LCI The LCI is a strong and simple heuristic in the task of longitudinal imaging: First, it is extremely simple and requires no additional labels. Secondly, in medical imaging, temporal changes are often sparse and gradual, making LCI particularly competitive. In Figure 2.1, we show an example longitudinal image sequence from the ACDC dataset [7]. The changes over time are minimal, making the prediction task inherently challenging to improve upon beyond LCI. As a result, LCI often performs well on standard pixel-level metrics, making it challenging for alternative methods to surpass its results. In the next section, we illustrate that models might even predict longitudinal change perfectly but still fail to achieve better performance on image metrics. This highlights a discrepancy between the metrics popularly used and the dynamic aspects we aim to capture.

2.3.3 Reverse QR Code Problem

This subsection introduces the *reverse* QR code problem, a synthetic example which illustrates the limitations of image-level metrics such as MSE. The term *reverse* refers to the inversion of the usual QR code information layout: in standard QR codes, essential information is distributed across the pattern, while the center can be reserved for arbitrary content. In our example, all the longitudinal information is within the center, hence the attribute *reverse*. We show that even under mild conditions, a model with perfect longitudinal prediction but low spatial resolution can still achieve a suboptimal MSE. In our example, the images resemble QR codes in structure, but the critical spatio-temporal information is concentrated in the central region.

Data Assumptions Images are of size $S \times S = 8 \times 8$, and temporal change appears in a $r \times r = 4 \times 4$ in the center of the image. For the sake of complexity, we assume a checkerboard pattern outside the center region, where no neighbouring pixel is the same. See Figure 2.2 for an illustration. **Problem Statement** We consider the longitudinal prediction problem, where we have the context I_0 and the target I_1. For further simplicity, we assume that we have sufficient context to predict the center of the target image.

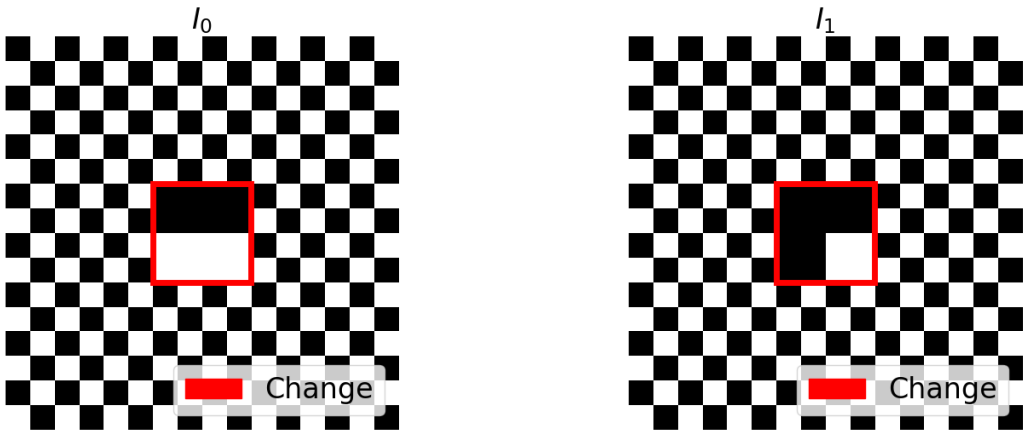


Figure 2.2: Illustration of the reverse QR code problem. (**Left**) Input image I_0 consisting of an 8×8 grid with a high-frequency checkerboard pattern. (**Right**) Target image I_1, which differs from I_0 only within the red boxed 4×4 central region. This central region contains the entire temporal change, while the surrounding pixels remain unchanged but retain high spatial detail.

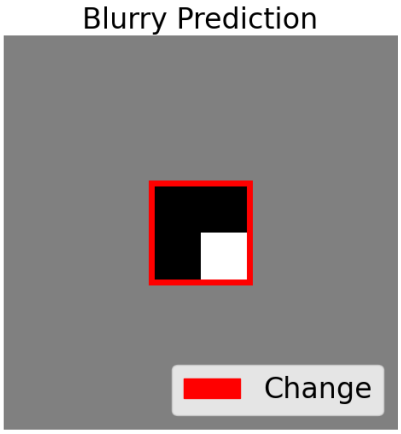


Figure 2.3: Predicted image $\hat{I}_1$ for the reverse QR code problem under a low spatial resolution constraint. The model correctly predicts the temporal change in the central 4×4 region, achieving perfect temporal accuracy there, but produces a coarse 2×2-equivalent resolution in the rest of the image. This setting demonstrates that a method can be temporally perfect yet still achieve a low pixel-level error (MSE) due to reduced spatial detail outside the change region.

Model Assumptions We assume a model with strong longitudinal prediction capability, i. e., the available context is sufficient to accurately predict the temporal changes in the target image. To isolate spatial effects, we constrain the model to lower (i. e. half) spatial resolution: its effective output corresponds to a 2×2 grid over the 8×8 image domain. Figure 2.3 illustrates such a prediction. In this configuration, the prediction $\hat{I}_1$ reproduces the central 4×4 change region of the target I_1 exactly, but deviates in the remaining 48 pixels, which are temporally static but fine-structured. We now quantify this discrepancy using the MSE between $\hat{I}_1$ and I_1, and compare it to LCI.

$$\begin{aligned} \mathrm{MSE}(I_1, \hat{I}_1) &= \frac{1}{64} \sum_{i=1}^{64} (I_{1,i} - \hat{I}_{1,i})^2 \\ &= \frac{1}{64} \left(24 \cdot (0.5 - 1)^2 + 24 \cdot (0.5 - 0)^2 + 16 \cdot (0 - 0)^2 \right) \\ &= \frac{48 \cdot 0.25}{64} = \frac{12}{64}. \end{aligned} \tag{2.29}$$

$$\begin{aligned} \mathrm{MSE}(I_1, I_0) &= \frac{1}{64} \sum_{i=1}^{64} (I_{1,i} - I_{0,i})^2 \\ &= \frac{1}{64} \left(48 \cdot (0 - 0)^2 + 4 \cdot (0 - 0)^2 + 4 \cdot (0 - 1)^2 + 8 \cdot (1 - 1)^2 \right) \\ &= \frac{4}{64} = \mathrm{LCI}. \end{aligned} \tag{2.30}$$

Paradoxically, a model which perfectly captures the longitudinal change can score ***worse*** *than the trivial LCI heuristic.* These results illustrate a fundamental limitation: pixel-level metrics reward overall similarity rather than temporal correctness. We will later return to this example and show how our proposed longtfm explicitly models temporal evolution, achieving an overall zero MSE, despite having the same architectural restrictions.

Generalizing Discrepancy For a general formulation, we can generalize: Let ϵ be the average squared spatial error outside the change region, and δ the mean spatial difference in the change region. Then

$$MSE_{\text{method}} = \frac{(S^d - r^d)\epsilon}{S^d} \tag{2.31}$$

being the difference of the method, and

$$MSE_{LCI} = \frac{r^d \delta}{S^d} \tag{2.32}$$

the MSE for the generalized problem. Then we find that LCI performs better whenever

$$\epsilon > \frac{r^d}{S^d - r^d}\delta. \tag{2.33}$$

This inequality shows that if the static regions dominate, even mild blurring in the unchanged areas outweighs perfect prediction in the dynamic region. In particular, we note that this inequality scales with the dimension d, making it even stronger for $3D$ medical imaging. Therefore, the local metrics we use may underestimate temporal modeling when spatial fidelity decreases, particularly in medical imaging sequences with small localized changes.

Conceptual Takeaways Albeit simply and only exemplary, this reverse QR problem demonstrates that pixel-level metrics (and possibly others) such as MSE *can* fail to reward correct temporal modeling when static spatial detail dominates. Since these metrics weight all image regions equally, large static areas overshadow small but clinically relevant dynamic regions. One approach would be to design new metrics that emphasize temporal change, though such metrics would require reinterpretation and extensive validation. Alternatively, we can add these metrics as additional information. Or we can retrain existing metrics while explicitly modeling the temporal differences themselves. As we will later show, even a spatially imperfect model can achieve a perfect MSE if it reconstructs temporal changes precisely. While we do not aim to redefine evaluation metrics here, this observation motivates our focus on modeling temporal evolution directly through changes via Flow Matching.

2.4 Related Works

In this section, we organize prior work into three categories. We begin by establishing natural imaging baselines, focusing on general spatio-temporal methods and related tasks. This groundwork allows for a smoother transition to the specialized medical imaging approaches covered in the next section. Following the natural imaging baselines, we next cover medical imaging methods, a subset which often faces technical limitations. After discussing these categories, we transition to continuous-time models that underpin much of this thesis. Finally, building on the previous sections, we review continuous-time models. These models form the methodological foundation of this thesis, bridging earlier approaches and informing our proposed methods. Additionally, to round out our overview, we include a brief introduction to data augmentation techniques. These methods will be revisited for deeper discussion in Section 6.2, providing essential context for their relevance. **Natural Imaging Baselines**: For natural image and video prediction, we review a general suite of methods. We then detail the experimental baselines and describe how these methods pool temporal information. ConvLSTMs extend standard LSTMs with convolutional operations for images. They process sequences recurrently and allow variable-length temporal inputs. Neural Processes[37] aggregate observations through a permutation invariant pooling operation, enabling flexible context sizes but discarding explicit temporal order. Vision Transformers[5] encode time via fixed-length temporal tokens, imposing a predefined horizon. SimVP[34] adopts a temporal U-Net architecture, where temporal depth is fixed during training but can be extended recurrently at inference. **Medical Imaging Baselines**: Approaches like Attentive Segmentation Processes Petersen et al. [102] and SADM Yoon et al. [154] are better suited to the clinical task but often incur high computational costs, lack generality, or are subject to sampling constraints. **Continuous Dynamics Generative Models**: Frameworks based on Neural ODEs [13] and Flow Matching [79] model continuous dynamics, and directly inspire our proposed method Temporal Flow Matching (TFM).

2.4.1 Natural Imaging Baselines

Before the deep learning era, statistical filtering and prediction theory provided the foundation for spatio-temporal forecasting. Notably, the Wiener-Kolmogorov filter, introduced in M. and Wiener [88]. This work was a predecessor to the Kalman filter by Kalman [59], which is a recursive algorithm for estimating the state of a dynamic system from noisy observations. The filter introduced recursive estimation for linear dynamical systems, enabling real-time state prediction and correction. Other classical methods like ARIMA and Kalman/Wiener filters are still used today and dominated time-series forecasting before deep learning. A major early theme in deep learning for temporal data was the use of recurrent neural networks (RNNs) and their

variants, such as Long Short-Term Memory (LSTM) by Hochreiter and Schmidhuber [49]. For spatio-temporal data, LSTMs were extended to ConvLSTMs (see [121] and Figure 2.4). These replace fully connected layers with convolutional ones. This was later extended to PredRNN by Wang et al. [142] and Wang et al. [143], allowing memory states to "zigzag" between layers for improved temporal modeling. Le Guen and Thome [75] introduced a method to disentangle physical dynamics from the data, where physical dynamics are disentangled from unknown complementary information. In the work by Gao et al. [34], a plain CNN encoder-decoder with an intermediate translator was proposed. Essentially, time is treated as an additional channel, and convolutional layers are applied to the temporal dimension. Remarkably, this method achieved state-of-the-art performance without recurrent layers, GAN losses, or optical flow, and was trained only on the MSE loss. Furthermore, this method was extended to SimVPv2 by Tan et al. [127]. With the rise of attention Vaswani et al. [137], Transformers were widely adopted for temporal data, and more specifically for video data. Xu et al. [147] used a graph-based spatial transformer to model spatial dependencies, achieving longer forecasting horizons than other methods. Earthformer was proposed by Gao et al. [35]. The method uses a cuboid space-time attention mechanism to tackle high-dimensional spatio-temporal data. It even outperformed ConvLSTMs and numeric simulation baselines. Weissenborn et al. [144] proposed the first autoregressive video transformer for the video generation task. VideoGPT by Yan et al. [150] uses a vector quantization approach to learn discrete video representations, which are then used for video generation.
Gordon and Parde [42] used a latent neural ODE and a GAN for video generation. Generative models have garnered significant attention in recent years, particularly in video prediction and generation. Early generative works propose a conditional diffusion video prediction framework Voleti et al. [138]. Efficient video prediction uses sparsely conditioned Flow Matching for latent video prediction [19]. By approximating the initial conditions of the flow ODE by a noisy version of the previous frame, this method speeds up the inference time. For more efficiency, pyramidal flow matching is suggested in Jin et al. [57]. Ye and Bilodeau [151] propose a continuous stochastic video prediction model; the inherent task is still based on a regular grid, despite the ability of the method to deal with irregular time. Most approaches still struggle with sparse, irregular sampling, or require large datasets. Few have been applied to 3D medical time series. Having introduced our proposed method, in the next section we focus on two families of approaches that we will compare with it. These methods have been applied to medical imaging, chosen for strong performance on natural imaging tasks, or selected for technical abilities.

Convolutional LSTM Long Short-Term Memory (LSTM) networks are a type of recurrent neural network (RNN) designed to learn long-term dependencies in sequential data. They were first introduced by Hochreiter and Schmidhuber [49],

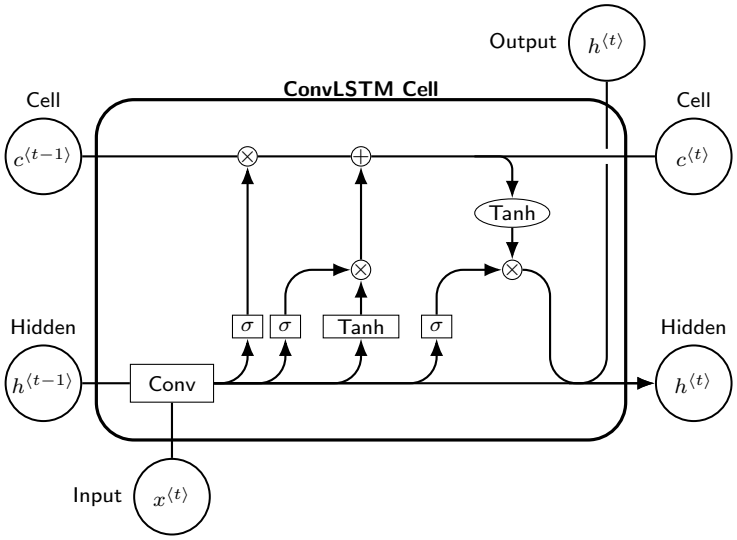


Figure 2.4: Architecture of a Convolutional LSTM (ConvLSTM) cell, which extends the traditional LSTM by replacing matrix multiplications with convolutions, allowing it to capture both spatial and temporal dependencies in sequential image data. At each time step t, the ConvLSTM takes as input the current frame $x^{\langle t \rangle}$, the previous hidden state $h^{\langle t-1 \rangle}$, and the previous cell state $c^{\langle t-1 \rangle}$. The internal gates (input, forget, and output) regulate the flow of information using convolutional operations, updating the hidden state $h^{\langle t \rangle}$ and the memory cell $c^{\langle t \rangle}$. This module is commonly used within the encoder-decoder framework of a UNet to model spatio-temporal sequences, where ConvLSTM layers serve as temporal processing blocks between the spatial downsampling and upsampling paths.

for long sequence modelling. Initially, they were popular for sequences, and tasks like language modelling, e.g. for translation [125]. But LSTMs have also been adapted for image data [121], particularly in video prediction tasks [142]. In Figure 2.4, we can see the general approach of an LSTM cell, together with the convolutional adaptation.

Neural Processes Neural Processes (NPs) are a family of meta learning models that combine the strengths of neural networks and Gaussian processes to learn distributions over functions. They were first introduced in [37], where a short overview can be seen in Figure 2.5. Conditional NPs refine the basic NP by conditioning on the set of observed data points [36]. However, NPs usually underfit, so [63] introduced the Attentive Neural Processes(ANPs), which use attention mechanisms to improve the model's ability to capture complex relationships in the data. [29] refine this further, by omitting the quadratic attention mechanism, and using a latent bottleneck attentive neural process (LBANP). [56] provide a comprehensive overview of

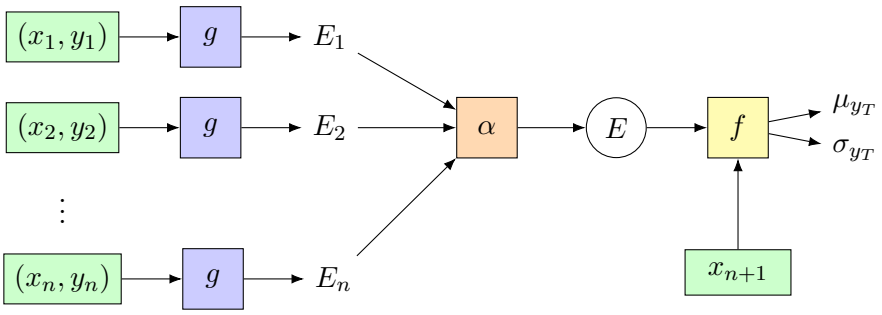


Figure 2.5: Neural Processes, see [37], figure adapted from [56]. (x_i, y_i) denotes indexed input of the data, g is an encoder producing encoded features E_i, α is an aggregator, E are the merged encodings, and f is a decoder. x_{n+1} is the *query* input, and μ_{y_T} is the predicted mean of the target output, with σ_{y_T} being the predicted standard deviation. In most of our cases, x corresponds to time and y to the image, but the basic NPs can be used for any kind of data.

the NP family. Furthermore, [102] uses an Attentive Segmentation Process (ASP) to segment gliomas in MRI scans, which is a specific application of the ANP. We will discuss the latter method in more detail in a later section.

SimVP SimVP is a state-of-the-art (SOTA) method for video prediction, see [34]. The basic concept of SimVP is to treat the temporal dimension on par with the spatial dimensions. In practice, this means the time dimension is used as an additional channel, and convolutional layers are applied to it as well. In the latent space, time is treated differently, where a translator learns the temporal evolution [34]. We can broadly describe these methods and variations as 4D CNNs.

Video Vision Transformers The Video Vision Transformer (ViViT) was first introduced for video classification in [5]. Later, it was adapted for video prediction in medical applications, such as in [154], which will be discussed in more detail in Section 2.4.2. We briefly describe how ViViT processes spatio-temporal data. Let $x \in \mathbb{R}^{T \times C \times S}$ denote a spatio-temporal sequence with T frames, C channels, and $S = H \cdot W \cdot D$ spatial voxels. Define a patch size $P = (p_1, p_2, p_3)$, and let $N = S/(p_1 p_2 p_3)$ be the number of patches. Each patch is flattened and linearly projected, resulting in $x \in \mathbb{R}^{(T \cdot N) \times m}$ where m is the internal dimension of the transformer. A temporal transformer operates on this sequence of length $T \cdot N$, and the dimensionality m is progressively reduced across layers. The output is then rearranged as $x \in \mathbb{R}^{m \times (T \cdot N)}$, and passed through several layers of a spatial transformer. Finally, the output is upsampled using a fixed scale factor of $p_1 \cdot p_2 \cdot p_3$, and reshaped to the original spatial dimensions. *Notably* this scale factor can be quite

large. For example, [154] uses a patch size of $8 \times 32 \times 32$, leading to extremely coarse representations of the input data.

2.4.2 Medical Imaging Baselines

One of the more prominent approaches to spatio-temporal modeling is classification for disease progression. One of the earlier works is by Lipton et al. [81], who use an LSTM to diagnose patients from multivariate EHR time series data, thereby establishing LSTMs as a solid baseline for temporal modeling. Hao and Negahdar [44] extend this idea by combining localized LSTM layers with a joint spatio-temporal attention mechanism, forecasting long-COVID outcomes from heterogeneous modalities including laboratory values, vital signs, demographics, medical history, and imaging. Konwer et al. [67] further improves image-based prediction by embedding explicit disease-stage representations into their networks. Ouyang et al. [99] propose a global pooling operation for RNNs that aggregates features across all time steps into a single representation, which they apply to predict Alzheimer's disease, and disease states from patients in the National Consortium on alcohol and Neurodevelopment in Adolescence (NCANDA) study. Xu et al. [149] employs a bi-directional GRU on serial CT scans to predict lung cancer treatment response, demonstrating consistent performance gains with each additional follow-up scan. Multimodal frameworks, such as those by Lu et al. [87] and Muksimova et al. [94], integrate imaging, clinical, and demographic data over extended intervals for early Alzheimer's disease detection, while [26] predict a scalar atrophy index capturing progressive hippocampal volume loss.
Convolutional LSTMs have been applied to pixel-wise forecasting Zhang et al. [156], use ConvLSTMs to model tumor-growth trajectories. Although they implicitly assume regularly spaced scans and report somewhat noisy volume predictions on a very "well-behaved" dataset (tumor volumes change at near-constant rates or change little at all), their work is influential, motivating our use of an open-source ConvLSTM baseline in our experiments. Lachinov et al. [73] propose a continuous-time Neural ODE framework for continuous image segmentation at future time points. However, while their method is limited to single images and their code is not open source, direct comparison with their specific method is not possible. Other recent research has explored multi-input and time-conditioned approaches. A method using multiple inputs and time conditioning by Chen et al. [17] predicts the growth of vestibular schwannomas using multiple longitudinal MRI scans. While their method is time-continuous and publicly available, it has primarily been evaluated on in-house data with limited information on the regularity or irregularity of the scan intervals. Their approach focuses on predicting segmentation masks by regressing signed distance fields and incorporates the LCI into their metrics. Beyond direct prediction, lesion-tracking methods, such as DeepLesion Tracker by Cai et al. [11], and change map networks for chest CT images by [64] have been explored for longitudinal tasks.

Self-supervised representation learning has emerged as a powerful approach for temporal medical imaging, leveraging the rich information available across multiple time points without requiring explicit annotations. Recent work by Shen et al. [120] demonstrates how spatiotemporal features can be effectively learned from longitudinal imaging data, capturing both spatial structures and their temporal evolution patterns. Generative architectures are increasingly leveraged for longitudinal synthesis. [116] combine a clinically informed latent vector with a temporal embedding to generate complete image sequences, albeit given only a single image. Yoon et al. [153] proposes a latent diffusion module for superresolution to improve AD diagnosis from single MRI scans. Zhu et al. [162] employ a diffusion model augmented with a temporal-consistency module and an age embedding to generate realistic inpaintings for adolescent brain MRI by interpolating between a preceding and a subsequent scan. Litrico et al. [82] introduce Temporally Aware Diffusion Model (TADM) to forecast future MRIs in AD and MCI patients; however, their method remains two-dimensional and accepts only a single input image per prediction. A more in-depth method for AD progression is BrLP from Puglisi et al. [106], which is a Latent Diffusion Model with a control net, and a latent auxiliary model. Again, this model only parses single images, and it is dependent on a latent module which is specific to Alzheimer's disease. A significant challenge in this domain is the limited availability of ready-to-use medical imaging datasets. Unlike natural imaging where numerous preprocessed datasets exist, medical datasets typically require substantial preprocessing and curation before analysis. This creates barriers to reproducibility as researchers implement inconsistent processing pipelines. For example, while Alzheimer's datasets like ADNI and OASIS contain valuable longitudinal data, they require extensive preprocessing, and publications often inadequately document patient selection criteria and preprocessing steps, hampering fair comparison between methods.

State of The Art Our review of the state of the art in spatio-temporal and longitudinal medical imaging identifies several critical technical requirements that no existing method satisfies simultaneously:

- **Limited Temporal Input Capacity:** Current methodologies predominantly utilize single-image inputs due to computational constraints. While adequate for simple dynamics, this fundamental limitation prevents effective modeling of complex temporal relationships. Theoretical analyses confirm that even for basic growth patterns, single-timepoint observations are insufficient to accurately characterize temporal evolution and progression trajectories.
- **Irregular Temporal Sampling:** Despite the abundance of methods for processing medical video data, few approaches effectively handle multiple irregularly-spaced timepoints. Most existing frameworks assume uniform temporal sampling intervals and encounter significant difficulties when processing

acquisitions with variable or missing timepoints—a common scenario in clinical settings.

- **Computational Inefficiency:** State-of-the-art approaches often require substantial computational resources, limiting their practical deployment and iterative refinement. More critically, there exists a fundamental inefficiency in how these models allocate capacity. As evidenced by information-theoretic analyses (per our LCI discussion), medical images exhibit minimal temporal changes relative to their static content. Consequently, models may disproportionately allocate resources to modeling invariant anatomical structures rather than the clinically relevant temporal variations, resulting in suboptimal prediction performance despite high computational demands.
- **Disease-Specific Architectural Biases:** A significant proportion of current methods are optimized specifically for Alzheimer's disease progression, incorporating implicit assumptions about characteristic neurodegenerative patterns. While effective for their intended application, these architectural biases limit generalizability to other pathologies with different progression patterns. A universally applicable model requires disease-agnostic design principles that can adapt to diverse temporal dynamics.
- **Discrete Temporal Modeling:** Predominant approaches employ discrete-time models that characterize transitions between fixed timepoints or intervals. This discretization presents significant limitations when modeling clinical data with highly variable acquisition intervals, as naive zero-filling approaches create extremely sparse, high-dimensional representations that are computationally inefficient and statistically suboptimal.

The proposed discrete TFM framework addresses the first four technical limitations directly, and the continuous variant solves the last. Regarding evaluation methodology, we encountered a significant gap in the literature. To our knowledge[4], no existing methods have been benchmarked on *exactly* the problem formulation described in Section 2.3.

2.4.3 Related Works: Synthetic Data Generation

Synthetic data generation has emerged as an increasingly important tool for deep learning applications. Goncalves et al. [41] explore the different types of synthetic data and their respective applications in machine learning contexts. Xu et al. [148] propose a theoretical framework to describe synthetic data generation processes. The authors demonstrate that "the synthetic feature distribution does not need to be

[4]While some segmentation approaches like [102] handle irregular sampling, and progression models such as [115] perform similar tasks, they lack detailed documentation of synthetic experiments and sampling strategies. Furthermore, their autoencoder-based methods without spatial residual connections may be suboptimal for MSE evaluation, as we demonstrate in Section 5.1.

similar to that of real data for ensuring comparable generalization of synthetic models, provided proper model specifications in downstream learning tasks." In Ouyang et al. [100], the authors propose a contrastive loss approach for data generation to improve robustness against adversarial attacks. Nguyen et al. [95] propose utilizing pre-trained networks, specifically Stable Diffusion, for semantic segmentation tasks, addressing the labor-intensive nature of dataset creation in natural imaging. Their approach generates pseudo labels that serve as a foundation for pre-training on MSCOCO and PASCAL VOC datasets. Kapania et al. [60] provide a general overview of synthetic data applications and their expanding role in deep learning research and applications. Synthetic data has become increasingly important in medical imaging due to real data limitations and privacy considerations. Giuffrè and Shung [40] discuss synthetic data's role in healthcare, emphasizing its importance for privacy preservation and model performance enhancement. The authors highlight the potential of digital twins for the healthcare sector while acknowledging risks that may limit practical applications. For specific tasks, anatomical phantoms have been developed to simulate realistic structures, as demonstrated in [32] and Segars et al. [119]. These phantoms enable researchers to address biological questions while maintaining complete control over the data generation process. Oakden-Rayner et al. [98] demonstrate the utility of synthetic data for detecting clinically relevant model failures. Segal et al. [118] propose methods for using synthetic data in controlled benchmarking of medical models, addressing model robustness, fairness, and generalizability while contributing a reproducible, interpretable, and configurable tool designed to advance reliable ML deployment in clinical settings.

2.5 Continuous Time Models

In this section, we present a comprehensive overview of Neural Ordinary Differential Equations (Neural ODEs) and Flow Matching (FM), adopting the notation from Lipman et al. [80]. We begin by examining continuous-time models based on Neural ODEs. These models are particularly suited for irregularly sampled time series, a scenario prevalent in longitudinal medical imaging. We next transition to continuous-time *generative* models. Although these generative models are not commonly applied to time series data, they are essential for the iterative generation of high-dimensional data categories such as images. To illustrate, we first introduce diffusion models, which have gained widespread attention in recent years for producing highly realistic samples and have become a staple in generative modeling literature. Subsequently, we explore Flow Matching, which serves as the methodological foundation of this thesis. Finally, we provide a brief introduction to Schrödinger Bridge Matching, a framework that conceptually unifies Flow Matching and diffusion models. Although Schrödinger bridges were historically introduced earlier, recent advances such as simulation-free Schrödinger Bridges [133] have significantly

improved their practical viability.
Unless otherwise noted, all definitions and notation in this chapter adhere strictly to those in the referenced works; any modifications are explicitly indicated. Despite appearing unrelated at first glance, both Neural ODEs and Flow Matching share a commonality: both methods are designed to learn continuous-time vector fields, although they target different aspects of the ODE. Somewhat incidentally, the development of this thesis begins with Neural ODEs and ultimately culminates in Flow Matching-based methods. We begin with an overview of Neural ODEs in Section 2.5.1, followed by a detailed discussion of Flow Matching in Section 2.5.3. We conclude the chapter by comparing the two approaches and demonstrating how Flow Matching *contains* the LCI, motivating its use for longitudinal image prediction.

2.5.1 Neural ODEs

Neural ODE (NODE) (proposed in [13]) are commonly introduced as continuous-time models, in which the evolution of a system is governed by a differential equation. The central idea is to replace a sequence of transformations (as in a residual network) with a continuous dynamical system whose evolution is parametrized by a neural network. In this formulation, we denote by X_t the state at time t and by θ the neural network parameters. This notation is adopted for consistency with the Flow Matching framework discussed later. While solving such equations numerically was a key technical challenge in the original work, we briefly summarize the core formulation as follows:

$$\frac{dX_t}{dt} = f(X_t, t, \theta), \tag{2.34}$$

given an initial state X_{t_0} at time t_0, and a function f which is parameterized by θ. The final state X_{t_1} is obtained by solving the ODE, see [13]:

$$X_1 = X_0 + \int_{t_0}^{t_1} f(X_t, t, \theta)dt = \text{ODESolve}(f, X_{t_0}, t_0, t_1, \theta). \tag{2.35}$$

Training NODE requires computing gradients of a loss with respect to the parameters θ. This is achieved using the adjoint method [13], which backpropagates through the ODE solver by solving another differential equation backward in time. Let $a_t = \frac{\partial L}{\partial X_t}$ denote the adjoint state, its dynamics are given by

$$\frac{da_t}{dt} = -a_t^T \frac{\partial f(X_t, t, \theta)}{\partial X_t}. \tag{2.36}$$

The gradient w.r.t. the parameters is then obtained from a third integral:

$$\frac{dL}{d\theta} = -\int_{t_1}^{t_0} a_t^T \frac{\partial f(X_t, t, \theta)}{\partial \theta} dt. \tag{2.37}$$

Algorithm 1 Reverse-mode derivative of an ODE initial value problem

Require: dynamics parameters θ, start time t_0, stop time t_1, final state X_1, loss gradient $\frac{\partial L}{\partial X_1}$

$s_0 \leftarrow \left[X_1,\ \frac{\partial L}{\partial X_1},\ 0_{|\theta|}\right]$

function AUG_DYNAMICS($[X_t,\ a_t,\ \cdot],\ t,\ \theta$)

  return $\left[f\big(X_t, t, \theta\big),\ -a_t^\top \frac{\partial f}{\partial X},\ -a_t^\top \frac{\partial f}{\partial \theta}\right]$

end function

$[X_0,\ \partial L/\partial X_0,\ \partial L/\partial \theta] \leftarrow \text{ODESolve}\big(s_0,\ \text{aug\_dynamics},\ t_1,\ t_0,\ \theta\big)$

return $\frac{\partial L}{\partial X_0},\ \frac{\partial L}{\partial \theta}$

As shown in [13], all quantities X, a and $\frac{\partial L}{\partial \theta}$ can be computed by a single call to the ODE. This formulation enables neural networks to model continuous-time dynamics while maintaining full differentiability.

Extensions A simple yet elegant extension is the Augmented Neural ODEs [28], which expands the latent state to a higher-dimensional space:

$$\frac{d}{dt}\begin{pmatrix} X_t \\ b_t \end{pmatrix} = f\left(\begin{pmatrix} X_t \\ b_t \end{pmatrix}, t, \theta\right), \tag{2.38}$$

where b_t is the augmented state with $b_0 = \mathbf{0}$. [28] demonstrate that Augmented NODEs enable modeling of a broader class of problems, thereby addressing specific topological limitations.

Continuous Normalizing Flows (CNFs) NODE are closely related to Continuous Normalizing Flows (CNFs), which extend discrete normalizing flows to the continuous domain. Traditional normalizing flows e. g. [107] model a sequence of invertible transformations between latent variables and data through discrete layers. By contrast, CNFs replace this discrete sequence with a continuous transformation governed by an ODE (2.35). This formulation enables a continuous change of variables that simplifies the computation of the log-density transformations:

Theorem 2.1. (Instantaneous change of variables (from [13, Theorem 1]))Let X_t be a finite continuous random variable with probability density function $p_t(X)$ dependent on time. Let $\frac{dX}{dt} = f(X_t, t)$ be a differential equation describing a continuous in time transformation of X_t. Assuming that f is uniformly Lipschitz continuous in X and continuous in t, then the change in log proba-

bility also follows a differential equation:

$$\frac{\partial \log p_t(X_t)}{\partial t} = -tr\left(\frac{df}{dX_t}\right). \tag{2.39}$$

This theorem provides the continuous analogue [13] of the discrete change-of-variables formula used in standard flows. For example, the planar normalizing flow introduced in [107] with

$$X(t+1) = Z(t) + uh(w^T X_t + b), \tag{2.40}$$

the continuous counterpart becomes

$$\frac{dX_t}{dt} = uh(w^T X_t + b), \quad \text{where } \frac{\partial logp(X_t)}{\partial t} = -u^T \frac{\partial h}{\partial X_t}. \tag{2.41}$$

Controlled Neural ODEs A related extension, particularly relevant for *irregularly sampled time series*, is the **Controlled Neural ODE** by [62]. Instead of solely as a function of time, the latent state is driven by an external control signal Z_τ:

$$X_t = X_0 + \int_{t_0}^{t} f(X_\tau, \theta) dZ_\tau, \tag{2.42}$$

Here, Z_τ represents the input trajectory, which can be constructed using a continuous interpolation such as a cubic spline with knots at the observation times $t_0, \ldots, t_1$. Under this formulation, the CDE can be equivalently expressed as

$$X_t = X_0 + \int_{t_0}^{t} f(X_\tau, \theta) \frac{dZ}{d\tau}(\tau) d\tau, \tag{2.43}$$

which can be solved using a standard ODE solver in the same manner as (2.35). CDEs thus generalize Neural ODEs by allowing multiple temporally distributed inputs to influence the latent trajectory, rather than just the initial state[5].

Practical Remarks and Limitations In practice, most NODE implementations rely on black-box ODE solvers, which can substantially increase both training and inference times, even with adapted solvers [103]. Several extensions have been proposed to address these limitations, such as Augmented Neural ODEs [28], locally regularized adaptive solvers [101], or hybrid discretization schemes. Nevertheless, memory and computational efficiency remain major bottlenecks, especially in the imaging domain. A representative example is ImageFlowNet [83], which applies NODE to 2D image forecasting. While interesting, even the 2D formulation is computationally demanding, and hence not easily applicable to typical 3D medical image time series.

[5]Recurrent NODEs could in practice still be used, but they impose jumps in the latent trajectory

Connection to Flow Matching Equation (2.35) also reveals a conceptual link to Flow Matching (FM): Both frameworks model data evolution via continuous-time vector fields. However, while NODE learn trajectories from known initial and final states via supervision, FM directly learns the vector field that transports one distribution to another. We will elaborate on this relationship in the following section.

2.5.2 Diffusion Models

Diffusion models constitute a prominent class of generative methods that learn to reverse a predefined forward-noising process. Formally, these models define a stochastic forward process via a stochastic differential equation (SDE) that gradually transforms data samples into Gaussian noise. This process also models the diffusion of particles, hence the sharing of name. A neural network, often referred to as the diffusion model or score network, is then trained to approximate the score function required to reverse this process through a corresponding reverse-time SDE. The widespread success of diffusion models in generative modeling stems from earlier state-of-the-art performance in high-dimensional data synthesis, particularly in image generation. Their connection to continuous-time generative modeling and to flow matching will be revisited later.

Remark: Although this thesis does not directly build on diffusion models, many of our baselines and comparative methods are. Furthermore, there is an interesting connection between the Flow and Diffusion models, as well as a mathematical background. We therefore include a brief mathematical overview to contextualize their relevance and to deepen the understanding. Diffusion models formulate generative modeling as the task of reversing a fixed noising process that progressively transforms data into Gaussian noise Ho et al. [47]. While these models have set the state of the art (SOTA) in image generation, more recent approaches, particularly those in FM, offer a conceptually more natural formulation for modeling medical temporal evolution.

Diffusion Background

Diffusion models define a probabilistic generative process by reversing a gradual noising procedure. Given a sample $x_0 \sim q$, a *forward* Markov chain progressively perturbs the data into pure Gaussian noise over T steps. The forward process is

$$\begin{aligned} q(x_{1:T} \mid x_0) &:= \prod_{t=1}^{T} q(x_t \mid x_{t-1}), \\ \text{where } q(x_t \mid x_{t-1}) &= \mathcal{N}(x_t \mid \sqrt{1-\beta_t} x_{t-1}, \beta I), \end{aligned} \tag{2.44}$$

according to a variance schedule $\beta_1, \ldots, \beta_T \in [0, 1]$, and I the identity matrix. Diffusion models approximate the posterior distribution $p_\theta(x_{0:T})$, referred to as the *forward* or *denoising* process. [6] The reverse or generative process is parametrized as

$$\begin{aligned} p_\theta(x_{0:T}) &\coloneqq p(x_T) \prod_{t=1}^{T} p_\theta(x_{t-1} \mid x_t), \\ \text{where } p_\theta(x_{t-1} \mid x_t) &= \mathcal{N}(x_{t-1} \mid \mu_\theta(x_t, t), \Sigma_\theta(x_t, t)). \end{aligned} \tag{2.45}$$

A notable property of the forward process is that it permits closed-form sampling at any arbitrary step t. Defining $\alpha_t \coloneqq 1 - \beta_t$ and $\bar{\alpha}_t \coloneqq \prod_{s=1}^{t} \alpha_s$, one can directly compute the marginal distribution $q(x_t \mid x_0)$ without simulating all previous transitions. The marginal distribution can be computed directly as

$$q(x_t \mid x_0) = \mathcal{N}(x_t \mid \sqrt{\bar{\alpha}_t} x_0, (1 - \bar{\alpha}_t) I). \tag{2.46}$$

Training minimized a simplified noise-prediction objective

$$\begin{aligned} &\mathcal{L}_\theta = \mathbb{E}_{x_0, \epsilon, t} \left\| \epsilon - \epsilon_\theta \left(\sqrt{\bar{\alpha}_t} x_0 + \sqrt{1 - \bar{\alpha}_t} \epsilon, t \right) \right\|, \\ &\text{where } x_0 \sim q, \epsilon \sim \mathcal{N}(0, I), t \sim U(\{1, \ldots, T\}). \end{aligned} \tag{2.47}$$

This objective, introduced by Ho et al. [47], encourages the neural network ϵ_θ to predict the noise added at a particular diffusion step t, rather than the denoised image itself.

