Longitudinal Flow Matching for Trajectory Modeling

Mohammad Mohaiminul Islam
QurAI, Univ. of Amsterdam

Thijs P. Kuipers
BMEP, Amsterdam UMC

Sharvaree Vadgama
AMLab, Univ. of Amsterdam

Coen de Vente
QurAI, Univ. of Amsterdam

Afsana Khan
Maastricht University

Clara I. Sánchez
QurAI, Univ. of Amsterdam

Erik J. Bekkers
AMLab, Univ. of Amsterdam

Open-source code: `github.com/niazoys/longitudinal-flow-matching`

Abstract

Generative models for sequential data often struggle with sparsely sampled and high-dimensional trajectories, typically reducing the learning of dynamics to pairwise transitions. We propose *Interpolative Multi-Marginal Flow Matching* (IMMFM), a framework that learns continuous stochastic dynamics jointly consistent with multiple observed time points. IMMFM employs a quadratic interpolation path as a smooth target for flow matching and jointly optimizes drift and a data-driven diffusion coefficient, supported by a theoretical condition for stable learning. This design captures intrinsic stochasticity, handles irregular sparse sampling, and yields subject-specific trajectories. Experiments on synthetic benchmarks and real-world longitudinal neuroimaging datasets show that IMMFM outperforms existing methods in both forecasting accuracy and further downstream tasks.

1 INTRODUCTION

Modeling trajectories of high-dimensional states is highly relevant in many domains, including climate and geophysical systems (Kidger et al., 2020), video generation (Voleti et al., 2021; Bossa and Sahli, 2023; Dang et al., 2023), and longitudinal biomedical imaging (Lachinov et al., 2023). Continuous-time generative modeling has advanced rapidly in recent years. Neural ODEs (Chen et al., 2018) enable end-to-end learning of continuous dynamics, while diffusion models (Shi et al., 2023; Liu et al., 2023), flow matching (Tong et al., 2023), and Schrödinger Bridge methods (Hamdouche et al., 2023) support transitions between complex distributions. Neural operator methods also allow for high-resolution data-driven solutions to PDE systems at scale (Yang et al., 2023b). Together, these developments make modeling complex high-dimensional trajectories increasingly feasible.

Recent advances such as SDE Matching (Bartosh et al., 2025) enable scalable, simulation-free training of latent SDEs, but modeling full high-dimensional trajectories remains challenging, particularly for very high-dimensional observations such as high-resolution images. Existing methods offer partial solutions: some rely on pairwise transitions (Liu et al., 2025), while others use rolling-window strategies (Zhang et al., 2024) that can lack temporal alignment and are mainly validated on low-dimensional periodic data. More broadly, many continuous generative frameworks remain formulated as two-marginal or pairwise transport problems. Applied naively, such pairwise approaches model a trajectory as independent transports for each segment, missing global constraints and dependencies across the full sequence. Extending them to a multi-marginal setting helps address these limitations (Pass, 2015).

Concurrent to our work, (Lee et al., 2025) and (Rohbeck et al., 2025) proposed SDE- and ODE-based multi-marginal flow matching methods relying on spline fitting or piecewise linear interpolation to construct conditional probability paths. However, piecewise linear interpolation can produce non-smooth behavior near intermediate marginals, while spline fitting on high-dimensional data such as images can be expensive and unreliable (Hastie et al., 2009; Wahba, 1990), complicating accurate path learning. Thus, practical modeling of sparse, high-dimensional data remains an open challenge.

This challenge is particularly pronounced in clinical medicine, where longitudinal imaging yields repeated

views of changing anatomy, yet observations are typically high-dimensional, irregularly and sparsely sampled, and subject-specific. Accurate modeling of such trajectories could improve treatment selection, streamline follow-up schedules, and support adaptive prognosis and trial design (Caruana et al., 2015; Locascio and Atri, 2011). Traditional pipelines simplify the high-dimensional image data to scalar biomarkers or regional volumes before applying sequence models, discarding rich spatial information (Lachinov et al., 2023; Lu et al., 2024; Ruan et al., 2024; Lyu et al., 2023; Sun and Yang, 2023; Dong et al., 2023; Liu et al., 2020; Lei et al., 2020; Nguyen et al., 2023). More recent generative approaches operate directly on images but focus primarily on population-level synthesis (Wolleb et al., 2022; Chen et al., 2025; Zhan et al., 2024; Cho et al., 2025), rather than subject-conditioned trajectory modeling needed for prognosis.

To address the challenges of modeling sparse, irregular, and high-dimensional trajectories, we introduce Interpolative Multi-Marginal Flow Matching (IMMFM), designed to capture subject-specific dynamics. IMMFM reframes longitudinal trajectory modeling as a multi-marginal path learning problem. Instead of learning only pairwise transitions, IMMFM jointly learns continuous stochastic dynamics that are consistent with multiple observed time points. We propose a quadratic interpolation with a lookahead scheme that yields a smooth target vector field for flow-matching, and we learn both drift and a data-driven diffusion coefficient within a stochastic flow framework. This enables the model to capture intrinsic stochasticity and observation uncertainty. We derive a theoretical condition that supports the joint drift-diffusion learning, ensuring identifiability and stable optimization. Practically, IMMFM respects irregular sampling, enforces temporal smoothness across segments, and yields subject-specific conditional trajectories rather than population synthetic samples.

We summarize our contributions as follows.

- We propose quadratic interpolation with a lookahead scheme that yields a smooth and tractable target for flow matching over multiple observations.
- We propose to learn a data-driven diffusion coefficient and prove the necessary theoretical condition for joint drift-diffusion optimization.
- IMMFM effectively models low- and high-dimensional trajectories, outperforming baselines on synthetic and longitudinal neuroimaging datasets.

2 BACKGROUND

Consider $z_{0:M} = (x_{t_0}, \ldots, x_{t_M})$ a sequence of observed data or their latent representations acquired at non-uniform and often sparse time points $t_0 < t_1 < \cdots < t_M \in [0,1]$, with $x_i \in \mathbb{R}^D$. Note that we do not require that all realizations of the time series are measured at identical, aligned time points; rather, we assume that each observation is sampled from an underlying continuous process over time. Let ρ_i be the probability distribution of the state at time t_i. We assume that these distributions lie on a smooth manifold embedded in $\mathbb{R}^d$, where $d \ll D$. The objective is to learn a continuous probabilistic flow $p_t(x)$ over $t \in [0,1]$ such that $p_{t_i} = \rho_i$ for all i which captures the individual trajectory-specific changes. Let $v(t,x,c)$ be a Lipschitz continuous time-dependent vector field $v : [0,1] \times \mathbb{R}^d \times \mathbb{R}^e \to \mathbb{R}^d$, where $v(t,x,c)$ is the velocity of the state at time t. The velocity depends on the current position x_t and conditioning variables $c \in \mathbb{R}^e$ (e.g. static covariates, baseline measurements, and/or the state at a previous timepoint). The associated flow operator $\psi_t(v)$ then pushes the initial distribution p_0 forward to p_t. This means that if a sample $x_0 \sim p_0$, the distribution of the transformed point $x_t = \psi_t(v)(x_0)$ is p_t.

2.1 Stochastic Differential Equations (SDEs)

We consider our trajectory modeling problem as a stochastic process that can be represented as an *Itô stochastic differential equation* of the form,

$$dx_t = u_t(x_t)\, dt + g(t, x_t)\, dW_t\,, \tag{1}$$

where W_t is a standard Brownian motion in $\mathbb{R}^d$ and $g : [0,1] \times \mathbb{R}^d \to \mathbb{R}_+$ is a diffusion coefficient. Evolving an initial density p_0 under Eq. (1) produces a collection of marginal densities $\{p_t\}_{t\in[0,1]}$, i.e., the density of x_t at each time t, governed by the Fokker–Planck equation:

$$\partial_t p_t \;=\; -\nabla \cdot \big(p_t\, u_t\big) \;+\; \frac{1}{2}\,\Delta\big(g(t,x)^2\, p_t\big). \tag{2}$$

where u_t is the drift term, and g is the diffusion coefficient. The drift u_t captures the mean progression of trajectories, while the diffusion g quantifies stochastic deviations, providing a general formulation for tracing trajectories across continuous change. In the deterministic limit when stochastic effects vanish ($g \equiv 0$), this system reduces to an ODE, and the Fokker-Planck equation (Eq.2) simplifies to the continuity equation of mass transport (Gardiner, 2009).

2.2 Learning SDEs via Probability Flow and Score Matching

A fundamental problem in learning SDE (Eq. (1)) dynamics is the high computational demand of conven-

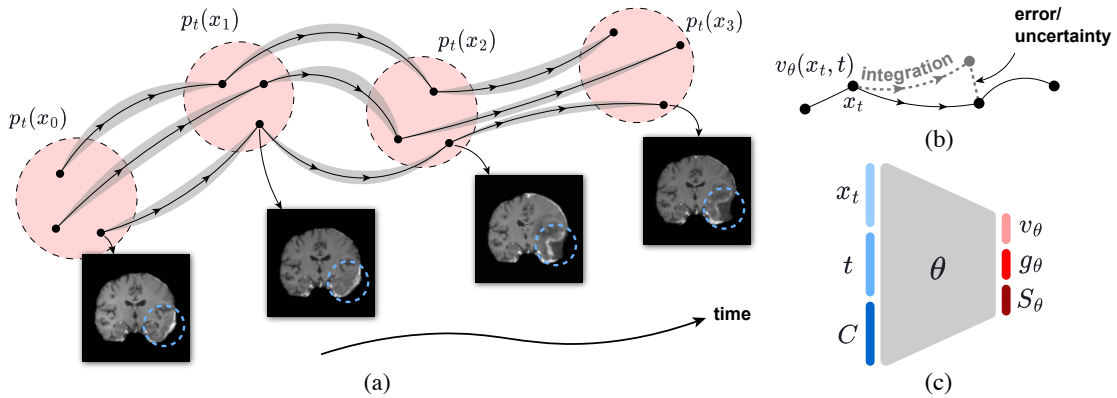

Figure 1: Overview of modeling problem and data. **(a)** Example of a sparsely and irregularly observed trajectory of disease progression over time. **(b)** IMMFM estimates positional uncertainty that informs the SDE's data-driven diffusion term. **(c)** IMMFM takes as input the position x_t, time t, and conditional variables C and predicts the velocity v_θ, diffusion term g_θ, and uncertainty S_θ.

tional training paradigms, often requiring the simulation of numerous full trajectories to estimate gradients (see also Kidger et al., 2021; Ryder et al., 2018; Anonymous, 2024; Li et al., 2020; Tzen and Raginsky, 2019). This simulation process can be prohibitively slow, particularly in high-dimensional settings. An effective alternative is to leverage the fundamental connection between the SDE and its deterministic counterpart: the *probability flow ODE*. This connection provides an insight central to score-based generative modeling and subsequent samplers (Song et al., 2021; Lu et al., 2022; Karras et al., 2022; Li et al., 2024; Cai et al., 2025). We therefore adopt the simulation-free training framework from (Tong et al., 2024) to learn the drift of the probability flow ODE. This ODE is governed by the continuity equation and shares the exact same marginals p_t as the SDE (Eq. 1):.