This formulation allows efficient training because the noisy sample at any diffusion step can be computed in closed form, avoiding sequential simulation. During inference, generation proceeds by iteratively reversing the diffusion process over T discrete steps. Starting from Gaussian noise $x_T \sim \mathcal{N}(0, I)$, the model applies the learned reverse transitions to progressively remove noise. Each reverse step reintroduces a small amount of randomness, which, although counterintuitive, improves sample diversity and stability. The denoising sample is calculated as

$$x_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(x_t - \frac{1 - \alpha_t}{\sqrt{1 - \bar{\alpha}_t}} \epsilon_\theta(x_t, t) \right) + \sigma_t z, \quad z \sim \mathcal{N}(0, I), \tag{2.48}$$

where σ_t controls the noise at step t and can follow different schedules [47, 122]. In continuous time, this process can be described by a reverse-time stochastic differential equation (SDE),

$$dX_t = \left[f(X_t, t) - g(t)^2 \nabla \log p_t(X_t) \right] dt + g(t) dW_t. \tag{2.49}$$

where the *score function* $\nabla \log p_t(X_t)$, the gradient of the log-density with respect to the current state, is approximated by a neural network s_θ. Furthermore, f is

[6] Interestingly, this noising process need not be limited to Gaussian perturbations; alternative forms such as blurring or cropping have also been explored [6].

the drift and g is the diffusion coefficient. As shown by Song et al. [122], for every diffusion process defined by a stochastic differential equation (SDE), there exists a corresponding *deterministic* process. This alternative formulation shares the same marginal distributions as the original SDE at all steps t. The resulting deterministic dynamics are governed by the so-called *probability flow ODE*, given by:

$$dX_t = \left[f(X_t, t) - \frac{1}{2} g(t)^2 \nabla \log p_t(X_t) \right] dt. \tag{2.50}$$

This equivalence links diffusion models to deterministic continuous-time approaches such as Flow Matching, providing a conceptual link between stochastic and ODE-based generative formulations.

Classifier Free Guidance Dhariwal and Nichol [22] demonstrated that diffusion models surpass GANs in image synthesis quality, particularly when incorporating classifier guidance to steer the generation process toward desired outputs. However, this approach introduces additional computational overhead, as it requires training and storing an additional classifier. An elegant alternative is **classifier-free guidance (CFG)** by Ho and Salimans [48], which removes the dependency on an auxiliary model. Here, the diffusion network is trained both *conditionally* and *unconditionally* by randomly dropping the conditioning signal during training. At inference, guidance is obtained by linearly combining the conditional and unconditional predictions:

$$\tilde{\epsilon}_\theta(x_t, c) = (1 + w)\epsilon_\theta(x_t, c) - w\epsilon_\theta(x_t, c = \varnothing), \tag{2.51}$$

where w controls the strength of conditioning. Larger w values produce samples that are more faithful to the conditioning, at the expense of diversity.

2.5.3 Flow Matching

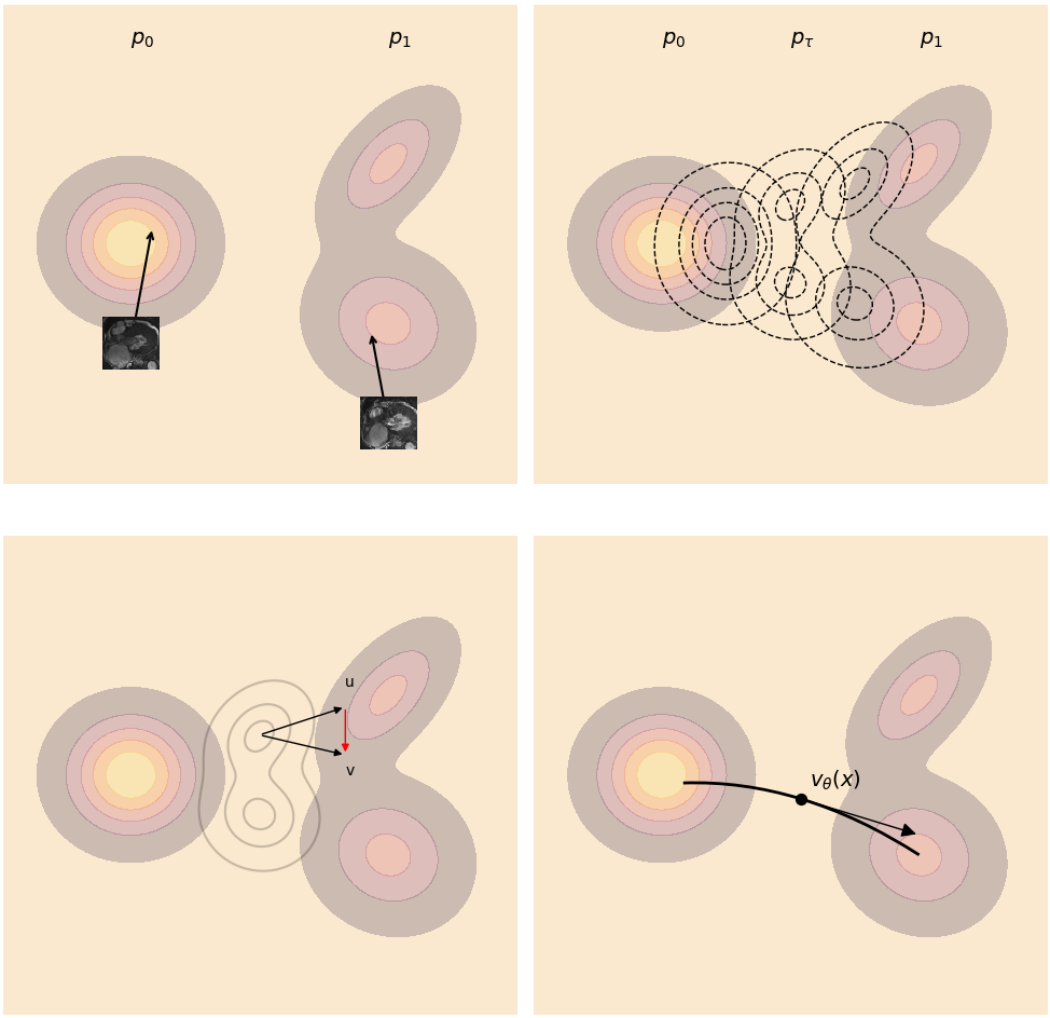


Figure 2.6: **Flow Matching schematic**. Shown is the *unconditional* Flow Matching process. (top left) Data: samples from source distribution p_0 and target distribution p_1, illustrated with medical images. (top right) Path design: interpolated densities p_τ between p_0 and p_1 define a continuous trajectory. (bottom left) Training: the model learns a time-dependent velocity field from the interpolated pairs. (bottom right) Sampling: new samples are generated by integrating the learned velocity field v_θ.

We will adopt FM as the central modeling framework for the main method proposed in this thesis. Our initial motivation for selecting FM was its recent success and novelty in image generation; subsequent results revealed properties that make it

particularly suitable for our setting. Most notably, its ability to model continuous dynamics.

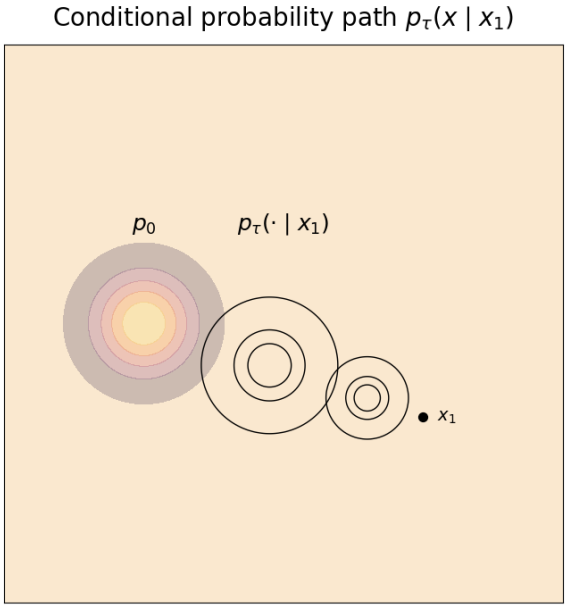


Figure 2.7: **Illustration of Flow Matching with conditional probability paths**. The diagram depicts the conditional distribution path $p_\tau(\cdot \mid x_1)$ evolving from the prior distribution p_0 (left) towards the target point x_1 (right) over time τ. Intermediate contour levels represent the progression of the conditional distribution as it transitions smoothly between the prior and the target, following the conditional Flow Matching process.

Flow Matching Background

We follow the notation of Lipman et al. [80]; Let $p_1 = q$ denote the data distribution on $\mathbb{R}^S$, and let $p_0 = p$ be a simple prior (for example $\mathcal{N}(0, I)$)[7]. Our goal is to generate new samples p_1 by learning a distribution path $(p_\tau)_{0 \leq \tau \leq 1}$ which transports $p_0 \to p_1$. FM learns a *step-dependent* vector field $u : [0, 1] \times \mathbb{R}^S \to \mathbb{R}^S$, which is parametrized and approximated via a neural network v_θ. The induced flow transports p_0 to p_1. **As a machine learning objective, we aim to learn an idealized ground truth flow.** This velocity field determines a *step* dependent

[7]For our application, the simple prior is not random noise but rather the patient's previous image in the sequence. This aligns the prior with an imaging modality (Dirac delta centered at that sample), differing from standard probabilistic priors.

flow field $\psi : [0,1] \times \mathbb{R}^S \to \mathbb{R}^S$, defined as

$$\frac{d}{d\tau}\psi_\tau(x) = u_\tau(\psi_\tau(x)), \tag{2.52}$$

satisfying $\psi_\tau(X_0) = X_\tau \sim p_\tau$. During inference, we draw $X_0 \sim p_0$, and integrate the learned velocity field from $\tau = 0$ to $\tau = 1$ in order to obtain $X_1 = \psi_1(X_0) \sim p_1$. We deliberately use τ to denote the FM step to avoid confusion with real acquisition time, and t is reserved exclusively for actual timesteps of image acquisition[8]. The velocity field u_τ *generates* the probability path p_τ if its flow ψ_τ satisfies the following transport equation:

$$X_\tau \coloneqq \psi_\tau(X_0) \sim p_\tau, \quad \text{for } X_0 \sim p_0. \tag{2.53}$$

For the basic Flow Matching setup, let the source distribution be $p_0 \sim \mathcal{N}(x|0, I)$ and construct the probability path p_τ as the aggregation of the conditional probability paths $p_{\tau|1}(X \mid x_1)$, each conditioned on one of the data examples x_1 from the target distribution p_1. The probability path p_τ therefore follows

$$p_\tau(x) = \int p_{\tau|1}(x|x_1)p_1(x_1)dx_1, \quad \text{where } p_{\tau|1}(x|x_1) = \mathcal{N}(x|\tau x_1, (1-\tau)^2 I). \tag{2.54}$$

An overview of each object is illustrated in Figure 2.6. Then, the unconditional Flow Matching loss reads:

$$\mathcal{L}_{FM} \coloneqq \mathbb{E}_{\tau, X_\tau}||v_\theta(X_\tau, \tau) - u_\tau(X_\tau)||_2^2, \quad \text{where } \tau \sim \mathcal{U}(0,1), X_\tau \sim p_\tau. \tag{2.55}$$

Importantly, this joint probability distribution is infeasible to calculate in practice, as we would need to evaluate (2.54) **for the whole dataset.** Instead, we consider the conditional velocity field. We define the random variable $X_\tau \sim p_\tau$ by drawing $X_0 \sim p_0$, $X_1 \sim p_1$ and then calculating their linear combination:

$$X_\tau = \tau X_1 + (1-\tau)X_0 \sim p_\tau. \tag{2.56}$$

This path, often called the *conditional optimal transport path* or *linear path*, has convenient properties discussed in [80]. Now, the training objective for Flow Matching is to learn the velocity field u_τ, which generates the probability path p_τ. We use equation (2.56) to manifest the conditional random variables

$$X_{\tau|1} = \tau x_1 + (1-\tau)X_0 \sim p_{\tau|1}(\cdot|x_1) = \mathcal{N}(\cdot|\tau x_1, (1-\tau)^2 I). \tag{2.57}$$

Plugging the conditional path into the ODE

$$\frac{d}{d\tau}X_{\tau|1} = u_\tau\big(X_{\tau|1} \mid x_1\big) \tag{2.58}$$

[8] See later 4.4

yields the *conditional velocity field*

$$u_\tau(x \mid x_1) = \frac{x_1 - x}{1 - \tau}. \tag{2.59}$$

An illustration of the conditional path and velocity is shown in Figure 2.7. Evaluated on samples from the path, this simplifies to the *sample-wise target*

$$u_\tau\big(X_\tau \mid x_1\big) = x_1 - X_0, \tag{2.60}$$

which we use for regression. Equipped with the conditional velocity field from equation (2.59), we can formulate a nice version of the Flow Matching loss from equation (2.55):

$$\mathcal{L}_{CFM} = \mathbb{E}_{\tau, X_0, X_1} ||v_\theta(X_\tau, \tau) - u_\tau(X_\tau | X_1)||_2^2, \tag{2.61}$$

where $\tau \sim \mathcal{U}(0,1), X_0 \sim \mathcal{N}(0, I), X_1 \sim p_1$. We repeat the following important results:

Theorem 2.2 (Theorem 1 in [80])**.** The training objective in equation (2.61) and the unconditional Flow Matching loss in equation (2.55) have the same gradients with respect to the parameters θ of the neural network v_θ, i. e.

$$\nabla_\theta \mathcal{L}_{FM}(\theta) = \nabla_\theta \mathcal{L}_{CFM}(\theta). \tag{2.62}$$

Finally, plugging in the conditional velocity field from equation (2.59) into the training objective in equation (2.61) leads to the final and simple Flow Matching loss:

$$\mathcal{L}_{CFM}^{OT} = \mathbb{E}_{\tau, X_0, X_1} ||v_\theta(X_\tau, \tau) - (X_1 - X_0)||_2^2, \tag{2.63}$$

with $\tau \sim \mathcal{U}(0,1), X_0 \sim \mathcal{N}(0, I), X_1 \sim p_1$.

Diffusion formulated as Flow Matching As noted in the introduction paper for Flow Matching [79], Gaussian diffusion models can be formulated as a special case of Flow Matching. There, the velocity field is calculated via [79, Eq. 19]:

$$u_t(x \mid x_1) = \frac{\alpha'_{1-\tau}}{1 - \alpha_{1-\tau}^2} (\alpha_{1-\tau} x - x_1) = \frac{B'(1-\tau)}{2} \left[\frac{e^{-B(1-\tau)} x - e^{-\frac{1}{2}B(1-\tau)} x_1}{1 - e^{-B(1-\tau)}} \right], \tag{2.64}$$

where $B(t) = \int_0^t \beta(s) ds$, and $f' = \frac{d}{d\tau} f$.

LCI and Flow Matching

Proposition 2.3. Let I_0 be the context image and I_1 the target image. If the velocity field is identically zero,

$$v_\theta(x, \tau) = 0 \quad \forall\, x, \tau, \tag{2.65}$$

then the ODE

$$\frac{dX_\tau}{d\tau} = v_\theta(X_\tau, \tau) = 0, \quad X_{\tau=0} = I_0 \tag{2.66}$$

has the constant solution $X_\tau \equiv I_0$. In particular,

$$\hat{X}_1 = I_0, \tag{2.67}$$

so a "zero" Flow-Matching model *predicts* the LCI.

As mentioned, LCI is a simple yet surprisingly strong heuristic, as changes are typically small relative to the static background. Proposition 2.3 formalizes the connection to *Flow Matching*: if the learned velocity field $u_\theta(x, \tau)$ vanishes for all x and τ, the dynamics reduce to the constant solution of X_0. Thus, every predicted frame remains identical to the starting point, and the model degenerates to the LCI. This equivalence is useful for two reasons:

1. It theoretically provides a *soft lower bound* for Flow Matching based time series forecasting methods. In the worst case, the method can fall back to LCI
2. It offers an *interpretability anchor*: The extent to which FM surpasses LCI, reflects the capacity to learn spatio-temporal relevant changes.

Hence, given that the LCI appears to be a strong empirical heuristic, Flow Matching correspondingly provides a inductive bias and starting point for modeling temporal evolution.

Aspect	**Neural ODEs**	**Flow Matching (FM)**
Learned object	Vector field $f_\theta(X, t)$	Velocity field $v_\theta(X, \tau)$
ODE Formulation	$\frac{dX_t}{dt} = f_\theta(X_t, t)$	$\frac{dX_\tau}{d\tau} = v_\theta(X_\tau, \tau)$
Training regression	Final state	Target velocity u_τ

Table 2.1: Conceptual comparison between Neural ODEs and Flow Matching (FM). Both approaches learn a vector field defining continuous dynamics, but differ in how the field is trained.

Comparing NODE to FM While Neural ODEs and Flow Matching share the same underlying ODE formulation, they differ in training strategy. NODE learn implicitly

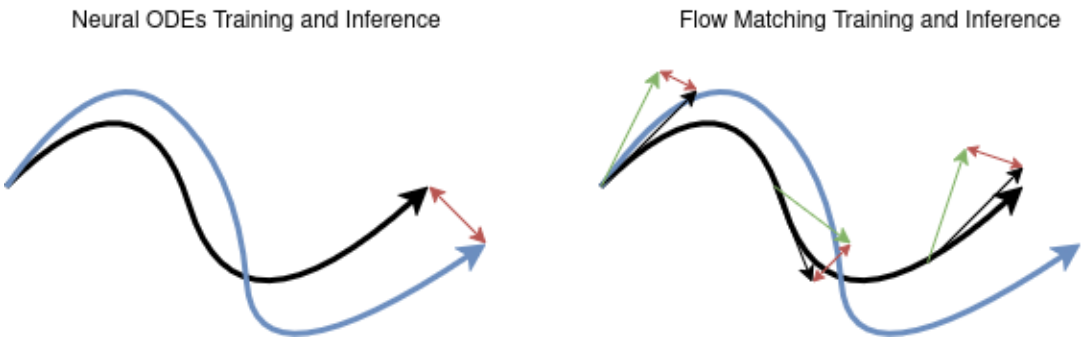


Figure 2.8: **Comparison of Neural ODEs (left) and Flow Matching (right)**. The black curve is the ground truth, together with its ground truth velocity u_τ as black arrows, and the blue curve is the trajectory over τ of the prediction. The red arrow indicates the training objective or the loss. For FM the green arrows indicates the instantaneous velocity field v_θ at time τ at the state X_τ

by enforcing that the integrated trajectory matches the final state, whereas Flow Matching learns explicitly by regressing toward a velocity field. Additionally, for Flow Matching, we explicitly construct the ground-truth velocity, whereas NODEs only learns the final state. Figure 2.8 and Table 2.1 summarize these conceptual differences.

2.5.4 Schrödinger Bridge Matching

Schrödinger Bridge models (SBs) offer a principled framework for generative modeling. Originally proposed in [117], a modern discussion can be found by Chen et al. [16], the problem is the following: Given two marginal distributions of objects at different states, what is the most likely evolution between them under the constraint that the dynamics remain close to a reference stochastic process, such as Brownian motion? Modern reformulation casts this as a problem of *stochastic optimal control*, where one seeks to interpolate between two distributions p and q, using a stochastic process P, while minimizing the Kullback-Leibler divergence to a reference process R:

$$P^* = \arg\min_{P \in \mathcal{P}} \mathrm{KL}(P|R) \quad \text{subject to} \quad P(x_0) = q,\ P(x_T) = p. \tag{2.68}$$

Classically, solving this required iterative algorithms such as *Iterative Proportional Filtering* (IPF), which alternates between a forward and backward conditioning to update drift terms. However, for image generation, this approach is computationally expensive and often impractical. Recent advances by Tong et al. [133] have introduced simulation-free Schrödinger bridges that leverage neural networks to approximate the optimal transport path between the two distributions. Then, the loss

is calculated via

$$\mathcal{L}_{SBM} = \mathbb{E}(\|v_\theta(t,x) - u_t^\circ(x|z)\|^2) + \mathbb{E}(\lambda(t)^2\|s_\theta - \nabla \log p_t(x|z)\|^2), \tag{2.69}$$

where u_t° is the optimal transport from the unique ODE solution for the marginal SDE, which is called the probability flow ODE.
SBs extend diffusion and flow matching by jointly modeling deterministic transport and the score. It employs two networks: one for the velocity field v_θ, capturing the mean dynamics (as in FM), and one for the score s_θ, capturing local uncertainty (as in diffusion). This unifies both flows and diffusion into a single stochastic optimal control framework.

Datasets and preprocessing 3

A central component of this thesis is the design and use of datasets that enable the evaluation of longitudinal generative models. Since real longitudinal medical datasets are often limited in scale, heterogeneous in acquisition, and incomplete in temporal coverage, we consider three complementary categories of data; First, fully synthetic datasets, which provide complete control over temporal dynamics; Second, semi-synthetic datasets, which augment real images with artificial but anatomically consistent longitudinal changes; And lastly real longitudinal datasets, which serve as the ultimate benchmark for medical realism. For the first two categories, the focus lies on the controllability and interpretability of temporal dynamics rather than on the clinical context. Accordingly, the medical background and dataset-specific clinical details are discussed only for the real longitudinal datasets. This section introduces the datasets employed in each category, along with their role in the broader experimental framework.

3.1 Synthetic Data

The primary motivation behind designing this dataset was to evaluate the longitudinal segmentation capabilities of different methods. Although this section appears before the real-data experiments, it was used after early medical data experiments produced unsatisfactory results, prompting the need for a more controlled benchmark. The dataset itself is conceptually simple: it consists of sequences of ellipses that grow over time, with growth governed by a single latent variable. This setup provides a straightforward way to assess whether methods can capture and reproduce the underlying temporal dynamics of a simple yet structured process. Despite the linear latent trajectory, diversity stems from random sampling of growth rates and initial sizes, as well as additional transformations such as shear and rotation, which influence the global appearance of each sequence.
More complex temporal trajectories, such as exponential, logistic, or sinusoidal growth, could further increase the task's difficulty, so we conducted small-scale tests with these variants. However, using more complex growth rates adds little value unless the goal is to further differentiate the methods' performance. Yet the main experiments presented later rely solely on the simpler linear formulation, as it was sufficient. By adjusting the parameter ranges of the ellipses, we can control the

diversity and statistical spread of the generated dataset.
In essence, this dataset serves as a diagnostic benchmark: it tests whether the proposed models can learn linear temporal trajectories from moderately complex spatial structures. While it still lacks the complexity that comes from medical data, such as intensity variation, acquisition noise, and biological variability, it is still useful for our experiments.
In 2 we see the algorithmic implementation, and in figures 3.1 we see two example series. We chose the option to generate the dataset on-the-fly, instead of saving it, and we fixed the amount of context time points to 4, and one target time point. Other examples are the moving MNIST or bouncing balls [86, 125].

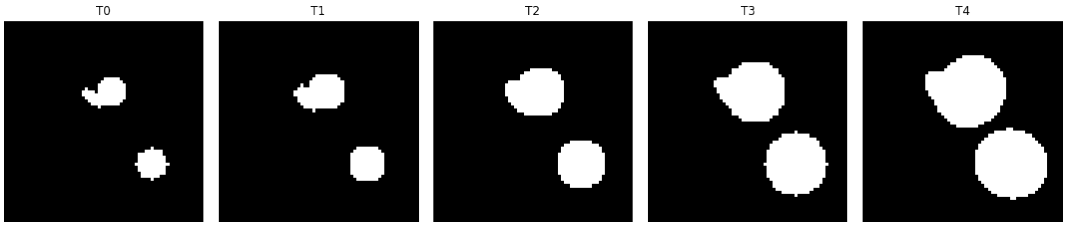
Algorithm 2 Synthetic Ellipses Dataset Generation

Require: image shape S, time grid $T = \{t_1, \ldots, t_m\}$, ranges *(low, high, start_low, start_high, shear_low, shear_high)*
Ensure: binary masks $\{M_t\}_{t \in T}$

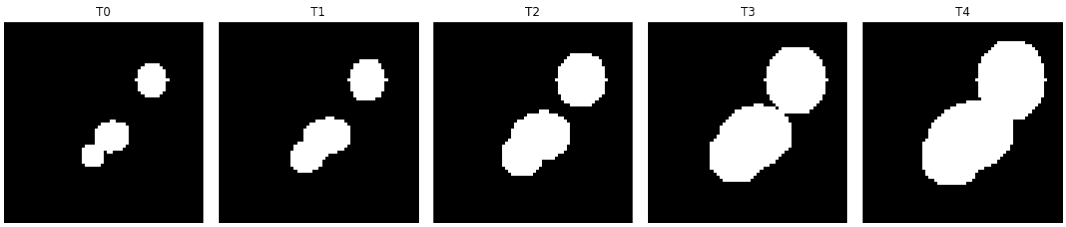
Stage 0: Global draws
$r \leftarrow \text{Uniform}(\text{low}, \text{high})$ ▷ growth rate
$K \leftarrow \text{RandInt}(1, 4)$ ▷ number of ellipses

Stage 1: Sample time-invariant ellipse templates
for $k \leftarrow 1$ to K **do**
  $c_k \leftarrow \textsc{SampleCenter}(S)$ ▷ center
  $s_k \leftarrow \mathcal{U}(\text{start\_low}, \text{start\_high})$ ▷ base radius
  $s_k^{\text{shear}} \leftarrow \mathcal{U}(\text{shear\_low}, \text{shear\_high})$ ▷ anisotropy
  $\theta_k \leftarrow \mathcal{U}(-\pi, \pi)$ ▷ orientation
end for

Stage 2: Render sequence over time
for each $t \in T$ **do**
  $M_t \leftarrow \mathbf{0}_S$ ▷ clear canvas
  for $k \leftarrow 1$ to K **do**
    $b_k(t) \leftarrow s_k + t \cdot r$ ▷ minor axis at time t
    $a_k(t) \leftarrow b_k(t) \cdot s_k^{\text{shear}}$ ▷ major axis via shear
    $E_k(t) \leftarrow \textsc{EllipseMask}(c_k, a_k(t), b_k(t), \theta_k; S)$ ▷ rasterize
    $M_t \leftarrow M_t \,\mathbf{OR}\, E_k(t)$ ▷ composite
  end for
end for
return $\{M_t\}_{t \in T}$



(a) Example series 1.



(b) Example series 2.

Figure 3.1: **Two examples of sequences from the synthetic ellipses dataset.** (a) and (b) show two distinct time series, each consisting of five binary segmentation masks captured at randomly sampled time points T_0 to T_4 within the interval $[0, 1]$. The sequences depict evolving shapes over time, controlled by different generation parameters such as motion complexity, deformation, or overlap. These examples illustrate the variability in temporal progression and structural changes used to evaluate model robustness and generalization.

3.2 Medical Data

3.2.1 Brain Tumor Segmentation Dataset

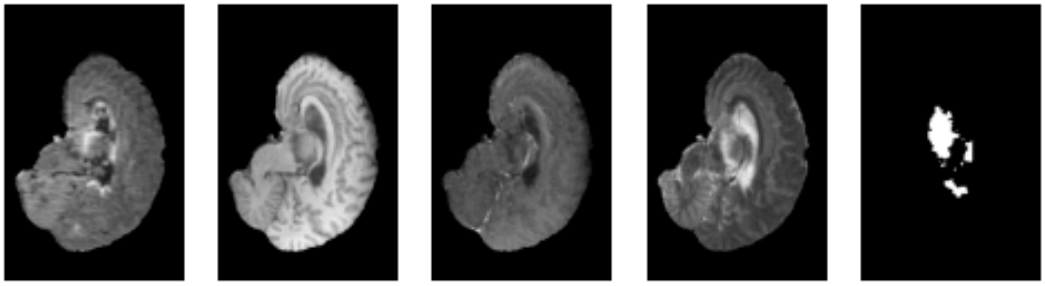


Figure 3.2: **BraTS Data Example:** Example axial slices from the Brain Tumor Segmentation (BraTS) dataset. The first four images correspond to the multi-modal MRI inputs: T1, T1-contrast-enhanced (T1ce), T2, and FLAIR. The final image shows the associated expert-annotated tumor segmentation mask. This dataset is used in our section for the extension to semi-synthetic longitudinal data.

While the previous sections focused on purely synthetic datasets, there remains a considerable gap between such toy datasets and real-world medical data. To address this gap, we extend our focus to the construction of semi-synthetic longitudinal datasets. A key motivation for this direction is that longitudinal datasets in medicine are often limited in size. Their temporal dynamics are not always well characterized [31]. By constructing semi-synthetic longitudinal series, we enrich training data and provide controlled benchmarks for evaluating generative models that capture temporal evolution. Specifically, our approach uses an existing medical imaging dataset as a structural backbone, augmented with artificial but anatomically consistent longitudinal changes. Through this, we aim to retain the controllability of synthetic datasets while incorporating the anatomical realism of real medical data. We employ the BraTS dataset [92] as the basis for our semi-synthetic data experiments. This resource is widely used for brain tumor segmentation research. This dataset aligns with our broader motivation of modeling temporal cancer evolution, offering multiple MRI modalities and expert-provided tumor annotations. An example case from BraTS is shown in Figure 3.2.
Building on this foundation, our augmentation method transforms a single static scan into a semi-synthetic temporal sequence that simulates a longitudinal progression. In contrast to the previously discussed synthetic datasets, this augmentation method relies on deformation fields derived from segmentation masks, which maintain anatomical plausibility while introducing measurable spatio-temporal variation.

Figure 3.3 illustrates this idea: the initial image is deformed into a later state, and the voxel-wise difference map highlights the regions most affected by the transformation. Section 3.2.1 details this method. It is training-free, computationally efficient (can be applied online), and works for any 3D medical dataset with segmentation masks. Thus, such augmentations provide a practical compromise between purely synthetic toy data and limited real longitudinal data. Ultimately, they enable systematic evaluation of temporal prediction models under controlled settings while preserving a closer resemblance to realistic medical image distributions.

(a) BraTS: Start of the time series

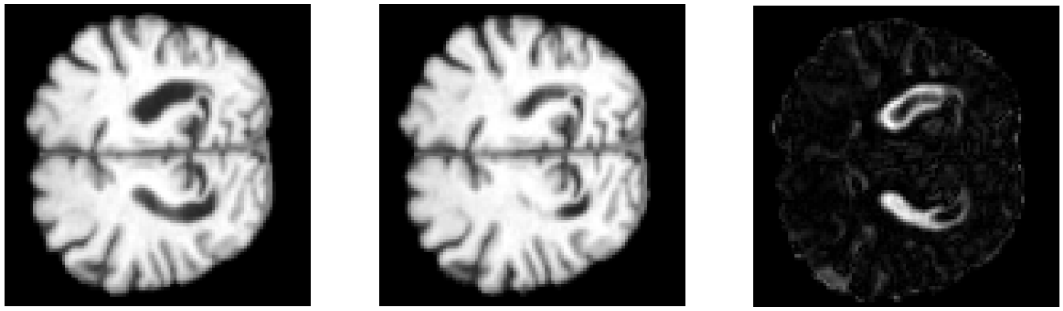
(b) BraTS: End of the time series

(c) Difference map of 3.3a and 3.3b

Figure 3.3: Example of longitudinal image augmentation using the longitudinal augmentations on the BraTS dataset. (a) shows the initial image in the synthetic time series, and (b) the final image after deformation. (c) visualizes the voxel-wise absolute difference between (a) and (b), highlighting the spatial regions most affected by the synthetic transformation. This illustrates LAUGEN's ability to generate significant temporal variation while maintaining a semblance of plausible medical images.

Since this semi-synthetic dataset is not inherently longitudinal, we omit detailed medical background information. In the following sections, we additionally focus on genuinely longitudinal and spatio-temporal medical datasets, discussing how we will use them to evaluate the proposed methods.

3.2.2 Alzheimer Disease



(a) Confirmed AD case, later stage.

(b) Confirmed AD case, early stage.

(c) Difference map of 3.4a and 3.4b

Figure 3.4: **Example images from the Alzheimer's Disease Neuroimaging Initiative (ADNI) dataset.** (a) and (b) show axial slices of T1-weighted MRIs from a confirmed Alzheimer's Disease (AD) case at later and earlier stages, respectively. (c) shows the absolute difference map between the two time points, highlighting structural brain changes associated with disease progression.

Alzheimer's disease (an example MRI is shown in Figure 3.4) is a progressive neurodegenerative disorder, resulting in memory loss, cognitive decline, behavioral changes, and eventually death [46]. The pathology of AD follows a typical spreading pattern [9, 52, 135]. This spreading process follows a well-behaved pattern, with intensity that can be modeled using well-established characteristics. Other areas, however, are only impaired in severe stages of the disease. In the earlier stages, changes are found in the medial temporal lobe structure, the entorhinal and perirhinal cortex, and the hippocampus [93]. However, one drawback is the fact that most known biomarkers, such as diminished hippocampal volume, are not unique to AD, and that atrophy is a downstream effect of amyloid plaques or NFTs [135]. In those cases, AD can be diagnosed post-mortem. Therefore, the goal in this research area is to identify biomarkers for diagnosing AD in vivo and monitoring disease progression. Deep learning has shown great promise, and AD is a well-studied disease in longitudinal imaging. A comprehensive review of recent AI advances in AD is given by [163]; Despite large cohort datasets such as OASIS [74] and ADNI [53], the review suggests that data scarcity remains a major issue.
Most longitudinal AD cohorts include subjects across the cognitive spectrum, from healthy controls to mild cognitive impairment (MCI) and diagnosed AD, captured over multiple years with follow-up intervals typically between 6 and 24 months. Structural T1-weighted MRI is the primary imaging modality. They are often complemented by PET or diffusion, as well as cognitive scores and metadata, but we

focus on T1 data. All scans undergo additional preprocessing steps, including brain extraction and inter-scan registration to ensure spatial alignment.

3.2.3 Cardiac Cine MRI

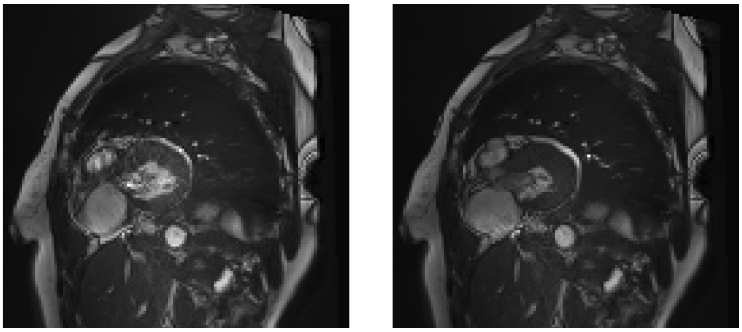
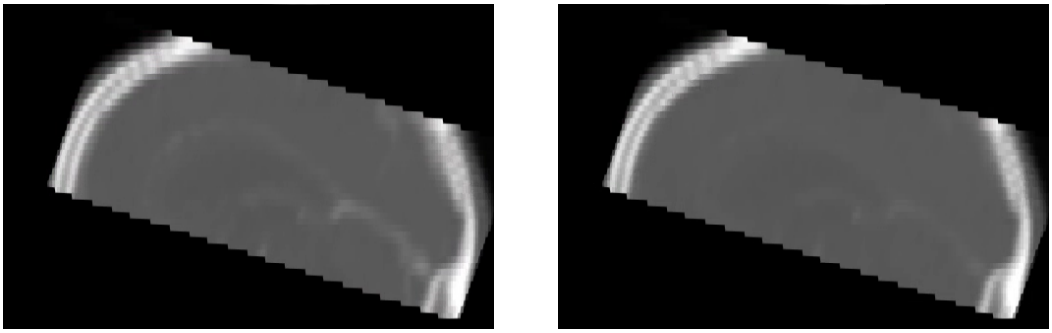


Figure 3.5: Example of a longitudinal image sequence, with the LCI on the right and the target (ground truth, GT) on the left. Data is from the ACDC dataset [7]. The lower row shows the pixel-level difference between the LCI and the GT.

Coronary Artery Disease (CAD) is the leading cause of death worldwide, and a major contributor to disability [123]. [12] provides an up-to-date overview of the current state of prevalence and attribution of mortality of CAD. For this purpose, the ACDC dataset [7] was created. This dataset measures segmentation results for the myocardium, left ventricle, and right ventricle, and also classifies different pathologies. The dataset consists of 150 patients, with varying pathologies and healthy subjects. [154] note that the task of generating the end diastolic (ED) and end systolic (ES) frames is of importance. These images are acquired with fine-grained temporal resolution. Interpolation or extrapolation may enable more rapid MR imaging, as discussed in [114]. The trajectories of the heart are well known, and the pathology groups are distinct. This makes the dataset well-suited for spatio-temporal benchmarking. Exemplary images of ACDC are shown in figure 2.1.
Because cardiac motion is periodic and well-characterized physiologically, it provides a suitable setting for evaluating longitudinal models. Each subject includes a short-axis cine MRI sequence, which is re-sampled to 12 frames between ED and ES phase, and to $128 \times 128 \times 32$ spatial resolution (following the pre-processing protocol from [154]). In total, the dataset contains 150 patients, with a 50 test and $80 - 20$ training-validation split.

3.2.4 Ischemic Stroke perfusion CT



(a) Perfusion CT, showing one frame of the time series.

(b) Showing the same patient of 3.6a, but a different frame of the time series .

Figure 3.6: **Example frames from the ISLES dataset**, which includes dynamic perfusion CT scans of stroke patients. (a) and (b) show two time points from the same patient, highlighting temporal variations in cerebral blood flow and contrast dynamics. These sequences are used to model disease progression and to evaluate predictive models under temporal imaging conditions.