$$\partial_t p_t = -\nabla \cdot (p_t u_t^\circ), \tag{3}$$

where u_t° is the drift of the probability flow. By combining the Fokker-Planck (Eq. (2)) and continuity (Eq. (3)) equations, we obtain the following identity (cf.(Tong et al., 2023, Eq. 5)):

$$u_t(x_t) = \underbrace{v_t(x_t) + \frac{1}{2}\nabla g(t, x_t)^2}_{\text{Prob. flow drift } u_t^\circ(x_t)} + \frac{g(t, x_t)^2}{2}\nabla \log p_t(x_t). \tag{4}$$

This identity is key to a simulation-free approach, as it decomposes the SDE drift u_t into two components that can be learned independently. The first component is the probability flow drift u_t° that captures the average velocity of the system's evolution[1] The second component is the score function $\nabla \log p_t$ that provides a corrective force to ensure the generated trajectories remain plausible. We utilize this decomposition, and learn the SDE drift by training two separate time-dependent neural networks to approximate both the flow drift $v_\theta(t, x) \approx u_t^\circ(x)$ and the score function $s_\theta(t, x) \approx \nabla \log p_t(x)$. Then the ideal training objective would be to minimize the true marginals p_t:

$$\mathcal{L}_{\text{SDE}}(\theta) = \mathbb{E}_{\substack{t\sim\mathcal{U}(0,1) \\ x\sim p_t}}\big[\|v_\theta(t, x) - u^\circ(t, x)\|_2^2 + \lambda(t)^2\|s_\theta(t, x) - \nabla \log p_t(x)\|_2^2\big], \tag{5}$$

However, this objective is intractable, as the true marginals p_t, the probability flow drift $u^\circ(t, x)$, and the score $\nabla \log p_t(x)$ are all unknown. Instead, we use the tractable conditional objective (Tong et al., 2023; Lipman et al., 2022), where targets can be derived analyically for constructed conditional paths $p_t(x \mid z)$:

$$\mathcal{L}_{\text{CSDE}}(\theta) = \mathbb{E}_{\substack{t\sim\mathcal{U}(0,1) \\ z\sim q \\ x\sim p_t(x|z)}}\Bigg[\underbrace{\big\|v_\theta(t, x, c) - u_t^\circ(x \mid z)\big\|_2^2}_{\text{Cond. Flow Matching}} + \lambda(t)^2\underbrace{\big\|s_\theta(t, x, c) - \nabla_x \log p_t(x \mid z)\big\|_2^2}_{\text{Cond. Score Matching}}\Bigg]. \tag{6}$$

The specific analytical forms of target velocity $u_t^\circ(x \mid z)$ and score $\nabla_x \log p_t(x \mid z)$ depend on how the conditional probability path is constructed.

3 METHOD

In this section, we present Interpolative Multi-Marginal Flow Matching (IMMFM). As established in 2.2, our SDE learning strategy relies on the construction of conditional probability path $p_t(x \mid z)$ and score

[1] In our formulation, we *absorb* the term $\frac{1}{2}\nabla(g(t, x)^2)$ into the learnable drift u_t°. This avoids expensive gradient computation w.r.t g during inference while leaving the training objective unchanged.

$\nabla_x \log p_t(x \mid z)$ defined in Eq. (6). An overview of our method is presented in Fig. 1.

3.1 Conditional Probability Path Construction

Conditional probability paths are commonly defined as Gaussian distributions with time-varying mean and covariance (Lipman et al., 2022). We follow this approach and define for each trajectory $z \sim q$ the conditional probability path p_t:

$$p_t(x \mid z) = \mathcal{N}\big(x \mid \mu_t(z),\ \sigma^2(t)I\big)\,, \tag{7}$$

where $\mu_t(z) : [0,1] \times Z \to \mathbb{R}^d$ and $\sigma(t) : [0,1] \to \mathbb{R}_+$ are the mean and standard deviation, respectively. Eq 7 induces a unique conditional velocity (cf. (Lipman et al., 2022, Thm. 3)):

$$u_t^\circ(x \mid z) = \frac{\sigma'(t)}{\sigma(t)}(x - \mu_t(z)) + \mu_t'(z). \tag{8}$$

In the multi-marginal case, the conditional path is often a simple linear interpolation between the two consecutive points in a trajectory. However, naively chaining these linear paths in a multi-marginal setting leads to a trajectory with discontinuous velocities, which fails to capture the underlying smooth dynamics of the system. We address this by generalizing the path construction, introducing a quadratic term that ensures a smoother transition in velocity. We condition this term on the subsequent trajectory segment to yield probability paths with greater dynamic fidelity. Crucially, this improved consistency is achieved *while preserving the analytical tractability*, which is essential for a simulation-free flow matching objective. We choose $\sigma(t)$ in such a way as to provide the path with a variance structure that is consistent with a Brownian bridge: zero at the observed endpoints (t_i, t_{i+1}) and maximum at midway between them. Formally, we define $\mu_t(z)$ and $\sigma(t)$ as:

$$\begin{aligned}\mu_t(z) &= x_{t_i} + v_i(t - t_i) + \tfrac{1}{2}\alpha_t(v_i - v_{i+1})(t - t_i)\\ \sigma(t) &= \sigma_0(t - t_i)\,\alpha_t\end{aligned} \tag{9}$$

where t is strictly between (t_i, t_{i+1}), $v_i = \frac{x_{t_{i+1}} - x_{t_i}}{t_{i+1} - t_i}$ is the velocity of the $[t_i, t_{i+1}]$ segment, and $\alpha_t = \frac{t_{i+1} - t}{t_{i+1} - t_i}$ is a time-dependent blending coefficient. This choice of $\mu_t(z)$ and $\sigma(t)$ leads to the associated conditional velocity $u_t^\circ(x|z)$ by substituting their respective derivatives into the general form of Eq. (8). This yields our *blended velocity field*:

$$\begin{aligned}u_t^\circ(x \mid z) = v_i + \frac{1}{2}(v_i - v_{i+1})(2\alpha_t - 1)\\ + \frac{\sigma'(t)}{\sigma(t)}\big(x - \mu_t(z)\big),\end{aligned} \tag{10}$$

where $\mu_t'(z)$ and $\sigma'(t)$ are the time derivatives of $\mu_t(z)$ and $\sigma(t)$, respectively. See Appendix A.1 for the full derivation. The corresponding conditional score function that is required for defining the full conditional SDE drift via Eq. (4), is analytically tractable for this Gaussian path:

$$\nabla_x \log p_t(x \mid z) = \frac{\mu_t(z) - x}{\sigma^2(t)}\,. \tag{11}$$

These expressions for the blended velocity field in Eq. (10) and the conditional score in Eq. (11) provide the analytically tractable targets for the neural networks v_θ and s_θ in our learning objective, Eq. (6).

3.2 Uncertainty as a Learned Diffusion Coefficient

With the target velocity $u_t^\circ(x \mid z)$ and score $\nabla_x \log p_t(x \mid z)$ of the SDE's drift term now fully specified, the final component of our model is the diffusion coefficient, $g(t, x_t, c)$. We learn the two components of the SDE drift as separate, independent functions. Because these components are learned independently of any specific diffusion schedule, they can be recombined at inference time with an *arbitrary* diffusion coefficient $g(t, x_t, c)$ to form a valid SDE (Tong et al., 2023). We use this flexibility and learn a data-driven diffusion coefficient g_θ[2]. More specifically, we let the stochasticity of the model reflect its predictive confidence by matching the diffusion g_θ^2 to the squared predictive error:

$$\begin{aligned}\mathcal{L}_{\text{uncertainty}}(\theta) = \mathbb{E}_{t,z,x}\Bigg[\Bigg\|g_\theta(t, x_t, c)^2 -\\ \underbrace{\|x_i + (t_{i+1} - t_i)u_\theta(t, x_t, c) - x_{t_{i+1}}\|_2^2}_{\text{Squared error of predictive construction}}\Bigg\|_2^2\Bigg].\end{aligned} \tag{12}$$

3.3 Training Objective

We formalize the overall training objective of our model, which follows a joint formulation of the primary conditional SDE objective from Eq. (6) and the uncertainty objective from Eq. (12):

$$\mathcal{L}_{\text{IMMFM}}(\theta) = \mathcal{L}_{\text{CSDE}}(\theta) + \beta\mathcal{L}_{\text{uncertainty}}(\theta), \tag{13}$$

where β is a small positive weight. This composite objective is designed for the joint optimization of the drift (v_θ), score (s_θ), and diffusion (g_θ) components.

We now establish the theoretical basis of our training objective. We begin with a key lemma regarding the residual of the drift approximation.

[2]In practice, g_θ outputs one scalar per dimension, i.e., a diagonal diffusion; Eq. 1 and Eq. 2 show the scalar form for simplicity.

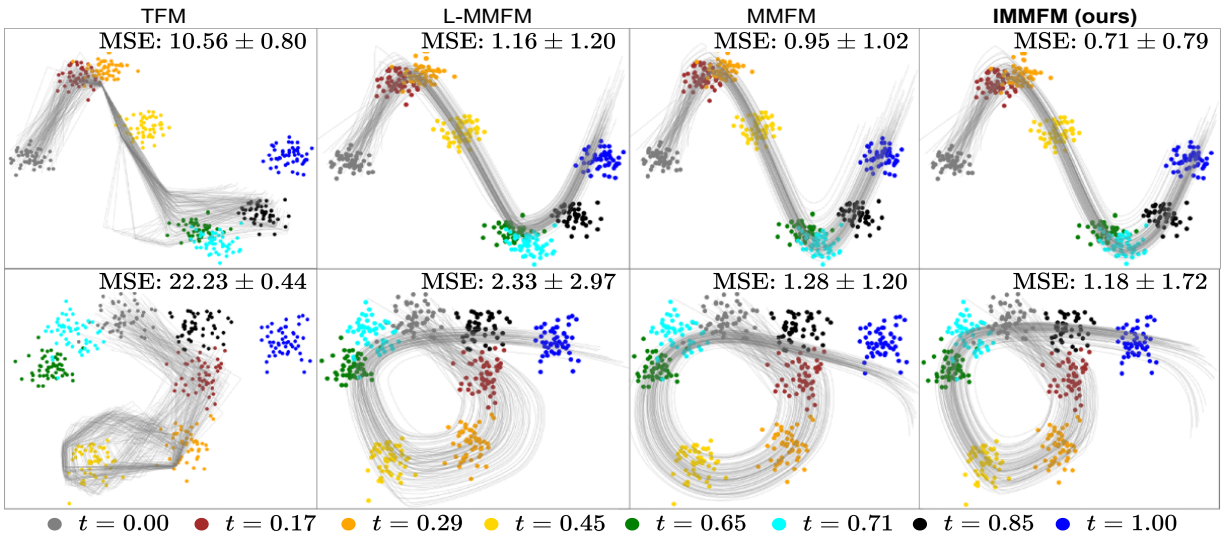

Figure 2: Trajectories on synthetic S-shaped (top row) and σ-shaped (bottom row) Gaussian datasets. Colored dots show subsets of training samples, and grey lines show the predicted trajectories. From left to right: TFM (Zhang et al., 2024), L-MMFM (Rohbeck et al., 2025), MMFM (Rohbeck et al., 2025), and IMMFM (ours).

Lemma 3.1 (Zero-mean residual)**.** *Let* $\Delta t := t_{i+1} - t_i$ *and define the residual for* $t \in (t_i, t_{i+1})$ *as* $r_\theta(t, x, z) = x_i + \Delta t\, u_\theta(t, x, c) - x_{t_{i+1}}$. *If the learned drift matches the true drift, i.e.,* $u_\theta(t, x, c) = u_t(x)$, *then* $\mathbb{E}_{z|x,t}[\, r_\theta(t, x, z)\,] = 0$.

This lemma states that a perfectly learned drift implies a zero-mean residual error. This property is central to our main result:

Proposition 3.1 (Gradient equivalence at stationary points)**.** *Under regularity, realizability, and optimality conditions, every stationary point of* $\mathcal{L}_{\text{SDE}}$ *(Eq. 5) is a stationary point of* $\mathcal{L}_{\text{IMMFM}}$ *(Eq. 13).*

To provide a formal proof, we first state the necessary assumptions and establish the relevant lemma on which the proof is based.

(A1) Regularity: The probability density $p_t(x)$, the drift u_θ, and the diffusion g_θ are continuously differentiable with respect to their parameters and state variables. This permits the interchange of expectation and differentiation.

(A2) Realizability: The function class for the drift v_θ and score s_θ is sufficiently expressive to contain the true probability flow drift u_t° and score $\nabla \log p_t$.

(A3) Optimality: We consider a parameter set θ^* that is a stationary point of the SDE objective $\mathcal{L}_{\text{SDE}}$. Specifically, we assume θ^* achieves the global minimum where $v_{\theta^*} = u^\circ$ and $s_{\theta^*} = \nabla \log p_t$.

Remark on Assumptions. These conditions are standard in the analysis of neural differential equations and flow matching. **(A1)** is a technical necessity to ensure that the loss gradients are well-defined and that operations such as differentiating under the integral sign are valid; it is satisfied by using smooth activation functions (e.g., SiLU, GELU) and standard diffusion processes. **(A2)** is a standard assumption in non-parametric estimation and learning theory, positing that our neural network architecture has sufficient capacity to approximate the true dynamics. Given the universal approximation capabilities of modern deep networks, this is a reasonable theoretical idealization. **(A3)** focuses the analysis on the properties of the ideal limit of training. It allows us to decouple the *consistency* of the objective from the *optimization dynamics*, ensuring that if the optimization succeeds (finding the global minimum of the SDE loss), the uncertainty quantification is also consistent.

Proof of Lemma 3.1. The Markov property of the target SDE implies that the expected future state given the current state $x_t = x$ evolves according to the drift: $\mathbb{E}[x_{t_{i+1}} \mid x_t = x] = x + \Delta t\, u_t(x)$. Substituting this into the expectation of the residual:

$$\mathbb{E}_{z|x,t}[r_\theta] = x_i + \Delta t\, u_\theta(t, x, c) - \mathbb{E}_{z|x,t}[x_{t_{i+1}}] \quad (14)$$
$$= x + \Delta t\, u_\theta - (x + \Delta t\, u_t). \quad (15)$$

When $u_\theta = u_t$, this difference is zero. □

Proof of Proposition 3.1. The total gradient is $\nabla_\theta \mathcal{L}_{\text{IMMFM}} = \nabla_\theta \mathcal{L}_{\text{CSDE}} + \beta \nabla_\theta \mathcal{L}_{\text{uncertainty}}$. From (Tong et al., 2023, Thm 3.2), we know that $\nabla_\theta \mathcal{L}_{\text{CSDE}} = \nabla_\theta \mathcal{L}_{\text{SDE}}$. By Assumption (A3), θ^* is a

stationary point of $\mathcal{L}_{\text{SDE}}$, so $\nabla_\theta \mathcal{L}_{\text{CSDE}}(\theta^*) = 0$. It remains to show that $\nabla_\theta \mathcal{L}_{\text{uncertainty}}(\theta^*) = 0$.

The uncertainty loss for a single sample is $\ell_{\text{unc}} = \|g_\theta^2 - r_\theta^2\|_2^2$. Its gradient is:

$$\nabla_\theta \ell_{\text{unc}} = 2(g_\theta^2 - r_\theta^2)\nabla_\theta g_\theta^2 - 2g_\theta^2 \nabla_\theta r_\theta^2.$$

Taking the expectation conditioned on x, t:

$$\mathbb{E}_{z|x,t}[\nabla_\theta \ell_{\text{unc}}] = 2(g_\theta^2 - \mathbb{E}_{z|x,t}[r_\theta^2])\nabla_\theta g_\theta^2 - 2g_\theta^2 \nabla_\theta \mathbb{E}_{z|x,t}[r_\theta^2].$$

At θ^*, we have $u_{\theta^*} = u_t$ (by A2, A3 and the drift decomposition). Thus, by Lemma 3.1, $\mathbb{E}[r_{\theta^*}] = 0$. The second moment is then the variance: $\mathbb{E}[r_{\theta^*}^2] = \text{Var}(r_{\theta^*})$. The diffusion g_θ is trained to match this variance (heteroscedastic regression). At the optimum θ^*, we have $g_{\theta^*}^2 = \mathbb{E}[r_{\theta^*}^2]$. Substituting this into the gradient equation:

1. The first term vanishes because $g_{\theta^*}^2 - \mathbb{E}[r_{\theta^*}^2] = 0$.
2. The second term involves $\nabla_\theta \mathbb{E}[r_\theta^2]$. Note that $\mathbb{E}[r_\theta^2] = \text{Var}(r_\theta) + (\mathbb{E}[r_\theta])^2$. Since u_{θ^*} is the true drift (conditional expectation), it minimizes the mean squared error of the prediction. Therefore, θ^* is a stationary point of $\mathbb{E}[r_\theta^2]$ with respect to the drift parameters. Thus, $\nabla_\theta \mathbb{E}[r_\theta^2]|_{\theta^*} = 0$.

Since both terms vanish, $\nabla_\theta \mathcal{L}_{\text{uncertainty}}(\theta^*) = 0$. Consequently, $\nabla_\theta \mathcal{L}_{\text{IMMFM}}(\theta^*) = 0$. □

At the stationary point of the SDE objective, the uncertainty objective has a zero gradient; the residual is zero-mean, and the learned diffusion matches the true variance. This guarantees that the uncertainty objective introduces no bias into the learned drift and score functions, allowing both the dynamics and uncertainty to be optimized jointly without compromising the fidelity of the generated trajectories.

Training Objective in Practice. The objective in Eq (13) requires complete trajectories for supervision. However, real-world longitudinal datasets rarely provide complete trajectories, but instead consist of sparsely sampled marginal distributions $(\rho_0, \rho_1, \ldots, \rho_M)$ without explicit pairings across time. We therefore outline a general solution based on multi-marginal optimal transport, which provides a framework to construct the necessary joint distribution $q(z)$ over these sparsely sampled trajectories.

3.4 Constructing Trajectories via Optimal Transport

Multi-marginal optimal transport (MMOT) finds the most probable cost-optimal couplings between observations. We formulate MMOT under the assumption of a pairwise additive structure for the global cost (Rohbeck et al., 2025) of the trajectory:

$$C(x_{t_0}, \ldots, x_{t_M}) = \sum_{i=0}^{M-1} k(x_{t_i}, x_{t_{i+1}}), \tag{16}$$

where $k(\cdot, \cdot)$ represents the transition cost between any two sequential states and $C(\cdots)$ denotes the total cost over the entire trajectory. Under this additive cost structure, the MMOT problem decomposes into a series of independent, simpler pairwise Optimal Transport (OT) problems. The optimal coupling between each consecutive pair of observations (ρ_i, ρ_{i+1}) can therefore be found separately. We apply this pairwise OT framework to address the sparsely sampled trajectories of real-world longitudinal datasets. More specifically, the cost $k(x_{t_i}, x_{t_{i+1}})$ becomes the squared Euclidean distance $\|x_{t_i} - x_{t_{i+1}}\|_2^2$, and the minimization is performed over the *augmented empirical distributions* $\rho_i^\dagger$ and $\rho_{i+1}^\dagger$. These represent the original data mapped through a set of alignment transformations to ensure smooth and invertible spatial mappings between timepoints, such as the diffeomorphic registrations described by (Mok and Chung, 2020). This effectively turns the MMOT task into a pairwise *spatial alignment problem*, i.e., registration, where we directly register pairs of images by finding a transformation that maps one onto the other. The resulting set of pairwise transformations then defines the optimal transport plan:

$$\pi_{i,i+1}^* = \underset{\pi \in \Pi(\rho_i^\dagger, \rho_{i+1}^\dagger)}{\arg\min} \int_{\mathbb{R}^d \times \mathbb{R}^d} \|x_{t_i} - x_{t_{i+1}}\|_2^2 \, d\pi(x_{t_i}, x_{t_{i+1}}) \tag{17}$$

As a result, the OT framework is a flexible tool to account for any kind of misalignment between trajectory states, given that it can be expressed through a cost function. After solving for all the optimal pairwise plans $(\pi_{0,1}^*, \pi_{1,2}^*, \ldots)$ (Zhou and Parno, 2024), they can be combined to reconstruct the full multi-marginal distribution $q(z)$ via the following proposition (proof in Appendix A.2).

Proposition 3.2 (Diffeomorphic MMOT Decomposition)**.** *Under the pairwise additive cost structure, MMOT decomposes into a series of independent pairwise OT problems. The resulting joint coupling is:*

$$q(z) = \pi^*(x_{t_0}, \ldots, x_{t_M}) = \frac{\prod_{i=0}^{M-1} \pi_{i,i+1}^*(x_{t_i}, x_{t_{i+1}})}{\prod_{i=1}^{M-1} \rho_i^\dagger(x_{t_i})}, \tag{18}$$

preserving the intermediate augmented marginals.

4 EXPERIMENTAL SETUP

This section details our experimental setup, including the baselines, model implementation, datasets, and evaluation metrics.

Table 1: Trajectory forecasting performance, averaged over three runs and all timepoints. Results show the average over all time points, three runs, and three seeds with best results in **bold**. Reported ± values indicate inter-subject standard deviation.