Stroke is the fifth leading cause of death in the US, where 87% of the total $795k$ are ischemic [155]. The ISLES dataset [108] (see Figure 3.6) was introduced to foster the development of machine learning algorithms for lesion identification, brain health quantification, and prognosis prediction from acute stroke imaging. Automated analysis in this context is clinically critical, as treatment decisions, such as thrombectomy, must often be made within 6 hours. Accurate prediction of lesion volume and growth may support decisions about whether to transfer patients to specialized centers.
Although perfusion CT is not inherently longitudinal, we can use its temporal attribute. Each sequence reflects cerebral blood flow evolution and contrast changes. This yields a short temporal series with spatially consistent intensity changes per patient across the brain. In this work, we use temporal signals to evaluate spatio-temporal models. To our knowledge, this constitutes the first use of ISLES for this specific prediction task.
From the normalized series, we sample seven consecutive frames, using the last as the target and randomly masking the remaining context frames. The resulting context tensor has shape $[T, H, D, W] = [7, 16, 128, 128]$. We use a split of 92 training, 23 validation, and 34 test volumes.

Target (GT) Difference to LIB Last Context (LIB)

Figure 3.7: **Example case from the LUMIERE dataset**. Shown are axial MRI slices of the same patient at two different time points. The left image is the target scan (GT) representing the later time point. The right image shows the last available context image (LIB), i.e., the most recent prior scan. The middle image is the absolute voxel-wise difference between the target and LIB, highlighting structural or intensity changes over time. This example illustrates the temporal variability that longitudinal models aim to capture.

3.2.5 Longitudinal Brain Tumor

Glioblastoma is the most aggressive primary brain tumor, associated with poor prognosis and limited survival [91]. The current standard of care combines maximal safe resection with adjuvant radiotherapy and chemotherapy. Tumor response and progression are evaluated according to the Response Assessment in Neuro-Oncology (RANO) criteria [145], which rely on post-contrast MRI measurements to assess changes in tumor size and enhancement over time. Accurate imaging and consistent follow-up are therefore essential for clinical management, as understanding tumor growth dynamics directly informs treatment scheduling. Recent advances in AI have shown promise for quantitative and automated tumor response assessment [61]. The LUMIERE dataset [124] was created to study the longitudinal evolution of glioblastoma under treatment. It provides expert RANO ratings across multiple follow-up timepoints, automated tumor segmentations, and complementary metadata for 3D MRI scans. This makes LUMIERE particularly well suited for longitudinal prediction tasks, especially when dealing with irregularly sampled sequences. In our setup, we aim to predict future MRI volumes from available context images, using this prediction as a proxy for modeling the underlying disease trajectory.
The dataset spans several months to years between follow-ups, leading to irregular temporal spacing across patients. Images are reshaped to a standardized tensor size of $[T, H, D, W] = [7, 96, 96, 64]$. For patients with fewer available acquisitions,

zero-padding is applied to ensure consistent preprocessing across all cases. The dataset is divided into 48 training, 12 validation, and 14 test volumes. All scans are co-registered to a common reference space to reduce inter-scan alignment errors.

3.2.6 Near Staticity of Medical Time Series

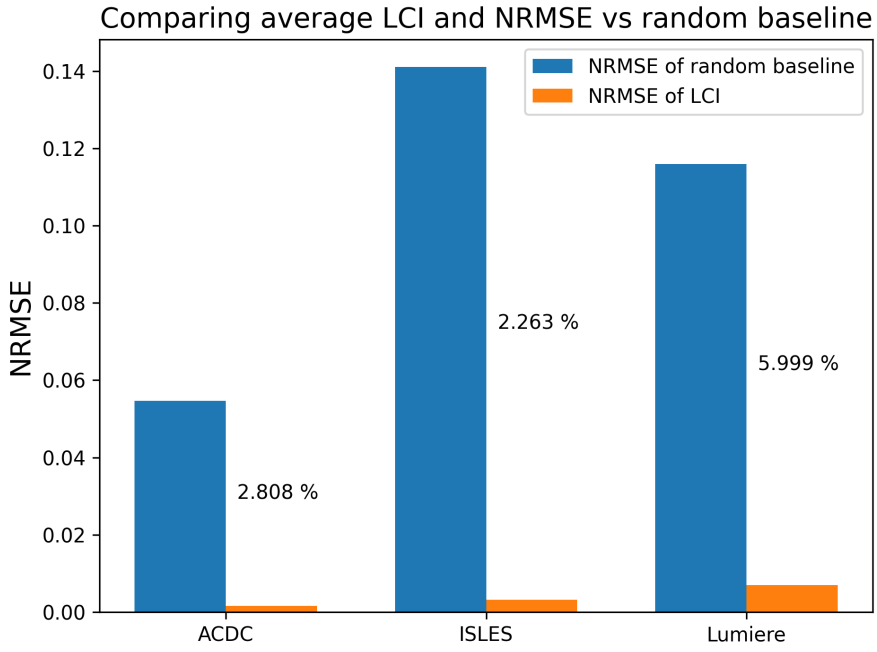


Figure 3.8: **LCI is *close* to the Target**: We report two $NRMSE$ values: one between LCI, I_{target}, and another between target and a random image. Percentages indicate the ratio of the LCI error to the random image baseline error. For reference, the $NRMSE$ between two random patients in the ACDC dataset is approximately 0.0287, which is about half the error of the random baseline.

When examining examples from the datasets (e. g. see Figure 3.5), we observe that the absolute changes within a sequence are relatively small. This pattern holds across most spatio-temporal medical datasets: temporal variation is low, cand changes are typically localized rather than global. Unlike natural video data, where dynamics arise from large motion, lighting, or texture variations, medical scans remain mostly visually stable across time, with gradual anatomical evolution.
To quantify this, we compute the average $NRMSE$ of the LCI relative to random noise across the three datasets used for our 3D spatio-temporal experiments: ACDC, ISLEs and LUMIERE (Figure 3.8). The LCI remains below 3% for ACDC

and ISLES, both of which show fine-grained temporal changes, and around 6% for LUMIERE, where supposedly larger tumors and registration inaccuracies lead to more visible differences (see Figure 3.7). These low values confirm that temporal evolution in medical imaging is smooth and spatially constrained, reinforcing the advantage that we can achieve when we focus mostly on these smaller changes.
This observation supports the use of Flow Matching as a canonical approach for spatio-temporal modeling. In fact, we will show that Flow Matching trivially contains the LCI. While this analysis does not directly quantify the issue discussed in Section 2.3.3, it provides additional evidence that focusing on modeling differences is well justified. To our knowledge, this behaviour has not been systematically analyzed in prior work, likely reflecting the limited attention longitudinal image generation has received in the literature. Together, these findings provide an empirical and conceptual foundation for the methodological choices in Section 4.3.

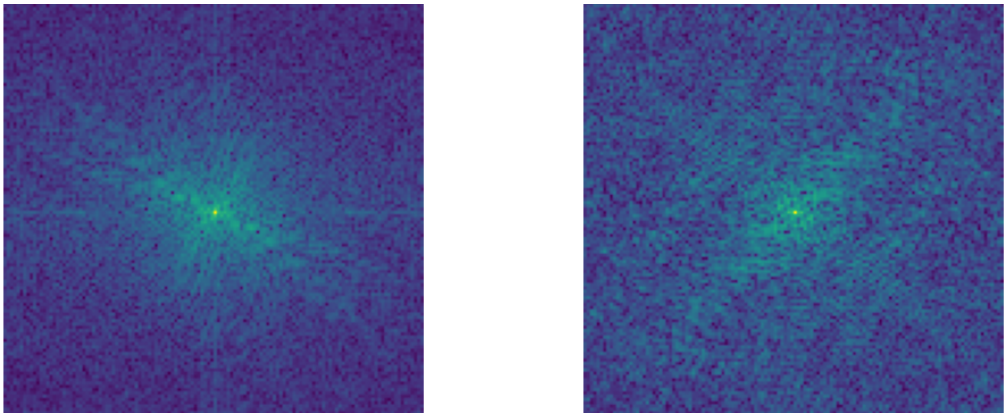


Figure 3.9: **Fourier magnitude spectra from the target image and the difference between the target image and the LCI. Left:** Fourier spectrum of a slice of a target image from the ACDC dataset. **Right:** Fourier spectrum of a slice of the difference between the target image and the LCI from the ACDC dataset. .

Fourier Spectra We analyzed the Fourier spectra to obtain further quantitative distinctions between the images and the LCI. The spectra show that the difference between the target and the LCI exhibits significantly reduced spectral energy, with most of the energy concentrated in low-amplitude frequency components. This suggests that temporal changes have a smaller amplitude than those in the full image. As a result, the model only needs to learn smaller residual dynamics instead of reconstructing the full frequency structure, likely simplifying prediction. Figure 3.10 shows the magnitude histogram, while Figure 3.9 provides two example Fourier transforms: one from a target image in the ACDC dataset and another of the difference between the target and the LCI.

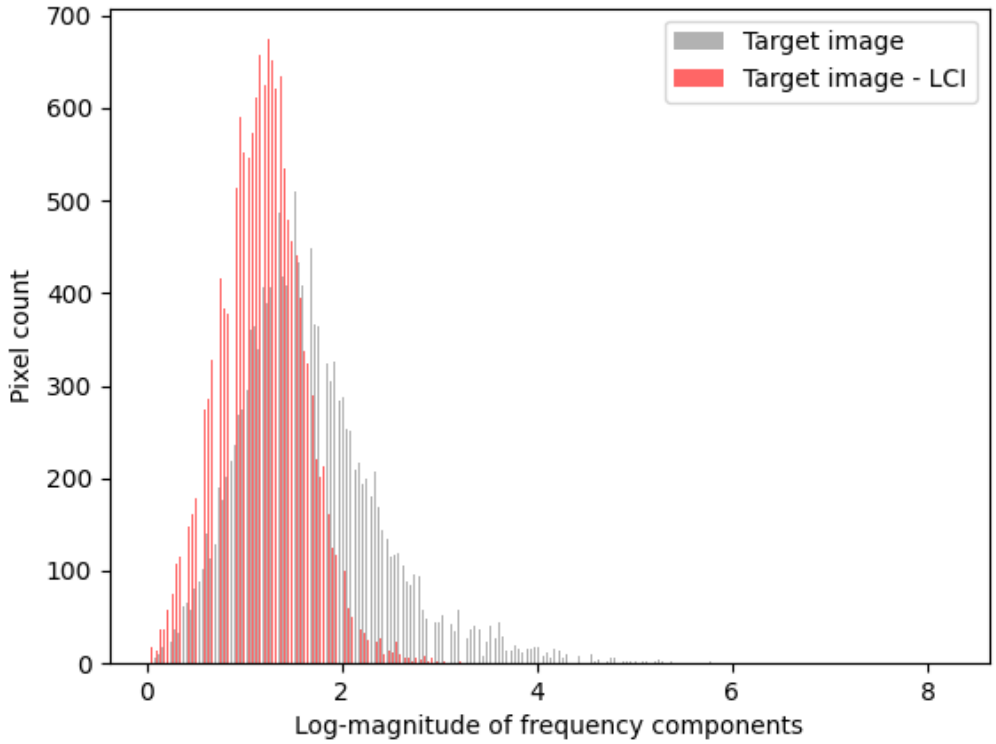


Figure 3.10: Comparison of the Fourier magnitude histograms for a target image and the difference between its difference to LCI. The red distribution (Target - LCI) shows a clear reduction in frequency amplitude compared to the gray (Target image) distribution. This shows an overall reduction in spectral energy.

Methods 4

In the previous Chapter 3 we discussed the datasets used, and before that we discussed the baselines in Chapter 2, including both discrete and time-continuous approaches. In this Chapter, we present the principal methodological contributions of this work. We first describe extensions of existing baselines, in particular improvements to ASP.
Next, we introduce a medical image augmentation strategy that generates longitudinal image sequences from static scans. Finally, we detail the main proposed method, Temporal Flow Matching (TFM), which extends Flow Matching to model temporal evolution across multiple time points. We further enhance this method with continuous-time modeling, deformation fields, and a generalization toward Schrödinger Bridges.

4.1 Neural Processes and Neural ODEs

In this section we summarize the Attentive Segmentation Process (ASP [102]) and the extensions we proposed for longitudinal image prediction. While the ASP provides a solid baseline for learning from sparse temporal observations through attention-based aggregation on image features, it faces limitations in both computational scalability and temporal expressiveness. Our goal is therefore twofold: first, we aim to make the attention mechanism more scalable for high-resolution or 3D data, and secondly, to improve temporal modeling through continuous-time representations based on Neural Ordinary Differential Equations (Neural ODEs [13]). Despite several architectural variations, the overall performance gains remains moderate, suggesting that the Neural Process backbone itself may impose structural constraints for our tasks (for a summary of Neural Processes see Section 2.5). The following subsections present the baseline ASP formulation 4.1.1,the Neural ODE integration designed to enhance temporal continuity in Section 4.1.2 the proposed scalability modifications 4.1.2.

4.1.1 Attentive Segmentation Process

The ASP is a Neural Process variant designed to predict future target segmentations from a set of continuous-time observations using attention based feature aggrega-

tion. Given context observations $\{(x_i, t_i)\}_{i=1}^{n}$, the model aims to reconstruct the corresponding target image x_{target} at specified time t_{target}. **Context Encoding**: Each context slice $x_i \in \mathbb{R}^{C\times H\times W}$ at time t_i is processed by a shared encoder E_ϕ to produce multi-scale feature maps $r_{i,\ell} \in \mathbb{R}^{d_\ell \times H_\ell \times W_\ell}$ for levels $\ell = 1, \ldots, L$, and a global vector $r_{i,G} \in \mathbb{R}^{d_G}$.

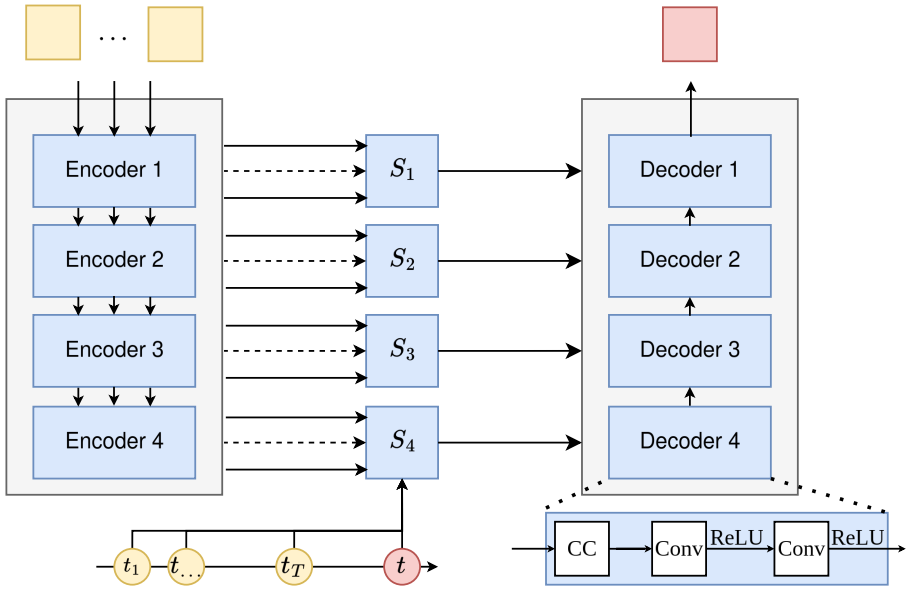


Figure 4.1: **Overview of the Attentive Neural Process Architecture:** Given a set of context observations $I_1, \ldots, I_T$ at times $t_1, \ldots, t_T$, we want to predict the target observation at a given time t_{target}. The basic backbone overview of a neural process is shown in Figure 2.5. This figure here shows the backbone of the ASP itself, which is a UNet architecture with attention summing the up the features. The features are the image features, the context times and the target time which are encoded via a linear layer. The features from the context times and target times are in different parts of the attention. In the bottleneck, time embeddings are summed up, and the output is then concatenated to the image embeddings. The Figure has been adapted from [102].

Time embedding. Acquisition times t_i and target time t_t are embedded via linear layer u to have the same dimensionality as the feature maps:

$$\tau_i = u(t_i) \in \mathbb{R}^{d_\tau}. \tag{4.1}$$

Spatio-Temporal Attention: At the two coarsest U-Net scales ($\ell = 1, 2$), we flatten the feature maps and apply attention only along the temporal axis using (2.9). For finer scales ($\ell = 3, \ldots, L$), attention is computed over both spatial and temporal dimensions. In other words, the top (coarse) levels use temporal-only attention,

while the bottom (fine) levels employ full spatio-temporal attention. The three matrices for the attention operation, key value and query correspond to the image features, the context times and the target time embeddings respectively. Figure 4.1 shows and overview of the ASP. **Computational Considerations**: Temporal attention scales quadratically with the number of context frames, whereas full spatio-temporal attention adds a quadratic cost in the number of spatial patches. This quickly becomes prohibitive for high-resolution or 3D data, motivating the development of lightweight alternatives, which we will discuss in the following subsection.

4.1.2 Neural Process Extensions

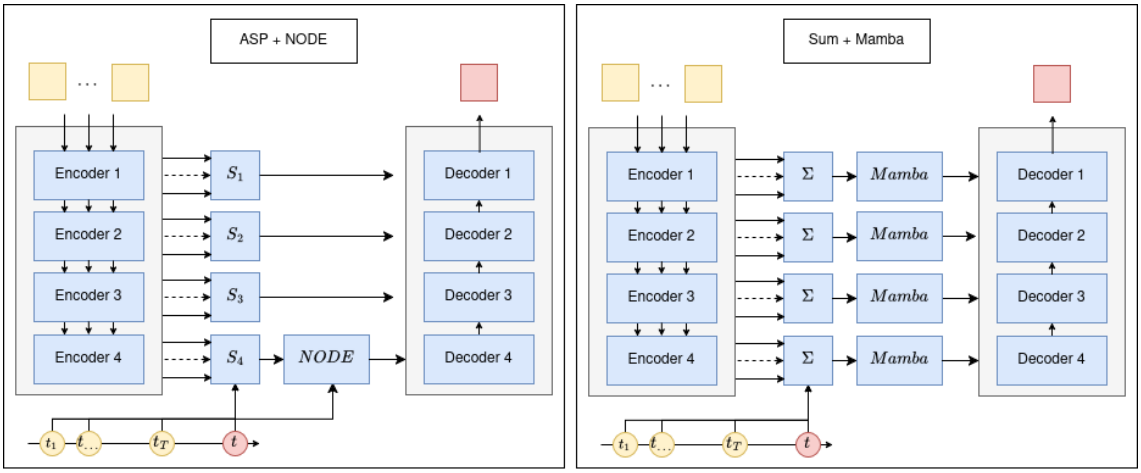


Figure 4.2: **The left** diagram shows the NODE variant of the ASP, where the skip connections can also be a simple sum instead of the attention mechanism. **The right** diagram shows the Mamba configuration, which replaces the attention-based skip connections with summed feature aggregation followed by Mamba blocks. Both architectures have the same encoder-decoder structure, see Figure 4.1.

Temporal Modeling

Neural ODE bottleneck. To model continuous-time dynamics in the latent space, we place a Neural ODE after feature aggregation. Let

$$h_{\text{sum}} = \sum_{i=1}^{k} E_\phi(x_i) \tag{4.2}$$

be the spatial feature at the lowest feature size (colloquially called the bottleneck). We solve

$$z_{\text{target}} = ODEintegrate(h_{\text{sum}}, f, \{t_0, \ldots, t_{\text{target}}\}), \tag{4.3}$$

where T_c and T_t are, respectively, the mean context time and the target time. The dynamics function f is a two-layer neural network with tanh activations and the same latent dimensionality as h_{sum}. Integration is performed using a fixed-step fourth-order Runge-Kutta solver [72, 112] with 20 steps, producing z_{target}, which is subsequently decoded to reconstruct the target image, together with the skip-connections. A diagram is shown in Figure 4.2 left.

Longitudinal VAE (l-VAE) As proposed by [115], the longitudinal VAE replaces the deepest U-Net bottleneck with a variational autoencoder whose latent interpolation follows a pre-defined geodesic. We add this l-VAE to the same ASP backbone, without the skip connections. Given context feature sum

$$h_{\text{sum}} = \sum_{i=1}^{k} E_\phi(x_i), \tag{4.4}$$

we encode to a Gaussian latent (μ, σ), sample $z_0 \sim \mathcal{N}(\mu, \sigma^2)$, and evolve $z(t)$ along

$$z_{i,j} = p_0 + \left[e^{\xi^i (t_{i,j} - \tau_i)}\right] v_0 + w_i + \varepsilon_{i,j}, \tag{4.5}$$

where ξ^i and τ_i, respectively the *acceleration factor* and the *onset age* of patient i, allow an affine time warp aligning all patients on a common pathological timeline, and $w_i \in \mathcal{Z}$ is the *space shift* that encodes inter-subject variability, while $\varepsilon_{i,j}$ represents residual noise [115]. But unlike neural ODEs, this model assumes a fixed trajectory, limiting it to diseases that align with that trajectory. We implemented the longitudinal VAE in the ASP backbone by setting all skip connections to zero and setting the bottleneck as described in (4.5).

Reducing Attention Complexity

Summed Image Representations We also test whether even simpler representations are sufficient. Let $\{h_i\}_{i=1}^n$, $h_i \in \mathbb{R}^{C' \times S'}$, be the encoder feature maps at context times t_i. Instead of multi-head attention over $\{h_i\}$, as well as context and target time, we compute

$$h_\ell = \sum_{i=1}^{k} r_{i,\ell}, , \tag{4.6}$$

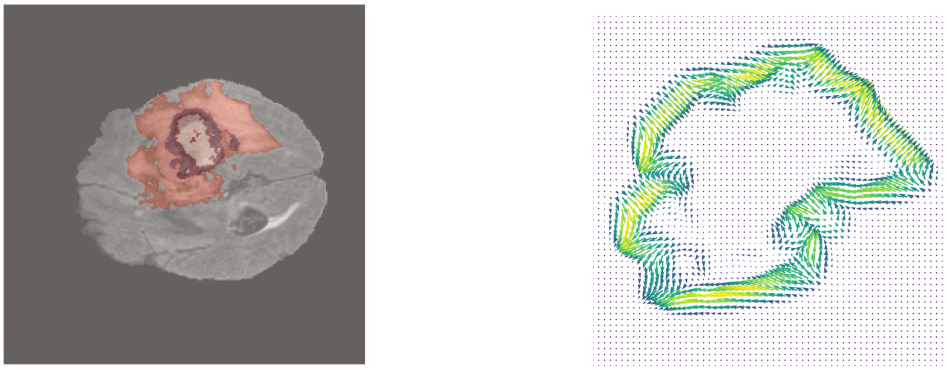
and pass h_ℓ to the decoder using the same skip-connection paths. This reduces memory and computation by replacing the $O(n^2)$ attention cost with a single element-wise sum. But this reduces model expressiveness, since no time representation reaches the decoder. In practice, this provides a minimal baseline for skip-connections while replacing attention.

Mamba for Skip Connections To clarify, to avoid the quadratic cost of attention in ASP, we replace all skip-connection ASP attention blocks ($S_{0...3}$) with Mamba blocks [43]. At each resolution level ℓ, we first sum context features:

$$h'_\ell = \text{Mamba}(h_\ell), \tag{4.7}$$

then apply a Mamba layer to h_ℓ (a sequence-to-sequence model with linear cost in sequence length); then merge the output via U-Net skip connections. The final bottleneck, S_4, remains a simple sum, as in a Neural Process. This was a preliminary approach. The original ASP used three different inputs for key, query, and value (2.9). We therefore tested first whether spatial features alone were sufficient.

4.2 Longitudinal Augmentation and Data Generation



(a) **Image with Segmentation Mask** (b) **Normal Vector**.

Figure 4.3: **Visualization of a selected slice from BraTS [92]**. (a) Shows an example slice from the BraTS dataset together with a segmentation mask overlayed. (b) This image shows a slice with a displacement vector derived from the segmentation mask boundary. The coloured arrows represent the direction and magnitude of the displacement field, computed as the normalized gradient of a Gaussian-blurred segmentation mask.

We encountered difficulties when experimenting with the ADNI dataset using Neural Processes and its variants, and it was unclear whether the poor results stemmed from methodological limitations or from the dataset's complexity. Controlled experiments on synthetic data confirmed that the methods struggled to capture temporal dynamics even under idealized conditions, indicating inherent model constraints.

Motivated by these findings, we developed the following longitudinal data augmentation and generation method, designed to bridge the gap between fully synthetic and hard-to-control real data.
This method alleviates these issues by synthetically generating longitudinal image sequences from static scans, allowing us to achieve the following objectives, which we will test: First, to pre-train methods or for online augmentations. Secondly, to benchmark spatio-temporal methods under controlled, yet realistic conditions. It therefore complements the synthetic benchmarks, whose use was motivated by the observations from these earlier experiments.

4.2.1 Longitudinal Augmentations via Deformation Fields.

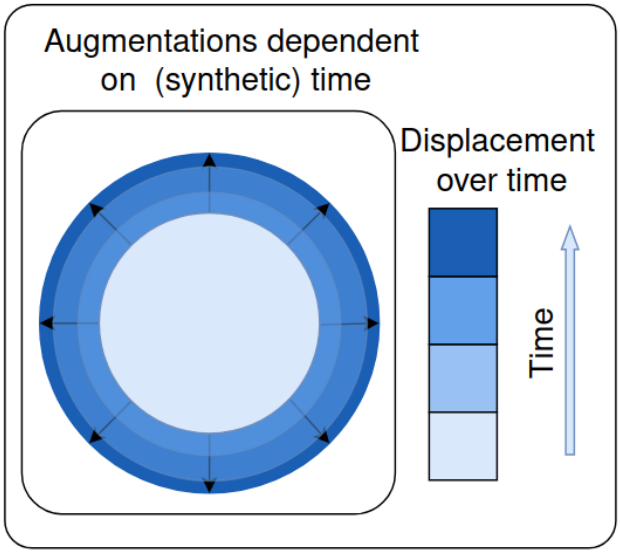


Figure 4.4: **Extending biological augmentations to generate longitudinal series.** This figure illustrates the process of time-dependent augmentation, where the displacement magnitude varies over synthetic time and amplitude $\alpha(t)$. The original segmentation mask is shown in light blue, and the subsequent masks are shown in darker blue. The concentric rings represent successive time points, with increasing displacement applied to the segmentation boundary at each step. This enables the generation of semi-synthetic longitudinal series from a single image and segmentation mask.

Method Longitudinal augmentations are generated from static images and their corresponding segmentation masks. Figure 4.3 shows an example image in (a) and its corresponding deformation field in (b). We begin by describing the simplest setting, *uniform*, where displacement is applied uniformly across the segmentation

mask boundaries, following the scheme of [68]. We extend this approach with two variants: *directional* mode, which introduces anisotropic displacements along a preferred axis, and *Gaussian* mode, where displacement strength decays spatially by a Gaussian function and produces localized growth around specific regions. In all cases, the original structure (blue) is transformed into a new shape (green) through a vector displacement field. Figure 4.4 depicts how we generate a time-dependent deformation based on the displacement fields.
Importantly, the method uses a segmentation mask solely to generate plausible vector fields and does not rely on it for validation. This approach remains practical in most settings since numerous automatic segmentation methods exist (see e. g. [66, 113] and their medical adaptations [161]). Thus, acquiring segmentations is not a significant barrier.

Biological Augmentations using Deformations Given a binary segmentation mask $\mathcal{S} \in \{0,1\}^{H\times D\times W}$, we first apply a Gaussian Filter to obtain a smoothed version $\mathcal{S}_G = G_\sigma * \mathcal{S}$, where G_σ is a 3D Gaussian kernel with a standard deviation σ. From this smoothed mask, we compute the spatial gradient:

$$\nabla \mathcal{S}_\sigma = \left(\frac{\partial \mathcal{S}_\sigma}{\partial x}, \frac{\partial \mathcal{S}_\sigma}{\partial y}, \frac{\partial \mathcal{S}_\sigma}{\partial z} \right), \tag{4.8}$$

which is then normalized to obtain unit normal vectors

$$\hat{n}(x,y,z) = \frac{\nabla \mathcal{S}_\sigma(x,y,z)}{\|\nabla \mathcal{S}_\sigma(x,y,z)\|}. \tag{4.9}$$

This ensures that each voxel's displacement direction is independent of gradient magnitude. A displacement field $D(x,y,z)$ is generated (e. g. uniform for the basic setting, Gaussian decaying for blob growth, or directional, see Figure 4.5). The final deformation field is

$$v(x,y,z) = D(x,y,z) \cdot \hat{n}(x,y,z). \tag{4.10}$$

The image I and segmentation masks $\mathcal{S}$ are then warped using v. The deformation map is given by

$$T(x,y,z) = (x,y,z) + v(x,y,z). \tag{4.11}$$

Longitudinal Evolution To simulate temporal evolution, we extend the deformation field by making the displacement magnitude time-dependent (see Figure 4.4):

$$v(x,y,z,t) = \alpha(t) \cdot D(x,y,z) \cdot \hat{n}(x,y,z), \tag{4.12}$$

where $\alpha(t)$ is the latent trajectory controlling deformation strength over time. For additional flexibility, the displacement field D may also vary with t to model non-stationary or region-specific dynamics. In our experiments, all three augmentation

settings shown in Figure 4.5 were used for image generation. For the ACDC dataset, we applied only the uniform deformation mode. In both cases, we adopted a linear latent growth, though the method allows for arbitrary functional forms $\alpha(t)$.

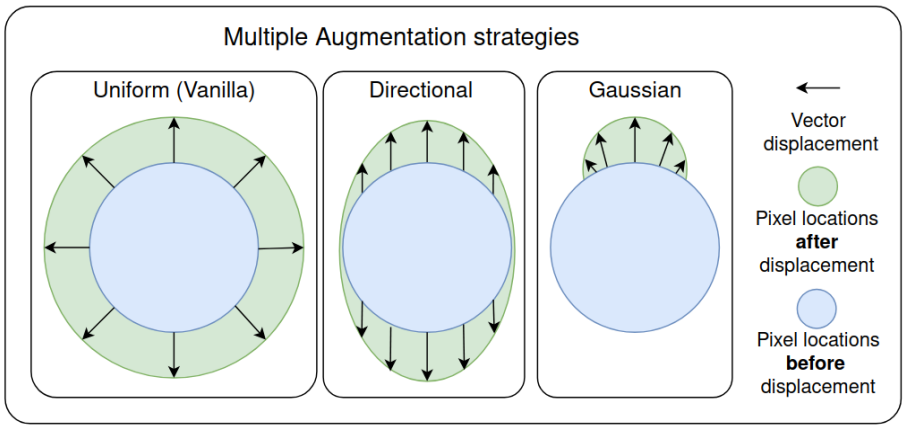


Figure 4.5: **Extension of [68] to more shapes**. This figure illustrates different displacement strategies applied to a segmentation mask for biological augmentations. In the Uniform setting, the displacement is uniform across the mask, as in the original work. The directional setting introduces an anisotropic displacement that simulates shape deformations along a preferred axis. The **Gaussian** strategy modulates the displacement strength spatially via a Gaussian decay, enabling very localized growth in specific regions. In each case, the original shape (blue) is transformed into a new shape (red) via vector displacements.

4.3 Flow Matching for Spatio-Temporal Series

In this section, we propose Temporal Flow Matching (TFM), a method for spatio-temporal medical image modelling. We begin by explaining why FM provides a natural formulation for modeling medical image time series. It's linear interpolation that formulates the velocity loss, which inherently captures *differences* between scans. Next, we extend the FM formulation to incorporate multiple context images within the same flow model, using a sparsity filling strategy (see Figure 4.7), yielding TFM (see Figure 4.6). We then discuss how temporal grid quantization enables discrete modeling but introduces a trade-off between temporal resolution and computational cost. Finally, we highlight how this limitation motivates the transition to a continuous-time formulation, addressed in the next section.

Why use Flows?: Difference Modeling

We add an explanation that clarifies why Flow Matching is so apt for medical imaging. In standard FM, transport is learned between two distinct distributions p_0 and p_1. But in our model, we reinterpret the two distinct distributions as the source I and the target I_{target}. Recall the ground truth velocity (4.19)

$$X_1 - X_0 = I_{\text{target}} - I, \tag{4.13}$$

i. e. *exactly the per-voxel change* per time step we wish to predict. We refer to this as **Difference Modeling**, the network v_θ *models* the spatio-temporal *difference*, rather than reconstructing entire images. Accordingly, recall the regression term from (2.63):

$$\mathcal{L}_{FM} = \mathbb{E}||v_\theta(X_\tau, \tau) - (X_1 - X_0)\,||_2^2, \tag{4.14}$$

which already encodes this subtraction, and the network still receives the full interpolated context X_τ as input. As seen in medical time series (Figure 3.8), most temporal variations are spatially localized. This is important: starting from the last available image and modeling just the residual change means the initial estimate is already close to the target by common image similarity metrics. Again, this ties back to the start, where changes in images are small, see Section 3.2.6. The whole motivation also ties back to Proposition 2.3.

Handling Irregular Temporal Contexts Irregular temporal sampling is common in longitudinal imaging. However, few works address this challenge in sequences. Most existing methods operate in an image-to-image fashion and thus fail to model multiple observations. In our context, when applying flow-based models, two canonical strategies can be considered:

1. **Temporal Pooling**: Compress the context sequence using a spatio-temporal encoder, or predict the flow from only the last available image.

2. **Dimension Padding**: Extend the target and context dimensionality to a fixed context sequence length.

The first approach is simple but discards earlier-frame information or pools it through a spatio-temporal model [154]. The second approach preserves all available context and the temporal information from earlier frames. The downside of the second one is the increased computational complexity. We adopt **dimension padding** along the time axis. This provides flexibility and stability by broadcasting the target image to match the number of context images. This choice is partly inspired by SimVP [34], which similarly treats different contexts and target times. For truly irregular data, we discuss how to embed it in a regular grid. The ACDC and ISLES datasets are spatiotemporal, meaning their data points are organized on a regular grid. In ADNI, acquisitions occur approximately every 6 months, establishing a canonical temporal sampling grid. Of the datasets, only LUMIERE is more irregular, with data collected at a weekly resolution and during longer acquisition periods. Most of the data are arranged on a regular, discrete grid. When we subsample to form an irregular grid, the original discrete grid structure remains present in the dataset.

Full-Resolution Modeling Unlike methods that compress all information into a latent space, our approach directly models images at **full spatial resolution**. We therefore avoid pre-training an autoencoder, thereby simplifying training and reducing overhead. Our approach maintains a computational footprint comparable to that of other spatio-temporal networks. This makes the operation feasible. By jointly processing the entire input sequence, the model can capture fine spatial and temporal dependencies.

4.3.1 Temporal Flow Matching Theory

For TFM, we again assume a known source distribution q and a target distribution p. Our goal is to learn a distribution path $(p_\tau)_{0\leq\tau\leq1}$, which connects the two distributions. We define an ODE via a *step*-dependent vector field $u : [0,1]\times\mathbb{R}^{T\times S} \to \mathbb{R}^{T\times S}$. Here, S is the spatial and T the temporal dimension. We approximate u with a neural network v_θ. The corresponding flow satisfies,

$$\frac{d}{d\tau}\psi_\tau(x) = u_\tau(\psi_{\tau(x)}), \tag{4.15}$$

with $\psi_0 = \mathrm{Id}$ and $p_\tau = (\psi_\tau)_{\#}q$. We form patient-coupled pairs $(X_0, X_1) \sim \Pi$, where X_0 is the context sequence $\mathcal{I}$, and X_1 is I_{target} stacked T times

$$X_1 \coloneqq [\underbrace{I_{\text{target}}, \ldots, I_{\text{target}}}_{T}], \in \mathbb{R}^{T\times S}. \tag{4.16}$$

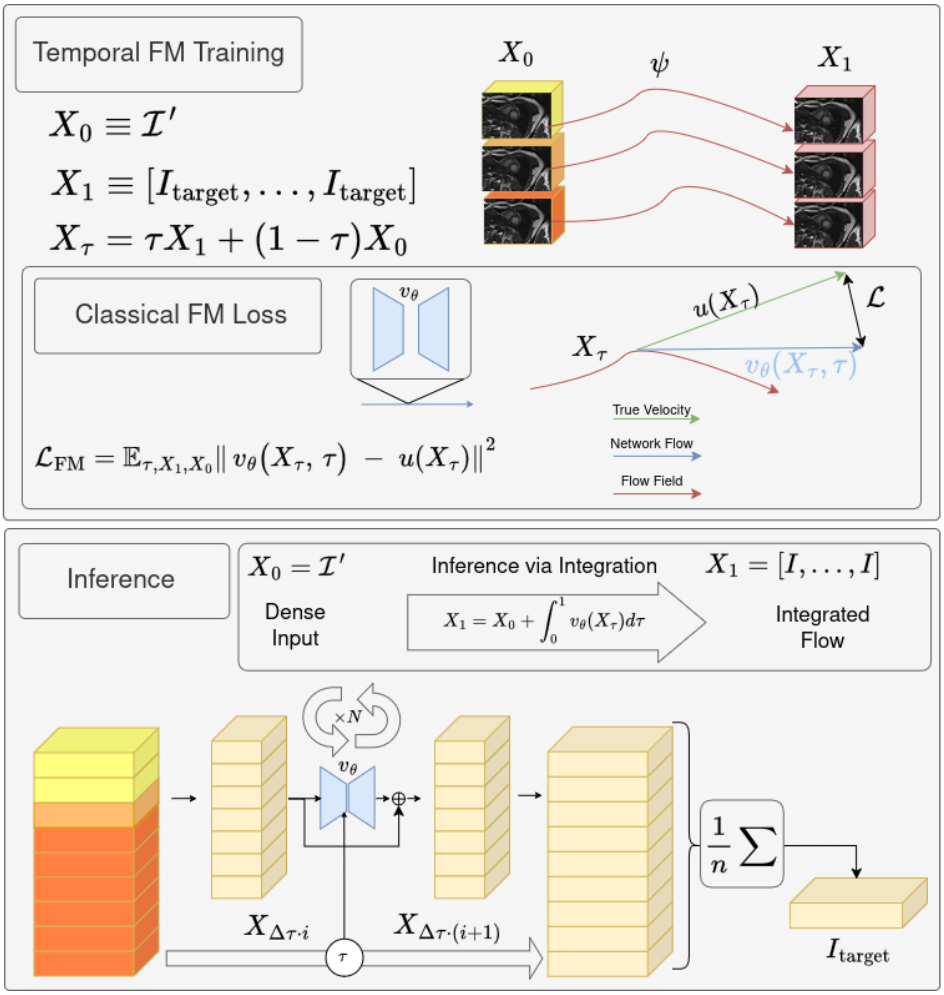


Figure 4.6: **Training and Inference of TFM** Here we see the training and inference for TFM. The **training** happens via generating the sparsity-filled sequence $\mathcal{I}'$. Then a linear interpolation between $\mathcal{I}'$ and $\mathcal{I}_{\text{target}}$. The training is done via sampling $\tau \sim \mathcal{U}(0, 1)$, and then calculating the MSE loss between true velocity u_τ and the predicted velocity $v_\theta(X_\tau, \tau)$. For **Inference**, we first have to choose an integration strategy. In the experiments, we used Runge-Kutta, but here an example for Euler Integration is shown. For this, again $\mathcal{I}'$ is calculated, and for each integration step, we calculate $X_{\tau_i+1} = X_{\tau_i} + \Delta v_\theta(X_{\tau_i}, \tau_i)$.