Dataset	Metric	ImageFlowNet	I2SB	TFM	L-MMFM	M-MMFM	O-IMMFM	S-IMMFM	SU-IMMFM
Starmen	MSE×10 ↓	0.26±0.02	0.57±0.37	0.23±0.09	0.11±0.13	0.12±0.14	**0.09±0.01**	0.09±0.12	0.09±0.21
ADNI	PSNR ↑	36.43±1.76	35.56±1.97	36.01±4.02	36.07±2.41	36.02±2.45	37.51±2.24	37.43±2.22	**37.52±2.24**
	SSIM ↑	0.94±0.02	0.97±0.01	0.93±0.11	0.96±0.04	0.96±0.04	0.97±0.06	0.97±0.05	**0.97±0.04**
	MSE×10 ↓	0.04±0.02	0.05±0.03	14.30±0.03	0.02±0.06	0.02±0.07	0.02±0.07	**0.02±0.05**	0.02±0.06
	DSC ↑	0.91±0.15	0.91±0.15	0.89±0.17	0.91±0.19	0.9±0.14	0.92±0.14	0.92±0.14	**0.92±0.14**
	HD ↓	10.70±48.27	8.78±46.82	11.00±21.11	8.89±26.74	11.78±25.17	6.73±25.01	7.17±29.73	**6.50±22.66**
Brain MS	PSNR ↑	31.72±1.83	32.77±0.26	31.4±2.73	34.57±3.24	33.75±3.32	36.67±3.37	36.63±3.28	**36.67±3.36**
	SSIM ↑	0.86±0.05	0.93±0.03	0.88±0.05	0.93±0.06	0.92±0.06	0.95±0.04	0.95±0.04	**0.95±0.04**
	MSE×10 ↓	0.14±0.10	0.05±0.00	13.69±0.00	**0.01±0.01**	0.02±0.01	**0.01±0.01**	**0.01±0.01**	**0.01±0.01**
	DSC ↑	0.37±0.23	0.43±0.22	0.51±0.22	0.70±0.10	0.66±0.11	0.73±0.10	0.72±0.10	**0.73±0.12**
	HD ↓	63.24±79.39	47.00±55.39	27.71±0.00	**20.57±10.84**	24.94±9.39	21.06±11.17	21.17±11.09	21.03±11.16
Brain GBM	PSNR ↑	**35.86±0.12**	35.49±0.17	27.47±12.49	30.08±6.35	30.17±6.86	31.78±5.45	31.48±5.45	31.94±5.55
	SSIM ↑	**0.94±0.00**	0.94±0.00	0.73±0.37	0.89±0.11	0.90±0.11	0.92 ±0.11	0.92±0.21	0.93±0.34
	MSE×10 ↓	**0.01±0.01**	0.02±0.01	0.03±0.01	**0.01±0.01**	0.01±0.02	**0.01±0.01**	**0.01±0.01**	**0.01±0.01**
	DSC ↑	0.30±0.02	0.25±0.00	0.34±0.28	0.42±0.28	0.41±0.28	0.45±0.28	0.45±0.28	**0.46±0.28**
	HD ↓	198.19±7.78	189.61±7.64	271.5±15.00	142.88±77.476	143.03±79.12	141.36±77.28	141.37±75.42	**135.08±74.42**

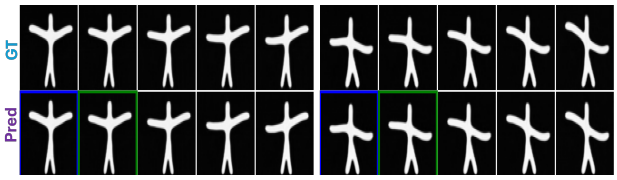

Figure 3: Trajectory on Starmen dataset. The conditioning frame is marked with green, and the reference starting frame is marked with blue. On the left *Hand-downward motion*, on the right *Hand-upward motion*.

Baselines. We benchmark our method against recent Neural ODE/SDE, denoising diffusion, and flow matching approaches, including ImageFlowNet (Liu et al., 2025), I^2SB (Liu et al., 2023), and Trajectory Flow Matching (TFM) (Zhang et al., 2024). While non-generative forecasting models form an important class of methods, we restrict our primary evaluation to state-of-the-art generative approaches, as they are conceptually closest to our framework. Furthermore, (Liu et al., 2025) recently demonstrated that ImageFlowNet outperforms standard non-generative forecasting baselines. By establishing that our method surpasses ImageFlowNet, our experiments transitively indicate an advantage over these non-generative approaches as well. Two concurrent works proposed related multi-marginal flow matching methods: (Lee et al., 2025), which combines spline fitting with a rolling-window strategy, and (Rohbeck et al., 2025), which also relies on spline fitting. Although the former appeared too late for inclusion, we benchmark against the latter (MMFM). However, TFM allows us to evaluate a similar rolling-window strategy. For consistency, we use the same network architecture across all flow matching and diffusion-based methods (see Appendix E.1).

Datasets. We validate our IMMFM variants using several longitudinal datasets, beginning with low-dimensional synthetic benchmarks, including S-shaped and σ-shaped Gaussians to test the learning of changing curvature and crossover points. Our evaluation also includes the controlled Starmen image dataset (Bône et al., 2018) and three real-world clinical cohorts with structural MRI scans of patients with Alzheimer's Disease (ADNI1, 317 participants, 4-6 visits (Mueller et al., 2005)), Multiple Sclerosis (MS) (Brain MS, 19 patients, 4-6 visits (Carass et al., 2017)), and Glioblastoma (GB) (Brain GBM, 91 patients, 2-18 visits) (Suter et al., 2022). We prioritize ADNI-1 over larger cohorts like ADNI-3 to maintain computational feasibility while demonstrating our model's efficacy in data-constrained clinical settings, where massive longitudinal datasets are often unavailable. Each visit represents one time point in the disease progression trajectories of the patients. These clinical datasets present challenges due to their high dimensionality, irregular and sparse sampling, and varying numbers of time points. For all image datasets, we first perform spatial alignment of the images as described in Section 3.4, using full volumes for 3D methods and extracted slices for 2D methods. Further details on datasets are provided in Appendix C.

Modeling Choices. IMMFM operates in the latent space obtained from a pre-trained UNet-style autoencoder. The flow matching dynamics are modeled by a U-ViT-based regressor network (Davtyan et al., 2023), that learns the components of the SDE. Specifically, we use the neural networks $v_\theta(t, x, c)$ and $s_\theta(t, x, c)$ to model the drift and score, respectively. We evaluate three variants of IMMFM: 1) a deterministic ODE version (O-IMMFM), where trajectories are generated using Euler integration, 2) a standard SDE version (S-IMMFM) simulated with the Euler-Maruyama method, and 3) an SDE with our learned, uncertainty-driven

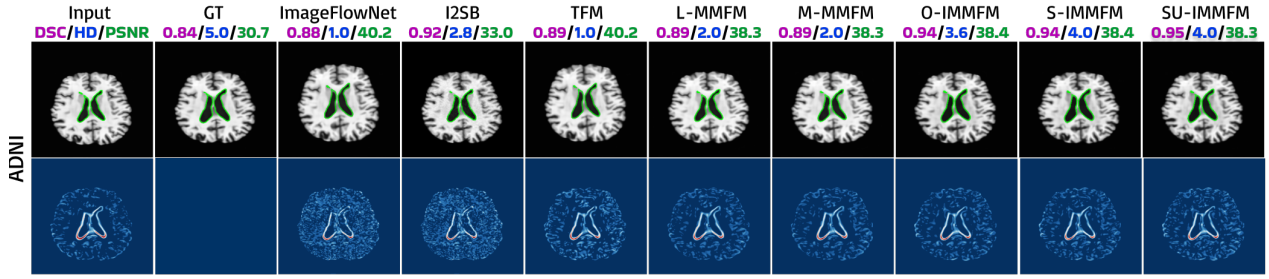

Figure 4: Visual comparison of forecasting results on the Alzheimer's (ADNI), the first row displays the forecasted image. The second row shows the corresponding pixel-wise difference map between the forecast and the ground truth. The different evaluation metrics DSC, HD, and PSNR are listed at the top.

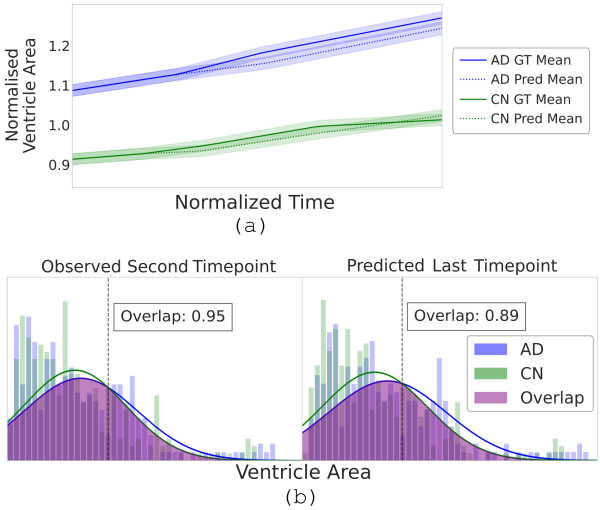

Figure 5: (a) Ground truth and predicted mean ventricle growth over time. (b) Ventricular areas at the second visit (∼18 months) versus model-predicted areas at the last visit (∼36 months) for Alzheimer's (AD) and cognitively normal (CN) subjects.

diffusion coefficient (SU-IMMFM). We emphasize that these three configurations represent independent, valid modeling choices rather than mere ablations. Depending on the application and computational budget, different variants may be preferable. The SDE variants introduce additional computational complexity through score matching and the prediction of the diffusion term. While this capacity is necessary for modeling highly stochastic clinical trajectories, it is not always warranted. For example, in cases where the underlying processes are smooth and largely deterministic, such as the arm movements in the Starmen dataset see Fig. 3, where the simpler O-IMMFM variant is theoretically sufficient. (Empirically it performs best.) Across all variants, we incorporate contextual information to generate subject-specific trajectories rather than population averages. The primary component of c is the latent representation of the preceding image $x_{t_{i-1}}$ when modeling the interval $[t_i, t_{i+1}]$. This provides an implicit initial velocity for the trajectory, enabling the learning of relationships between current and previous states to better predict future evolution. Detailed model implementations are provided in Appendix E.1.

Evaluation Metrics. We evaluate model performance with 1) image-level similarity metrics and 2) downstream segmentation accuracy metrics. Image quality is assessed with standard synthesis metrics. For pixel-wise error, we use *Mean Squared Error (MSE)* and *Peak Signal-to-Noise Ratio (PSNR)*, a logarithmic measure of reconstruction quality (Hore and Ziou, 2010). To assess the preservation of anatomical features, we also report the *Structural Similarity Index (SSIM)* (Wang et al., 2004). To evaluate the model's ability to forecast clinically relevant changes, we measure the geometric accuracy of key regions of interest (ROI) using baseline metrics to validate medical image segmentation (Taha and Hanbury, 2015; Isensee et al., 2021; Menze et al., 2015; Simpson et al., 2019; Maier-Hein et al., 2022; Zou et al., 2004; Crum et al., 2006). The *Dice–Sørensen coefficient (DSC)* measures the volumetric overlap between the predicted and ground-truth ROIs (Dice, 1945; Sørensen, 1948). The *Hausdorff distance (HD)* complements this by measuring the maximum distance between the boundaries of the two segmentations, quantifying contour accuracy (Huttenlocher et al., 1993). For details, see Appendix D.

Training and Inference. The model is trained to learn the components of the SDE. Following (Tong et al., 2023), the score weighting $\lambda(t)$ in Eq. 6 is set to $2\sigma(t)/\sigma_0^2$. At inference, trajectories are simulated by constructing the full SDE drift from Eq. (4) and the system evolves according to Eq. (1). For all experiments, we simulate the trajectory *autoregressively*. Training and inference algorithms are provided in Appendix B.

5 EXPERIMENTS

We begin the evaluation of our proposed method by demonstrating trajectory learning performance on a

simple multi-marginal dataset of temporally arranged Gaussians representing S-shape and σ-shape (see Appendix C). We show in Fig. 2 that our method can learn these simple trajectories effectively with improved mean squared error over the baselines. We exclude ImageFlowNet (Liu et al., 2025) and I^2SB (Liu et al., 2023) from this experiment since they were not developed to handle multi-marginal paths.

5.1 Subject Specific Trajectories via Conditioning

Starmen have three distinct motion classes: 1) hand-downward motion, 2) hand-upward motion, and 3) static. We train our model conditioned on the image of the previous time point. As shown in Fig. 3 (and appendix Fig. 7), our model simulates the trajectory of the correct class solely based on previous frame conditioning. Our method achieves the best MSE compared to all the baselines, see appendix Tab. 1.