Or specifically,

$$X_1 = \mathbf{1}_T \otimes I_{\text{target}} \in \mathbb{R}^{T \times S}, \tag{4.17}$$

where $(\mathbf{1}_T \otimes I)_T = I \; \forall t \in T$. We define the linear path interpolant

$$X_\tau = (1 - \tau)X_0 + \tau X_1, \tag{4.18}$$

yielding

$$u_\tau \coloneqq \frac{dX_\tau}{d\tau} = X_1 - X_0. \tag{4.19}$$

We train $v_\theta : \mathbb{R}^{T \times S} \times [0, 1] \to \mathbb{R}^{T \times S}$ with the linear interpolant regression loss

$$\mathcal{L}_{TFM} = \mathbb{E}_{\tau \sim \mathcal{U}(0,1), \{X_0, X_1\} \sim \Pi} ||v_\theta(X_\tau, \tau) - (X_1 - X_0)||_2^2, \tag{4.20}$$

At inference, we solve the inverse value problem

$$\frac{dX_\tau}{d\tau} = v_{\theta(X_\tau, \tau)}, \tag{4.21}$$

where X_0 is the context sequence and $\tau \in [0, 1]$. Which, in practice, is calculated numerically via

$$X_1 = X_0 + \int_0^1 v_\theta(X_\tau, \tau) d\tau, \tag{4.22}$$

to obtain X_1 (e. g.via Euler or Runge-Kutta with suitable step size or tolerance). Using Euler as example, we calculate

$$X_{\tau_i + 1} = X_{\tau_i} + \Delta v_\theta(X_{\tau_i}, \tau_i), \tag{4.23}$$

where $\Delta\tau$ is the integration step size, and i the total steps.
Since the prediction is larger than what we want to predict, we need to reduce the dimension from $S \times T$ to S. We extract the target image by projecting along the time axis. Let $w \in \Delta^{T-1}$ be a one-hot selector ($w \in \mathbb{R}^T$ and $\sum_t w_t = 1$). We define

$$R_w(X) = \sum_{i=1}^{T} w_i X^i \in \mathbb{R}^S, \tag{4.24}$$

where X^i is the ith channel. Then $\hat{I}_{\text{target}} = R_w(X_1)$. In the experiments, we tested using the last channel ($w_T = 1$) or the mean ($w_i = 1/T$).

Sparsity Filling

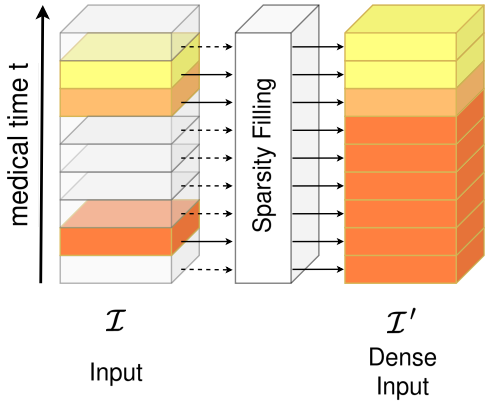


Figure 4.7: **Illustration of the sparsity filling mechanism**. The original input sequence $\mathcal{I}$ contains missing time points (shown as transparent slices), which are replaced by the most recent available scan through a forward-filling process. The resulting dense sequence $\mathcal{I}'$ ensures a temporally complete input for downstream modeling. This approach preserves temporal structure while addressing common sparsity in longitudinal medical imaging data.

Irregular sampling in longitudinal imaging leads to missing acquisition at certain timesteps, as we model time via the discrete grid. This creates gaps along the temporal axis and distorts the estimated flow between I_i and I_{target}. To address this, we apply **sparsity filling**, where missing frames (zero images) are replaced by the most recent available scan (see Figure 4.7). If missing frames occur before the first observed scan, they are filled using the earliest available image. This forward-filling strategy ensures temporally dense inputs, resulting in smoother flows. Formally, we set

$$X_0 = \mathcal{I}' \quad \text{and} \quad X_1 = [\underbrace{\tilde{I}, \ldots, \tilde{I}}_{\times n}], \tag{4.25}$$

where $\mathcal{I}'$ denotes the filled sequence. More rigorously: each missing frame is initialized as zero and iteratively replaced by the nearest past available image in time:

$$\hat{I}_1 = \tilde{I}_{k_0}, \quad \hat{I}_k = m_k \tilde{I}_k + (1 - m_k)\hat{I}_{k-1} \quad k \in \{2, \ldots, K\}, \tag{4.26}$$

where k_0 defines the first observed image in the grid and $m_k \in \{0, 1\}$ indicating frame availability. Then we have

$$k_0 = \min\{k \in \{1, \ldots, K\}\} | m_k = 1\}. \tag{4.27}$$

We define the sparsity-filling operator as

$$\mathcal{F}^{\mathrm{SF}}\left([\tilde{I}_1, \ldots, \tilde{I}_K]\right) = [\hat{I}_1, \ldots, \hat{I}_K]. \tag{4.28}$$

Empirically, this step proves highly beneficial: each filled image in $\mathcal{I}'$ lies closer to the target image than a zero-filled frame, producing more homogeneous flow fields and improving convergence stability as well as performance. Conceptually, the motivation is similar to that of difference modeling; both approaches aim to reduce the effective gap between input and target representations, thereby simplifying the learning problem. The caveat is that this assumes motion is static; more advanced options include linear interpolation to give a more realistic prior. However, we note that the continuous version can skip this sparsity-filling step, as it can utilize arbitrary time steps.

Grid Quantization

We briefly describe how irregular temporal data may be quantized into a fixed grid. This procedure, while necessary for discrete processing, introduces inaccuracies and additional computational burden, which motivate our transition to continuous-time formulations.

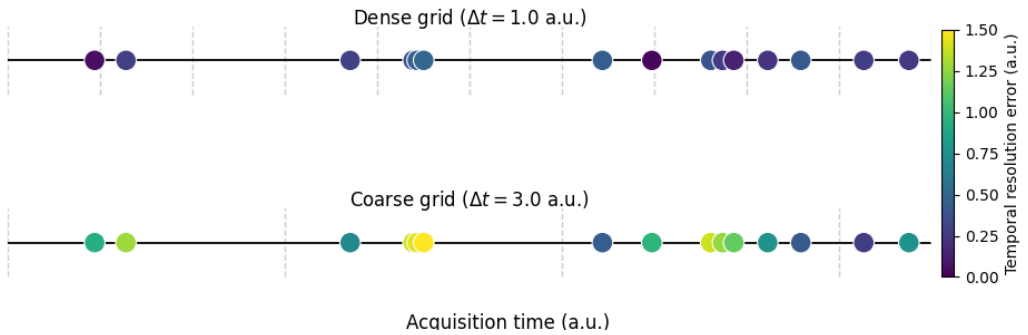


Figure 4.8: **Illustration of the temporal resolution issue.** When continuous times are discretized onto a fixed grid, each observation is assigned to its nearest grid point (dotted vertical lines), introducing quantization errors (indicated by color). **Top:** With a dense grid ($\Delta t = 1$), most acquisitions align closely to the grid, though already small inaccuracies remain, and some points are excluded. **Bottom:** A coarser grid ($\Delta t = 3$) yields larger temporal mismatches and fewer effective observations. This highlights a fundamental trade-off: finer grids improve temporal accuracy but increase computational cost and the amount of zero-filled frames, whereas coarser grids reduce inputs but distort temporal alignment and reduce the total number of time-points that can be included.

Assume we want to embed data on a grid with spacing $\Delta > 0$, and a maximum size of $K \in \mathbb{N}_+$, where

$$g_k = g_1 + (k-1)\Delta, \quad k \in \{1, \dots, K\}. \tag{4.29}$$

In total, we define the grid as $\mathbf{g} = [g_1, \dots, g_k]$. We define the grid quantizer as

$$q(t_i) = \text{clip}\left(1 + \left\lfloor \frac{t_i - g_1}{\Delta} + \tfrac{1}{2} \right\rfloor, 1, K\right), \tag{4.30}$$

where $\text{clip}(a, b, c) := \min(\max(a, b), c)$. I.e., we clip the value t to the closest grid point g_k. We define the Kronecker delta for indices

$$\delta_{k,q(t_i)} = \begin{cases} 1, & q(t_i) = k \\ 0, & \text{else.} \end{cases} \tag{4.31}$$

Since for some grid sizes Δ, there can be too many items, we define the occupancy:

$$m_k = \min\left(1, \sum_{i=1}^{T} \delta_{k,q(t_i)}\right) \in \{0, 1\}. \tag{4.32}$$

We use the last available index for filling:

$$i^*(k) = \text{argmax}_i(\delta_{k,q(t_i)} t_i), \tag{4.33}$$

i.e. the index which is *latest* while still falling into the index k. Finally, our embedded images are

$$\tilde{I}_k = \begin{cases} I_{i^*(k)}, & m_k = 1 \\ 0, & \text{else.} \end{cases} \tag{4.34}$$

Lastly, we define the grid quantization as

$$\mathcal{E}_{\mathbf{g}}^{\text{grid}}\left(\{(I_i, t_i)\}_{i=1}^{T}\right) = [\tilde{I}_1, \dots, \tilde{I}_K]. \tag{4.35}$$

We present a conundrum, illustrated later by an example. Equation (4.30) shows a tradeoff when we keep the grid size K constant. First, we have an effective context range of $\frac{K}{\Delta}$. Secondly, with a larger Δ, more grid points are assigned to the same index and, through occupancy (4.32), are removed when multiple time points are present. For discrete series, this is not an issue. For very irregular series, a trade-off arises. Figure 4.8 illustrates this grid quantization with its quantizer and quantizer error.

4.3.2 Architecture Details

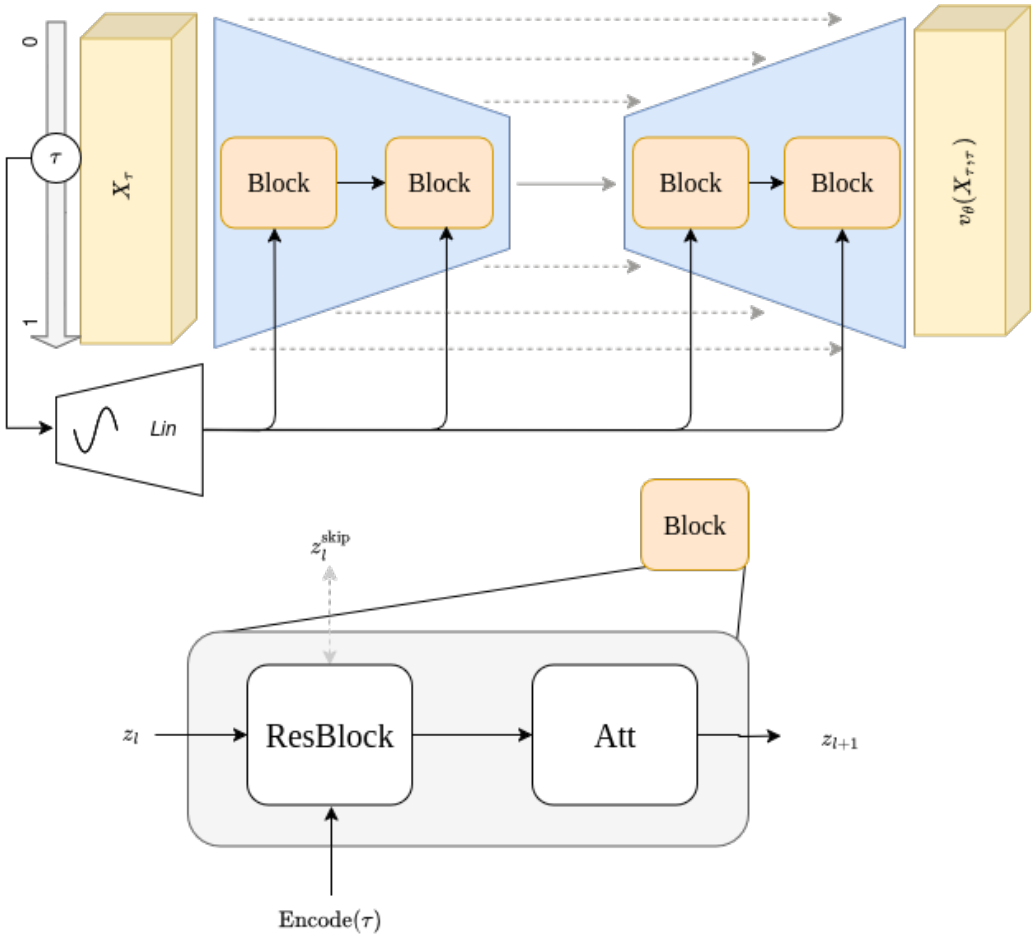


Figure 4.9: **TFM Architecture.** The model takes as inputs a sample X_τ and a time τ. The scalar τ is embedded into the model dimensions using sinusoidal embeddings, followed by a linear layer. This produces a high-dimensional vector, which is then added to each ResBlock. The core of the network is a classical U-Net architecture, consisting of a downsampling path, a bottleneck, and an upsampling path. After each ResBlock, there is an attention block that allows the model to focus on different parts and time steps of the input. Finally, a 1×1 convolutional layer is used to produce the output velocity field.

The TFM network follows a standard U-Net backbone adapted from the TorchCFM library [131]. The scalar flow step τ is embedded via sinusoidal time encoding followed by a two-layer MLP and added to each ResBlock through feature-wise modulation. The encoder-decoder consists of ℓ ResBlocks ($[1, 1, 2, 4]$ expansion ratio)

with progressive downsampling and upsampling, while attention layers are applied after each block to utilize the embedding τ. Implementation specifics, including channel sizes and embedding dimensions, are provided in Appendix 7.2.

Complete Algorithm Algorithm 3 outlines the complete training and inference process of Temporal Flow Matching. During training, random flow steps τ and corresponding interpolation X_τ get sampled uniformly. The network is trained via a MSE loss between true velocity and the network $v_\theta(X_\tau, \tau)$. At inference, the learned flow field is integrated over $\tau \in [0, 1]$ via an ODE solver to reconstruct the target image.

Algorithm 3 Temporal Flow Matching: Training and Inference

Require: Patients $\Pi = \{[\mathcal{I}_1, I_{\text{target},1}], \ldots, [\mathcal{I}_p, I_{\text{target},p}]\}$ and initial network v_θ

while training **do**
  $[\mathcal{I}, I_{\text{target}}] \sim \Pi$ ▷ pick a random patient
  $\tau \sim \mathcal{U}(0, 1)$ ▷ pick a random flow step
  $\mathcal{I}_{\text{target}} \leftarrow [I_{\text{target}}, \ldots, I_{\text{target}}]$ ▷ Extend the dimension of I_{target} T times
  $\mathcal{I}' \leftarrow$ Sparsity Filling$(\mathcal{I})$ ▷ Fill empty images
  $X_\tau \leftarrow (1 - \tau)\,\mathcal{I}' + \tau\,\mathcal{I}_{\text{target}}$ ▷ Calculate the linear interpolation between each context and target
  $\mathcal{L}_{\text{TFM}} \leftarrow \left\| v_\theta\big(\tau,\, X_\tau\big) \;-\; (\mathcal{I}_{\text{target}} - \mathcal{I}')\right\|_2^2$ ▷ Calculate the velocity loss
  Update $\theta \;\leftarrow\; \text{AdamW}\,(\nabla_\theta \mathcal{L}_{\text{TFM}})$
end while
return v_θ
if inference **then**
  Initialize $X_0 \leftarrow \mathcal{I}'$
  Define integration grid $\{\tau_0 = 0, \ldots, \tau_N = 1\}$ with n steps
  $\hat{X}_{0:N} \leftarrow \text{ODEInt}\big(v_\theta, X_0,\, \{\tau_0, \ldots, \tau_N\}\big)$ ▷ numerically integrate the network
  return $\hat{X}_N$
end if

4.4 Extensions

While our method achieves strong results on image reconstruction metrics, several technical limitations remain to be addressed. The most critical of these is the handling of continuous-time modeling. Fortunately, Flow Matching naturally lends itself to this setting, an insight which aligns well with its connection to NODEs. Beyond this, we introduce additional extensions aimed at enhancing performance, rather than overcoming core limitations. Specifically, we explore deformation-aware modeling and Schrödinger Bridge formulations as potential future directions. In the following, we outline their conceptual motivations and present preliminary experimental results to illustrate their promise.

4.4.1 Continuous Time

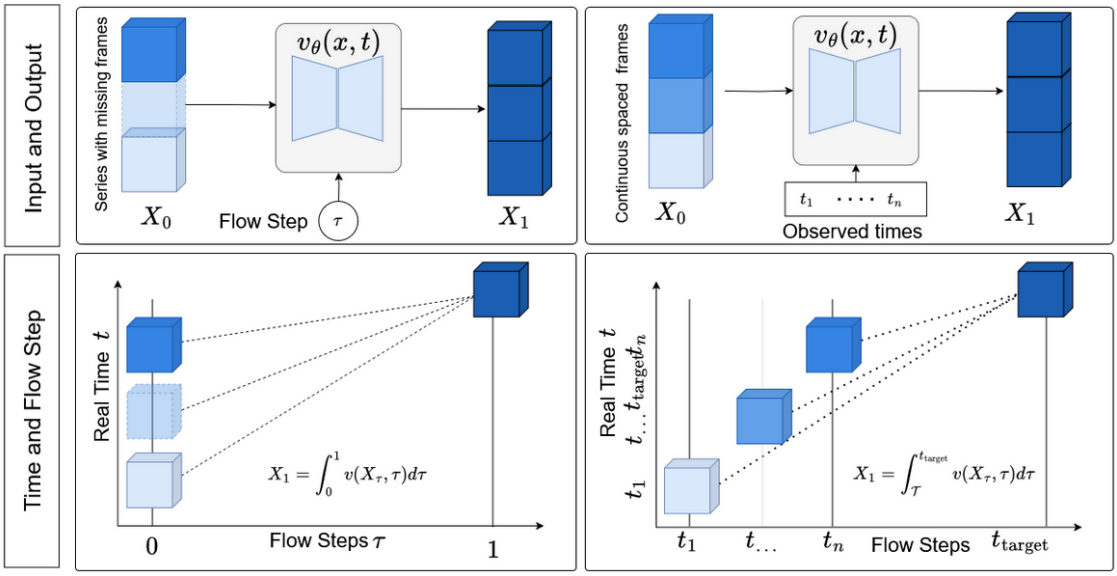


Figure 4.10: **Comparison between discrete TFM time embedding (left) and continuous TFM time embedding (right)**. In the discrete formulation, all input frames are discretized onto a regular grid, and the Flow Matching step $\tau \in [0, 1]$ interpolates between the start and target frames. In the continuous formulation, each input frame I_i is associated with its true time step t_i. This is done via a per-frame continuous coordinate $t_i = (1 - \tau)t_i + \tau t_{\text{target}}$. Using real-time time steps enables the model to learn the dynamics even under irregular time intervals.

While the discrete Flow Matching formulation performs well on regularly sampled data, it remains constrained by the assumptions of a fixed, uniformly spaced tem-

poral grid. This design choice simplifies implementation. However, it limits applicability when acquisitions are highly irregular or when the temporal resolution differs across subjects. We address this limitation by extending TFM to operate in **continuous time**, allowing the model to directly condition on real-valued timesteps.

Flow Steps as Continuous Time The Flow Matching U-Net (see Figure 4.9) already embeds the scalar FM step τ, through a sinusoidal encoding and feature-wise modulation at each ResBlock. In the original FM formulation, this step is treated as an abstract interpolation variable between two distributions. Here, we re-interpret τ as a representation of **actual observed time**. [1] This modification requires only a minimal change to the conditioning mechanism: Instead of embedding a uniform scalar step, the network now conditions on continuous time coordinates derived from the true acquisition times [2].

Continuous Time Conditioning

We extend the discrete scalar step τ into a continuous time vector that directly contains the true acquisition times. Let $[t_0, \dots, t_n]$ denote the timepoints associated with the input images, and let t_{target} represent the target time. We define the transformation

$$\tau \mapsto \mathcal{T}(\tau) \coloneqq (1-\tau) \begin{bmatrix} t_0 \\ \vdots \\ t_n \end{bmatrix} + \tau \begin{bmatrix} t_{\text{target}} \\ \vdots \\ t_{\text{target}} \end{bmatrix}. \tag{4.36}$$

Each entry in $\mathcal{T}(\tau)$ represents the interpolated time corresponding to one channel or input image. With this reparameterization, the network input becomes

$$v(X_\tau, \tau) \to v(X_\tau, \mathcal{T}(\tau)) \tag{4.37}$$

The underlying flow dynamics remain identical to the discrete case:

$$\frac{d}{d\tau}\psi_\tau(x) = u_\tau(\psi_\tau(x)). \tag{4.38}$$

Accordingly, the interpolation between source and target images also remains unchanged:

$$X_\tau = (1-\tau)X_0 + \tau X_1. \tag{4.39}$$

[1] This is the reason why we were explicit to call τ as step before, and why we renamed t in the literature to τ.

[2] A concurrent approach, Zhang et al. [158], also addresses irregular temporal data by modeling the flow between consecutive samples. However, their formulation operates recurrently and is currently suited for low-dimensional trajectories. Extending it to our high-dimensional 3D network would require substantial architectural modifications

This interpolated sample X_τ is passed through the network, while the corresponding time vector conditioning is now derived from $\mathcal{T}(\tau)$ instead of τ.
The time vector is encoded via elementwise sinusoidal embedding, followed by summation across all frames, in order to preserve the dimensionality of the embedding independent of n:

$$E(\mathcal{T}(\tau)) = \frac{1}{n}\sum_{i=0}^{n} \text{SinEnc}(\mathcal{T}(\tau)_i). \tag{4.40}$$

This highlights a potential drawback: many time points may drown out any temporal signal. The overall Flow Matching loss remains structurally identical, except that the velocity field now depends on the continuous time vector:

$$\mathcal{L}^{OT}_{\text{continuous } TFM} = \mathbb{E}_{\tau\sim\mathcal{U}(0,1),\{X_0,X_1\}\sim\Pi}||v_\theta(X_\tau, \mathcal{T}(\tau)) - (X_1 - X_0)||, \tag{4.41}$$

At inference, we solve the same ODE inverse integration problem, now parametrized via:

$$\frac{dX_\tau}{d\tau} = v_{\theta(X_\tau,\mathcal{T}(\tau))}, \tag{4.42}$$

This formulation yields one timestamp per input frame, allowing the model to directly condition on real acquisition times without relying on a fixed temporal grid. As a result, this variation of TFM **can naturally handle irregularly sampled sequences** while preserving the original structure, and even improve the efficiency. Moreover, since no dense zero-filling is required, the continuous variant even reduces memory usage and improves scalability, particularly in datasets with highly varying acquisition schedules (see Figure 4.8 as an example).

Further Addenda.

Since the previous methods were not that reliant on the explicit time conditioning, we are looking for ways to improve.

Velocity Attenuated Loss Classical Flow Matching assumes transport between a random distribution and another distribution, whereas in our longitudinal case, large portions of the velocity field are close to zero. This imbalance can bias the network toward static flows. To counteract this, we can reweight the loss by the magnitude of the true velocity field:

$$\mathcal{L}_{\text{Velocity Attenuated}} = \frac{1}{S*T}\sum_{i=1}^{S*T} \|u_i * (u_i - v_{\theta,i})\|_2^2, \tag{4.43}$$

This attenuates the contribution of stationary regions, emphasizing voxels that change significantly.

Contrastive Learning. To further enhance temporal sensitivity, we integrate a contrastive self-supervised term inspired by [15] and which was used in [83]. Unlike classical contrastive methods such as SimCLR [14], which require large batches of negatives, or MoCo [45], which maintains a memory queue, SimSiam operates without explicit negatives. It promotes representational consistency between two different views, in our case, between close adjacent temporal embeddings. Formally, let $f_\theta(\cdot)$ denote the encoder (here, our U-Net backbone). Given an input x, we compute two stochastic augmentations x_1, x_2 (in our case, temporal augmentations). The encoder produces latent features $h_1 = f_\theta(x_1)$ and $h_2 = f_\theta(x_2)$. A projection head $g(\cdot)$ maps these features into a lower-dimensional embedding space, yielding $z_1 = g(h_1)$ and $z_2 = g(h_2)$. A predictor $q(\cdot)$ is then applied asymmetrically to one branch, producing $\hat{z}_1 = q(z_1)$. The loss is the negative cosine similarity:

$$\mathcal{L}_{\text{SimSiam}} = -\frac{\hat{z}_1^\top z_2}{\|\hat{z}_1\|\|z_2\|}, \tag{4.44}$$

symmetric over both directions ($1 \to 2$ and $2 \to 1$). In practice, $g(\cdot)$ is a linear projection with batch normalization, and $q(\cdot)$ is a lightweight MLP.

Temporal Augmentations. Since absolute time is often less relevant than relative progression, we introduce **temporal jittering** as a stochastic augmentation. We perturb the input timesteps by small Gaussian offsets:

$$\vec{t} \mapsto \vec{t} + \mathcal{N}(0, \sigma^2), \tag{4.45}$$

thereby encouraging invariance to global time shifts while preserving local ordering.

4.4.2 Schrödinger Bridges

Schrödinger Bridges generalize deterministic Flow Matching by introducing a stochastic regularization. Schrödinger Bridge (SB) have been proposed by [8, 140], and the more modern, simulation-free variant [133] and their generalized SB variant from [84]. Our preliminary results confirm the technical feasibility. However, they also highlight challenges, particularly with our approach, which calculates Flows at the voxel level. This section briefly shows the training and inference code. We also describe changes needed to adapt voxel-based Flows. It is important to note that our results remain exploratory but promising.

Training and Inference To provide specifics, Algorithms 4 and 5 outline the training and inference schemes for the Schrödinger Bridge TFM. In contrast to the standard TFM, where the velocity field is estimated via optimal transport, the SB formulation learns a different velocity and score [133]. During inference, we use the Euler-Maruyama method. This allows us to simulate forward trajectories that combine deterministic drift and data-dependent noise.

Algorithm 4 Score-and-Flow Matching Training

Require: Source prior $q(z)$, conditional path $\{p_t(x \mid z)\}_{t\in[0,1]}$, velocity field $u_t(x \mid z)$, weighting schedule $\lambda(t)$, initial networks v_θ, s_θ

while not converged **do**
  $z \sim q(z)$
  $t \sim \mathcal{U}(0,1)$
  $x \sim p_t(x \mid z)$
  $\mathcal{L}_v \leftarrow \left\| v_\theta(x,t) - u_t(x \mid z) \right\|_2^2$ ▷ Flow Matching (velocity) loss
  $\mathcal{L}_s \leftarrow \left\| s_\theta(x,t) - \nabla_x \log p_t(x \mid z) \right\|_2^2$ ▷ Score Matching loss
  $\mathcal{L}_{SBM} \leftarrow \mathcal{L}_v + \lambda(t)^2 \mathcal{L}_s$ ▷ Combine with time-dependent weighting
  $\theta \leftarrow \text{Update}(\theta, \nabla_\theta \mathcal{L}_{SBM})$
end while
return v_θ, s_θ

Algorithm 5 Simulation-Free Schrödinger TFM (Euler–Maruyama integration)

Require: Flow network v_θ, score network s_θ, diffusion schedule $g(t)$, step size Δt, source prior q_0

Sample $x_0 \sim q_0(x)$
for $n = 0$ to $N-1$ **do**
  $t \leftarrow n\,\Delta t$
  $u_t \leftarrow v_\theta(x_t,t) + \frac{g(t)^2}{2} s_\theta(x_t,t)$
  $x_{t+\Delta t} \sim \mathcal{N}\big(x_t + u_t\,\Delta t,\ g(t)^2\,\Theta(v_\theta(x_t,t) > T)\Delta t\, I\big)$ ▷ Euler–Maruyama step with noise mask
end for
return $x_{N\,\Delta t}$

Noise masking for voxel-level changes A consideration is that in most medical spatio-temporal series, most voxels remain static over time, with only small regions exhibiting change. To avoid perturbations in voxel space where change is scarce, we restrict the diffusion noise to these dynamic regions. We apply a spatial mask M derived from the magnitude of the predicted velocity field:

$$\sigma = \Theta(u_\theta > T) \cdot g(t) \cdot \sqrt{\Delta t}, \tag{4.46}$$

where T is a threshold, and Θ is the Heaviside step function. Additionally, inference-time noise is scaled relative to the training noise

$$\sigma_{\text{inference}} = C \cdot \sigma_{\text{train}}, \tag{4.47}$$

with C controlling the strength of stochasticity. This selective noise stabilizes reconstructions and better aligns stochastic sampling with regions of change.

4.4.3 Beyond Mass Generation: From Flows to Deformations

Not all voxel-space flows correspond to physically meaningful transformations. In some cases, the predicted flow fields represent changes in intensity rather than actual spatial motion, leading to the appearance or disappearance of mass. Such behaviour can be problematic when modeling physical or biological processes that are expected to conserve intensity or that describe deformations. Deformation-based formulations, in contrast, preserve mass by explicitly transporting intensity through space[3]. Both formulations are theoretically valid, but they model different physical or image-based processes. Figure 4.11 illustrates this: the left panel shows the ground truth, the middle panel depicts a flow-based transformation that alters mass, and the right panel demonstrates a deformation-based mapping that conserves it.

4.4.4 Deformation-aware Flow Matching

This section presents an exploratory extension of our method using deformation fields. The ideas herein are **preliminary** and self-contained, reflecting a conceptual hunch rather than a fully developed method. Although we lack rigorous mathematical guarantees for the combined deformation and FM formulation, we nonetheless provide the following math as a guiding hypothesis for future investigations. While Flow Matching effectively models transformations between arbitrary distributions, its standard formulation assumes pixel-wise evolution, where each spatial position

[3]Strictly speaking, standard deformation fields preserve spatial correspondence rather than physical mass. True mass conservation requires either a volume-preserving deformation or reweighting of intensities. On the other hand, all mass-conserved deformations are contained within deformations.

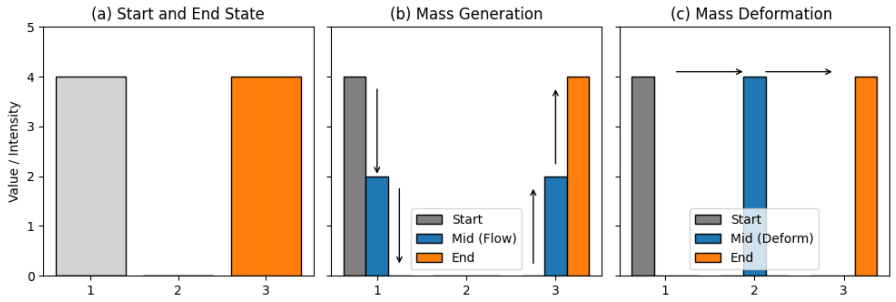


Figure 4.11: **Deformation vs Mass Generation** This figure illustrates the conceptual difference between voxel-wise mass generation and deformation. (a) Shows the starting and the end state, where gray is the starting state and orange is the final state. Blue bars represent intermediate states. (b) Mass generation and deletion represent the intensity transport as a creation and destruction process. Values at the start decrease while new intensity appears elsewhere. (c) Shows a mass displacement, where the same intensity is spatially shifted, preserving the mass.

changes independently over time. In medical imaging, however, many physiological processes are better described as spatial deformations rather than direct intensity changes. For instance, cardiac motion in ACDC corresponds mainly to tissue displacement and volume change, while brain atrophy manifests as slow, spatially coherent shrinkage. In such cases, pure Flow Matching requires large voxel-wise velocity updates to approximate geometric motion, which may be inefficient. (Yet, how to quantify this possible inefficiency is not trivial, and we leave confirmation for future work. Here, we leave this solely as motivation.)
To address this possible limitation, we implement Deformation-aware Flow Matching, an extension that first models geometric deformation and then intensity evolution. Conceptually, the image evolution is decomposed into two coupled components:

1. a deformation field capturing spatial displacement, and
2. a Flow Matching term accounting for local intensity or mass changes.

Figure 4.11 illustrates this relationship: classical optimal transport (OT) displaces mass but does not create, while flow matching interpolates vertically in intensity space (in our formulation). An additional perspective of optimal transport with mass generation can be found in Figure 7.1
Although biological changes are not strictly mass-preserving, the deformation provides a physically meaningful prior that could stabilize learning and better represent motion. We therefore treat the deformation before the flow, so that the flow is calculated after the images have been transformed. The underlying network is unchanged, except for increasing the channel size.

Optimal Transport vs. Flow Matching We approach this problem from a complementary perspective. Since deformation fields correspond to mass-preserving transformations, they are naturally described via optimal transport. In its dynamic formulation, OT defines a transport plan that minimizes kinetic energy while conserving mass:

$$\mathcal{W}_2(\rho_0, \rho_1) = \inf_{\gamma, v} \int_0^1 \int_\Omega \rho(t,x) \|v(t,x)\|^2 dx dt, \tag{4.48}$$

subject to the continuity equation

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho v) = 0. \tag{4.49}$$

This purely **deformational model** enforces strict mass preservation, making it inadequate for problems which involve local intensity changes (for us, perfusion CT is a counterexample). To address this, one can introduce a source term $s(t,x)$, which allows mass creation over time, leading to extensions such as the Fisher-Rao metric:

$$\mathcal{W}_{FR}(\rho_0, \rho_1) = \inf_{\gamma, p} \int_0^1 \int_\Omega \frac{p(t,x)^2}{\rho(t,x)} + \gamma \frac{s(t,x)^2}{\rho(t,x)} dx dt, \tag{4.50}$$

subject to

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho v) = s(t,x). \tag{4.51}$$

[4] In contrast, Flow Matching operates directly in the distribution space, transporting probability mass vertically in density space rather than horizontally in spatial coordinates (Fig. 7.1).

3D Warping. Given the last context image

$$x_{\text{prev}} \in \mathbb{R}^S \tag{4.52}$$

and a predicted voxel flow field

$$f \in \mathbb{R}^{3 \times S}, \tag{4.53}$$

(with components f_x, f_y, f_z in voxel units along (W, H, D)). We define a normalized sampling grid in $[-1, 1]^3$:

$$g_{\text{base}}(i,j,k) = \left(\frac{2k}{W-1} - 1,\ \frac{2j}{H-1} - 1,\ \frac{2i}{D-1} - 1 \right). \tag{4.54}$$

[4] As shown in [18, Eq. 5.7], the relative weighting γ governs the trade-off between transport and mass variation. For small γ, creation and destruction dominate, whereas for large γ, transport is favored. In practice, this balance must be assessed empirically.

The flow field is normalized to the same range:

$$\Delta g_x = f_x \cdot \frac{2}{W-1}, \quad \Delta g_y = f_y \cdot \frac{2}{H-1}, \quad \Delta g_z = f_z \cdot \frac{2}{D-1}. \tag{4.55}$$

The deformed sampling grid is then

$$g = g_{\text{base}} + (\Delta g_x,\, \Delta g_y,\, \Delta g_z), \tag{4.56}$$

and the warped volume is obtained via trilinear resampling:

$$x_{\text{warp}} = \text{grid\_sample}(x_{\text{prev}},\, g,\, \texttt{mode} = \text{bilinear},\, \texttt{padding\_mode} = \text{border}). \tag{4.57}$$

This operation produces x_{warp} at the deformed coordinates specified by the flow.

Technical Implementation. We adopt a minimal yet effective extension of the continuous TFM model to incorporate deformation fields. Recall that the continuous model predicts a velocity field for each time step over the spatial domain.

$$\mathcal{M} : \mathbb{R}^{T \times S} \to \mathbb{R}^{T \times S}, \tag{4.58}$$

To reduce memory requirements, we apply the deformation only to the last context image. Formally, we extend the Flow Matching model to

$$\mathcal{M}' : \mathbb{R}^{T \times S} \to \mathbb{R}^{(T+3) \times S}, \tag{4.59}$$

where

$$\mathcal{M}'(\mathcal{I}, \mathcal{T}) \coloneqq \big[v_{\text{FM}},\, f_{\text{LCI}}\big], \tag{4.60}$$

with v_{FM} denoting the velocity prediction from the TFM model and f_{LCI} the deformation field that maps the last context image to the target image. Thus, the only architectural modification is increasing the output channel dimension to accommodate the deformation field, while all other settings remain unchanged. While we implemented this deformation-aware formulation and verified its technical feasibility, comprehensive large-scale experiments are left for future work.

Results 5

In this chapter, we present both quantitative and qualitative results obtained with the methods introduced in the previous sections. We begin in Section 2.5.1 by discussing the results of Neural Processes, Neural ODEs, and the Attentive Segmentation process. These models are evaluated on the ADNI dataset, which contains longitudinal MRI scans of Alzheimer's patients, and on a controlled synthetic ellipses dataset. Since the baselines *showed limited performance*, we further evaluated several natural imaging baselines on this synthetic ellipses dataset to verify that the task itself is learnable. After confirming this, we extended the evaluation to the ACDC dataset, where these baselines demonstrated that the prediction task can indeed be learned on medical data.
Next, Section 4.2 introduces our longitudinal data augmentation approach. We analyze both qualitative realism and quantitative impact as augmentations for methods on the ACDC dataset. Although not the primary focus of this thesis, these experiments provided valuable benchmarks that aided the development of the proposed flow-based approaches. We also note that this augmentation and generation strategy may have broader applicability beyond the scope of this work.
Finally, Section 4.3 presents the results of the core contribution of our work: benchmarking the proposed TFM method across three datasets, ACDC, Lumiere, and ISLES. We compare the approach against established spatio-temporal natural-image baselines, including Convolutional LSTM (ConvLSTM), Video Vision Transformer (ViViT), and SimVP (SimVP), and demonstrate consistent improvements across datasets. In Section 5.4, we further investigate several methodological extensions; Most crucially, the continuous variant addresses one of the main limitations of its discrete formulation. Furthermore, we also explore a deformation-field model and a potential extension towards Schrödinger Bridges. Together, these experiments demonstrate that our flow-based method is scalable, efficient, and applicable across diverse medical imaging scenarios.

5.1 Spatio Temporal Experiments with Neural Processes

In this section, we present the experimental results for Neural Processes, the Attentive Segmentation Process, and other variants using Neural ODEs (NODEs). We

begin with the results from the longitudinal brain MRI data in the ADNI dataset. There, the first goal was to determine whether NODEs can improve spatio-temporal modeling. Second, after the improvements were marginal, we tried to reduce the attention cost by replacing the computationally intensive attention modules in the skip connections. followed by evaluations on synthetic segmentation experiments designed to probe for failures in spatio-temporal learning. Finally, we extend these synthetic experiments to natural-imaging methods to confirm that they work as intended and test them on the ACDC dataset.

5.1.1 Neural Processes and Neural ODEs

We first evaluate the impact of integrating Neural ODEs into the image reconstruction variant of ASP. The experiments are conducted on longitudinal brain MRI from the ADNI cohort. Preprocessing involves DICOM to Nifti conversion using Plastimatch[2], rigid registration to MNI space using FLIRT[55], and brain extraction using HD-BET[51]. During training and testing, one 2D slice per subject is sampled along a random anatomical plane, with one target time and up to three context times. Networks are optimized using MSE and SSIM and the Adam optimizer [65] trained on a batch size of 8. Performance is measured on a held-out validation set. Integrating NODEs into ASP only yields marginal quantitative improvement, compared to the baseline ASP (see Table 5.1). However, the overall SSIM values are lower than those reported for similar experiments.

Model description	MSE	$SSIM$
ASP (adapted)	0.071	0.309
ASP (adapted) + neural ODE	0.070	0.309
Summed representations	0.072	0.305
Summed representations + neural ODE	0.066	0.308

Table 5.1: Results comparing different model architectures. The task is predicting a $2D$ slice (randomly chosen from transverse, coronal, or sagittal axis) of a MRI image with a randomly chosen time-point. The loss is comparing the reconstruction error between the network prediction and the ground truth. Trained using the $SSIM$ loss.

5.1.2 Neural Processes: Attention Alternatives

Model	Time	$MSE(10^{-2}) \downarrow$	$LPIPS \downarrow$	$PSNR \uparrow$	$SSIM(10^{-2}) \uparrow$
ASP	✓	**3.199** (0.126)	5.74 (1.66)	**34.50** (0.38)	**47.40** (3.62)
	x	**3.221** (0.143)	6.32 (3.04)	**34.44** (0.42)	**46.98** (5.43)
l-VAE	✓	4.955 (0.005)	14.65 (0.05)	30.06 (0.01)	27.78 (0.03)
	x	4.982 (0.024)	14.91 (0.05)	29.99 (0.03)	27.50 (0.10)
Mamba skip	✓	3.435 (0.016)	5.82 (0.22)	33.78 (0.04)	45.95 (0.29)
	x	3.436 (0.024)	5.87 (0.19)	33.80 (0.05)	45.77 (0.36)
Random I_i	x	3.839	**3.43**	32.69	**47.41**

Table 5.2: **Quantitative ADNI results**. The label *With time* indicates that the models receive the relative time input, whereas *masked time* denotes that temporal information is not provided. We compare the three models against a simple baseline in which a random input image I_i, with $i \in [k]$, is inserted in place of the prediction; metrics are then computed between I_i and the target image I_{target}. Missing time is denoted via x.

We further assess spatial feature encoding strategies on the ADNI dataset (Table 5.2). Since the previous attempt performed rather poorly, we omitted the random sampling from all axes, and instead only sampled slices from a specific axis. Models are tested both with explicit relative time inputs and with masked time information. Including time conditioning does not improve performance, which was the first negative signal. Interestingly, a simple random-context baseline achieves comparable SSIM and way better LPIPS, suggesting that the models are not better than a random baseline in terms of these metrics. This led us to question whether the models were unable to learn or whether our experimental setup was suboptimal, prompting us to perform synthetic segmentation experiments.

5.1.3 Synthetic Segmentation Experiments

Since the image reconstruction experiments yielded negative results, we wanted to isolate temporal consistency effects. For this reason, we design the synthetic segmentation experiments (growing ellipses, see 3.1). Each sequence simulates a linear growth of an ellipse, with shear and rotation, while only the size varies over time (see Table 5.3 for parameter settings). Models are trained with Dice loss for 100 epochs using Adam (1×10^{-4} learning rate with cosine decay), using a batch size of 32.

Results in Table 5.4 ([24]) show that while ASP achieves the highest Dice scores across all difficulty settings, *however* the LCI heuristic still is better in experiment

ID 5. The remaining methods perform significantly worse than ASP and exhibit inconsistent behavior across different experimental settings. These findings suggest two key observations: first, that the NP backbone may be inherently unstable when used in isolation; and second, that replacing the attention mechanism within ASP is not trivial. Even ASP shows signs of instability under certain conditions, indicating that the overall backbone may not be fully optimal. Motivated by these results, we next evaluate natural image baselines on the very same experimental settings to determine whether these limitations arise from model design or from the intrinsic difficulty of the synthetic experiments.

ID	**Shape**	**N**	**Growth (m)**	**start**	**shear (s)**	**time**
0	Ellipse	$[1, 4]$	Var ($m \in [3, 8]$)	$[4, 6]$	$[0, 3, 1.0]$	regular
1	Circle	$[1, 4]$	Var ($m \in [3, 8]$)	$[4, 6]$	1	irregular
2	Ellipse	$[1, 4]$	Fixed ($m = 6$)	$[4, 6]$	$[0, 3, 1.0]$	irregular
3	Ellipse	$[1, 4]$	Var ($m \in [3, 8]$)	$[4, 6]$	$[0, 3, 1.0]$	regular & masked
4	Ellipse	$[1, 4]$	Var ($m \in [3, 8]$)	$[4, 6]$	$[0, 3, 1.0]$	irregular
5	Ellipse	[1,3]	Var ($m \in [2, 10]$)	$[3, 8]$	$[0.3, 1.3]$	irregular

Table 5.3: **Parameters for the synthetic experiments.** The results are shown in Table 5.4. The nomenclature is given in 3.1. N is the number of objects, *growth* is the growth rate, *shear* is the shear of the ellipses, *start* is the minimum size of the object, *time* indicates whether the time points are irregular and whether they are masked. *Var* means that the growth is uniformly sampled from the interval, *Fixed* means the objects have a fixed growth rate.

ID	**Shape**	**Growth**	**Time**	**NP**	**Mamba**	**ASP**	$\boldsymbol{I_k}$
0	Ellipse	Var ($m \in [3, 8]$)	reg	76.66 (0.75)	81.47 (0.76)	**94.32** (0.57)	84.98
1	Circle	Var ($m \in [3, 8]$)	irreg	84.81 (0.83)	84.87 (1.12)	**90.36** (0.08)	85.33
2	Ellipse	Fixed ($m = 6$)	irreg	83.42 (2.92)	86.41 (0.51)	**89.29** (0.31)	83.38
3*	Ellipse	Var ($m \in [3, 8]$)	reg $*0$	76.89 (0.88)	80.67 (2.46)	**90.27** (0.14)	84.98
4**	Ellipse	Var ($m \in [3, 8]$)	irreg	84.90 (0.84)	84.50 (0.97)	**89.87** (0.30)	84.98
5**	Ellipse	Var ($m \in [2, 10]$)	irreg	73.84 (1.99)	76.42 (1.62)	83.62 (0.34)	**86.26**

Table 5.4: **Dice score results (%) of synthetic experiments**. I_k represents the dice score between the previous input image and the target image, averaged over more than 2k random object generation runs for comparison. The rounded brackets are the standard deviation for the models over 3 runs. The resolution is 64×64. The exact values for each parameter can be found in Table 5.3. For shorthand, we will denote each experiment by an ID, which can be interpreted as a *relative difficulty*. * time is *masked*. ** See Table 5.3 for differences.

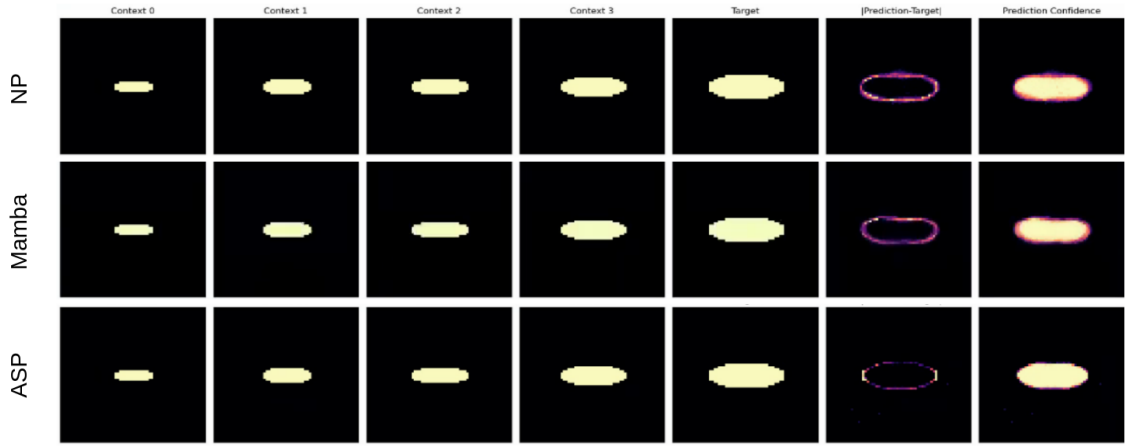


Figure 5.1: **Qualitative results on the synthetic ellipse dataset.** Each row shows the context images and the predicted image by a different model (NP, Mamba, ASP) given our four context frames. The target columm shows the ground truth, while the $|Prediction - Target|$ highlights segmentation errors, and the prediction confidence shows uncertainty in the predicted regions. NP and Mamba exhibit noticable deviations at the object boundaries, whereas ASP yields sharper predictions, albeit not perfect one. Mamba, despite having skip-connections, does not perform better than the NP.

Qualitative Results Interestingly, although the reconstructions for our Mamba variant shown in Figure 5.2 seem better than the ones of ASP shown in Figure 5.3, the quantitative metrics tell a different story. Despite producing images which look sharper, the reconstruction metrics are mostly better for the ASP. However, for the ellipses dataset, the Mamba variant performs significantly worse, see Figure 5.1.

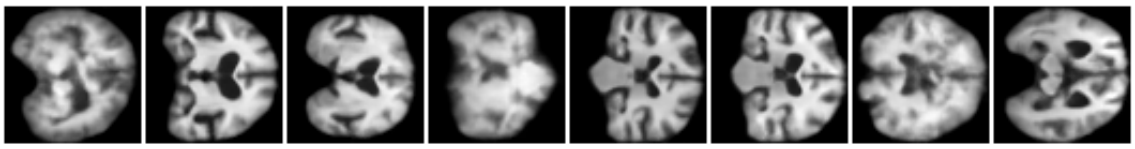


Figure 5.2: **Qualitative Results for Mamba on the ADNI dataset**

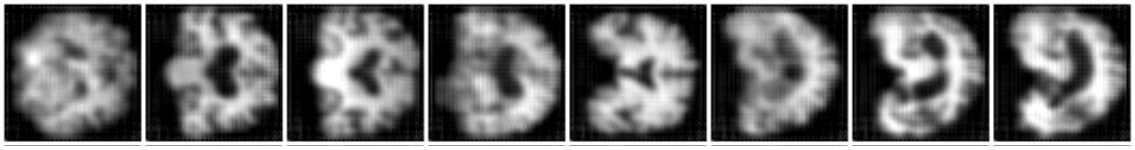


Figure 5.3: **Qualitative Results for ASP on the ADNI dataset**

Natural Imaging Baselines

We next evaluate the natural-imaging architectures within our synthetic experimental settings, focusing first on SimVP [34]. Our goal is twofold: first, to test whether high-performing video prediction methods can solve the synthetic task where our previous methods struggled; and second, to assess whether these methods generalize to medical image time series. As a note, all experimental settings are kept identical, but the optimization is performed in the OpenSTL framework [126], as it was more convenient for testing the available methods.
To ensure a fair comparison, we maintain identical data configurations and input distributions, as well as identical evaluation protocols. We focus primarily on the two most challenging configurations, where ASP failed to significantly outperform the LCI heuristic. As shown in Table 5.5, SimVP achieves substantially higher and more stable performance across both settings. These findings lead to two conclusions: the synthetic dataset is not inherently too difficult, and the Neural Process appears to be a suboptimal backbone for this type of task. Consequently, subsequent experiments are based primarily on the natural-imaging baselines.

ID	Shape	Growth	Time	NP	Mamba	ASP	SimVP	LCI
4**	Ellipse	Var ($m \in [3, 8]$)	irreg	84.90	84.50	89.87	**96.60**	84.98
5**	Ellipse	Var ($m \in [2, 10]$)	irreg	73.84	76.42	83.62	**95.60**	86.26

Table 5.5: **Dice scores (%) for difficult synthetic ellipses with SimVP.** Each configuration (ID) defines a specific object shape, growth rate range, and temporal sampling pattern. Columns show mean Dice scores over 2,000 runs for Neural Processes (NP), Mamba, ASP, and SimVP, with I_k denoting the Dice score between the last input and target frame. Parameter details are listed in Table 5.3; all experiments use 64×64 resolution.

2D ACDC Experiments

To further validate these findings on real data, we conducted experiments on the 2D ACDC dataset using natural-imaging baselines. Our main objective was to determine whether the strong performance of SimVP on the synthetic ellipses also generalizes to medical imaging. The regularity of the cardiac cycle makes this dataset particularly suitable for testing temporal models. Another practical motivation was the availability of an established pre-processing pipeline from [154], ensuring consistency from raw data to dataloader level, unlike ADNI, which required several manual preprocessing steps.
We follow the aforementioned patient-wise splits (90 for training, 10 for validation, and 50 for test). Each sequence contains 12 frames from the end-diastole(ED) to

end-systole (ES). Volumes are resampled to $128 \times 128 \times 32$ and normalized, and eight frames are randomly sampled per sequence for training. The slice showing the largest temporal change is selected for evaluation. We use four context frames to predict four future frames and evaluate on NRMSE, SSIM, and PSNR.
In addition to SimVP, we benchmarked against several widely used sequence models, including ConvLSTM [121], MIM [141], and SwinLSTM [129], all of which were from the OpenSTL benchmark framework by Tan et al. [126]. We also report the LCI heuristic, which highlights the amount of change within the sequence.
The results in Table 5.6 show that all methods outperform LCI, confirming that the task is learnable. Moreover, the performance spread across methods indicates that the dataset is not trivial. Interestingly, the best performing method is not the strongest natural-imaging baseline (see [126] results), highlighting that there exists a gap between natural imaging and medical imaging baselines. The results also indicate that recurrence-based methods remain performant, while small refinements such as convolutional or Swin provide only incremental differences.

Model	NRMSE (↓)	SSIM (↑)	PSNR (↑)
LCI	0.1055	83.40	23.96
SimVP [34]	0.0859	84.39	25.22
ConvLSTM [121]	**0.0631**	**90.77**	28.37
MIM [141]	0.0648	**90.77**	**28.40**
SwinLSTM [129]	0.0677	89.03	27.47

Table 5.6: **Quantitative comparison of different methods on the 2D ACDC dataset.** We report three common image quality metrics: normalized root mean squared error (NRMSE, ↓), structural similarity (SSIM, ↑), and peak signal-to-noise ratio (PSNR, ↑). Lower NRMSE and higher SSIM/PSNR values indicate better performance.

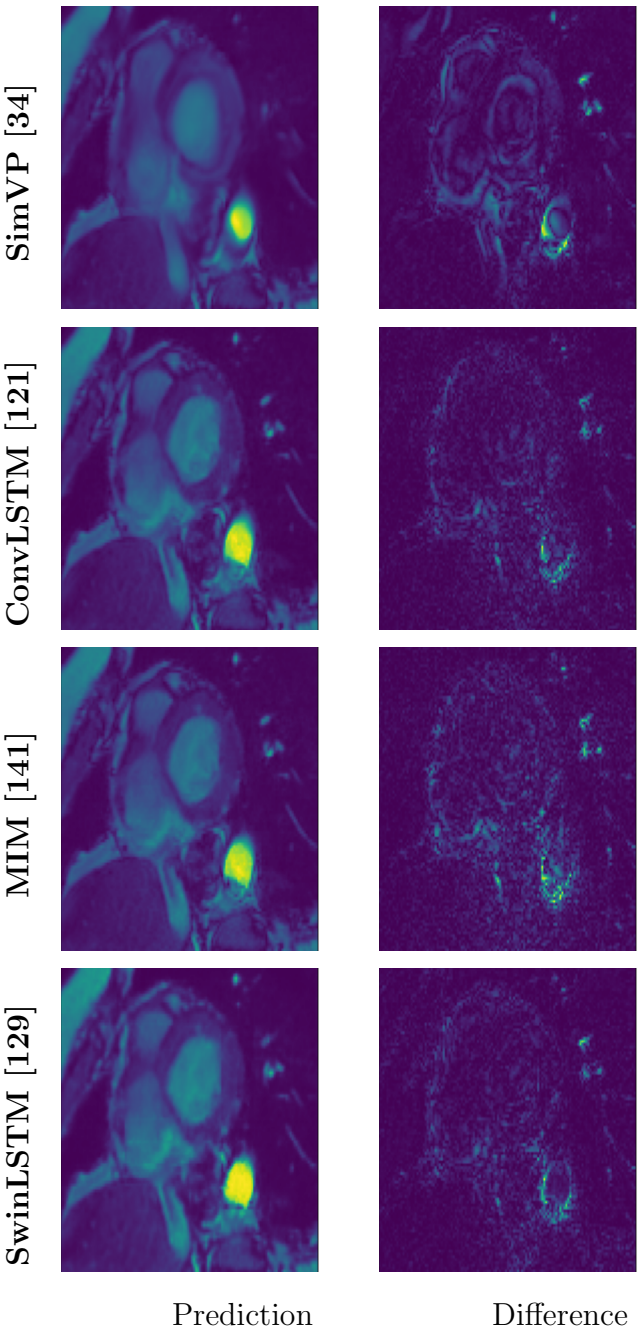


Figure 5.4: **Qualitative results on the 2D ACDC dataset.** Each row shows one model, with the predicted frame (left) and the corresponding difference map (right). The rotated labels indicate the method. For clarity, images are cropped to remove axes and padding.

5.2 Applying Augmentation Strategies to Longitudinal Series

We evaluate our longitudinal data generation as both a data augmentation method and for generating synthetic series on its own. Evaluating with a newly generated synthetic dataset is inherently challenging, as there is no ground truth and we must infer its utility from other sources. For this, we will conduct in the next section. For the augmentation experiments, quantitative evaluation is possible by comparing model performance when trained on real data alone versus when augmented with the synthetic series.

(a) ACDC: ED heart phase (real). (b) ACDC: Synthetically augmented end. (c) Difference map of 5.5a and 5.5b

Figure 5.5: **Visual example for a real example, an augmented one and the difference**. (a) Real end-diastolic (ED) cardiac frame. (b) Synthetically deformed frame produced by our longitudinal augmentations from the same ED phase. (c) Difference map between (a) and (b), showing that changes are localized near the anatomical boundary. The deformation magnitude was chosen for performance; it is not fully realistic anatomically but still provides a useful augmentation signal.

Experimental Setup We will use two datasets: BraTS [92], which contains only single time-point scans and is repurposed here to simulate longitudinal series, and ACDC [7], which provides regularly acquired 3D cardiac images and serves as the primary benchmark for augmentation. For ACDC, we train on 90 subjects and validate on 10, which is the original split used in [154]. The prediction backbone for this experiment is SimVP [34], implemented within OpenSTL [126], trained on four context and four target frames. During the mixing experiments, a fraction of the training data, defined as the mixing ratio, is *added* as a semi-synthetic frame. In other words, from the true data, we take a single frame, augment it longitudinally, and add this to the sequence. So the effective batch size and epoch time are the same, but some real samples are replaced.

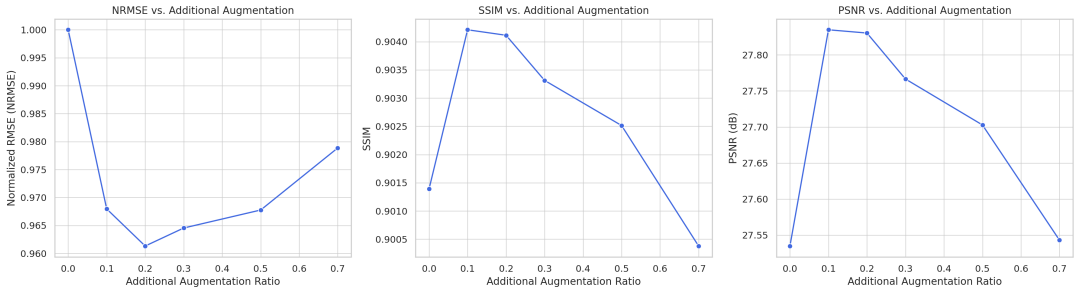


Figure 5.6: **Performance in terms of synthetic mixing ratio.** The plots show the effect of incorporation additional synthetic ausgmentations at different mixing ratios on three evaluation metrics, $NRMSE$, $SSIM$, and $PSNR$. The method is the SimVP method, trained on 2D ACDC cine MRI data. On the left, the normalized RMSE decreases as a small fraction of augmentations is added, reaching its lowest point around a ratio of 0.2–0.3, before increasing again for larger ratios. The middle panel shows SSIM, where structural similarity improves with moderate augmentation (peaking between 0.1–0.2) but gradually declines as the synthetic fraction increases further. Similarly, the right panel demonstrates that PSNR follows the same trend, with the best performance for moderate augmentation and a drop at higher ratios. This Figure was taken from [23].

Quantitative Results: Mixing Ratio Analysis Figure 5.6 shows the effect of varying the synthetic-to-real data ratio on the three reported metrics. Performance improves when a small to moderate proportion of synthetic data is included, peaking at ratios between $0.1 - 0.3$, after which performance gradually declines. For reference, the LCI attains RMSE = 12.144, SSIM = 0.8408 and PSNR = 24.00. These preliminary findings suggest that we can augment longitudinal data with this approach.

Qualitative results on ACDC. Figure 5.5 presents visual examples from ACDC, showing a real end-diastolic frame, its augmented version, and the corresponding difference map. Changes are solely localized around the segmentation boundaries. Although the augmented frame is not anatomically perfect, it still seems to help the method. These images suggest that the segmentation and the extent of blurring can be tuned to the dataset in question, so as to generate more faithful images (in Figure 5.5, the whole heart shrinks rather than the chamber).

Qualitative results on BraTS. Figure 5.7 illustrates two examples from the BraTS dataset augmented (or generated) via our method. Even with simple linear deformations, the generated follow-ups appear visibly distinct yet not overly distorted. The method in these figures is TFM, which we will present in the following section. For the first row: Context SSIM = 0.946, PSNR = 24.8; Prediction SSIM = 0.985, PSNR = 36.9. For the second example: Context SSIM = 0.949, PSNR = 23.7; Prediction SSIM = 0.980, PSNR = 33.2.

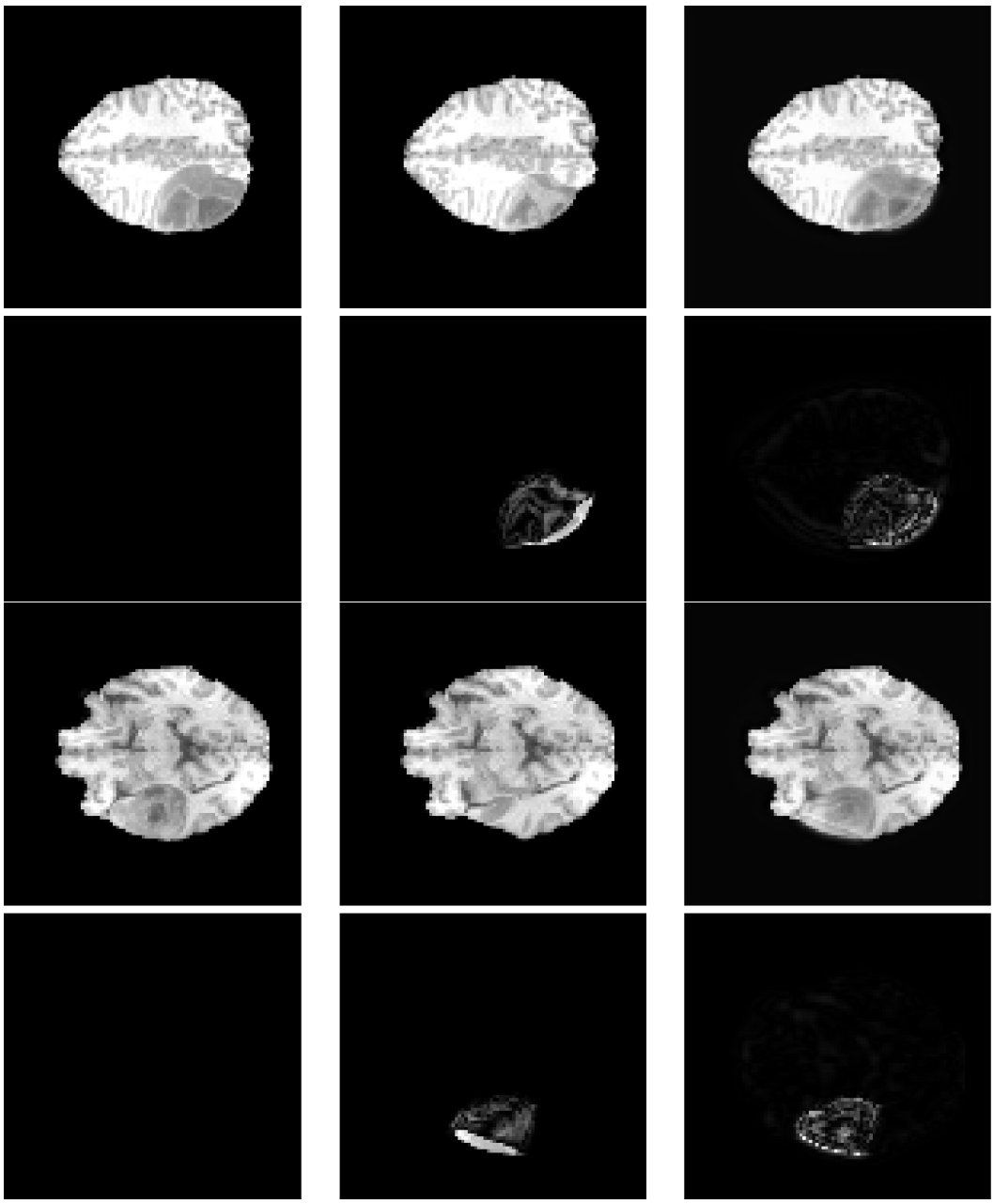


Figure 5.7: **Qualitative examples from BraTS + longitudinal augmentations.** Each column within an example shows the ground truth (GT), the input context frame, and the prediction of TFM. The first two rows is the image, and below the residuals. For the first example: Context SSIM = 0.946, PSNR = 24.8; Prediction SSIM = 0.985, PSNR = 36.9. For the second example: Context SSIM = 0.949, PSNR = 23.7; Prediction SSIM = 0.980, PSNR = 33.2. All images are normalized with identical windowing for fair visual comparison, and top text from the original images has been cropped for cleaner presentation.

5.3 Temporal Flow Matching

In this section, we evaluate the performance of Temporal Flow Matching in spatio-temporal prediction tasks. We begin by revisiting the dataset preparation and series pre-processing steps (see 5.3.1), as these design choices *strongly affect experimental results* in low-data regimes. We then present the main quantitative and qualitative results, comparing our approach to established spatio-temporal baselines. Next, we discuss the challenges encountered in reproducing one specific diffusion-based baseline. Finally, we report results on the semi-synthetic dataset method introduced in the previous Section and conclude with ablation studies demonstrating the robustness of our method.

5.3.1 Experimental settings

Data Preprocessing

ACDC [7] is a cardiac MRI dataset that captures dynamic anatomical changes of the heart across different phases of the cardiac cycle. Each subject sequence consists of multiple time points, where each time point corresponds to a 3D cine MRI scan. Due to the acquisition protocol, one spatial dimension has lower resolution. The end-diastolic (ED) and end-systolic (ES) phases mark the start and end of each sequence, respectively. All images are reshaped to a tensor of size $[T, H, D, W] = [12, 32, 128, 128]$, where T denotes the number of time points, and H, D, and W are the spatial dimensions. The prediction task is to generate the final frame given a subset of context frames. To simulate missing observations and irregular sampling, context frames are randomly masked during training.

ISLES [108] provides perfusion CT scans of stroke patients, acquired as dynamic 4D volumes with high temporal resolution. Given the high frame count and the often minimal change between adjacent frames, we temporally downsample the data by selecting every second timepoint. From this reduced sequence, we randomly sample a 8-frame segment, where the final frame is treated as the target for prediction. The context consists of the preceding three frames, which are further masked randomly to simulate sparse and irregular observations. The resulting context has dimensions $[T, H, D, W] = [8, 16, 128, 128]$, where the temporal padding accounts for the masked and missing slices. The ISLES dataset is split into 92 training, 23 validation, and 34 test subjects. **Lumiere** [124] is a longitudinal 3D MRI dataset that tracks tumor evolution in glioma patients across multiple clinical visits. Each subject contains a time series of volumetric scans, which we spatially reshape and temporally pad to a uniform tensor of size $[T, H, D, W] = [8, 96, 96, 64]$. Due to the varying number of available timepoints per patient, we prepend zero-valued volumes as padding to maintain a consistent input shape across the dataset. The final dataset includes 48 training, 12 validation, and 14 test subjects. Figure3.7 shows visual examples of

image pairs from different timepoints for LUMIERE.

Experimental Details

All methods were trained with the AdamW [65, 85] optimizer, a cosine-annealed learning rate schedule, and a batch size of 4. The learning rate was fixed at 1e−4 for all experiments. For TFM, we used 10 integration steps during inference. Our method builds on the standard FM U-Net from the TorchCFM library [131–133]. It incorporates cross-attention between time embeddings and spatial feature maps. See Figure 4.6 for an overview. To ensure a fair comparison, all experiments were repeated three times with different validation splits, while keeping the same random seed within each split.

Fixing Irregularity Sampling: During our experiments, we *encountered a crucial but underexplored* issue: handling irregular temporal subsampling during validation. [1] Our training setup randomly subsamples context frames, which improves robustness for training, but complicates validation. If validation frames are also randomly sampled, small validation sets can lead to highly variable results, since the distance between the last available image and the target frame affects performance. This instability is especially evident for the LCI heuristic, which fluctuates between near-optimal (when the last frame is included) and poor results (when the distance is maximal).
To address this, we fix the validation context frames and reuse them across all runs. This stabilizes evaluation and ensures fair method comparison, albeit at the cost of potential bias, since the validation (and hence the best performing save) depends on the chosen subsampling pattern. However, some overfitting may remain, suggesting that future work should explore optimal validation masking. Designing a consistent, yet simple and unbiased sampling strategy remains a non-trivial problem.
Finally, it is worth noting that both Flow Matching (and Diffusion) differ from the predictive methods we used in how their losses relate to their validation metrics. For the Flow-based models, training loss and validation metrics are not directly comparable: e. g.ConvLSTM minimize pixel-wise MSE during both training (which is the same metrics as during validation), whereas FM and Diffusion optimize velocity or score-matching objectives, and generate predictions only through ODE-based inference. This distinction makes it inherently harder to compare the values of training and validation performance.

[1] Only a few works explicitly address subsamples or sparse longitudinal data. For those works, there was no such discussion.

Quantitative Results

Table 5.7: **Quantitative Evaluation on Test Sequences:** Reported values are mean (standard deviation) over three runs. Metrics include normalized root MSE, $NRMSE$, structural similarity index ($SSIM$[%]) and peak signal-to-noise-ratio $PSNR$. *ViViT on Lumiere ran out of $40GB$ memory, despite having a smaller batch size and the lowest possible feature size.

Dataset	Model	NRMSE	SSIM[%]	PSNR
ACDC	LIB	0.056	93.3	28.49
	ConvLSTM	0.112 (0.005)	50.4 (1.5)	19.12 (0.31)
	SimVP	0.124 (0.001)	52.8 (1.6)	21.21 (0.13)
	ViViT	0.120 (0.008)	30.1 (6.9)	18.47 (0.53)
	TFM (ours)	**0.040 (0.012)**	**94.5 (0.8)**	**30.51 (1.56)**
ISLES	LIB	0.057	95.6	28.39
	ConvLSTM	0.182 (0.005)	40.8 (0.9)	17.85 (0.23)
	SimVP	0.124 (0.001)	52.8 (1.6)	21.21 (0.13)
	ViViT	0.162 (0.003)	32.5 (0.8)	18.84 (0.21)
	TFM (ours)	**0.041 (0.007)**	**97.6 (0.8)**	**31.03 (1.08)**
Lumiere*	LIB	0.085	89.3	21.55
	ConvLSTM	0.352 (0.009)	7.9 (4.2)	9.12 (0.22)
	SimVP	0.711 (0.028)	-2.5 (0.8)	2.98 (0.34)
	TFM (ours)	**0.069 (0.007)**	**89.7 (1.2)**	**23.73 (0.82)**

Comparison to Diffusion Model

We compare our approach against the official Sequence-Aware Diffusion Model (SADM) implementation [154]. While SADM is theoretically well-suited as a baseline for our Flow Matching approach, we encountered major practical issues reproducing meaningful results. Using the official implementation and the provided data, along with their training and pre-processing pipeline, our runs produced outputs visually indistinguishable from random noise.

To address this, we adopted a two-step strategy. First, we report the official SADM implementation as published, acknowledging that deviations may stem from differences in random seeding (the preprocessing and training pipelines were identical). Second, we implemented an adopted version of SADM, based on the modifications proposed by [106]. We also note that similar reproducibility issues have been reported on the official SADM repository, highlighting the broader need for robust and verifiable baselines.

Adapted SADM implementation: The original ViViT backbone for their spatio-

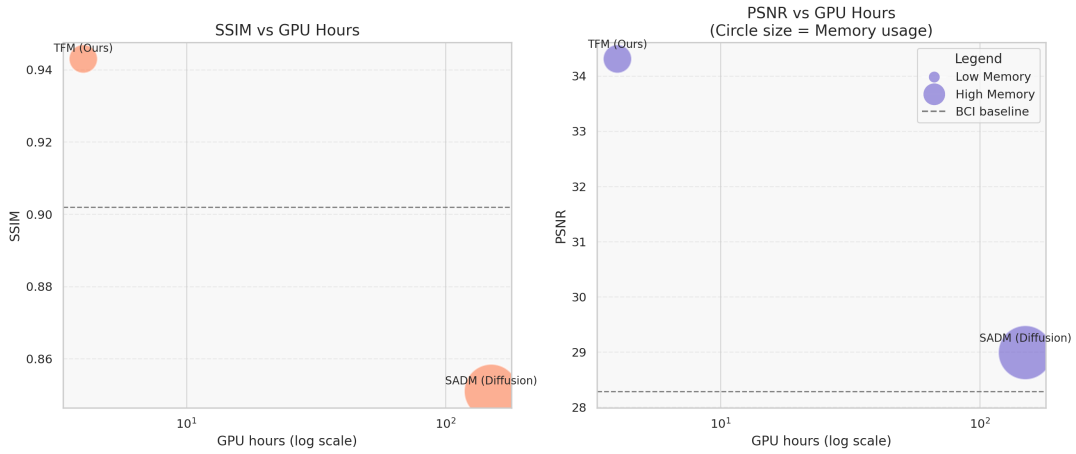


Figure 5.8: Comparison of model efficiency and performance. Left: SSIM versus GPU hours (log scale). Right: PSNR versus GPU hours, with circle size representing memory usage. Our method (TFM) achieves higher SSIM and PSNR than the diffusion-based SADM model, while requiring significantly fewer GPU hours and less memory. The dashed line indicates the baseline performance of the LCI. These results highlight the efficiency of TFM in achieving strong image quality with substantially lower computational cost.

temporal branch used patch sizes of $8 \times 32 \times 32$, which are excessively large for $4D \rightarrow 3D$, causing blurry predictions. To improve modeling performance, we introduced skip connections and let the ViViT *model* learn the difference between the input and target frames. Furthermore, our diffusion setup was built on latent Diffusion, as opposed to the original SADM, which used pixel-level Diffusion. The autoencoder is pre-trained and then frozen, while the diffusion model is trained independently in a three-stage process. Our adapted version of SADM achieves SSIM $= 0.873$, PSNR $= 23.96$ dB, and NRMSE $= 0.066$, demonstrating a substantial improvement over the unmodified SADM baseline.

Head-to-head comparison For a fairer comparison between TFM and the official SADM implementation, we used their publicly available pre-processing pipeline and trained on the same data split.[2] Performance is reported using the two primary metrics from their paper—PSNR and SSIM—alongside maximum memory usage. NRMSE is omitted due to the unspecified normalization factor in the original work. Both methods were evaluated on the same set of 50 test cases, enabling a fairer comparison. We focus on the single-frame prediction setting, in which the first frame of the sequence is used to predict the final frame. Finally, in Figure 5.8 we see the

[2]For this comparison, we used a 90-10 train–test split, instead of the 80-20 split employed in previous sections. This difference in partitioning explains the discrepancy in results between Figure 5.14 and Figure 5.8.

performance with respect to computational needs, with massive differences in SSIM and PSNR, orders of magnitude lower training time, and a substantially smaller memory footprint.