5.2 Real-World Application: Disease Progression Modeling

Tab. 1 summarizes the quantitative results on the disease progression datasets. IMMFM variants consistently outperform baselines across datasets, with gains of **1**–**4.4**% in Dice score, **1.5**–**2.2** dB in PSNR, and **1.2**–**4.5**% in SSIM. On noisy, stochastic clinical datasets, the uncertainty-aware SU-IMMFM provides the best performance. We examine the contribution of each model's individual components in an ablation study in Appendix F.1. We further show in Tab. 5 that IMMFM extends from 2D image data to 3D volumetric data, improving Dice by $\sim$ **3**% and reducing inter-subject variability compared to 2D.

Qualitatively, IMMFM generates consistent images over both short- and long-term horizons (Fig. 4 and appendix Fig. 8; see also temporal generalizability in Appendix F.2). The model makes clinically meaningful updates, especially around the lesion and tumor regions, where we observe an improved overlap of these regions with their ground-truth counterparts.

Finally, we examine the specific progression of the disease. For the ADNI dataset, we focus on the ventricle region, as ventricular enlargement is a well-studied biomarker of AD progression and its underlying neurodegenerative process (Nestor et al., 2008). IMMFM captures distinct ventricular growth trajectories for AD versus CN groups without explicit class conditioning (Fig. 5). Using only observed data at the second visit ($\sim$18 months), AD–CN classification reaches **71.7**%. When using trajectories forecasted by our model to $\sim$36 months, AD-CN classification accuracy increases to **80.8**%, a **9.1**% gain. In practical terms, this corresponds to correctly classifying about nine additional subjects per 100, *18 months earlier* than would be possible using only the observed data (see Appendix G.1.2 for additional details).

6 CONCLUSION

We introduced Interpolative Multi-Marginal Flow Matching (IMMFM), a framework for learning high-dimensional trajectories from sparse and irregular data. By combining a smooth quadratic conditional path with a data-driven diffusion coefficient, IMMFM captures both structured progression and uncertainty. Across synthetic and real datasets, including challenging longitudinal medical imaging benchmarks, IMMFM consistently outperformed prior methods in forecasting accuracy and downstream tasks, demonstrating its improved reliability and clinical relevance.

The model's quadratic path construction inherently smooths high-frequency oscillations between sparse observations. We view this low-pass filtering effect as a necessary implicit regularization. It ensures stable flow matching by preventing numerical instabilities in chaotic regimes. Crucially, SU-IMMFM compensates for this smoothing by absorbing unmodeled chaotic volatility into its learned diffusion component, preserving the representation of local stochasticity.

Naturally, the approach still depends on high-quality data and sufficient training coverage. To address these constraints and expand the framework's capabilities, several promising future directions exist. Enhancing the latent space with temporally aware autoencoders (Yang et al., 2023a; Blattmann et al., 2023) could significantly improve trajectory coherence. Furthermore, incorporating multi-modal conditioning (Shaik et al., 2024) and causal representation learning would allow the model to simulate counterfactual interventions, advancing its utility from a prognostic tool to a comprehensive decision support system (von Kügelgen et al., 2024). Finally, integrating domain-specific knowledge offers a powerful avenue to improve generalization in data-scarce settings. Depending on the application, this could range from strict Physics-Informed Neural Network (PINN) constraints for systems with well-established governing equations (Qian et al., 2025; Cuomo et al., 2023) to approximate mechanistic priors, such as network diffusion models of pathology spread, for complex clinical domains like Alzheimer's disease progression. A detailed discussion of these limitations and future directions is provided in Appendix H.

In conclusion, IMMFM offers a promising foundation for robust, clinically useful trajectory modeling.

References

Anonymous (2024). An efficient high-dimensional gradient estimator for stochastic differential equations. In *Advances in Neural Information Processing Systems*. (NeurIPS 2024, double-blind submission; update with final authors when available).

Avants, B. B., Tustison, N. J., and Song, G. (2009). Advanced normalization tools (ants). *The Insight Journal*, 2(365):1–35.

Bartosh, G., Vetrov, D., and Naesseth, C. A. (2025). Sde matching: Scalable and simulation-free training of latent stochastic differential equations. *arXiv preprint arXiv:2502.02472*.

Blattmann, A., Rombach, R., Ling, H., Dockhorn, T., Kim, S. W., Fidler, S., and Esser, P. (2023). Align your latents: High-resolution video synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22563–22575.

Bône, A., Colliot, O., and Durrleman, S. (2018). Learning distributions of shape trajectories from longitudinal datasets: A hierarchical model on a manifold of diffeomorphisms. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9271–9280.

Bossa, M. N. and Sahli, H. (2023). A multidimensional ode-based model of alzheimer's disease progression. *Scientific reports*, 13(1):3162.

Cai, W., Yang, M., and Xie, Y. (2025). Minimax optimality of the probability flow ODE sampler. *arXiv preprint arXiv:2501.04567*.

Carass, A., Roy, S., Jog, A., Cuzzocreo, J. L., Magrath, E., Gherman, A., Button, J., Nguyen, J., Prados, F., Sudre, C. H., and et al. (2017). Longitudinal multiple sclerosis lesion segmentation: Resource and challenge. *NeuroImage*, 148:77–102.

Caruana, E. J., Roman, M., Hernández-Sánchez, J., and Solli, P. (2015). Longitudinal studies. *Journal of thoracic disease*, 7(11):E537.

Chen, R. T., Rubanova, Y., Bettencourt, J., and Duvenaud, D. K. (2018). Neural ordinary differential equations. *Advances in neural information processing systems*, 31.

Chen, Y. et al. (2025). Generative ai for synthetic data across multiple medical modalities. *Computers in Biology and Medicine*, 170:107981.

Cho, H., Wei, Z., Lee, S., Dan, T., Wu, G., and Kim, W. H. (2025). Conditional diffusion with ordinal regression: Longitudinal data generation for neurodegenerative disease studies. In *The Thirteenth International Conference on Learning Representations*.

Crum, W. R., Camara, O., and Hill, D. L. G. (2006). Generalized overlap measures for evaluation and validation in medical image analysis. *IEEE Transactions on Medical Imaging*, 25(11):1451–1461.

Cuomo, S., Di Cola, V. S., Giampaolo, F., Rozza, G., Raissi, M., and Piccialli, F. (2023). Scientific machine learning. *Physics Reports*, 1027:1–74.

Dang, T., Han, J., Xia, T., Bondareva, E., Siegele-Brown, C., Chauhan, J., Grammenos, A., Spathis, D., Cicuta, P., and Mascolo, C. (2023). Conditional neural ode processes for individual disease progression forecasting: a case study on covid-19. In *Proceedings of the 29th ACM SIGKDD Conference On Knowledge Discovery and Data Mining*, pages 3914–3925.

Davtyan, A., Sameni, S., and Favaro, P. (2023). Efficient video prediction via sparsely conditioned flow matching. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 23263–23274.

Dhariwal, P. and Nichol, A. (2021). Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794.

Dice, L. R. (1945). Measures of the amount of ecologic association between species. *Ecology*, 26(3):297–302.

Dong, X., Wong, R., Lyu, W., Abell-Hart, K., Deng, J., Liu, Y., Hajagos, J. G., Rosenthal, R. N., Chen, C., and Wang, F. (2023). An integrated lstm-heterorgnn model for interpretable opioid overdose risk prediction. *Artificial Intelligence in Medicine*, 135:102439.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. (2021). An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations (ICLR)*.

Gardiner, C. W. (2009). *Stochastic Methods: A Handbook for the Natural and Social Sciences*. Springer, Berlin, Heidelberg, 5 edition.

Hamdouche, M., Henry-Labordere, P., and Pham, H. (2023). Generative modeling for time series via schr {\" o} dinger bridge. *arXiv preprint arXiv:2304.05093*.

Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer Series in Statistics. Springer, New York, 2nd edition.

Hore, A. and Ziou, D. (2010). Image quality metrics: A survey. In *2010 20th International Conference on Pattern Recognition*, pages 2366–2369. IEEE.

Huttenlocher, D. P., Klanderman, G. A., and Rucklidge, W. J. (1993). Comparing images using the hausdorff distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(9):850–863.

Inman, H. F. and Bradley Jr, E. L. (1989). The overlapping coefficient as a measure of agreement between probability distributions and point estimation of the overlap of two normal densities. *Communications in Statistics-Theory and Methods*, 18(10):3851–3874.

Isensee, F., Jaeger, P. F., Kohl, S. A., Petersen, J., and Maier-Hein, K. H. (2021). nnU-Net: a self-configuring method for deep learning-based biomedical image segmentation. *Nature Methods*, 18(2):203–211.

Karras, T., Aittala, M., Laine, S., Herva, A., and Lehtinen, J. (2022). Elucidating the design space of diffusion-based generative models. *arXiv preprint arXiv:2206.00364*.

Kidger, P., Foster, J., Li, R. T. Q., and Lyons, T. (2021). Efficient and accurate gradients for neural SDEs. In *International Conference on Machine Learning*, pages 5562–5572.

Kidger, P., Morrill, J., Foster, J., and Lyons, T. (2020). Neural controlled differential equations for irregular time series. *Advances in neural information processing systems*, 33:6696–6707.

Lachinov, D., Chakravarty, A., Grechenig, C., Schmidt-Erfurth, U., and Bogunović, H. (2023). Learning spatio-temporal model of disease progression with neuralodes from longitudinal volumetric data. *IEEE Transactions on Medical Imaging*, 43(3):1165–1179.

Lee, J., Moradijamei, B., and Shakeri, H. (2025). Multi-marginal stochastic flow matching for high-dimensional snapshot data at irregular time points. *arXiv preprint arXiv:2508.04351*.

Lei, B., Yang, M., Yang, P., Zhou, F., Hou, W., Zou, W., Li, X., Wang, T., Xiao, X., and Wang, S. (2020). Deep and joint learning of longitudinal data for alzheimer's disease prediction. *Pattern Recognition*, 102:107247.

Li, G., Zhang, Y., and Cheng, X. (2024). A sharp convergence theory for the probability flow ODE. *arXiv preprint arXiv:2404.01234*.

Li, R. T. Q., Kidger, P., Foster, J., and Lyons, T. (2020). Scalable gradients for stochastic differential equations. In *Advances in Neural Information Processing Systems*, volume 33, pages 3515–3527.

Lipman, Y., Chen, R. T., Ben-Hamu, H., Nickel, M., and Le, M. (2022). Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*.

Liu, C., Xu, K., Shen, L. L., Huguet, G., Wang, Z., Tong, A., Bzdok, D., Stewart, J., Wang, J. C., Del Priore, L. V., et al. (2025). Imageflownet: Forecasting multiscale image-level trajectories of disease progression with irregularly-sampled longitudinal medical images. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE.

Liu, G.-H., Vahdat, A., Huang, D.-A., Theodorou, E. A., Nie, W., and Anandkumar, A. (2023). I^2sb: Image-to-image schrödinger bridge. *arXiv preprint arXiv:2302.05872*.

Liu, P., Fu, B., Yang, S. X., Deng, L., Zhong, X., and Zheng, H. (2020). Optimizing survival analysis of xgboost for ties to predict disease progression of breast cancer. *IEEE Transactions on Biomedical Engineering*, 68(1):148–160.

Locascio, J. J. and Atri, A. (2011). An overview of longitudinal data analysis methods for neurological research. *Dementia and geriatric cognitive disorders extra*, 1(1):330–357.

Lu, C., Zhou, Z., Bao, F., Chen, J., Li, C., and Zhu, J. (2022). DPM-solver: A fast ODE solver for diffusion probabilistic model sampling in around 10 steps. *Advances in Neural Information Processing Systems*, 35:5775–5788.