5.3.2 Qualitative results

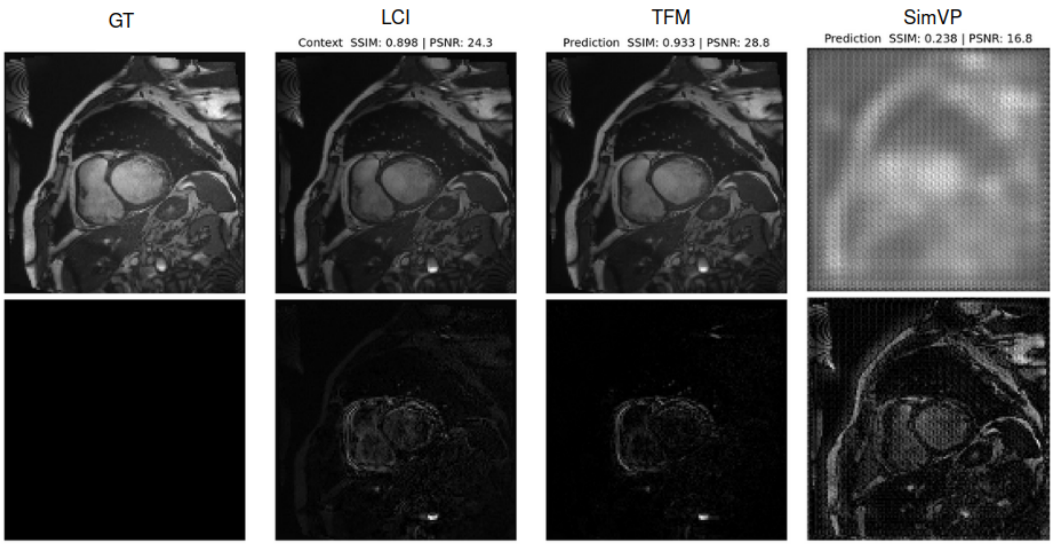


Figure 5.9: **Comparison of longitudinal prediction quality on ACDC**. Columns show: ground truth (GT), LCI, TFM prediction, and SimVP prediction. Top row displays the reconstructed or predicted frames with reported SSIM and PSNR values; bottom row shows corresponding difference residual maps (empty for GT). TFM produces a high-fidelity prediction with substantially higher SSIM and PSNR, whereas SimVP fails to recover coherent structure. LCI provides a reasonable frame and its residual highlights smaller deviations. This layout illustrates both absolute appearance and localized errors across methods.

We present qualitative results comparing TFM with image sequence prediction baselines. A clear difference is observed between SimVP and our Flow Matching approach. This gap may arise from memory limitations during training when applying SimVP to $3D + T$ data, which necessitated reducing feature dimensionality compared to the $2D$ experiments. Despite its overall small computational requirements, TFM produces high-fidelity predictions across the entire volume. We suppose this stems from the fact that the majority of the volume remains unchanged; hence, TFM can learn the differences more effectively. In contrast, SimVP captures the overall anatomy, but suffers from noticeable blurring and loss of fine detail. Overall, TFM

achieves sharper, more anatomically consistent predictions that closely approximate the ground truth volumes.

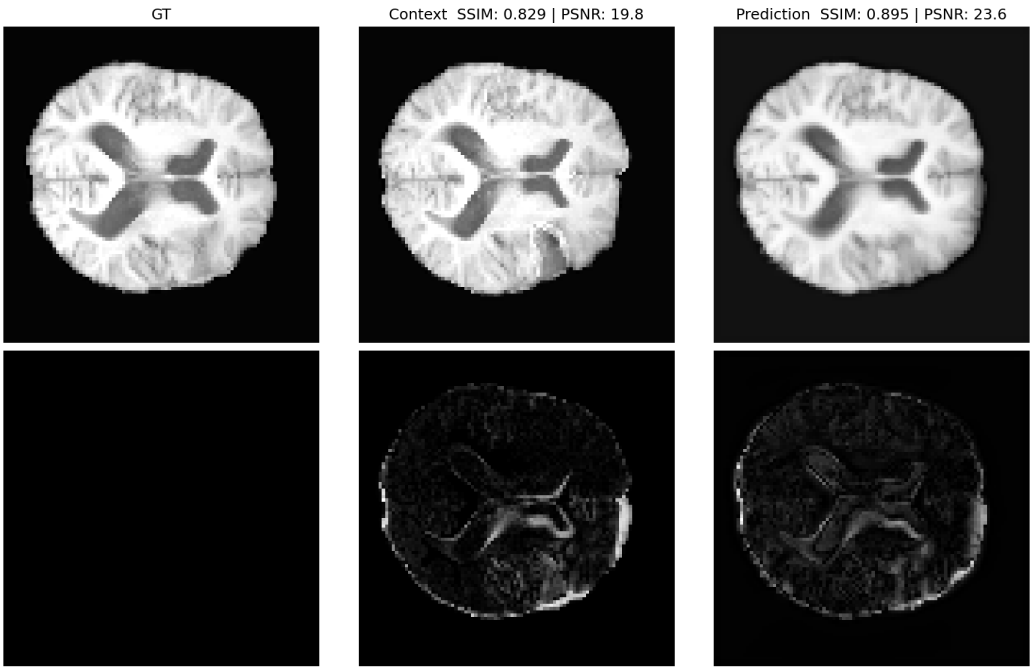


Figure 5.10: **Longitudinal prediction quality on LUMIERE**. Selected example from the same evaluation as Figure 5.9; shown because it is especially illustrative, though not cherry-picked to mislead. Columns: ground truth (GT), context frame (with reported SSIM and PSNR), and model prediction. Top row shows the image appearance, bottom row shows the high-frequency residual / difference map (empty for GT).

We show qualitative examples of TFM on the Lumiere dataset in Figure 5.10, on ACDC in Figure 5.9, and a zoomed-in example for ISLES in Figure 5.11. These results are particularly noteworthy, as the Lumiere dataset poses a significant challenge due to its pronounced spatial and temporal variability. While not all cases achieve this level of precision, the presented examples demonstrate **a clear and promising trend**, indicating that TFM can reconstruct medical longitudinal data well.

Synthetic Experiments

We conducted synthetic experiments to evaluate both the performance of various methods and the realism of the generated synthetic data. This task presents a chicken-and-egg problem: assessing method quality depends on the data, yet data

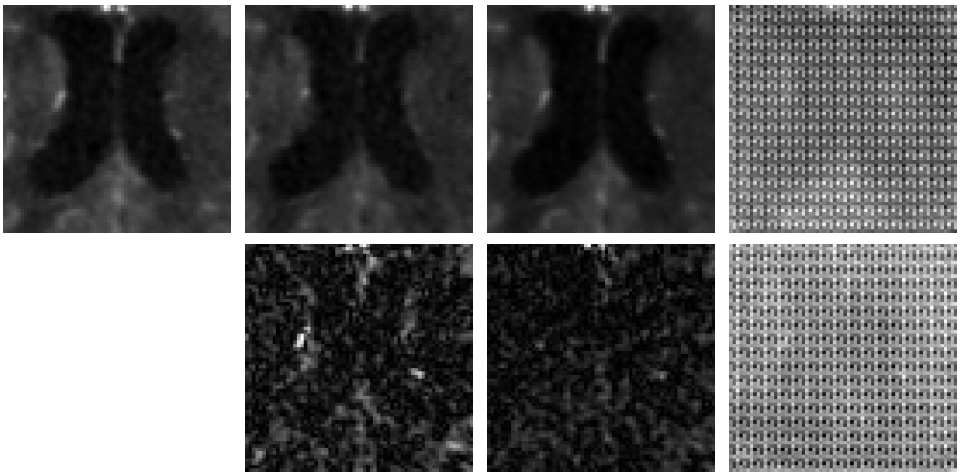


Figure 5.11: **Zoomed-in prediction Examples from the ISLEs Dataset**: **Top row**: Visual comparison (from left to right) of the ground truth, LCI, TFM, and SimVP, alongside the Ground Truth, *focusing on a high-resolution crop for clarity.* **Bottom row**: residuals of absolute value of top row. The second row shows the absolute values from the residuals. TFM preserves the spatial quality, while SimVP struggles to recover the underlying details. This shows that even in fine details. Furthermore, in the residuals, we can see that they are lower and less noisy. For this specific patch, the $NRMSE$ are in order: $0.0028, 0.0016, 0.0751$.

(a) GT (b) LCI (c) TFM (d) SimVP (e) ConvLSTM

Figure 5.12: **Qualitative comparison on the Semi-synthetic BraTS dataset**. We show the ground truth (GT), the last context image (LCI), and predictions from TFM, SimVP, and ConvLSTM. The upper row are the corresponding images, the lower row the residuals.

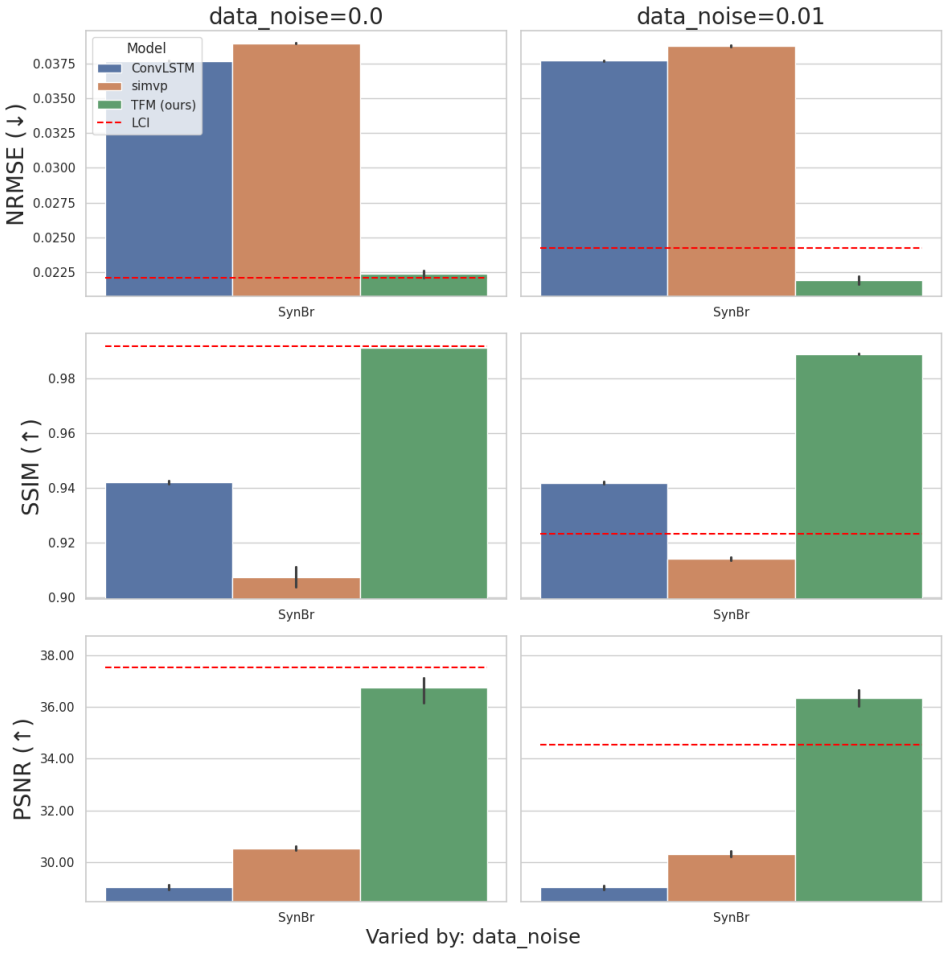


Figure 5.13: **Different context noise schedule for Semi-synthetic BraTS** Performance comparison of ConvLSTM, SimVP, and TFM on the SynBr dataset under different Gaussian *context* noise levels (0.0 and 0.01) for NRMSE (↓), SSIM (↑), and PSNR (↑). Except for noises, the experimental setups are identical. The dashed red line indicates the LCI baseline, notably different with different noise levels. Error bars represent standard deviation over test samples.

realism can be easily judged only through method behavior. However, since all methods were in the previous subsection tested on real medical datasets, their relative performance provides a reference for interpreting the synthetic results. Figure 5.13 summarizes these findings. Note that we added noise to the context frames but not to the target image.
Overall, SimVP performs worst, while ConvLSTM achieves slightly better results, consistent with the patterns observed on the real LUMIERE dataset 5.7. Qualitative examples in Figure 5.12 show that TFM yields minimal residuals and realistic predictions, particularly in regions of change compared to the last LCI. Neither SimVP nor ConvLSTM produce sharp results, nor beat the LCI heuristic. TFM consistently performs best, maintaining stable results and beating the LCI under small noise. These results suggest that we can at least add this semi-synthetic dataset as an additional benchmark, given the data scarcity.

5.3.3 Ablations for Temporal Flow Matching

In this section, we present a series of ablation studies to analyze the behavior and robustness of TFM. We begin with *crucial ablations* that evaluate the core design choices, such as the attention U-Net, sparsity filling, and sequence learning. Then, we investigate *secondary factors*, including feature size, training noise, etc, to quantify their practical influence on performance and efficiency. Together, these experiments provide insight into which design choices are crucial and how our Flow Matching approach is robust to variations in other design choices.

Crucial Design Choices

The ablations in Table 5.8 highlight two particularly critical components: sparsity filling and full-sequence learning. Removing sparsity filling leads to a sharp drop in performance across all metrics, confirming its necessity. Similarly, restricting the model to use only the last available frame causes instability, underscoring the importance of temporal context. While the attention UNet improves results slightly, its effect is secondary compared to these core design choices.

Other Design Choices We conducted an ablation study to test how training noise (TN) affects prediction performance. As shown in Table 5.9, a small amount of training noise (TN $\in [0.01, 0.05]$) slightly improved NRMSE, SSIM, and PSNR compared to the Last Context Image (LCI) baseline, with the best overall performance observed at TN $= 0.01$. Moderate noise helps regularize the model, but too much noise (TN ≥ 0.3) lowers reconstruction quality. Interestingly, the improvements were robust across a range of small TN values, indicating that the method is not overly sensitive to this hyperparameter.

Change	NRMSE	SSIM [%]	PSNR
Att UNet & Mean	**0.0261**	**96.04**	**32.30**
No Att: Mean	0.0270	95.77	31.88
No Att: Last	0.0271	95.77	31.87
No Sparsity Filling	0.0444	90.92	27.30
LIB + FM*	0.1029	66.83	19.97
LCI	0.0380	93.50	29.49

Table 5.8: **Crucial Ablation Results for TFM on ACDC:**This table compares TFM under different design changes, showing the performance under each scenario. The ablations were done on an ACDC validation set, and 10 NEF. We evaluate the effect of using a more lightweight version of the UNet which does not use attention('No Att'). Instead, τ and image embeddings are merged via concatenation in the bottleneck. We also compare aggregating via the mean and the last image, but these results are only for inference. Training is still done the same way. Third, we compare sparsity filling with the alternative of using the image sequences $\mathcal{I}$ as they are given. This notable reduces performance. *Limiting the model to only see LCI during training and perform FM on this is unstable, which highlights the importance of temporal context.

Facet	NRMSE (↓)	SSIM (↑)	PSNR (↑)
LCI	0.039	93.53	29.42
TN=0.0	0.026 (-0.013)	96.06 (+2.49)	32.24 (+2.90)
TN=0.01	**0.026 (-0.014)**	**96.12 (+2.56)**	32.32 (+2.98)
TN=0.025	0.026 (-0.013)	96.06 (+2.49)	**32.34 (+3.00)**
TN=0.05	0.026 (-0.013)	96.04 (+2.47)	32.26 (+2.92)
TN=0.1	0.026 (-0.013)	96.06 (+2.49)	32.24 (+2.90)
TN=0.3	0.026 (-0.013)	95.78 (+2.21)	32.26 (+2.92)

Table 5.9: **Different Masked** validation performance Ablation on training noise (TN) levels for the proposed method, compared to the last context image (LCI) baseline. Values in parentheses indicate the difference relative to LCI. Moderate TN levels (0.0-0.05) slightly improve NRMSE, SSIM, and PSNR, with TN = 0.01 yielding the best overall performance.

Facet	NRMSE (↓)	SSIM (↑)	PSNR (↑)	Max Mem. [GB]
LCI	0.038	93.50	29.49	-
FS=8.0	0.027 (-0.012)	95.94 (+2.38)	32.07 (+2.74)	5.40
FS=16.0	0.026 (-0.013)	95.99 (+2.43)	32.17 (+2.83)	6.54
FS=32.0	0.026 (-0.013)	**96.06 (+2.49)**	32.24 (+2.90)	11.68
FS=64.0	**0.026 (-0.014)**	95.97 (+2.41)	**32.35 (+3.01)**	22.42

Table 5.10: Ablation on feature size (FS) for the proposed method, compared to the last context image (LCI) baseline. Values in parentheses indicate the difference relative to LCI. Larger FS values generally improve NRMSE, SSIM, and PSNR, with FS = 32.0 achieving the best overall performance.

Table 5.10 reports the maximum GPU memory consumption observed during training on the ACDC dataset for different feature sizes (FS), as well as their performance. We note that the increase from 8 to 16 is marginal, possibly because the input size T is larger. FSs larger than 16 almost linearly increase memory consumption. All other training settings, including batch size, optimizer configuration, and network architecture, were kept constant to isolate the effect of FS on memory usage. The results show a clear upward trend, with larger FS values substantially increasing the memory requirements due to the greater number of intermediate feature maps and parameters. This observation highlights the practical hardware constraints when scaling model capacity.

Number of Function Evaluations (NFEs) In 5.11, we evaluate how the number of function evaluations (NFEs) affects the SSIM performance on one ACDC validation set. We see that The first NFE already achieves a good SSIM, but the value plateaus after 10 steps.

Number of Context Frames To assess the influence of temporal masking, we conduct experiments where varying numbers of context frames are masked. We use two masking orders: forward ($1 \rightarrow T$) and reverse ($T \rightarrow 1$). The forward strategy starts masking from the earliest frame, t_1. The reverse strategy begins from the final frame, t_T. This is a zero-shot performance evaluation. The model was trained on irregular frames and is now tested when a specific number of context frames are masked. As shown in Figure 5.14, the forward masking order maintains high SSIM values (≥ 0.98) at all masking levels. This indicates stable reconstructions even when early frames are missing. The highest performance, though marginal, is observed when about half of the frames are masked. *We hypothesize this is due to the masking ratio occurring most during training.* The model may have adapted

Facet	NRMSE (↓)	SSIM (↑)	PSNR (↑)
LCI	0.038	93.50	29.49
NE=1.0	0.030 (-0.009)	95.43 (+1.87)	31.47 (+2.13)
NE=5.0	**0.025 (-0.014)**	**96.18 (+2.62)**	**32.47 (+3.13)**
NE=10.0	0.026 (-0.013)	96.06 (+2.49)	32.24 (+2.90)
NE=100.0	0.026 (-0.013)	96.04 (+2.48)	32.30 (+2.97)
NE=200.0	0.025 (-0.014)	96.18 (+2.62)	32.47 (+3.13)

Table 5.11: **Evaluating Metrics vs. Number of Function Evaluations:** We evaluate how the number of function evaluations (NFEs) affects $SSIM$ performance on one ACDC validation set. $SSIM$ increases with more evaluations and peaks at 25 NFEs, after which it plateaus. However, the improvement becomes marginal after beyond just 5 NFEs.

implicitly to this regime. *This suggests performance may improve if the model is fine-tuned to specific masking patterns used at test time, e. g. for the dense series.* In contrast, the reverse masking strategy shows a steady decline in SSIM as more frames are masked. SSIM values drop below 0.91. This is expected. As context frames move further from the prediction time point, it becomes more difficult for the model to infer accurate temporal dynamics.

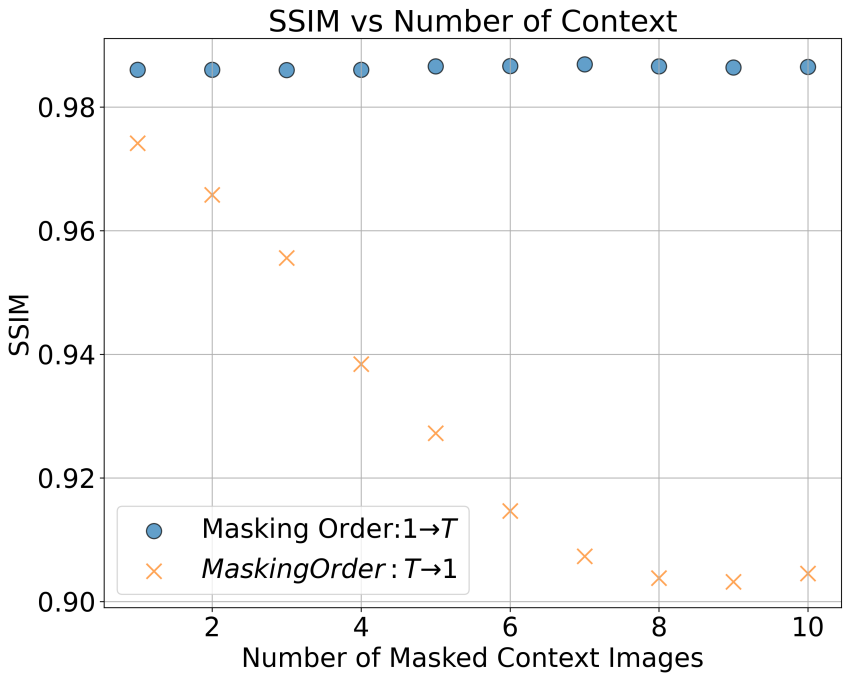


Figure 5.14: **Effect of masking on method performance.** Effect of masking on method performance, measured by SSIM. We evaluate how the model's reconstruction quality varies with the number of masked context images. Two masking orders are compared: forward $1 \rightarrow T$ and reverse $T \rightarrow 1$. The forward masking order maintains consistently high SSIM across all levels of masking, indicating robust prediction. In contrast, the reverse masking order shows a steady decline in SSIM as more context frames are masked, suggesting reduced temporal coherence when earlier frames are missing, as expected.

5.4 Extension to Temporal Flow Matching

In this section, we explore the extension of Temporal Flow Matching (TFM) from discrete time grids to continuous temporal modeling, by using the simulation-free Schrödinger Bridge formulation. The goal is to demonstrate that the continuous formulation (1) matches or surpasses discrete baselines on standard benchmarks, and (2) more effectively handles cases with irregular or continuous timesteps than the discrete models. We compare performance on medical datasets with the discrete TFM baseline. The continuous model achieves similar or better results and is more computationally efficient. Then, we present evidence that, when evaluated on real-world continuous data, the continuous TFM outperforms the discrete version. We outline preliminary results toward a Schrödinger Bridge formulation. These results illustrate potential directions and the restrictions of our current setup.

5.4.1 Continuous Time Extension

In this subsection, we present additional experiments that extend beyond the main benchmarking results (Table 5.7). We aim to show how discrete temporal embedding in TFM generalizes to continuous time. We also analyze its impact on performance. As introduced in Section 4.3, the discrete model treats the flow step parameter τ abstractly and not as explicit time. Here, we explore a continuous variant where real-valued timesteps replace discrete steps.
We begin by directly comparing the continuous method to the discrete method in discrete settings (that is, regularly sampled time grids with missing entries). We show that the continuous approach achieves comparable performance and reduced computation. Next, we simulate continuous experiments by resampling timesteps from regular sequences. In these cases, where the temporal signal is pronounced, the continuous Flow Matching variant outperforms the discrete counterpart, which struggles to represent highly irregular temporal dependencies. This direct comparison highlights the advantage of handling irregular timing. For all experiments, we retain the standard TFM configuration as a reference baseline, fixing the hyperparameters.

Discrete Experiments To ensure consistency, we first test the continuous formulation on datasets sampled at relatively regular time intervals. Under these conditions, the continuous model performs on par or slightly better than the discrete variant, confirming that both can handle uniform (but masked) time grids (Table 5.12) This validates that the continuous formulation does not suffer when applied to discrete grids. Computational performance can be seen in Table 7.1.

Dataset	Model	NRMSE $[10^{-2}] \downarrow$	SSIM [%] $\uparrow$	PSNR $[dB] \uparrow$
ACDC	LCI	4.48	92.79	28.918
	ConvLSTM	11.20 ± 0.48	50.44 ± 1.53	19.123 ± 0.312
	SimVP	9.27 ± 0.29	49.08 ± 4.01	20.715 ± 0.267
	NODE + LSTM	11.59 ± 0.18	36.41 ± 2.94	18.946 ± 0.186
	ViViT	13.90 ± 2.66	17.06 ± 8.60	17.252 ± 1.738
	Flow Matching discrete	3.97 ± 1.23	**94.51 ± 0.79**	**30.510 ± 1.560**
	Flow Matching continuous	**3.74 ± 0.21**	94.34 ± 0.45	29.750 ± 0.528
ISLES	LCI	5.25	96.29	29.002
	ConvLSTM	19.31 ± 0.18	39.92 ± 0.66	17.644 ± 0.014
	SimVP	13.06 ± 0.19	48.82 ± 1.60	20.799 ± 0.112
	ViViT	16.54 ± 0.30	36.76 ± 1.49	18.671 ± 0.134
	NODE + LSTM	15.10 ± 0.87	40.55 ± 7.15	19.481 ± 0.515
	Flow Matching discrete	4.50 ± 0.76	**97.33 ± 0.93**	30.542 ± 1.540
	Flow Matching continuous	**4.38 ± 0.48**	97.31 ± 0.38	**30.809 ± 1.099**
Lumiere	LCI	8.38	88.35	21.631
	ConvLSTM	34.79 ± 0.67	9.21 ± 2.81	9.217 ± 0.171
	SimVP	71.03 ± 0.89	-1.92 ± 0.51	2.989 ± 0.109
	ViViT*	OOM	OOM	OOM
	NODE+LSTM	13.07 ± 1.03	48.66 ± 2.26	17.742 ± 0.659
	Flow Matching discrete	7.92 ± 0.92	**91.43 ± 1.84**	22.427 ± 0.969
	Flow Matching continuous	**7.55 ± 0.86**	89.32 ± 1.83	**22.551 ± 0.979**

Table 5.12: **Discrete Time: Quantitative Evaluation on Many-to-One Sequences:** Reported values are mean (standard deviation) over three runs. Metrics include normalized root MSE, $NRMSE$, structural similarity index ($SSIM$[%]) and peak signal-to-noise-ratio $PSNR$. *ViViT OOM on a 40 GB GPU, despite having a smaller batch size and the lowest possible feature size. Standard deviation of LIB omitted for visual clarity. Blue row: only method to beat LIB and our proposed Flow Matching.

Continuous Time When time intervals become truly irregular, the advantages of continuous conditioning becomes evident. In cases with sparse or unevenly sampled observations, such as single or two context settings with irregular sampling, the discrete variant struggles, as it relies on relative frame indices rather than actual timesteps. By contrast, the continuous variant encodes real-valued timesteps, enabling improved predictive fidelity (see Tables 5.13 and 5.14).

Method	SSIM ↑	PSNR ↑	NRMSE ↓
LCI	93.27	29.77	0.0349
NODE + LSTM	57.50	22.87	0.0728
Flow Matching discrete	93.27	29.77	0.0348
Flow Matching Continuous	**93.86**	**30.09**	**0.0330**

Table 5.13: **Single Image Continuous ACDC**, where discrete TFM lacks explicit timestamp conditioning, and therefore fails to outperform LCI. Here we have a large time distance to the target.

Method	SSIM (%) ↑	PSNR ↑	NRMSE ↓
LCI (baseline)	94.99	31.727	0.03038
TFM discrete	96.67	33.534	0.02211
TFM continuous	**97.81**	**36.007**	**0.01670**

Table 5.14: **Two Image Continuous ACDC** Comparison of the LCI heuristic with TFM under discrete (implicit time) and continuous (explicit timestamps) settings. Here, we have two context images and a small target horizon.

Model	**NRMSE**	**SSIM (%)**	**PSNR (dB)**	**Change**
LCI	0.038	93.50	30.25	-
Continuous	0.038	94.24	30.80	-
Continuous	**0.037**	**94.31**	**30.98**	+ Time Augmentation
Continuous	**0.037**	94.28	30.95	+ Difference Loss
Continuous	**0.037**	94.23	30.79	+ SimSiam

Table 5.15: **Ablation on Continuous** auxiliary components on ACDC. Heuristic denotes the LCI. The base is the basic continuous TFM. Each additional row adds a single modification: SimSiam contrastive with 0.2, the additional Difference Loss 0.5 and a Time Augmentation of 0.2.

Ablations for Auxiliary Tasks We performed exploratory ablations to demonstrate that auxiliary objectives can improve the continuous variant. Table 5.15 summarizes these ablations, discussed in Section 4.4.1. Each modification was tested in isolation. We evaluated SimSiam, which may improve temporal representations; Difference Scaled Loss, which attenuates important location differences; and Time Augmentations, which jitter time. Considering all those changes, it appears that—except

for SimSiam—the modifications provided slight improvements over the vanilla version. However, we ran only a single experimental test. These outcomes hint at potential gains, suggesting room for further improvement of the continuous method. These experiments confirm the feasibility and indicate some benefit from the modifications. Previous main results use the vanilla continuous version, which serves as a baseline for comparison. The aim was to highlight opportunities for further measurable improvements.

5.4.2 Schrödinger Bridge TFM

In this subsection, we present preliminary experiments extending Temporal Flow Matching to a Schrödinger Bridge formulation. While conceptually appealing, the voxel-based setting poses challenges that limit direct applicability in its current form. We experimented with noisy and masked variants of the model, starting from a relatively low training noise level (0.01). Although we report conventional image quality metrics, we note that they may not be ideal for measuring coverage and statistical variability, since the ground-truth image represents only one of many valid future realizations. A principled evaluation would instead require distribution-aware metrics and latent sampling. One alternative may be to perform the bridge dynamics on a latent formulation.
Quantitatively (Table 5.16), moderate noise scaling and flow-based masking (i. e.only noising where the flow is sufficiently large) improved reconstruction stability, compared to the vanilla Schrödinger method. Qualitative masked samples (Figure 5.16) further indicate that the scaled noise suppresses noisy artifacts. However, even under stochastic sampling (Figure 5.15), image-level variability remains limited, suggesting that, in its current form, the voxel-level bridge dynamics do not yet induce meaningful probabilistic diversity. Overall, these results demonstrate that while Schrödinger Bridges can, in principle, be combined with TFM, their effective deployment in image space remains non-trivial.

Table 5.16: Performance comparison of Schrödinger bridge TFM variants with different noise and masking strategies using image quality metrics (PSNR, SSIM, and NRMSE). Shown are large noise schedules with a training noise of 0.1, to show the effects of noise masking and scaling.

Model	**PSNR (dB)**	**SSIM** (%)	**NRMSE** 10^{-2}
Vanilla Schrödinger TFM	25.68	67.89	5.291
+ Flow Masked TFM	25.69	67.92	5.287
+ Scaled Noise $\times 0.25$	**29.89**	**92.07**	**3.485**

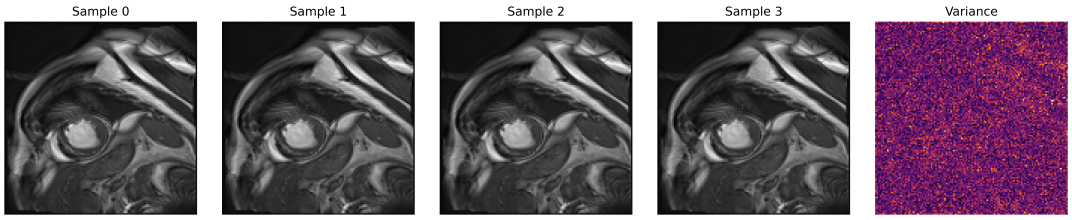
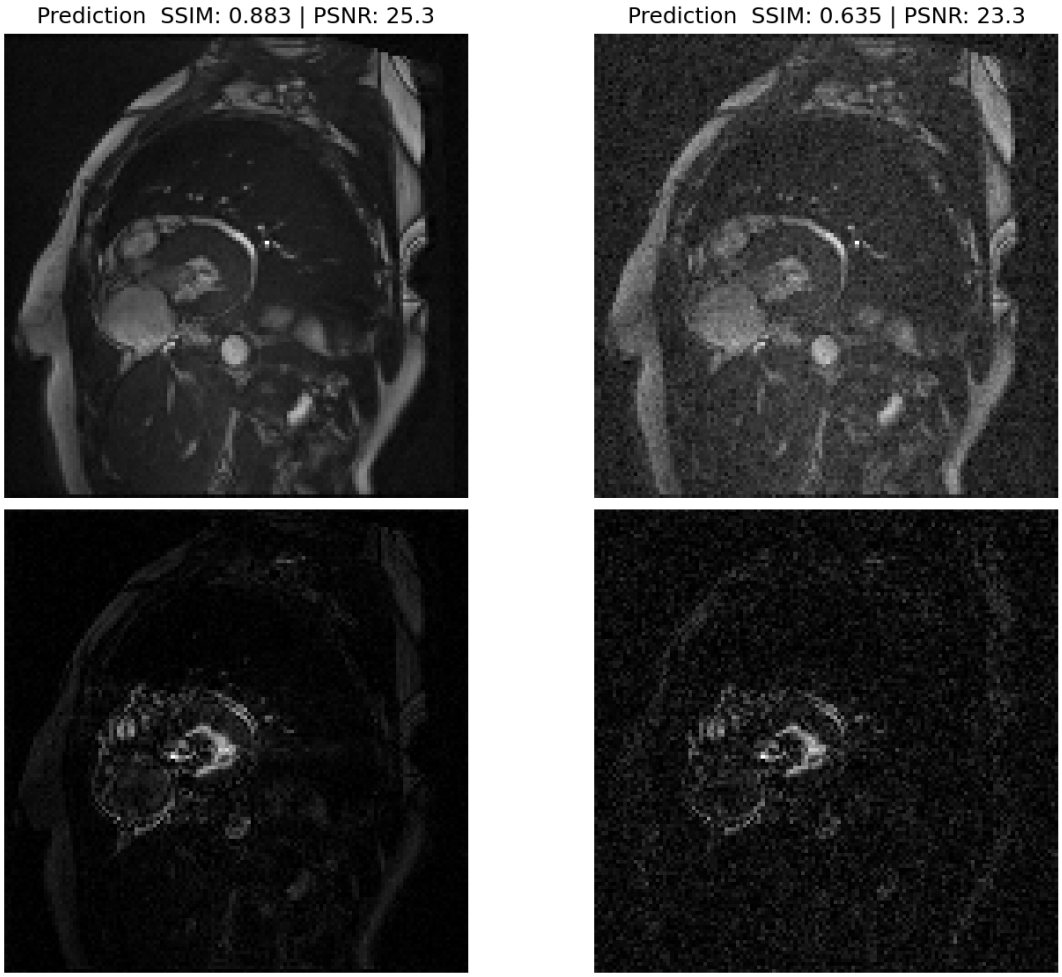


Figure 5.15: **Stochastic samples and variance from the Schrödinger bridge TFM model on ACDC data.** Here we see four samples and the variance across them. The variance is rather uniformly distributed, rather than where temporal change is expected.



(a) Noise scaled by 0.25 with TFM and masked

(b) Vanilla Schrödinger bridge TFM without masking

Figure 5.16: **Qualitative comparison of Schrödinger bridge TFM predictions.** Each column shows results from a single timepoint prediction task. The top row depicts the model prediction along with the corresponding SSIM and PSNR metrics, while the bottom row shows the absolute voxel-wise error maps compared to the ground truth. We omit ground-truth images for brevity, as they are not the focus here. These examples illustrate the impact of noise and masking on the model's generative fidelity. The important takeaway is that the masking helps focus on only the important regions of change.

Discussion 6

In this chapter, we discuss the implications of the results presented in the preceding sections. We begin with the ADNI experiments and the adaptations of Attentive Segmentation Process (ASP), highlighting potential limitations in the NP backbone. Next, we discuss semisynthetic longitudinal augmentations, comparing them with related approaches and assessing their impact on spatio-temporal medical imaging. Finally, we discuss our proposed method TFM in both its discrete and continuous variants. We reflect on how these findings impact the understanding and advancement of spatio-temporal modeling in medical imaging.

6.1 Discussion Neural Processes

In this section, we discuss the results of the Neural Processes (NP) experiments (Section 5.1) across both the synthetic segmentation task and the Alzheimer's disease prediction task. We began with the 2D Alzheimer's disease forecasting experiment (Table 5.1), where neural ODEs were explored as an extension of the ASP. In addition, we explored architectural improvements, such as Mamba blocks, as attention proved to be a major bottleneck when scaling this architecture to 3D. For clarity, results are presented in chronological order in which they were obtained, rather than from simpler to more complex tasks. Although it might appear more natural to start with the synthetic experiments, our initial assumption was that the model would generalize well to ADNI; only later did we uncover the limitations.
Across all experiments, two findings emerged: first, explicit temporal information was not always beneficial to the models; second, NP-based methods often failed to outperform the simple Last Context Image (LCI) heuristic. A similar trend was observed in the synthetic experiments, where the LCI again achieved comparable metric values. To contextualize results and assess whether it was a data-specific problem, we compared against natural imaging baselines, particularly the SimVP, which consistently outperformed NP-based methods. Finally, to reestablish a connection to medical data, we evaluated the best-performing natural imaging baselines on the ACDC dataset in a 2D setting and showed that all tested baselines outperform the LCI.

6.1.1 Alzheimer' Disease Prediction

Neural ODEs Table 5.1 summarizes the results of extending the ASP with Neural ODE (NODE). While incorporating the NODE yielded a slight improvement over both the baseline ASP and its summed representation variant, the overall performance remained weak. We attribute this to the complexity of the setup, in which slices were randomly sampled from 3D volumes along arbitrary axes, which likely exceeded the method's representational capacity. Both the baseline ASP and the ODE-augmented variant have a similar number of parameters, suggesting that the limitation stems from the architecture rather than the temporal modeling. To mitigate this, we simplified the task by restricting the axis to a single one, while also exploring alternatives to the attention mechanism to improve the method's dimensional scaling.

Attention Alternatives Replacing the attention module in ASP with Mamba blocks showed initial promise, suggesting that the architecture could benefit from more efficient skip connections. However, ablation studies revealed a critical issue: removing the explicit time input (see Table 5.2) did not degrade performance. This finding undermines the intended contribution of continuous-time modeling and raises concerns about whether the model truly leverages temporal information. One possible interpretation is that image biomarkers already encode sufficient cues for disease progression, allowing the network to infer temporal information. Nevertheless, this observation was concerning, especially since other baselines demonstrated clear benefits from explicit time conditioning for modeling AD (e. g. [106]). To probe further, we introduced a simple heuristic that used a random image as a gauge. This experiment emphasized a broader issue, discussed earlier in Section 2.3.3: *whether our evaluation metrics truly capture temporal learning.* Since none of the tested methods, which were all based on NPs, surpassed this LCI, the limitation likely reflected that these methods appeared insensitive to explicit temporal information. To disentangle the factors underlying the difficulty of interpreting image reconstruction metrics, we next conducted simplified synthetic experiments designed to isolate temporal learning.

6.1.2 Synthetic Segmentation

We adopted the synthetic experiments to evaluate how effectively the methods learn spatio-temporal dynamics. A key finding was that the LCI heuristic *outperformed*[1] all other methods on the synthetic segmentation task under specific settings, raising concerns about the robustness of the underlying backbone. While ASP performed reasonably well in several experiments, it struggled with certain synthetic settings.

[1]It is paradoxical to say that LCI *outperforms* a method while it essentially does nothing

These limitations motivated us to consider alternative baselines beyond NP-based methods. Specifically, we turned to natural imaging models such as SimVP from Gao et al. [34], which have demonstrated strong performance in video prediction, to test whether the weaknesses observed in NP-based models are of an architectural nature.

Natural Imaging Baselines Using the same experimental setup, we evaluated SimVP, which massively outperformed NP-based methods on the synthetic dataset (Table 5.5). This likely confirms the suspicion that the NP backbones were suboptimal for this task. Following this, we adopted natural imaging methods as backbones for subsequent medical experiments, with a particular focus on spatio-temporal prediction. The key takeaway is the importance of testing multiple backbone variations and assessing which architectures generalize effectively across settings. Notably, SimVP contains roughly the same number of parameters as the NP-based methods, yet remains computationally efficient as it avoids attention over long image token sequences.
Table 5.6 extends this comparison to several baselines on the 2D ACDC dataset. Two main observations emerge: first, all methods outperform the LCI; second, there is sufficient performance spread, so the data is not too easy to learn. Interestingly, ConvLSTM performed best, whereas MIM and SwinLSTM, despite their more complex and more recent design, perform slightly worse. All methods were trained on the same data with the OpenSTL framework [126]. Interestingly, although SimVP consistently outperforms other methods on benchmarks such as Moving MNIST and various synthetic or natural imaging datasets, it performs the worst on ACDC. *This discrepancy highlights a clear gap between the effectiveness of natural imaging baselines and their transferability to medical datasets.*

6.1.3 Summary

In summary, the sequence of experiments underscores the limitations of Neural Process-based models and highlights the importance of systematically testing models on controlled (semi-)synthetic datasets, in addition to applying them to clinical data. The controllability of the data motivated us to pursue a semi-synthetic augmentation and data-generation strategy. NP-based methods exhibited limited robustness, often failing to surpass the LCI baseline and showing minimal sensitivity to explicit temporal conditioning. In contrast, natural imaging baselines such as SimVP achieved consistently stronger results on synthetic tasks, even without explicit time embeddings, suggesting that their backbone is more apt for spatio-temporal modeling.
These findings suggest that while NP formulations are conceptually elegant, their practical benefit for longitudinal medical and synthetic image generation remains modest. The observed performance gap and the relative strength of natural-image

baselines motivated us to explore flow-based methods that directly model continuous dynamics.

6.2 Longitudinal Augmentation and Data Generation

This section presents results from our longitudinal augmentation experiments and their broader implications. Our preliminary experiments with longitudinal augmentation and data generation clarify how these methods fit within synthetic data generation.

6.2.1 Findings for Longitudinal Augmentations

Synthetic augmentation has been widely studied in machine learning and medical imaging. Interest comes from data scarcity, privacy concerns, and the need for controlled evaluation (see Section 2.4.3). In that section, we reviewed earlier work and examined how it fits the broader trend of using synthetic data for model development and benchmarking. The longitudinal augmentation approach produced an additional dataset suitable for validating forecasting methods. We evaluated its effectiveness in two ways: First, we integrated the generated sequences online as additional training data to improve SimVP forecasting on the 2D ACDC dataset. Improvements were modest but consistent. This is remarkable given the simple augmentations and strong artefacts, which reduced visual realism (see 5.5). Several studies have shown the benefit of even imperfect synthetic distributions. These benefits occur when they are appropriately aligned with model assumptions [40, 41]. This supports our hypothesis, based on prior literature: synthetic data may be a useful inductive prior even if it does not match real data perfectly [60, 148]. Our results match this view; Longitudinal augmentations need not perfectly replicate disease progression to be useful. As emphasized earlier, the objective here was functional utility rather than realistic quality. Higher realism and stronger performance could come from using more anatomically relevant segmentations, which would better capture actual appearance changes.
In addition to quantitative evaluation, we also visually inspected the generated sequences. Despite their simplicity, the results appeared realistic, demonstrating clear progression between images. In Section 5.3, we benchmarked two spatio-temporal baselines and TFM on the semi-synthetic longitudinal BraTS dataset. However, whether those results fully validate the longitudinal augmentation approach remains an open question, as we did not use it for downstream tasks.

6.2.2 Limitations of Longitudinal Augmentations

Current limitations include a simple linear latent trajectory for our experiments. We also used coarse segmentation units, such as whole-organ boundaries instead of finer substructures. Both factors limit the anatomical realism of sequences. They may also reduce applicability for tasks needing fine spatial or morphological detail. It is important to note that this limitation is not foundational, but rather a

simplification made to demonstrate the feasibility of using Longitudinal Augmentations. Future work could add more realistic, disease-specific temporal patterns, like sigmoidal growth or decay curves. These could come from clinical data or disease trajectories. Adding finer-grained segmentation, down to sub-organ or lesion annotations, may increase plausibility and improve alignment with observed progression. We applied these augmentations only once, but using them multiple times could produce finer deformations, given the Gaussian and directional settings. These enhancements should preserve controllability, flexibility, and low data needs. This balance makes the augmentations attractive for augmentation and benchmarking. The latent trajectory can be better fit to clinical patterns, using longitudinal studies or expert annotations. We acknowledge that we qualitatively evaluated results only superficially and by a non-medical expert. Our experiments show longitudinal augmentations reliably generate synthetic trajectories that match key visual and structural features of real data.

6.2.3 Summary and Outlook for Longitudinal Augmentations

Overall, these exploratory longitudinal augmentation experiments primarily serve as proof-of-concept. The generated sequences demonstrate realistic evolution, even with a basic latent growth model. Due to their computational efficiency and flexibility, these methods are practical for augmentation and benchmarking.
In Section 5.3, we show that such sequences assist in evaluating longitudinal methods. Nonetheless, rigorous validation of these methods remains necessary. A comprehensive evaluation should compare augmentations to real datasets, assess their alignment with biomarkers, or include expert assessment.
Nevertheless, our biologically-informed longitudinal augmentations offer a lightweight, controllable way to generate spatio-temporal data. While still in an early stage, these augmentations have the potential to advance scalable, interpretable, and computationally efficient semi-synthetic data generation for medical imaging, with the intent to complement existing data-driven and generative approaches.

6.3 Medical Spatio-Temporal Learning using Flow Matching Discussion

We proposed and then Temporal Flow Matching (TFM) for image prediction on three medical spatio-temporal datasets (ACDC, LUMIERE, ISLES). Across all datasets, TFM consistently surpassed the Last Context Image (LCI) and competing methods in terms of PSNR, SSIM, and NRMSE. It remained stable under ablations and was computationally efficient. The main findings are

- **Accuracy:** TFM outperformed all baselines and consistently beat the LCI across all metrics.
- **Robustness:** Results are stable across ablations, and most design choices had a minor impact
- **Major Impact:** Learning the whole sequence and adding sparsity filling made the model very performant. This highlights the value of modeling within the same distribution.
- **Efficiency:** Supports Single-GPU training with modest memory and runtime requirements, requires no heavy pretraining or augmentations, and remains well below diffusion baselines and comparable to natural imaging methods.

6.3.1 Revisiting the Reverse QR Problem

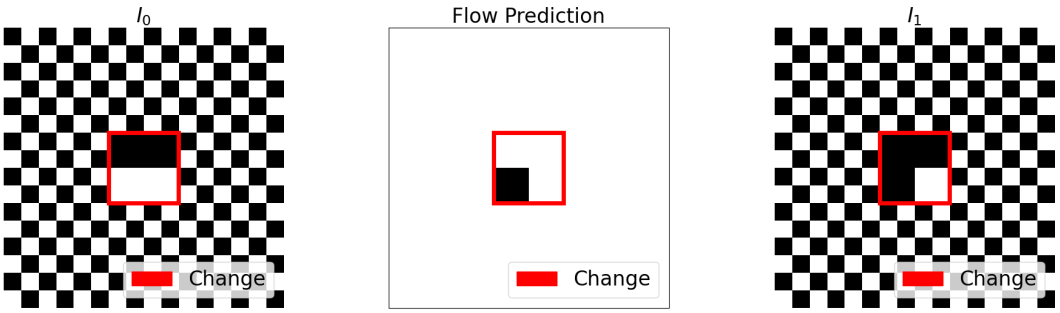
In Section 2.3.3, we introduced the *reverse QR Code problem*, which describes a toy problem, where there is a high-dimensional image time series, but the spatio-temporal change is locally bounded. We will illustrate here how FM can address this issue, even when operating on the same low spatial resolutions. Recall that this toy example showed a key limitation of current evaluation practices: Even a perfect longitudinal model cannot achieve a perfect score if its spatial predictions are slightly misaligned. Most standard pixel-wise metrics treat all pixels equally, without accounting for change. As a result, models that correctly predict dynamic regions can still be penalized for mainly irrelevant background errors.

This raises two important questions:

- *Are the current metrics fit for purpose?*
- *Can modeling the difference mitigate this issue?*

While we cannot definitely claim that Flow Matching (FM) resolves these problems, we can see that it does address them in part. Since FM explicitly models voxel-level displacements, it focuses on regions of change. This can potentially reduce the bias of static regions. As illustrated in Figure 6.1, FM perfectly solves the aforementioned reverse QR code problem.

A potential next step would be to design a *difference-weighted* loss function, one that gives higher importance to regions exhibiting change. Under such a metric, a



(a) Start Context **(b)** Flow **(c)** Prediction

Figure 6.1: Comparison between the starting last context, the predicted flow, and the target *as well as* the prediction.

perfect longitudinal model would achieve a perfect difference-weighted MSE, while the LCI would not. However, interpreting such metrics may be difficult. Possibly, they are used only in conjunction with regular metrics. In summary, this example highlights a key lesson: Flow-based approaches can solve this toy problem. Whether this result generalizes to actual time series remains open. Ultimately, we conclude that progress in spatio-temporal modeling must also include better ways to measure temporal accuracy.

Ablations

We conducted a series of ablation studies to evaluate how individual components of our tfm design influence performance. The experiments reveal two findings. Some design choices are crucial for model stability and accuracy. Others show that our approach is resilient to architectural or hyperparameter variation.

Critical Design Choices Table 5.8 summarizes the critical ablation results. The most impactful factor is the sparsity-filling strategy; removing it leads to a substantial drop in performance. This is expected, as sparsity filling directly aligns with the motivation to leverage samples from the same distribution. In contrast, training exclusively on non-zero frames leads to unstable training.

Robust Design Choices Other factors appear to impact our method less: Using the last frame or mean aggregation leads to only minor metric differences. Further work could test which strategy generalizes better across datasets. Replacing the attention U-Net backbone with a lighter variant slightly reduces performance. However, it is computationally lighter. Finally, the set of ablations on function evaluations, feature dimensionality, etc. has little effect on performance. This suggests

that TFM is stable across different configurations.

Experimental Setup and Sampling Challenges

During experimentation, we found that the **order and strategy of temporal sampling** play a far more significant role than initially expected. If the sampling order during validation is not fixed, the performance of the LCI heuristic and each model fluctuates dramatically, depending on which frames are visible. For example, when only the last frame is retained, results appear strong. This skews the validation distribution toward overfilled sequences when choosing the best performing validation seed. Such conditions may not reflect realistic settings.
To illustrate, suppose each index is masked with probability p across n frames. The probability of keeping the last frame is p, whereas the probability of keeping only the first frame is p^{n-1}. We observed this issue with ACDC and ISLES because we artificially subsampled these datasets. One fix is to randomly sample which frame is the last available frame (=LCI), the first frame, and then fill in between. This practice mitigates this bias. We emphasize this as a critical experimental detail for reproducibility, especially in irregularly sampled data.
A related challenge concerns picking the best **validation performance**. Even with a fixed dataset split, random resampling of context frames can shift the validation distribution between runs and validation steps. This behavior, resembling *validation bootstrapping*, can substantially distort comparability across reported results. To ensure consistency, we fixed the random subsampling pattern across all runs. This arguably improved comparability and fixed the value of LCI between validation epochs.
Random subsampling can vary validation performance and the choice of 'best' validation epoch. This variability is higher if it is not explicitly controlled. Even the seemingly simple decision to use multiple context images introduces additional randomness that is rarely discussed in prior literature. In most studies, image time series are used either in full or only for single-context images. However, in longitudinal medical datasets where data are scarce, such sampling effects can disproportionately influence validation scores. In contrast, natural image datasets typically have sufficient sample sizes. As a result, stochastic effects tend to average out.
This highlights the importance of **explicitly sampling protocols** in medical spatio-temporal research. Without such transparency, small differences in sampling strategies can cause large discrepancies in reported performance. This is especially true for methods operating on sparse but grid-quantized data.
One possible protocol to standardize evaluation would be to iteratively mask each index in the sequence, producing T distinct validation results per sequence. While averaging these results is not straightforward, this approach would ensure that all temporal distances are covered systematically. Alternatively, one could fix the first and the last frame while randomly sampling the intermediate context frames. This

strategy offers greater coverage, though at the cost of reduced interpretability when aggregating performance into a single scalar metric.

6.3.2 Baseline Selection and Comparison

Several baselines, such as SADM [154], fell short of expectation, to a degree that SADM failed to even converge under *their* experimental settings and datasets. The standalone ViViT model also performed extremely poorly, both qualitatively and quantitatively. This partially explains SADM's performance, as ViViT produces the temporal predictions as an intermediate step, as conditioning for the diffusion stage. Given the ViViT's patch size of $8 \times 32 \times 32$, it is likely that this coarse representation limited its ability to capture fine image details. The discrepancy between prior work's reported SADM results and our empirical findings was substantial. To obtain a functional sequence-based diffusion baseline and reduce this discrepancy, we explored multiple strategies to improve SADM's performance.

1. Pretrain the autoencoder and vision transformer separately before diffusion training.

2. Freezing pretrained components and training the diffusion model in the latent space.

3. Adding a skip connection to the vision transformer to improve the diffusion model's input quality.

Despite these efforts, our adapted SADM only reached SSIM = 0.873, PSNR = 23.96 dB, and NRMSE = 0.066, still far below TFM's performance and visually more blurry. Ironically, we spent more time attempting to stabilize SADM than was needed to train TFM, despite TFM having a substantially smaller computational footprint (Fig. 5.8). As shown in the same figure, TFM not only achieves superior predictive accuracy but also requires less memory than SADM. For completeness, the original SADM implementation could not be trained in our environment with their code; we therefore report their published results alongside those from our adapted implementation.
Identifying robust medical baselines proved challenging, prompting us to include natural imaging methods. While approaches like SimVP represent the state-of-the-art (SOTA) in spatio-temporal modeling, they transfer poorly to the 3D medical domain. SimVP performed well on 2D ACDC, but the performance degraded immensely on 3D experiments. One likely reason is the reduced number of parameters we had to impose in order to make 3D training feasible, along with the fact that we were the first to extend this method to 3D. A deeper investigation into how to recover the lost performance is left for future work. Notably, even in the 2D setting, recurrent approaches consistently outperformed SimVP, underscoring the

gap between results achieved on natural imaging benchmarks and those observed in medical datasets.
In contrast, truly medical imaging specific baselines designed for sparse or irregular temporal modeling remain scarce. To the best of our knowledge, mainly SADM and ConvLSTM variants directly address comparable problem settings based on sequences on discrete grid settings. This highlights the lack of suitable baselines and further motivates the development of TFM as a lightweight and reliable foundation for longitudinal image modeling.

Key Technical Contributions

In this section, we summarize the key technical contributions of TFM and explain why it is a strong candidate as a baseline for spatio-temporal modeling tasks in medical imaging.

- **Efficient and Scalable Architecture:** We demonstrate that a U-Net-based architecture can effectively model spatio-temporal medical imaging data using a single local GPU, without extreme restrictions on input size. While ACDC is relatively small, LUMIERE is representative of typical 3D medical imaging datasets in terms of resolution and complexity.
- **Handling Sparse and Irregularly Sampled Data:** TFM is explicitly designed to work with sparse and irregularly sampled time series, albeit on a discrete grid (4.30), common in real-world medical imaging scenarios. Many existing methods implicitly assume regular sampling or fixed time intervals and fail to address this challenge. By incorporating a sparsity-filling strategy, we enable the same model to operate effectively in both sparse and irregularly sampled regimes, making TFM a versatile choice across a wide range of clinical acquisition settings.
- **Flexibility and Generalizability:** TFM is not tied to a specific imaging modality, anatomy, or prediction task. It can be adapted to different datasets and problem settings with minimal architectural modifications. We demonstrate this flexibility across multiple datasets, highlighting TFM's potential as a universal baseline for medical imaging tasks involving temporal dynamics.
- **Strong Baseline Performance:** TFM achieves state-of-the-art performance on several challenging datasets, outperforming existing baselines in both quantitative metrics and qualitative evaluations. This consistent performance across tasks, hyperparameters, and datasets underscores its effectiveness and reliability.
- **Methodological Simplicity:** While there are opportunities for further improvement, TFM is conceptually and practically straightforward to implement and train. Unlike other models that require multi-stage pipelines—such as pretraining separate autoencoders, control nets or vision transformers, TFM can be trained

end-to-end from scratch. This simplicity lowers the barrier to adoption and makes it accessible to a broader range of researchers and practitioners.

6.3.3 Temporal Flow Matching Limitations

Our prediction task serves as a proxy for disease progression, but the practice may be more complex. Fully modeling disease dynamics is beyond the scope of this work; here, we focus on image-based forecasting. Additionally, we have the following limitations;

- **Implicit Temporal Modeling:** Temporal information in TFM is represented implicitly through the relative position of the input, rather than being modeled explicitly in the architecture. While this approach is effective in practice, it may not fully capture more complex temporal dependencies, particularly in cases with highly irregular or nonlinear dynamics.
- **Deterministic Predictions and Uncertainty:** As discussed in Section 5.4.2, the standard TFM formulation produces deterministic predictions, even though this is rather an evaluation than a method issue. This limitation concerns evaluation rather than modeling itself: *How can we reliably judge a variation of predictions to be valid, while having only a single ground truth?*
- **Limited Contextual Information:** Temporal context in TFM is encoded through position within the tensor, which introduces computational constraints on the input size. Consequently, the amount of temporal context that can be processed for long sequences is limited, potentially affecting performance.
- **Limited Interpretability:** While TFM achieves strong performance according to standard quantitative metrics, the link between these metrics and true clinical insight remains uncertain. As discussed previously regarding difference-attenuated metrics, high metric performance does not necessarily translate into meaningful temporal or pathological understanding. This reflects a broader challenge in medical AI, where numerical accuracy may not fully capture clinically relevant progression dynamics. Addressing this limitation will likely require the development of domain-specific or change-localized evaluation metrics that better reflect clinical interpretability.

6.3.4 TFM Summary

Temporal Flow Matching (TFM) offers a simple, computationally efficient approach to longitudinal image prediction. Across all evaluated datasets, it consistently surpasses the LCI heuristic and spatio-temporal competitors, while training on modest hardware. By supervising a velocity field on a sparsity-filled sequence, TFM handles irregular sampling without specialized components. These properties make it a

dependable baseline for medical spatio-temporal modeling, and a solid foundation for future work.

6.4 TFM Extension

While the discrete TFM achieves strong results, it left several open challenges. The most relevant challenge for medical imaging is handling continuous time. Clinical longitudinal follow-ups almost never lie on a fixed grid. We addressed this by extending TFM to a continuous time formulation. After that, we discuss two further extensions may be useful for medical data: A deformation-based variant better aligns anatomical changes, and a Schrödinger Bridge formulation which describes a regularized optimal transport formulation.

6.4.1 Continuous Temporal Flow Matching

The motivation for introducing a continuous-time formulation is that longitudinal medical data are often irregularly sampled (not easily quantized). Discretizing these timesteps into a fixed grid introduces quantization errors. It increases computational overhead, especially when many timepoints are empty or unevenly spaced. Furthermore, the discrete TFM relied on sparsity filling and implicit temporal information, which can cause further errors. Thus, the continuous TFM replaces discrete quantization with direct conditioning on real-valued timesteps. Despite this change, the architecture and loss function remain identical. Overall, the method learns flow directly in continuous time, without needing a predefined temporal grid. The main observations are:

- On highly irregular ACDC splits, the continuous version of TFM yielded clear and consistent improvements.
- On regularly sampled datasets, its performance matched that of the discrete TFM.

In the discrete TFM, the step embedding for τ mainly served as a technical mechanism to induce flow between frames. It did not encode meaningful temporal information. Replacing this component with continuous conditioning is both natural and elegant. It extends the method's applicability to real-world, irregularly sampled data. Further work is needed to regularize and interpret the learned continuous dynamics in practice, even though we have taken some first steps in that direction.

6.4.2 Schrödinger Bridge TFM

Spatio-temporal prediction in medical imaging involves uncertainty; the available context frames often do not uniquely determine future anatomy. Deterministic processes, such as in the ACDC or ISLES datasets, are suitable when disease progression allows a pre-defined trajectory. Other diseases, such as glioblastoma, exhibit more stochastic evolution with multiple plausible futures. In these cases, we want to model a distribution of possible futures, not just a single outcome.

A Schrödinger Bridge (SB) formulation offers a principled framework for *introducing stochasticity* into Flow Matching while remaining under *well-defined constraints*. Unlike deterministic Flow Matching, SB identifies the most likely stochastic process between two distributions under Brownian motion and adds an entropic regularization term. Intuitively, this represents a soft version of optimal transport. The trajectory is smooth and regularized by uncertainty. Such regularization could, in future work, be derived from anatomical or learned priors [84]. It would constrain the stochastic trajectories to remain physiologically plausible.
Evaluating stochastic models remains a major challenge. Deterministic predictions use voxel-wise errors, but evaluating distributions over futures requires comparing probability distributions. Common generative metrics, such as FID or Inception Score, are limited because they fail to assess whether the predicted modes correspond to clinically meaningful variations. A more theoretically grounded option is to compute divergences like the Wasserstein distance or KL divergence between the learned and ground-truth trajectory distributions. However, this is highly infeasible in voxel space. Because we intentionally implemented SB TFM in voxel space, we now need to move to a more compact latent representation to address this limitation.
Future work could draw on the ideas of our *longitudinal augmentations* 4.2, which already generate realistic image trajectories. A stochastic version of this framework could test probabilistic forecasting. This would enable sampling from anatomically regularized stochastic flows.
In summary, integrating the Schrödinger Bridge formulation within our TFM method represents a *natural and mathematically grounded next step*. Our exploration faces evaluation difficulties and the challenge of using high-dimensional voxel space instead of a compact latent domain. Still, it offers a clear direction for future work.

6.4.3 Deformation and Flow Matching

The deformation-based extension of TFM remains conceptual for this work, yet it introduces a mathematically and intuitively appealing perspective on modeling spatio-temporal changes in medical imaging. Instead of moving intensities through voxel space, we move mass using geometric deformations. The reframing replaces intensity interpolation with spatial transport, possibly ensuring that intermediate states remain anatomically consistent. Though preliminary, this is a more elegant extension for settings where structural consistency matters, e. g. for datasets like ACDC.

6.4.4 Summary of Extensions

We extended the discrete TFM formulation to continuous time, providing a new interpretation of the flow step τ as a real-valued temporal variable. This continuous

formulation represents the conceptual and technical culmination of our work, addressing all the key limitations identified in existing methods throughout the literature. It thus constitutes the core methodological contribution of this thesis. Building on this, we presented initial results using Schrödinger Bridges; while they function technically, their effectiveness is limited by our current evaluation approach. These SB models, in particular, enable a more natural modeling of image prediction under Brownian motion constraints. Additionally, we explored how deformation-based TFM could leverage the inherent anatomical changes to model image distributions more naturally. These extensions clearly demonstrate the flexibility of the Temporal Flow Matching framework and establish new directions for developments in longitudinal medical imaging.

Conclusion and Outlook 7

This work addressed the challenge of forecasting longitudinal medical images under sparse and irregular sampling. We began by assessing both medical and natural imaging baselines and found that forecasting entire images or segmentations is difficult, to the point that even a simple heuristic such as Last Context Image (LCI) remains hard to surpass. Beyond these empirical findings, our main contributions include the introduction of semi-synthetic augmentation strategies and, most importantly, the Temporal Flow Matching (TFM) framework, which enables forecasting of 3D medical image time series at arbitrary timepoints. TFM is computationally efficient, robust across different parameter settings, and consistently outperforms all tested baselines, often using equal or significantly fewer computational resources. We further explored theoretically appealing extensions, including Schrödinger Bridges and deformation-aware formulations, both of which aim to capture plausible image evolution. Future work could extend this foundation by leveraging large pre-trained or foundation models, exploring semi-synthetic pre-training strategies, and developing methods and evaluations for generating multiple plausible futures. Additional directions include designing metrics that focus on localized differences rather than global similarity. In summary, this work addresses a previously underexplored problem—modeling of sparse and irregular medical image time series—and provides a strong methodological and conceptual foundation for modeling disease trajectories. While substantial progress has been made, achieving robust clinical translation will require continued effort in scaling, evaluation, and integration with real-world workflows.

Bibliography

[1] "(PDF) Advancements in Computer Vision: A Comprehensive Survey of Image Processing and Interdisciplinary Applications". In: *ResearchGate* (Aug. 2025) (cit. on p. 1).

[2] "(PDF) PLASTIMATCH– AN OPEN SOURCE SOFTWARE SUITE FOR RADIOTHERAPY IMAGE PROCESSING." In: *ResearchGate* (cit. on p. 82).

[3] "(PDF) SYSTEMATIC LITERATURE REVIEW ON ARTIFICIAL INTELLIGENCE APPLICATIONS IN SUPPLY CHAIN DEMAND FORECASTING". In: *ResearchGate* (Aug. 2025) (cit. on p. 2).

[4] Josh Abramson et al. "Accurate Structure Prediction of Biomolecular Interactions with AlphaFold 3". In: *Nature* 630.8016 (June 2024), pp. 493–500 (cit. on p. 1).

[5] Anurag Arnab et al. "ViViT: A Video Vision Transformer". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 6836–6846 (cit. on pp. 10, 21, 24).

[6] Arpit Bansal et al. "Cold Diffusion: Inverting Arbitrary Image Transforms Without Noise". In: (2022) (cit. on p. 33).

[7] Olivier Bernard et al. "Deep Learning Techniques for Automatic MRI Cardiac Multi-Structures Segmentation and Diagnosis: Is the Problem Solved?" In: *IEEE Transactions on Medical Imaging* 37.11 (Nov. 2018), pp. 2514–2525 (cit. on pp. 16, 17, 49, 89, 93).

[8] Valentin De Bortoli et al. "Diffusion Schrödinger Bridge with Applications to Score-Based Generative Modeling". In: *ArXiv* (June 2021) (cit. on p. 75).

[9] H. Braak and E. Braak. "Neuropathological Stageing of Alzheimer-related Changes". In: *Acta Neuropathologica* 82.4 (Sept. 1991), pp. 239–259 (cit. on p. 48).

[10] J. Bushberg. "The Essential Physics of Medical Imaging". In: Dec. 2001 (cit. on p. 7).

[11] Jinzheng Cai et al. "Deep Lesion Tracker: Monitoring Lesions in 4D Longitudinal Imaging Studies". In: *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2021), pp. 15154–15164 (cit. on p. 25).

[12] CDC. *Heart Disease Facts*. https://www.cdc.gov/heart-disease/data-research/facts-stats/index.html. Feb. 2025 (cit. on p. 49).

[13] Ricky T. Q. Chen et al. *Neural Ordinary Differential Equations*. Dec. 2019. arXiv: 1806.07366 [cs] (cit. on pp. 21, 29–31, 55).

[14] Ting Chen et al. "A Simple Framework for Contrastive Learning of Visual Representations". In: *ArXiv* (Feb. 2020) (cit. on p. 75).

[15] Xinlei Chen and Kaiming He. "Exploring Simple Siamese Representation Learning". In: *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2021), pp. 15745–15753 (cit. on p. 75).

[16] Yongxin Chen, Tryphon T. Georgiou, and Michele Pavon. "Optimal Transport in Systems and Control". In: *Annual Review of Control, Robotics, and Autonomous Systems* 4.1 (May 2021), pp. 89–113 (cit. on p. 40).

[17] Yunjie Chen et al. "Vestibular Schwannoma Growth Prediction from Longitudinal MRI by Time-Conditioned Neural Fields". In: *Medical Image Computing and Computer Assisted Intervention – MICCAI 2024*. Ed. by Marius George Linguraru et al. Vol. 15003. Cham: Springer Nature Switzerland, 2024, pp. 508–518 (cit. on p. 25).

[18] Lenaic Chizat et al. *Unbalanced Optimal Transport: Dynamic and Kantorovich Formulation*. https://arxiv.org/abs/1508.05216v3. Aug. 2015 (cit. on p. 79).

[19] Aram Davtyan, Sepehr Sameni, and Paolo Favaro. "Efficient Video Prediction via Sparsely Conditioned Flow Matching". In: *2023 IEEE/CVF International Conference on Computer Vision (ICCV)* (Oct. 2023), pp. 23206–23217 (cit. on p. 22).

[20] Jia Deng et al. "ImageNet: A Large-Scale Hierarchical Image Database". In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. June 2009, pp. 248–255 (cit. on p. 9).

[21] Prafulla Dhariwal and Alex Nichol. "Diffusion Models Beat GANs on Image Synthesis". In: *ArXiv* (May 2021) (cit. on p. 1).

[22] Prafulla Dhariwal and Alex Nichol. "Diffusion Models Beat GANs on Image Synthesis". In: *ArXiv* (May 2021) (cit. on p. 34).

[23] Nico Disch et al. "Applying Longitudinal Augmentation and Data Generation (LAUGEN) in Medical Imaging". In: *Synthetic Data for Computer Vision Workshop @ CVPR 2025*. May 2025 (cit. on pp. 4, 5, 90).

[24] Nico Albert Disch et al. "Back to the Future: Challenges of Sparse and Irregular Medical Image Time Series". In: *Medical Image Computing and Computer Assisted Intervention – MICCAI 2024 Workshops*. Ed. by Anna Schroder et al. Vol. 15401. Cham: Springer Nature Switzerland, 2025, pp. 15–25 (cit. on p. 83).

[25] Nico Albert Disch et al. *Temporal Flow Matching for Learning Spatio-Temporal Trajectories in 4D Longitudinal Medical Imaging*. Aug. 2025. arXiv: 2508.21580 [cs] (cit. on p. 5).

[26] Mengjin Dong et al. "DeepAtrophy: Teaching a Neural Network to Detect Progressive Changes in Longitudinal MRI of the Hippocampal Region in Alzheimer's Disease". In: *NeuroImage* 243 (Nov. 2021), p. 118514 (cit. on p. 25).

[27] Alexey Dosovitskiy et al. *An Image Is Worth 16x16 Words: Transformers for Image Recognition at Scale*. June 2021. arXiv: 2010.11929 [cs] (cit. on p. 10).

[28] Emilien Dupont, Arnaud Doucet, and Yee Whye Teh. *Augmented Neural ODEs*. Oct. 2019. arXiv: 1904.01681 [stat] (cit. on pp. 30, 31).

[29] Leo Feng et al. *Latent Bottlenecked Attentive Neural Processes*. Mar. 2023. arXiv: 2211.08458 [cs] (cit. on p. 23).

[30] Fábio J. N. Ferreira and Agnaldo S. Carneiro. "AI-Driven Drug Discovery: A Comprehensive Review". In: *ACS Omega* 10.23 (), pp. 23889–23903 (cit. on p. 1).

[31] Alejandro F. Frangi, Sotirios A. Tsaftaris, and Jerry L. Prince. "Simulation and Synthesis in Medical Imaging". In: *IEEE transactions on medical imaging* 37.3 (Mar. 2018), pp. 673–679 (cit. on p. 46).

[32] Wanyi Fu et al. "iPhantom: A Framework for Automated Creation of Individualized Computational Phantoms and Its Application to CT Organ Dosimetry". In: *IEEE journal of biomedical and health informatics* 25.8 (Aug. 2021), pp. 3061–3072 (cit. on p. 28).

[33] K. Fukushima. "Neocognitron: A Self Organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position". In: *Biological Cybernetics* 36.4 (1980), pp. 193–202 (cit. on p. 10).

[34] Zhangyang Gao et al. "SimVP: Simpler Yet Better Video Prediction". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 3170–3180 (cit. on pp. 21, 22, 24, 64, 86–89, 115).

[35] Zhihan Gao et al. "Earthformer: Exploring Space-Time Transformers for Earth System Forecasting". In: *Advances in Neural Information Processing Systems* 35 (Dec. 2022), pp. 25390–25403 (cit. on p. 22).

[36] Marta Garnelo et al. "Conditional Neural Processes". In: *Proceedings of the 35th International Conference on Machine Learning*. PMLR, July 2018, pp. 1704–1713 (cit. on p. 23).

[37] Marta Garnelo et al. *Neural Processes*. https://arxiv.org/abs/1807.01622v1. July 2018 (cit. on pp. 21, 23, 24).

[38] Gilles E. Gignac and Eva T. Szodorai. "Defining Intelligence: Bridging the Gap between Human and Artificial Perspectives". In: *Intelligence* 104 (May 2024), p. 101832 (cit. on p. 1).

[39] Nicole Gillespie et al. *Trust, Attitudes and Use of Artificial Intelligence: A Global Study 2025*. Tech. rep. The University of Melbourne, 2025, 4974511 Bytes (cit. on p. 1).

[40] Mauro Giuffrè and Dennis L. Shung. "Harnessing the Power of Synthetic Data in Healthcare: Innovation, Application, and Privacy". In: *npj Digital Medicine* 6.1 (Oct. 2023), p. 186 (cit. on pp. 28, 117).

[41] Andre Goncalves et al. "Generation and Evaluation of Synthetic Patient Data". In: *BMC Medical Research Methodology* 20.1 (Dec. 2020), p. 108 (cit. on pp. 27, 117).

[42] Cade Gordon and Natalie Parde. "Latent Neural Differential Equations for Video Generation". In: *Preregister@NeurIPS*. Nov. 2020 (cit. on p. 22).

[43] Albert Gu and Tri Dao. *Mamba: Linear-Time Sequence Modeling with Selective State Spaces*. May 2024. arXiv: `2312.00752 [cs]` (cit. on pp. 10, 11, 59).

[44] Degan Hao and Mohammadreza Negahdar. "Predicting Outcomes in Long COVID Patients with Spatiotemporal Attention". In: *2023 IEEE 11th International Conference on Healthcare Informatics (ICHI)* (June 2023), pp. 162–167 (cit. on p. 25).

[45] Kaiming He et al. "Momentum Contrast for Unsupervised Visual Representation Learning". In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2020), pp. 9726–9735 (cit. on p. 75).

[46] Mark S. Henry et al. "The Development of Effective Biomarkers for Alzheimer's Disease: A Review". In: *International Journal of Geriatric Psychiatry* 28.4 (Apr. 2013), pp. 331–340 (cit. on p. 48).

[47] Jonathan Ho, Ajay Jain, and P. Abbeel. "Denoising Diffusion Probabilistic Models". In: *ArXiv* (June 2020) (cit. on pp. 1, 32, 33).

[48] Jonathan Ho and Tim Salimans. "Classifier-Free Diffusion Guidance". In: (2022) (cit. on p. 34).

[49] Sepp Hochreiter and Jürgen Schmidhuber. "Long Short-Term Memory". In: *Neural Computation* 9.8 (Nov. 1997), pp. 1735–1780 (cit. on p. 22).

[50] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. "Multilayer Feedforward Networks Are Universal Approximators". In: *Neural Networks* 2.5 (Jan. 1989), pp. 359–366 (cit. on p. 9).

[51] Fabian Isensee et al. “Automated Brain Extraction of Multisequence MRI Using Artificial Neural Networks”. In: *Human Brain Mapping* 40.17 (Dec. 2019), pp. 4952–4964 (cit. on p. 82).

[52] C. R. Jack et al. “Antemortem MRI Findings Correlate with Hippocampal Neuropathology in Typical Aging and Dementia”. In: *Neurology* 58.5 (Mar. 2002), pp. 750–757 (cit. on p. 48).

[53] Clifford R. Jack et al. “The Alzheimer’s Disease Neuroimaging Initiative (ADNI): MRI Methods”. In: *Journal of Magnetic Resonance Imaging* 27.4 (Apr. 2008), pp. 685–691 (cit. on p. 48).

[54] Sonain Jamil, Md Jalil Piran, and Oh-Jin Kwon. “A Comprehensive Survey of Transformers for Computer Vision”. In: *Drones* 7.5 (May 2023), p. 287 (cit. on p. 1).

[55] Mark Jenkinson and Stephen Smith. “A Global Optimisation Method for Robust Affine Registration of Brain Images”. In: *Medical Image Analysis* 5.2 (June 2001), pp. 143–156 (cit. on p. 82).

[56] Saurav Jha et al. *The Neural Process Family: Survey, Applications and Perspectives*. Oct. 2023. arXiv: `2209.00517 [cs]` (cit. on pp. 23, 24).

[57] Yang Jin et al. “Pyramidal Flow Matching for Efficient Video Generative Modeling”. In: (2024) (cit. on p. 22).

[58] John Jumper et al. “Highly Accurate Protein Structure Prediction with AlphaFold”. In: *Nature* 596.7873 (Aug. 2021), pp. 583–589 (cit. on p. 1).

[59] R. E. Kalman. “A New Approach to Linear Filtering and Prediction Problems”. In: *Journal of Basic Engineering*. Vol. 82. Mar. 1960, pp. 35–45 (cit. on p. 21).

[60] Shivani Kapania et al. “Examining the Expanding Role of Synthetic Data Throughout the AI Development Pipeline”. In: *Proceedings of the 2025 ACM Conference on Fairness, Accountability, and Transparency* (June 2025), pp. 45–60 (cit. on pp. 28, 117).

[61] Philipp Kickingereder et al. “Automated Quantitative Tumour Response Assessment of MRI in Neuro-Oncology with Artificial Neural Networks: A Multicentre, Retrospective Study”. In: *The Lancet Oncology* 20.5 (May 2019), pp. 728–740 (cit. on p. 51).

[62] Patrick Kidger et al. *Neural Controlled Differential Equations for Irregular Time Series*. Nov. 2020. arXiv: `2005.08926 [cs]` (cit. on p. 31).

[63] Hyunjik Kim et al. *Attentive Neural Processes*. July 2019. arXiv: `1901.05761 [cs]` (cit. on p. 23).

[64] Seong Tae Kim et al. "Longitudinal Quantitative Assessment of COVID-19 Infection Progression from Chest CTs". In: *Medical Image Computing and Computer Assisted Intervention – MICCAI 2021*. Ed. by Marleen De Bruijne et al. Vol. 12907. Cham: Springer International Publishing, 2021, pp. 273–282 (cit. on p. 25).