Lu, J., Han, X., Sun, Y., and Yang, S. (2024). Cats: Enhancing multivariate time series forecasting by constructing auxiliary time series as exogenous variables. In *Proceedings of the Forty-first International Conference on Machine Learning (ICML)*. arXiv preprint arXiv:2403.01673.

Lyu, W., Dong, X., Wong, R., Zheng, S., Abell-Hart, K., Wang, F., and Chen, C. (2023). A multimodal transformer: Fusing clinical notes with structured ehr data for interpretable in-hospital mortality prediction. In *AMIA Annual Symposium Proceedings*, volume 2022, page 719.

Maier-Hein, L., Reinke, A., et al. (2022). Metrics reloaded: recommendations for image analysis validation. *Nature Methods*, 19(9):1127–1140.

Menze, B. H., Jakab, A., Bauer, S., Kalpathy-Cramer, J., Farahani, K., Kirby, J., Burren, Y., Porz, N., Slotboom, J., Wiest, R., et al. (2015). The multimodal brain tumor image segmentation benchmark (brats). In *IEEE Transactions on Medical Imaging*, volume 34, pages 1993–2024.

Mok, T. C. and Chung, A. C. (2020). Fast symmetric diffeomorphic image registration with convolutional neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Mueller, S. G., Weiner, M. W., Thal, L. J., Petersen, R. C., Jack, C. R., Jagust, W., Trojanowski, J. Q.,

Toga, A. W., and Beckett, L. (2005). Ways toward an early diagnosis in Alzheimer's disease: The Alzheimer's Disease Neuroimaging Initiative (ADNI). *Alzheimer's & Dementia*, 1(1):55–66.

Nestor, S. M., Rupsingh, R., Borrie, M., Smith, M., Accomazzi, V., Wells, J. L., Fogarty, J., Bartha, R., and Initiative, A. D. N. (2008). Ventricular enlargement as a possible measure of alzheimer's disease progression validated using the alzheimer's disease neuroimaging initiative database. *Brain*, 131(9):2443–2454.

Nguyen, H. H., Blaschko, M. B., Saarakkala, S., and Tiulpin, A. (2023). Clinically inspired multi-agent transformers for disease trajectory forecasting from multimodal data. *IEEE Transactions on Medical Imaging*.

Pass, B. (2015). Multi-marginal optimal transport: theory and applications. *ESAIM: Mathematical Modelling and Numerical Analysis*, 49(6):1771–1790.

Qian, Y., Marty, É., Basu, A., O'Dea, E. B., Wang, X., Fox, S., Rohani, P., Drake, J. M., and Li, H. (2025). Physics-informed deep learning for infectious disease forecasting. *ArXiv*, pages arXiv–2501.

Reiser, B. and Faraggi, D. (1999). Measuring the effectiveness of diagnostic markers in the presence of measurement error through the use of roc curves. *Statistics in medicine*, 18(17-18):2583–2600.

Rohbeck, M., De Brouwer, E., Bunne, C., Huetter, J.-C., Biton, A., Chen, K. Y., Regev, A., and Lopez, R. (2025). Modeling complex system dynamics with flow matching across time and conditions. In *The Thirteenth International Conference on Learning Representations*.

Ruan, C., Huang, C., and Yang, Y. (2024). Comprehensive evaluation of multimodal ai models in medical imaging diagnosis: From data augmentation to preference-based comparison. *arXiv preprint arXiv:2412.05536*.

Ryder, T., Dondelinger, F., and Husmeier, D. (2018). Black-box variational inference for stochastic differential equations. In *International Conference on Artificial Intelligence and Statistics*, pages 1281–1290.

Shaik, T., Tao, X., Li, L., Xie, H., and Velásquez, J. D. (2024). A survey of multimodal information fusion for smart healthcare: Mapping the journey from data to wisdom. *Information Fusion*, 102:102040.

Shi, Y., De Bortoli, V., Campbell, A., and Doucet, A. (2023). Diffusion schrödinger bridge matching. *Advances in Neural Information Processing Systems*, 36:62183–62223.

Simpson, A. L., Antonelli, M., Bakas, S., Bilello, M., Farahani, K., Van Ginneken, B., Kopp-Schneider, A., Landman, B. A., Litjens, G., Menze, B. H., et al. (2019). A large annotated medical image dataset for the development and evaluation of segmentation algorithms. *arXiv preprint arXiv:1902.09063*.

Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. (2021). Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*.

Sørensen, T. (1948). A method of establishing groups of equal amplitude in plant sociology based on similarity of species and its application to analyses of the vegetation on danish commons. *Biologiske Skrifter*, 5:1–34.

Sun, Y. and Yang, S. (2023). Manifold-constrained gaussian process inference for time-varying parameters in dynamic systems. *Statistics and Computing*, 33(6):142.

Suter, Y., Knecht, U., Valenzuela, W., Notter, M., Hewer, E., Schucht, P., Wiest, R., and Reyes, M. (2022). The lumiere dataset: Longitudinal glioblastoma mri with expert rano evaluation. *Scientific Data*, 9(1):768.

Taha, A. A. and Hanbury, A. (2015). Metrics for evaluating 3d medical image segmentation: analysis, selection, and tool. *BMC Medical Imaging*, 15(1):29.

Tong, A., Malkin, N., Fatras, K., Atanackovic, L., Zhang, Y., Huguet, G., Wolf, G., and Bengio, Y. (2023). Simulation-free schrödinger bridges via score and flow matching. *arXiv preprint arXiv:2307.03672*.

Tong, A., Shi, Y., Maddison, C. J., and Amos, B. (2024). Simulation-free Schrödinger bridges via score and flow matching. In *Proceedings of the International Conference on Machine Learning*.

Tzen, B. and Raginsky, M. (2019). Neural stochastic differential equations: Deep latent gaussian models in continuous time. In *International Conference on Machine Learning*, pages 5474–5483.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.

Voleti, V., Kanaa, D., Kahou, S. E., and Pal, C. (2021). Simple video generation using neural odes. *arXiv preprint arXiv:2109.03292*.

von Kügelgen, J., Gresele, L., and Schölkopf, B. (2024). Causal representation learning. *arXiv preprint arXiv:2401.14411*.

Wahba, G. (1990). *Spline Models for Observational Data*, volume 59 of *CBMS–NSF Regional Conference*

Series in Applied Mathematics. SIAM, Philadelphia, PA.

Wang, Z., Bovik, A. C., Sheikh, H. R., and Simoncelli, E. P. (2004). Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612.

Wolleb, J. et al. (2022). Medfusion: Diffusion models for medical image synthesis. *Medical Image Analysis*, 79:102479.

Wu, Y., Ku, Z., Yin, P., and Zhang, Y. (2024). Q-fusion: A modular framework for multimodal multi-task foundation models. *arXiv preprint arXiv:2402.13220*.

Yang, D., Wang, Y., Kong, Q., Dantcheva, A., Garattoni, L., Francesca, G., and Brémond, F. (2023a). Self-supervised video representation learning via latent time navigation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 3118–3126.

Yang, Q., Hernandez-Garcia, A., Harder, P., Ramesh, V., Sattegeri, P., Szwarcman, D., and Rolnick, D. (2023b). Fourier neural operators for arbitrary resolution climate data downscaling. arxiv. *arXiv preprint arXiv:2305.14452*, 10.

Yu, S., Chen, T., Shen, J., Yuan, H., Tan, J., Yang, S., Liu, J., and Wang, Z. (2022). Unified visual transformer compression. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Zhan, Y. et al. (2024). Medm2g: Unifying medical multi-modal generation via cross-guided diffusion with visual invariance. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Zhang, X., Pu, Y., Kawamura, Y., Loza, A., Bengio, Y., Shung, D. L., and Tong, A. (2024). Trajectory flow matching with applications to clinical time series modeling. *arXiv preprint arXiv:2410.21154*.

Zhou, B. and Parno, M. (2024). Efficient and exact multimarginal optimal transport with pairwise costs. *Journal of Scientific Computing*, 100(1):25.

Zou, K. H., Warfield, S. K., Bharatha, A., Tempany, C. M. C., Kaus, M. R., Haker, S. J., Wells, W. M., Jolesz, F. A., and Kikinis, R. (2004). Statistical validation of image segmentation quality based on a spatial overlap index: scientific reports. *Academic Radiology*, 11(2):178–189.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes]
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Not Applicable]
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Yes]
2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [Yes]
   (b) Complete proofs of all theoretical results. [Yes]
   (c) Clear explanations of any assumptions. [Yes]
3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes]
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes]
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes]
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. [Yes]
   (b) The license information of the assets, if applicable. [Not Applicable]
   (c) New assets either in the supplemental material or as a URL, if applicable. [Yes]
   (d) Information about consent from data providers/curators. [Not Applicable]
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable]
5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. [Not Applicable]
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable]
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable]

Longitudinal Flow Matching For Trajectory Modeling: Supplementary Materials

A Derivation and Proof

A.1 Derivative of Mean μ_t and Varinace $\sigma(t)$

We formulate our *blended velocity field* as:

$$u_t^{\circ}(x \mid z) = v_i + \frac{1}{2}(v_i - v_{i+1})(2\alpha_t - 1) + \frac{\sigma'(t)}{\sigma(t)}\big(x - \mu_t(z)\big), \tag{19}$$

where $\mu_t'(z)$ and $\sigma'(t)$ are the time derivatives of $\mu_t(z)$ and $\sigma(t)$, respectively. The mean function for this case is:

$$\mu_t = x_i + \frac{x_{i+1} - x_i}{t_{i+1} - t_i}(t - t_i) + \frac{1}{2}\left(\frac{x_{i+1} - x_i}{t_{i+1} - t_i} - \frac{x_{i+2} - x_{i+1}}{t_{i+2} - t_{i+1}}\right)\frac{(t - t_i)(t_{i+1} - t)}{t_{i+1} - t_i} \tag{20}$$

$$= x_i + v_i(t - t_i) + \frac{1}{2}(v_i - v_{i+1})\frac{(t - t_i)(t_{i+1} - t)}{t_{i+1} - t_i} \tag{21}$$

where the scaled segment velocities are:

$$v_i = \frac{x_{i+1} - x_i}{t_{i+1} - t_i} \tag{22}$$

$$v_{i+1} = \frac{x_{i+2} - x_{i+1}}{t_{i+2} - t_{i+1}} \tag{23}$$

Derivative of μ'_t function:

$$\mu'_t = \frac{d}{dt}\left[x_i + v_i(t-t_i) + \frac{1}{2}(v_i - v_{i+1})\frac{(t-t_i)(t_{i+1}-t)}{t_{i+1}-t_i}\right] \tag{24}$$

$$= v_i + \frac{1}{2}(v_i - v_{i+1})\frac{d}{dt}\left[\frac{(t-t_i)(t_{i+1}-t)}{t_{i+1}-t_i}\right] \tag{25}$$

$$= v_i + \frac{1}{2}(v_i - v_{i+1})\frac{1}{t_{i+1}-t_i}\left[(t_{i+1}-t) - (t-t_i)\right] \tag{26}$$

$$= v_i + \frac{1}{2}(v_i - v_{i+1})\frac{t_{i+1}-t-t+t_i}{t_{i+1}-t_i} \tag{27}$$

$$= v_i + \frac{1}{2}(v_i - v_{i+1})\frac{t_{i+1}-t_i-2(t-t_i)}{t_{i+1}-t_i} \tag{28}$$

$$= v_i + \frac{1}{2}(v_i - v_{i+1})\left[1 - \frac{2(t-t_i)}{t_{i+1}-t_i}\right] \tag{29}$$

$$= v_i + \frac{1}{2}(v_i - v_{i+1})\left[1 - 2\left(1 - \frac{t_{i+1}-t}{t_{i+1}-t_i}\right)\right] \tag{30}$$

$$= v_i + \frac{1}{2}(v_i - v_{i+1})\left[1 - 2(1-\alpha_t)\right] \tag{31}$$

$$= v_i + \frac{1}{2}(v_i - v_{i+1})(2\alpha_t - 1) \tag{32}$$

$$\tag{33}$$

The variance function defined in Eq. (9):

$$\sigma(t) = \sigma_0 \cdot \frac{(t-t_i)(t_{i+1}-t)}{(t_{i+1}-t_i)} \tag{34}$$

Let $\Delta t = t_{i+1} - t_i$:

$$\sigma(t) = \sigma_0 \cdot \frac{(t-t_i)(t_{i+1}-t)}{\Delta t} \tag{35}$$

We compute the derivative with respect to t:

$$\sigma'(t) = \sigma_0 \cdot \frac{d}{dt}\left[\frac{(t-t_i)(t_{i+1}-t)}{\Delta t}\right] \tag{36}$$

$$= \sigma_0 \cdot \frac{1}{\Delta t} \cdot \frac{d}{dt}\left[(t-t_i)(t_{i+1}-t)\right] \tag{37}$$