[65] Diederik P. Kingma and Jimmy Ba. *Adam: A Method for Stochastic Optimization*. Jan. 2017. arXiv: `1412.6980 [cs]` (cit. on pp. 14, 82, 94).

[66] Alexander Kirillov et al. *Segment Anything*. Apr. 2023. arXiv: `2304.02643 [cs]` (cit. on p. 61).

[67] Aishik Konwer et al. "Temporal Context Matters: Enhancing Single Image Prediction with Disease Progression Representations". In: *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2022), pp. 18802–18813 (cit. on p. 25).

[68] Balint Kovacs et al. "Anatomy-Informed Data Augmentation for Enhanced Prostate Cancer Detection". In: *Medical Image Computing and Computer Assisted Intervention – MICCAI 2023*. Ed. by Hayit Greenspan et al. Cham: Springer Nature Switzerland, 2023, pp. 531–540 (cit. on pp. 61, 62).

[69] Oleg Kovalevskiy, Juan Mateos-Garcia, and Kathryn Tunyasuvunakool. "AlphaFold Two Years on: Validation and Impact". In: *Proceedings of the National Academy of Sciences* 121.34 (Aug. 2024), e2315002121 (cit. on p. 1).

[70] Moez Krichen. "Generative Adversarial Networks". In: *2023 14th International Conference on Computing Communication and Networking Technologies (ICCCNT)* (July 2023), pp. 1–7 (cit. on p. 1).

[71] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. "ImageNet Classification with Deep Convolutional Neural Networks". In: *Communications of the ACM* 60.6 (May 2017), pp. 84–90 (cit. on pp. 1, 9, 10).

[72] W. Kutta. "Beitrag Zur Naherungsweisen Integration Totaler Differentialgleichungen". In: (cit. on p. 58).

[73] Dmitrii Lachinov et al. "Learning Spatio-Temporal Model of Disease Progression With NeuralODEs From Longitudinal Volumetric Data". In: *IEEE Transactions on Medical Imaging* 43.3 (Mar. 2024), pp. 1165–1179 (cit. on p. 25).

[74] Pamela J. LaMontagne et al. "OASIS-3: Longitudinal Neuroimaging, Clinical, and Cognitive Dataset for Normal Aging and Alzheimer Disease". In: (Dec. 2019) (cit. on p. 48).

[75] Vincent Le Guen and Nicolas Thome. "Disentangling Physical Dynamics From Unknown Factors for Unsupervised Video Prediction". In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2020), pp. 11471–11481 (cit. on p. 22).

[76] Y. Lecun et al. "Gradient-Based Learning Applied to Document Recognition". In: *Proceedings of the IEEE* 86.11 (Nov. 1998), pp. 2278–2324 (cit. on p. 10).

[77] S. Legg and Marcus Hutter. "A Collection of Definitions of Intelligence". In: *Artificial General Intelligence*. June 2007 (cit. on p. 1).

[78] Tianyang Lin et al. "A Survey of Transformers". In: *AI Open* 3 (Jan. 2022), pp. 111–132 (cit. on p. 1).

[79] Yaron Lipman et al. *Flow Matching for Generative Modeling*. Feb. 2023. arXiv: 2210.02747 [cs] (cit. on pp. 2, 21, 38).

[80] Yaron Lipman et al. *Flow Matching Guide and Code*. Dec. 2024. arXiv: 2412.06264 [cs] (cit. on pp. 28, 36–38).

[81] Zachary Chase Lipton et al. "Learning to Diagnose with LSTM Recurrent Neural Networks". In: *CoRR* (Nov. 2015) (cit. on p. 25).

[82] Mattia Litrico et al. "TADM: Temporally-Aware Diffusion Model for Neurodegenerative Progression on Brain MRI". In: (2024) (cit. on p. 26).

[83] Chen Liu et al. *ImageFlowNet: Forecasting Multiscale Image-Level Trajectories of Disease Progression with Irregularly-Sampled Longitudinal Medical Images*. Apr. 2025. arXiv: 2406.14794 [eess] (cit. on pp. 31, 75).

[84] Guan-Horng Liu et al. *Generalized Schrödinger Bridge Matching*. Apr. 2024. arXiv: 2310.02233 [stat] (cit. on pp. 75, 127).

[85] Ilya Loshchilov and Frank Hutter. *Decoupled Weight Decay Regularization*. Jan. 2019. arXiv: 1711.05101 [cs] (cit. on p. 94).

[86] William Lotter, Gabriel Kreiman, and David Cox. *Unsupervised Learning of Visual Structure Using Predictive Generative Networks*. Jan. 2016. arXiv: 1511.06380 [cs] (cit. on p. 44).

[87] Donghuan Lu et al. "Multimodal and Multiscale Deep Neural Networks for the Early Diagnosis of Alzheimer's Disease Using Structural MR and FDG-PET Images". In: *Scientific Reports* 8.1 (Apr. 2018), p. 5697 (cit. on p. 25).

[88] P. A. M. and Norbert Wiener. "The Extrapolation, Interpolation and Smoothing of Stationary Time Series, with Engineering Applications." In: *Journal of the Royal Statistical Society. Series A (General)*. Vol. 113. 1950, p. 413. JSTOR: 10.2307/2981007 (cit. on p. 21).

[89] Lena Maier-Hein et al. "Metrics Reloaded: Recommendations for Image Analysis Validation". In: *Nature Methods* 21.2 (Feb. 2024), pp. 195–212 (cit. on pp. 8, 12).

[90] Kelly D. Martin and Johanna Zimmermann. "Artificial Intelligence and Its Implications for Data Privacy". In: *Current Opinion in Psychology* 58 (Aug. 2024), p. 101829 (cit. on p. 2).

[91] Jawad M. Melhem et al. "Updates in IDH-Wildtype Glioblastoma". In: *Neurotherapeutics* 19.6 (Oct. 2022), pp. 1705–1723 (cit. on p. 51).

[92] Bjoern H. Menze et al. "The Multimodal Brain Tumor Image Segmentation Benchmark (BRATS)". In: *IEEE Transactions on Medical Imaging* 34.10 (Oct. 2015), pp. 1993–2024 (cit. on pp. 46, 59, 89).

[93] Lisa Mosconi. "Glucose Metabolism in Normal Aging and Alzheimer's Disease: Methodological and Physiological Considerations for PET Studies". In: *Clinical and Translational Imaging* 1.4 (Aug. 2013), pp. 217–233 (cit. on p. 48).

[94] Shakhnoza Muksimova et al. "Multi-Modal Fusion and Longitudinal Analysis for Alzheimer's Disease Classification Using Deep Learning". In: *Diagnostics* 15.6 (Mar. 2025), p. 717 (cit. on p. 25).

[95] Quang Nguyen et al. "Dataset Diffusion: Diffusion-based Synthetic Dataset Generation for Pixel-Level Semantic Segmentation". In: (2023) (cit. on p. 28).

[96] ThiThuyHanh Nguyen. "Applications of Artificial Intelligence for Demand Forecasting". In: *Operations and Supply Chain Management: An International Journal* 16.4 (Nov. 2023), pp. 424–434 (cit. on p. 2).

[97] "nnU-Net Revisited: A Call for Rigorous Validation in 3D Medical Image Segmentation". In: *Lecture Notes in Computer Science*. Cham: Springer Nature Switzerland, 2024, pp. 488–498 (cit. on p. 12).

[98] Luke Oakden-Rayner et al. "Hidden Stratification Causes Clinically Meaningful Failures in Machine Learning for Medical Imaging". In: *Proceedings of the ACM Conference on Health, Inference, and Learning* 2020 (Apr. 2020), pp. 151–159 (cit. on p. 28).

[99] Jiahong Ouyang et al. "Longitudinal Pooling & Consistency Regularization to Model Disease Progression From MRIs". In: *IEEE Journal of Biomedical and Health Informatics* 25.6 (June 2021), pp. 2082–2092 (cit. on p. 25).

[100] Yidong Ouyang, Liyan Xie, and Guang Cheng. "Improving Adversarial Robustness Through the Contrastive-Guided Diffusion Process". In: *International Conference on Machine Learning*. Oct. 2022 (cit. on p. 28).

[101] Avik Pal, Alan Edelman, and Chris Rackauckas. "Locally Regularized Neural Differential Equations: Some Black Boxes Were Meant to Remain Closed!" In: (2023) (cit. on p. 31).

[102] Jens Petersen et al. *Continuous-Time Deep Glioma Growth Models*. July 2021. arXiv: `2106.12917 [eess]` (cit. on pp. 4, 21, 24, 27, 55, 56).

[103] Michael Poli et al. *Hypersolvers: Toward Fast Continuous-Depth Models*. Dec. 2020. arXiv: `2007.09601 [cs]` (cit. on p. 31).

[104] Ilan Price et al. "Probabilistic Weather Forecasting with Machine Learning". In: *Nature* 637.8044 (Jan. 2025), pp. 84–90 (cit. on p. 2).

[105] Jerry L. Prince and J. Links. "Medical Imaging Signals and Systems". In: Apr. 2005 (cit. on p. 7).

[106] Lemuel Puglisi, Daniel C. Alexander, and Daniele Ravì. "Brain Latent Progression: Individual-based Spatiotemporal Disease Progression on 3D Brain MRIs via Latent Diffusion". In: *Medical Image Analysis* (July 2025), p. 103734 (cit. on pp. 26, 95, 114).

[107] Danilo Jimenez Rezende and S. Mohamed. "Variational Inference with Normalizing Flows". In: *ArXiv* (May 2015) (cit. on pp. 30, 31).

[108] Evamaria O. Riedel et al. "ISLES 2024: The First Longitudinal Multimodal Multi-Center Real-World Dataset in (Sub-)Acute Stroke". In: (2024) (cit. on pp. 50, 93).

[109] Robin Rombach et al. "High-Resolution Image Synthesis with Latent Diffusion Models". In: *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2022), pp. 10674–10685 (cit. on p. 1).

[110] Felipe Romero Moreno. "Generative AI and Deepfakes: A Human Rights Approach to Tackling Harmful Content". In: *International Review of Law, Computers & Technology* 38.3 (Sept. 2024), pp. 297–326 (cit. on p. 2).

[111] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. "Learning Representations by Back-Propagating Errors". In: *Nature* 323.6088 (Oct. 1986), pp. 533–536 (cit. on pp. 9, 14).

[112] C. Runge. "Ueber die numerische Aufloesung von Differentialgleichungen". In: *Mathematische Annalen* 46.2 (June 1895), pp. 167–178 (cit. on p. 58).

[113] *SAM 2: Segment Anything in Images and Videos — Research - AI at Meta*. https://ai.meta.com/research/publications/sam-2-segment-anything-in-images-and-videos/ (cit. on p. 61).

[114] Jörg Sander, Bob D. De Vos, and Ivana Išgum. "Autoencoding Low-Resolution MRI for Semantically Smooth Interpolation of Anisotropic MRI". In: *Medical Image Analysis* 78 (May 2022), p. 102393 (cit. on p. 49).

[115] Benoît Sauty and Stanley Durrleman. "Progression Models for Imaging Data with Longitudinal Variational Auto Encoders". In: *Medical Image Computing and Computer Assisted Intervention – MICCAI 2022*. Ed. by Linwei Wang et al. Vol. 13431. Cham: Springer Nature Switzerland, 2022, pp. 3–13 (cit. on pp. 27, 58).

[116] Julian Schön et al. "Explicit Temporal Embedding in Deep Generative Latent Models for Longitudinal Medical Image Synthesis". In: (2023) (cit. on p. 26).

[117] E. Schrödinger. "Sur La Théorie Relativiste de l'électron et l'interprétation de La Mécanique Quantique". In: 1932 (cit. on p. 40).

[118] Bradley Segal et al. "Bridging the Generalisation Gap: Synthetic Data Generation for Multi-Site Clinical Model Validation". In: (2025) (cit. on p. 28).

[119] W. Paul Segars et al. "Application of the 4D XCAT Phantoms in Biomedical Imaging and Beyond". In: *IEEE transactions on medical imaging* 37.3 (Mar. 2018), pp. 680–692 (cit. on p. 28).

[120] Chengzhi Shen et al. "Spatiotemporal Representation Learning for Short and Long Medical Image Time Series". In: *Medical Image Computing and Computer Assisted Intervention – MICCAI 2024*. Ed. by Marius George Linguraru et al. Vol. 15011. Cham: Springer Nature Switzerland, 2024, pp. 656–666 (cit. on p. 26).

[121] Xingjian SHI et al. "Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting". In: *Advances in Neural Information Processing Systems*. Vol. 28. Curran Associates, Inc., 2015 (cit. on pp. 22, 23, 87, 88).

[122] Yang Song et al. "Score-Based Generative Modeling through Stochastic Differential Equations". In: *ArXiv* (Nov. 2020) (cit. on pp. 33, 34).

[123] Benjamin Stark, Catherine Johnson, and Gregory Andrew Roth. "Global Prevalence of Coronary Artery Disease: An Update from the Global Burden of Disease Study". In: *JACC* 83.13_Supplement (Apr. 2024), pp. 2320–2320 (cit. on p. 49).

[124] Yannick Suter et al. "The LUMIERE Dataset: Longitudinal Glioblastoma MRI with Expert RANO Evaluation". In: *Scientific Data* 9.1 (Dec. 2022), p. 768 (cit. on pp. 51, 93).

[125] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. "Sequence to Sequence Learning with Neural Networks". In: *Advances in Neural Information Processing Systems*. Vol. 27. Curran Associates, Inc., 2014 (cit. on pp. 23, 44).

[126] Cheng Tan et al. "OpenSTL: A Comprehensive Benchmark of Spatio-Temporal Predictive Learning". In: (2023) (cit. on pp. 86, 87, 89, 115).

[127] Cheng Tan et al. "SimVPv2: Towards Simple yet Powerful Spatiotemporal Predictive Learning". In: *IEEE Transactions on Multimedia* (2025), pp. 1–15 (cit. on p. 22).

[128] Chen Tang et al. "Deep Reinforcement Learning for Robotics: A Survey of Real-World Successes". In: *Annual Review of Control, Robotics, and Autonomous Systems* 8.Volume 8, 2025 (May 2025), pp. 153–188 (cit. on p. 1).

[129] Song Tang et al. "SwinLSTM: Improving Spatiotemporal Prediction Accuracy Using Swin Transformer and LSTM". In: *2023 IEEE/CVF International Conference on Computer Vision (ICCV)* (Oct. 2023), pp. 13424–13433 (cit. on pp. 87, 88).

[130] *The Global Risks Report 2024: Insight Report*. 19th ed. Geneva: World Economic Forum, 2024 (cit. on p. 2).

[131] Alexander Tong. *Atong01/Conditional-Flow-Matching*. June 2025 (cit. on pp. 70, 94).

[132] Alexander Tong et al. *Improving and Generalizing Flow-Based Generative Models with Minibatch Optimal Transport*. Mar. 2024. arXiv: 2302.00482 [cs] (cit. on p. 94).

[133] Alexander Tong et al. *Simulation-Free Schrödinger Bridges via Score and Flow Matching*. Mar. 2024. arXiv: 2307.03672 [cs] (cit. on pp. 28, 40, 75, 94).

[134] Maria Trigka and Elias Dritsas. "A Comprehensive Survey of Deep Learning Approaches in Image Processing". In: *Sensors* 25.2 (Jan. 2025), p. 531 (cit. on p. 1).

[135] Wieke M. van Oostveen and Elizabeth C. M. de Lange. "Imaging Techniques in Alzheimer's Disease: A Review of Applications in Early Diagnosis and Longitudinal Monitoring". In: *International Journal of Molecular Sciences* 22.4 (Feb. 2021), p. 2110 (cit. on p. 48).

[136] László Vancsura, Tibor Tatay, and Tibor Bareith. "Navigating AI-Driven Financial Forecasting: A Systematic Review of Current Status and Critical Research Gaps". In: *Forecasting* 7.3 (Sept. 2025), p. 36 (cit. on p. 2).

[137] Ashish Vaswani et al. "Attention Is All You Need". In: *Advances in Neural Information Processing Systems*. Vol. 30. Curran Associates, Inc., 2017 (cit. on pp. 10, 11, 22).

[138] Vikram Voleti, Alexia Jolicoeur-Martineau, and Christopher Pal. "MCVD: Masked Conditional Video Diffusion for Prediction, Generation, and Interpolation". In: (2022) (cit. on p. 22).

[139] A. Waibel, Hanazawa G. Hinton, and Ic Shikano Ic. "Phoneme Recognition: Neural Networks Vs". In: 1988 (cit. on p. 10).

[140] Gefei Wang et al. "Deep Generative Learning via Schrödinger Bridge". In: *International Conference on Machine Learning*. June 2021 (cit. on p. 75).

[141] Yunbo Wang et al. "Memory in Memory: A Predictive Neural Network for Learning Higher-Order Non-Stationarity From Spatiotemporal Dynamics". In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2019), pp. 9146–9154 (cit. on pp. 87, 88).

[142] Yunbo Wang et al. "PredRNN: Recurrent Neural Networks for Predictive Learning Using Spatiotemporal LSTMs". In: *Advances in Neural Information Processing Systems* 30 (2017) (cit. on pp. 22, 23).

[143] Yunbo Wang et al. "PredRNN++: Towards A Resolution of the Deep-in-Time Dilemma in Spatiotemporal Predictive Learning". In: () (cit. on p. 22).

[144] Dirk Weissenborn, Oscar Täckström, and Jakob Uszkoreit. "Scaling Autoregressive Video Models". In: *ArXiv* (June 2019) (cit. on p. 22).

[145] Patrick Y. Wen et al. "Updated Response Assessment Criteria for High-Grade Gliomas: Response Assessment in Neuro-Oncology Working Group". In: *Journal of Clinical Oncology: Official Journal of the American Society of Clinical Oncology* 28.11 (Apr. 2010), pp. 1963–1972 (cit. on p. 51).

[146] P. Werbos. "Beyond Regression : "New Tools for Prediction and Analysis in the Behavioral Sciences". In: 1974 (cit. on p. 14).

[147] Mingxing Xu et al. "Spatial-Temporal Transformer Networks for Traffic Flow Forecasting". In: *ArXiv* (Jan. 2020) (cit. on p. 22).

[148] Shirong Xu, Will Wei Sun, and Guang Cheng. "Utility Theory of Synthetic Data Generation". In: (2023) (cit. on pp. 27, 117).

[149] Yiwen Xu et al. "Deep Learning Predicts Lung Cancer Treatment Response from Serial Medical Imaging". In: *Clinical Cancer Research* 25.11 (June 2019), pp. 3266–3275 (cit. on p. 25).

[150] Wilson Yan et al. "VideoGPT: Video Generation Using VQ-VAE and Transformers". In: *ArXiv* (Apr. 2021) (cit. on p. 22).

[151] Xi Ye and Guillaume-Alexandre Bilodeau. "STDiff: Spatio-temporal Diffusion for Continuous Stochastic Video Prediction". In: (2023) (cit. on p. 22).

[152] Melike Nur Yeğin and Mehmet Fatih Amasyalı. "Generative Diffusion Models: A Survey of Current Theoretical Developments". In: *Neurocomputing* 608 (Dec. 2024), p. 128373 (cit. on pp. 1, 2).

[153] Dan Yoon et al. "Latent Diffusion Model-Based MRI Superresolution Enhances Mild Cognitive Impairment Prognostication and Alzheimer's Disease Classification". In: *NeuroImage* 296 (Aug. 2024), p. 120663 (cit. on p. 26).

[154] Jee Seok Yoon et al. "SADM: Sequence-Aware Diffusion Model for Longitudinal Medical Image Generation". In: *Information Processing in Medical Imaging Series Title: Lecture Notes in Computer Science*. Ed. by Alejandro Frangi et al. Vol. 13939. Cham: Springer Nature Switzerland, 2023, pp. 388–400 (cit. on pp. 21, 24, 25, 49, 64, 86, 89, 95, 122).

[155] Mohammed Yousufuddin and Nathan Young. "Aging and Ischemic Stroke". In: *Aging* 11.9 (May 2019), pp. 2542–2544 (cit. on p. 50).

[156] Ling Zhang et al. "Spatio-Temporal Convolutional LSTMs for Tumor Growth Prediction by Learning 4D Longitudinal Patient Data". In: *IEEE Transactions on Medical Imaging* 39.4 (Apr. 2020), pp. 1114–1126 (cit. on p. 25).

[157] Richard Zhang et al. *The Unreasonable Effectiveness of Deep Features as a Perceptual Metric*. Apr. 2018. arXiv: `1801.03924 [cs]` (cit. on p. 13).

[158] Xi Zhang et al. *Trajectory Flow Matching with Applications to Clinical Time Series Modeling*. Feb. 2025. arXiv: `2410.21154 [cs]` (cit. on p. 73).

[159] Jingyuan Zhao et al. "Autonomous Driving System: A Comprehensive Survey". In: *Expert Systems with Applications* 242 (May 2024), p. 122836 (cit. on p. 1).

[160] Xuanru Zhou et al. "Generative Artificial Intelligence in Medical Imaging: Foundations, Progress, and Clinical Translation". In: () (cit. on p. 2).

[161] Jiayuan Zhu et al. *Medical SAM 2: Segment Medical Images as Video via Segment Anything Model 2*. Dec. 2024. arXiv: `2408.00874 [cs]` (cit. on p. 61).

[162] Zihao Zhu et al. "LoCI-DiffCom: Longitudinal Consistency-Informed Diffusion Model for 3D Infant Brain Image Completion". In: *Medical Image Computing and Computer Assisted Intervention – MICCAI 2024*. Ed. by Marius George Linguraru et al. Cham: Springer Nature Switzerland, 2024, pp. 249–258 (cit. on p. 26).

[163] Zia-Ur-Rehman et al. "Recent Advancements in Neuroimaging-Based Alzheimer's Disease Prediction Using Deep Learning Approaches in e-Health: A Systematic Review". In: *Health Science Reports* 8.5 (May 2025), e70802 (cit. on p. 48).

7.1 Contributions

7.1.1 Submitted

[1] **Disch, Nico Albert** and Kirchhoff, Yannick and Peretzke, Robin and Rokuss, Maximilian and Roy, Saikat and Ulrich, Constantin and Zimmerer, David and Maier-Hein, Klaus *Temporal Flow Matching for Learning Spatio-Temporal Trajectories in 4D Longitudinal Medical Imaging*, arXiv
Contains most of the Temporal Flow Matching method 4.3.

[2] **Disch, Nico Albert** and Saikat Roy, Constantin Ulrich, Yannick Kirchhoff, Maximilian Rouven Rokuss, Robin Peretzke, David Zimmerer, Klaus Maier-Hein *CRONOS: Continuous time reconstruction for 4D medical longitudinal series* , Submitted to ICLR
Constitutes the continuous extension to TFM 4.4.

7.1.2 Accepted Papers

[1] **Disch, Nico Albert** and Kovacs, Balint and Ulrich, Constantin and Peretzke, Robin and Roy, Saikat and Rokuss, Maximilian Rouven and Kirchhoff, Yannick and Zimmerer, David and Maier-Hein, Klaus *Applying Longitudinal Augmentation and Data Generation (LAUGEN) in Medical Imaging*, Synthetic Data for Computer Vision Workshop @ CVPR 2025.
Constitutes the longitudinal augmentation chapter 4.2.

[2] **Disch, Nico Albert** and Peretzke, Robin and Roy, Saikat and Ulrich, Constantin and Zimmerer, David and Stiefelhagen, Rainer and Kleesiek, Jens and Maier-Hein, Klaus *Back to the Future: Challenges of Sparse and Irregular Medical Image Time Series*, Medical Image Computing and Computer Assisted Intervention – MICCAI 2024 Workshops
Contains the NP backbones with Ellipses experiments and ADNI 5.1.

Method	Runtime [10^3s]	Max Memory [GB]
ViViT	16	9.0
NODE+LSTM	60	6.6
SimVP	20	5.2
ConvLSTM	14	3.5
TFM discrete	18	7.7
TFM continuous	12	7.0

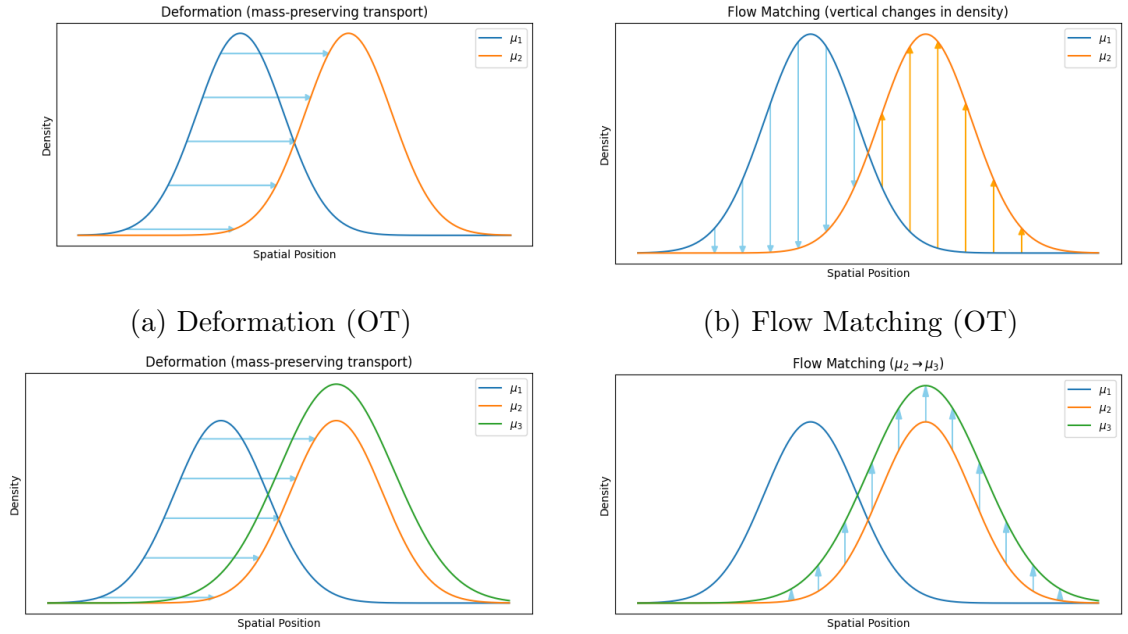
Table 7.1: Wall-clock runtime in thousands of seconds and maximum memory usage for a single run.

7.2 Conditional U-Net Mechanics.

Our implementation of the U-Net backbone follows the conditional architecture of the Flow Matching library. The model consists of two main components: an encoder (plus a bottleneck), and a decoder (expansive path), with skip connections between encoder and decoder stages. At each resolution, the feature width is scaled according to the multipliers $[C_1, C_2, C_3]$, relative to the base width C_0. **Timestep embeddings.** Each input timepoint t is mapped to a high-dimensional embedding using sinusoidal features. This raw embedding is passed through a two-layer MLP with SiLU activation to obtain a time embedding vector of size $2 * C_i$. Per-level projections of this embedding are injected additively into residual blocks. If class or image conditioning is present, it is combined with the time embedding. We do not actively use the class conditioning, but it is nice to keep that in mind. **Residual blocks.** Each block consists of GroupNorm, SiLU activation, and 3×3×3 convolutions. The time embedding is added channel-wise after the first normalization, through a FiLM-like scale–shift transformation. Residual connections use 1×1 convolutions. Dropout may be applied before the final convolution. Additionally, a temporal FiLM adapter modulates features according to the global time embedding. We added an optional spatial feature adapter. **Down- and upsampling.** Downsampling between encoder stages is performed by residual downsampling blocks. Upsampling in the decoder mirrors this process, using nearest-neighbor. At each decoder stage, the upsampled feature map is concatenated with the corresponding encoder skip connection, fused with a 1×1 convolution, and processed by residual blocks with conditioning. **Attention blocks.** At selected resolutions, self-attention is inserted after residual blocks. Features are normalized, projected into Q, K, V, and processed with multi-head attention. The output is projected back and added residually. **Decoder and output.** The decoder inverts the channel schedule $C_2 \to C_1 \to C_0$. Each stage applies upsampling, skip fusion, residual blocks with time conditioning, and optional attention. After the final decoder stage, GroupNorm and SiLU are applied, fol-

lowed by a 3×3×3 convolution projecting to the desired number of output channels.
Normalization and activation. GroupNorm is used throughout the network to accommodate small batch sizes, and SiLU serves as the activation function. All convolutions use "same" padding to preserve spatial alignment.
Summary UNet Architecture In total, the architecture follows a U-shaped encoder–decoder with skip connections, residual and attention blocks, FiLM-style time conditioning, and optional spatial adapters for context. This design enables efficient spatio-temporal modeling of volumetric medical data, with explicit temporal conditioning and multi-scale feature fusion.

7.3 Qualitative Results



(a) Deformation (OT)

(b) Flow Matching (OT)

(c) Deformation (UOT)

(d) Flow Matching (UOT)

Figure 7.1: **Comparison and motivation** between deformation-based transport and Flow Matching. The upper row illustrates the classical optimal transport framework, where mass is preserved during deformation. The lower row illustrates a source term that allows for mass change, either in the deformation framework or in Flow Matching. μ_1 signifies the initial distribution, μ_3 the target distribution. μ_2 is an intermediate distribution, which is closest to μ_3 via OT. (a) **Deformation (mass-preserving transport):** Classical optimal transport corresponds to a mass-preserving deformation field. Probability mass is displaced horizontally in space, such that the initial distribution μ_1 is mapped into the target distribution μ_2. (b) **Flow Matching (vertical transport at fixed x):** At each spatial position x, the density is interpolated vertically from $\mu_1(x)$ to $\mu_2(x)$. (c) **Deformation with mass change:** Extending the deformation formulation with a source term allows both displacement of existing mass and the creation or removal of density. (d) + **Flow Matching with mass change:** This panel shows how to combine Flow Matching after having applied a deformation field. The overall arrow sizes are lower, and

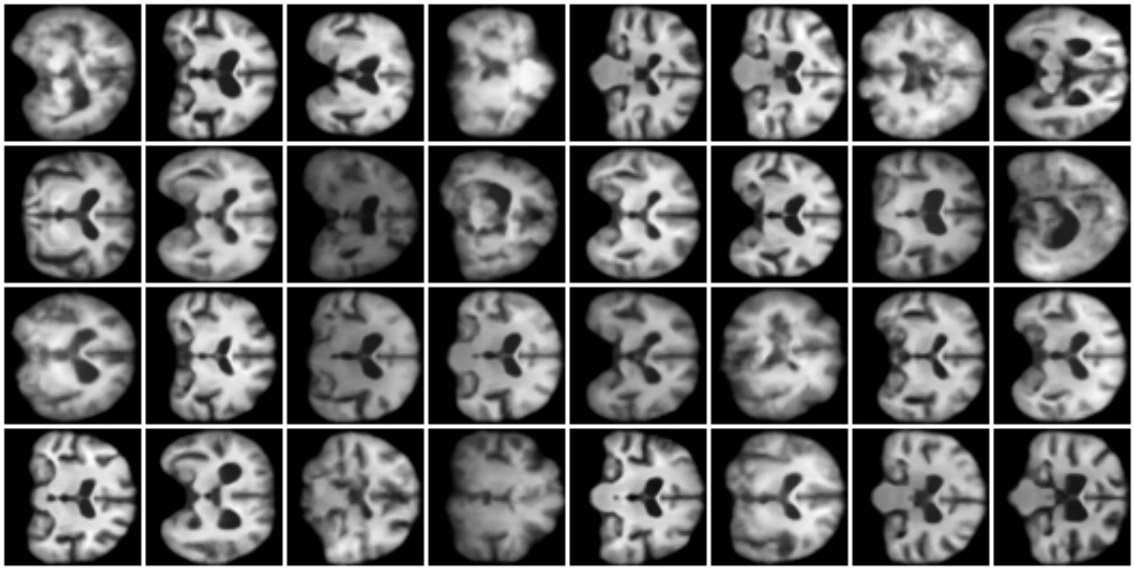


Figure 7.2: **Qualitative Results for Mamba on the ADNI dataset**

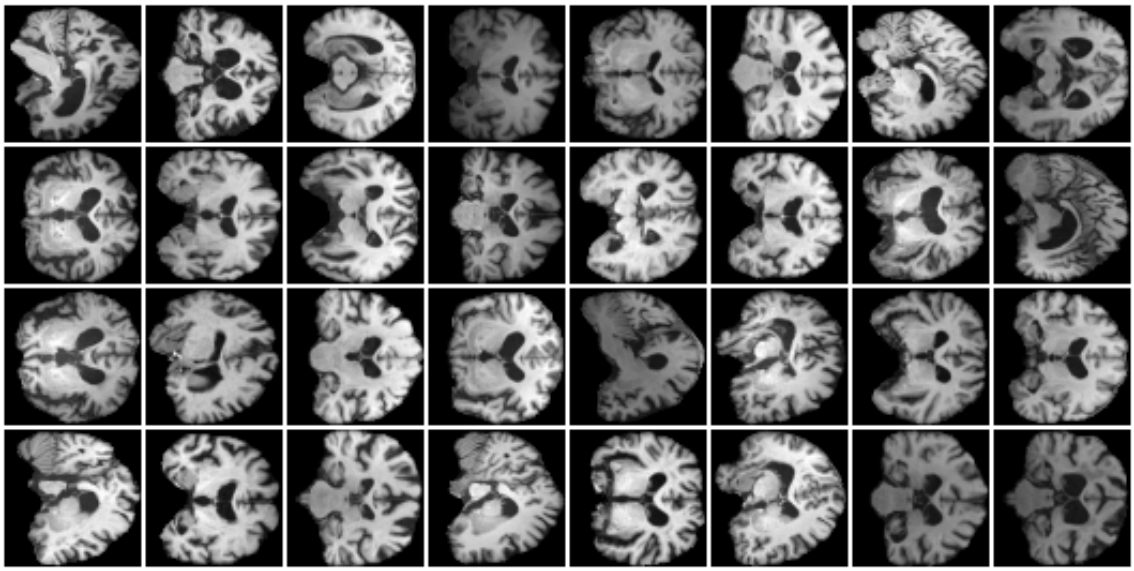


Figure 7.3: **Target for Mamba on the ADNI dataset**

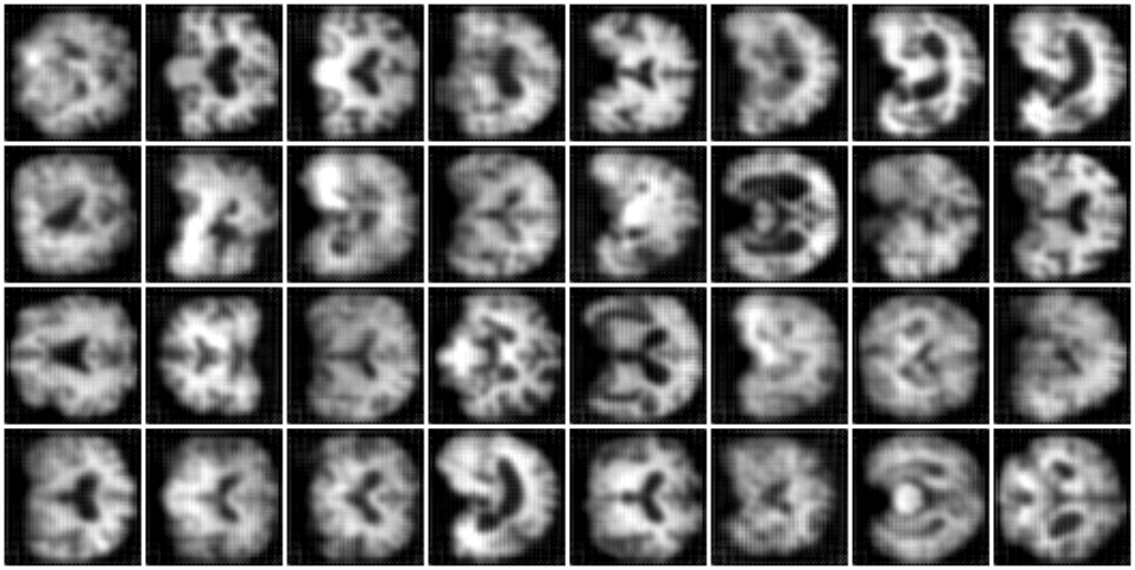


Figure 7.4: **Qualitative Results for ASP on the ADNI dataset**

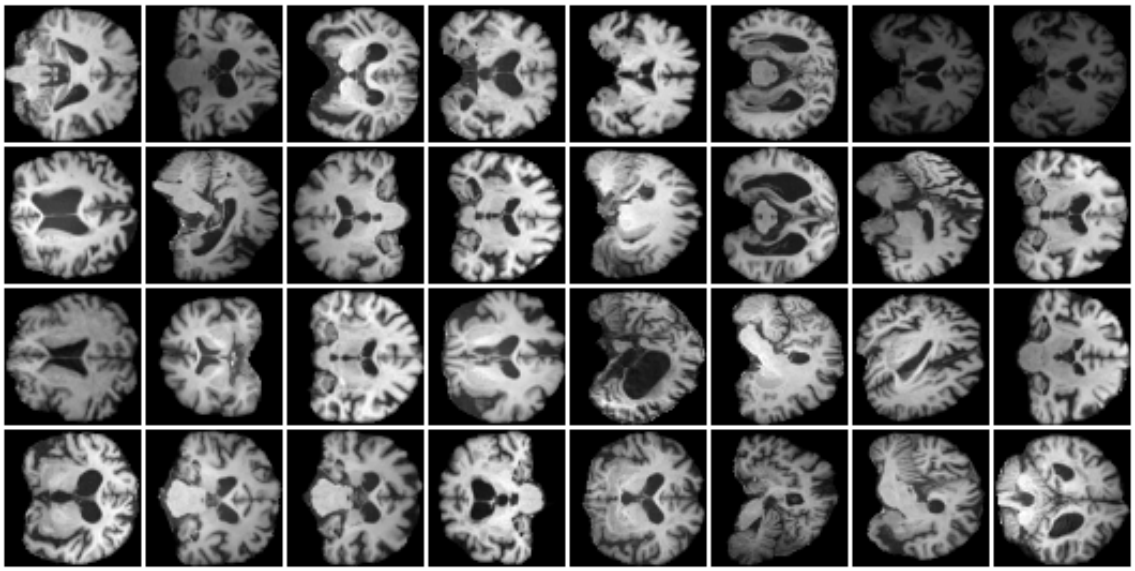


Figure 7.5: **Target for ASP on the ADNI dataset**