Using the product rule for $(t-t_i)(t_{i+1}-t)$:

$$\frac{d}{dt}\left[(t-t_i)(t_{i+1}-t)\right] = (1)(t_{i+1}-t) + (t-t_i)(-1) \tag{38}$$

$$= (t_{i+1}-t) - (t-t_i) \tag{39}$$

$$= t_{i+1} - t - t + t_i \tag{40}$$

$$= t_{i+1} + t_i - 2t \tag{41}$$

Therefore:

$$\sigma'(t) = \sigma_0 \cdot \frac{t_{i+1} + t_i - 2t}{\Delta t} = \sigma_0 \,(2\alpha_t - 1), \quad \text{where } \alpha_t = \frac{t_{i+1} - t}{t_{i+1} - t_i}. \tag{42}$$

A.2 Proof of Proposition 3.2

Proposition (Restatement of Proposition 3.2)**.** *Under the pairwise additive cost structure, the MMOT problem decomposes into a series of independent pairwise OT problems. The resulting joint coupling $q(z)$ is:*

$$q(z) = \pi^*(x_{t_0}, \ldots, x_{t_M}) = \frac{\prod_{i=0}^{M-1} \pi^*_{i,i+1}(x_{t_i}, x_{t_{i+1}})}{\prod_{i=1}^{M-1} \rho_i^\dagger(x_{t_i})} \tag{43}$$

This coupling preserves the intermediate augmented marginals.

Proof. We consider the multi-marginal optimal transport problem with augmented distributions $\rho_i^\dagger$ that incorporate all possible diffeomorphic transformations:

$$\min_{\pi \in \Pi(\rho_0^\dagger, \ldots, \rho_M^\dagger)} \int C(x_{t_0}, \ldots, x_{t_M}) \, d\pi(x_{t_0}, \ldots, x_{t_M}) \tag{44}$$

where $\Pi(\rho_0^\dagger, \ldots, \rho_M^\dagger)$ denotes couplings with prescribed augmented marginals.

Using the pairwise additive structure, the objective becomes:

$$\min_{\pi \in \Pi(\rho_0^\dagger, \ldots, \rho_M^\dagger)} \int \sum_{i=0}^{M-1} c(x_{t_i}, x_{t_{i+1}}) \, d\pi(x_{t_0}, \ldots, x_{t_M}) = \min_{\pi \in \Pi(\rho_0^\dagger, \ldots, \rho_M^\dagger)} \sum_{i=0}^{M-1} \int c(x_{t_i}, x_{t_{i+1}}) \, d\pi_{i,i+1}(x_{t_i}, x_{t_{i+1}}) \tag{45}$$

where $\pi_{i,i+1}$ is the marginal of π on coordinates $(x_{t_i}, x_{t_{i+1}})$.

We claim that the coupling:

$$q(z) = \frac{\prod_{i=0}^{M-1} \pi^*_{i,i+1}(x_{t_i}, x_{t_{i+1}})}{\prod_{i=1}^{M-1} \rho_i^\dagger(x_{t_i})} \tag{46}$$

solves the multi-marginal problem. To verify this, we must show it preserves marginals and minimizes cost.

For marginal preservation, we verify that $q(z)$ has the correct marginals. For interior points $i \in \{1, \ldots, M-1\}$:

$$\int q(z) \prod_{j \neq i} dx_{t_j} = \int \frac{\prod_{k=0}^{M-1} \pi^*_{k,k+1}(x_{t_k}, x_{t_{k+1}})}{\prod_{j=1}^{M-1} \rho_j^\dagger(x_{t_j})} \prod_{j \neq i} dx_{t_j} \tag{47}$$

$$= \frac{1}{\rho_i^\dagger(x_{t_i})} \int \pi^*_{i-1,i}(x_{t_{i-1}}, x_{t_i}) \pi^*_{i,i+1}(x_{t_i}, x_{t_{i+1}}) \prod_{k \notin \{i-1,i,i+1\}} \pi^*_{k,k+1}(x_{t_k}, x_{t_{k+1}}) \prod_{j \neq i} dx_{t_j} \tag{48}$$

Using Fubini's theorem to integrate out variables that appear in only one factor:

$$= \frac{1}{\rho_i^\dagger(x_{t_i})} \left[\int \pi^*_{i-1,i}(x_{t_{i-1}}, x_{t_i}) \, dx_{t_{i-1}} \right] \left[\int \pi^*_{i,i+1}(x_{t_i}, x_{t_{i+1}}) \, dx_{t_{i+1}} \right] \tag{49}$$

$$= \frac{1}{\rho_i^\dagger(x_{t_i})} \cdot \rho_i^\dagger(x_{t_i}) \cdot \rho_i^\dagger(x_{t_i}) = \rho_i^\dagger(x_{t_i}) \tag{50}$$

For boundary points:

$$\int q(z) \prod_{j \neq 0} dx_{t_j} = \int \pi^*_{0,1}(x_{t_0}, x_{t_1}) \, dx_{t_1} = \rho_0^\dagger(x_{t_0}) \tag{51}$$

$$\int q(z) \prod_{j \neq M} dx_{t_j} = \int \pi^*_{M-1,M}(x_{t_{M-1}}, x_{t_M}) \, dx_{t_{M-1}} = \rho_M^\dagger(x_{t_M}) \tag{52}$$

For cost minimization, the cost under $q(z)$ is

$$\int C(z)\, dq(z) = \sum_{i=0}^{M-1} \int c\big(x_{t_i}, x_{t_{i+1}}\big)\, d\pi^*_{i,i+1}\big(x_{t_i}, x_{t_{i+1}}\big),$$

which is the sum of optimal pairwise costs. Since each $\pi^*_{i,i+1}$ minimizes its respective term, $q(z)$ minimizes the total cost. □

B IMMFM Training and Forecasting Algorithms

This section provides the detailed procedures for the IMMFM framework. **Algorithm 1** outlines the complete training process, from sampling trajectories to the stochastic gradient update. **Algorithm 2** details the autoregressive forecasting method used for inference, which can perform either deterministic (ODE) or stochastic (SDE) simulations. A practical consideration for the training procedure is the use of Multi-Marginal Optimal Transport (MMOT) for constructing the ground-truth trajectories. This potentially expensive MMOT problem can be solved offline as a one-time preprocessing step for datasets where the set of longitudinal observations is fixed. The resulting optimal transport plans can be loaded during training, which significantly accelerates the optimization process.

Algorithm 1 IMMFM Training

Input: Training data $\mathcal{D} = \{\mathbf{z}_n\}_{n=1}^N$, initial variance parameter σ_0, loss weight β.
Initialize networks: Drift v_θ, Score s_θ, Diffusion g_θ.
while training **do**
Sample a mini-batch of full trajectories $\{(x_{t_0}, \dots, x_{t_M})\}$ from $\mathcal{D}$.
for each trajectory $\mathbf{z} = (x_{t_0}, \dots, x_{t_M})$ in the mini-batch **do**
    Sample $t \sim \mathcal{U}(t_0, t_M)$.
    Find index j such that $t \in [t_j, t_{j+1})$.
    Select segment data: $t_a \leftarrow t_j$, $x_a \leftarrow x_{t_j}$, $t_b \leftarrow t_{j+1}$, $x_b \leftarrow x_{t_{j+1}}$.
    Set conditioning variable $c \leftarrow x_{t_{j-1}}$ (or zero vector if $j = 0$).
    Define local interp. velocities $v_j \leftarrow \frac{x_b - x_a}{t_b - t_a}$ and $v_{j+1} \leftarrow \frac{x_{t_{j+2}} - x_b}{t_{j+2} - t_b}$ (or v_j if $j = M-1$).
    Compute $\alpha_t \leftarrow \frac{t_b - t}{t_b - t_a}$.
    Compute $\mu_t \leftarrow x_a + v_j(t - t_a) + \frac{1}{2}\alpha_t(v_j - v_{j+1})(t - t_a)$.
    Compute $\mu'_t \leftarrow v_j + \frac{1}{2}(v_j - v_{j+1})(2\alpha_t - 1)$.
    Sample $x \sim \mathcal{N}(x \mid \mu_t, \sigma(t)^2 I)$, with $\sigma(t) = \sigma_0(t - t_a)\alpha_t$.
    Compute target velocity $u_t^\circ(x \mid \mathbf{z}) \leftarrow \frac{\sigma'(t)}{\sigma(t)}(x - \mu_t) + \mu'_t$, with $\sigma'(t) = \sigma_0(2\alpha_t - 1)$.
    Compute target score $\nabla_x \log p_t(x \mid \mathbf{z}) \leftarrow \frac{\mu_t - x}{\sigma(t)^2}$.
    Compute $\mathcal{L}_{\text{CSDE}} \leftarrow \|v_\theta(t, x, c) - u_t^\circ(x \mid \mathbf{z})\|_2^2 + \lambda(t)^2 \|s_\theta(t, x, c) - \nabla_x \log p_t(x \mid \mathbf{z})\|_2^2$.
    Assemble SDE drift $u_\theta(t, x, c) \leftarrow v_\theta(t, x, c) + \frac{g_\theta(t,x,c)^2}{2} s_\theta(t, x, c)$.
    Predict endpoint $\hat{x}_{t_b} \leftarrow x + (t_b - t)u_\theta(t, x, c)$.
    Compute $\mathcal{L}_{\text{uncertainty}} \leftarrow \|g_\theta(t, x, c)^2 - \|\hat{x}_{t_b} - x_b\|_2^2\|_2^2$.
    Compute $\mathcal{L}_{\text{IMMFM}}(\theta) \leftarrow \mathcal{L}_{\text{CSDE}} + \beta\mathcal{L}_{\text{uncertainty}}$.
    Update θ using $\nabla_\theta \mathcal{L}_{\text{IMMFM}}(\theta)$.
end for
end while
Output: Trained networks $v_\theta, s_\theta, g_\theta$.

Algorithm 2 IMMFM Forecasting

Input: Trained networks $v_\theta, s_\theta, g_\theta$; observed prefix $(x_{t_0}, \ldots, x_{t_k})$ with $k \geq 1$; forecast end time t_{end}; integration step size Δt; mode $\in$ 'ODE', 'SDE'.
Initialize: Trajectory $\mathcal{T} \leftarrow [x_{t_0}, \ldots, x_{t_k}]$.
Set current time $t \leftarrow t_k$.
Set current state $x \leftarrow x_{t_k}$.
Set conditioning variable $c \leftarrow x_{t_{k-1}}$.
Set number of steps $N_{steps} \leftarrow \lfloor (t_{end} - t)/\Delta t \rfloor$.
for $i = 0$ to $N_{steps} - 1$ **do**
    if mode = 'ODE' **then**
        Compute ODE drift $u_t \leftarrow v_\theta(t, x, c)$.
        Evolve state $x_{new} \leftarrow x + u_t \Delta t$.
    else (mode = 'SDE')
        Sample noise $\mathbf{z} \sim \mathcal{N}(0, I)$.
        Compute SDE drift $u_t \leftarrow v_\theta(t, x, c) + \frac{g_\theta(t,x,c)^2}{2} s_\theta(t, x, c)$.
        Get SDE diffusion $g_t \leftarrow g_\theta(t, x, c)$.
        Evolve state $x_{new} \leftarrow x + u_t \Delta t + g_t \sqrt{\Delta t}\mathbf{z}$.
    end if
    Update conditioning variable $c \leftarrow x$.
    Update current state $x \leftarrow x_{new}$.
    Update current time $t \leftarrow t + \Delta t$.
    Append x to $\mathcal{T}$.
end for
Output: Complete trajectory $\mathcal{T}$ (observed prefix + forecast).

C Datasets

S-shape and σ-shape Gaussian Dataset. The S-shaped and σ-shaped Gaussians both consist of 8 marginal distributions in $\mathbb{R}^2$ at arbitrary timepoints $T = (0, 0.17, 0.29, 0.45, 0.65, 0.71, 0.85, 1)$. We select these two datasets because S-shaped Gaussians involve learning a flow with changing curvature, and the σ-shaped Gaussians have a crossover point for some x where the flow $u_{t_i}(x) = u_{t_j}(x)$ and $i \neq j$.

ADNI1 Dataset. ADNI-1, the inaugural phase of the Alzheimer's Disease Neuroimaging Initiative (Mueller et al., 2005), launched in October 2004 as a five-year multicenter study, enrolling 317 participants—100 cognitively normal (CN) elderly controls, 117 with amnestic mild cognitive impairment (MCI), and 100 with early Alzheimer's disease(AD) —across 57 sites in the US and Canada. Participants underwent serial 1.5T structural MRI at approximately six-month intervals. For the sake of simplicity, we use only CN and AD subjects. The dataset is provided with the segmentation mask for the ventricle. Because the actual clinical acquisitions were not imaged at those exact times, we bin the measurements to standard 6-month intervals to facilitate a better and more clinically relevant analysis. For the sake of simplicity, we use only CN and AD subjects. The dataset is provided with the segmentation mask for the ventricle.

Brain Multiple Sclerosis Dataset. We used longitudinal FLAIR-weighted MRI scans from the Brain MS dataset (Carass et al., 2017), monitoring 19 patients with multiple sclerosis (MS) over an average of 4.4 time points spanning approximately five years. The Training set included manual delineations by two experts, identifying and segmenting the lesions.

Brain Glioblastoma Dataset. We used longitudinal contrast-enhanced T1-weighted MRI scans from the LUMIERE dataset (Suter et al., 2022), tracking 91 glioblastoma (GBM) patients who underwent a pre-operative scan followed by repeated post-operative scans over up to five years. This resulted in 795 longitudinal image series, each comprising 2–18 time points. This also comes with segmentation labels for necrosis, contrast enhancement, and edema.

Starmen. The public synthetic Starmen dataset comprises 1 000 sequences of 10 images each and is commonly used to benchmark longitudinal frameworks (Bône et al., 2018). In every sequence, the sole temporal change is the raising of the left arm, with each subject's motion encoded via an affine time parameterization: $t^* = \alpha\,(t - \tau)$,

where α and τ are subject-specific parameters. To introduce additional variability, sequences are randomly rotated (uniformly between $-10°$ and $10°$) and translated by up to ± 6.8 pixels. Of the 1 000 sequences, 400 are reserved for training, 100 for validation, and the remaining 500 for testing. The ground-truth progression values have a mean of -0.12 and a standard deviation of 4.25. Finally, we augment the original dataset by adding two more classes of motion. From only a hand going up motion for each of the splits, we create an equal number of trajectories for two more classes (Hand-downward motion, and static) by reversing the order of the trajectory and replicating the first image 10 times over.

D Evaluation Metrics

We assess the performance of the model using three primary evaluation metrics: image similarity, residual magnitude, and regions of interest (ROI) similarity.

Image Similarity: Image similarity between the real future image x_{t_j} and the predicted future image $\hat{x}_{t_j}$ is quantified using two widely recognized metrics: Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index Measure (SSIM). These metrics are standard for image-to-image tasks, including super-resolution, denoising, and inpainting. PSNR is a logarithmic metric that normalizes the Mean Squared Error (MSE) between two images using their dynamic range. It is defined as:

$$\mathrm{PSNR}(x_a, x_b) = 10 \log_{10} \left(\frac{R^2}{\mathrm{MSE}(x_a, x_b)} \right) \tag{53}$$

where R is the common dynamic range of the images. The Mean Squared Error (MSE) is calculated as:

$$\mathrm{MSE}(x_a, x_b) = \frac{1}{H \times W} \sum_{h \in H, w \in W} \| x_{a(h,w)} - x_{b(h,w)} \|^2 \tag{54}$$

SSIM measures the perceptual similarity between two images, capturing structural changes. The formula is defined as:

$$\mathrm{SSIM}(x_a, x_b) = \frac{(2\mu_{x_a}\mu_{x_b} + c_1)\,(2\sigma_{x_a x_b} + c_2)}{\left(\mu_{x_a}^2 + \mu_{x_b}^2 + c_1\right)\left(\sigma_{x_a}^2 + \sigma_{x_b}^2 + c_2\right)} \tag{55}$$

where μ_{x_a} and μ_{x_b} are the pixel sample means, $\sigma_{x_a}^2$ and $\sigma_{x_b}^2$ are the variances, $\sigma_{x_a x_b}$ is the covariance of x_a and x_b, and $c_1 = (0.01R)^2, c_2 = (0.03R)^2$ are constants for numerical stability.

Residual Magnitude: We also assess the magnitude of residual differences between the predicted and real images using Mean Squared Error (MSE).

Region of Interest (ROI) Similarity: To accurately capture the ventricle for ADNI and lesion/tumor regions, we use two primary metrics: Dice Similarity Coefficient (DSC) and Hausdorff Distance (HD). These metrics are computed on the binarized atrophy region masks of the real future image x_{t_j} and the predicted future image $\hat{x}_{t_j}$. The DSC and HD are defined as follows, respectively:

$$\mathrm{DSC}(X, Y) = \frac{|X \cap Y|}{|X| + |Y|} \tag{56}$$

$$\mathrm{HD}(X, Y) = \max \left(\sup_{x \in X} d(x, Y), \sup_{y \in Y} d(X, y) \right) \tag{57}$$

E Implementation Details

E.1 Architecture

Flow Regressor: The flow regressor network for our proposed model, IMMFM, is built upon the U-ViT (Yu et al., 2022) architecture; we specifically adapt the implementation from (Davtyan et al., 2023). The network consists of 14 standard ViT(Dosovitskiy et al., 2021) blocks. These are interconnected by 4 long skip connections that link the first 4 blocks to the last 4 blocks. Along each skip connection, feature maps are channel-wise

concatenated and subsequently projected to the inner dimension of the ViT blocks. Within each ViT block, Layer Normalization is applied before both the Multihead Self-Attention (MHSA) layer and the subsequent MLP. The inner dimension for all ViT blocks is 512, and 8 heads are used in every MHSA layer. For processing the inputs first, the input image x_{t_i} and conditioning $c = x_{t_{i-1}}$ are channel-wise concatenated and then linearly projected to the inner dimension of the ViT blocks. As for non-imaging conditioning, such as demographics (e.g., sex, age, etc), we use separate projection blocks to project them to the inner dimension and subsequently add them to the imaging inputs. If no preceding image exists, a standard prior (zero vector) was used. Before passing it to the ViT blocks, we add learned positional embedding and a sinusoidal time embedding (Vaswani et al., 2017) of corresponding time t_i and t_{i-1} of image latents x_{t_i} and x_{t_i}. Finally, the network outputs velocity v_θ, score s_θ, and uncertainty g_θ.

Auto Encoder: For encoding the image to the latent space, we use the UNet architecture proposed in (Dhariwal and Nichol, 2021). However, we drop the long skip connection from encoder to decoder to transform it into an autoencoder so that all the information of the input image is contained within a single latent vector. For the Starman dataset, we use a latent dimension of 256, and for all the clinical datasets, we use 4096.

Segmentation Network: For ROI segmentation, we trained three auxiliary image segmentation networks, each tailored to one of the datasets. These networks follow a UNet (Isensee et al., 2021) architecture.

E.2 Data Preprocessing and Augmentation

For all three clinical datasets, we register the 3D volumes to have spatial alignment in each trajectory. For this, we use ANTS (Avants et al., 2009) to perform Affine followed by Diffeomorphic registration to align each scan towards the first scan in the sequence. To evaluate the computational cost of this OT alignment step, we measured processing times using an Intel Xeon Gold 5118 (2017) CPU running at 2.30Ghz. The full preprocessing pipeline for the ADNI dataset takes approximately 6 hours at a resolution of $256 \times 256 \times 128$ and 1 hour for $128 \times 128 \times 64$. For the GBM dataset, preprocessing takes roughly 9 hours for $256 \times 256 \times 128$ and 2.5 hours for $128 \times 128 \times 64$. Registering a single volume is very fast, taking roughly 6.85 seconds for $256 \times 256 \times 128$ and 1.12 seconds for $128 \times 128 \times 64$. These low inference costs make the approach practical for real-time inference. We emphasize that MRI images follow standardized protocols and are already reasonably aligned. As a result, the registration requires only a few iterations to converge. In imaging domains without such protocols, the initial misalignment could be larger, consequently requiring more iterations and increasing the computational cost.

Following registration, we employed two distinct processing pipelines. For our 2D analysis, we extracted all axial slices containing the primary region of interest (e.g., ventricles in ADNI), a selection guided by the provided ground truth segmentation masks. Each of these 2D intensity slices was then resized to 256×256 pixels using cubic interpolation. For our 3D analysis, we used the entire registered volume, resizing each intensity volume to $128 \times 128 \times 128$ voxels with cubic interpolation. All corresponding segmentation mask volumes were resized using nearest-neighbor interpolation to preserve label integrity.

The timeline for each trajectory, originally in days and weeks, was scaled to a range between 0 and 1 by dividing by the maximum duration of the respective dataset. For the Starmen dataset, no registration was needed. We employed a two-stage augmentation strategy. During autoencoder pre-training, we used standard image-level augmentations, including flipping, shifting, scaling, and rotation. However, for the main Flow Matching (FM) training, no image-level augmentation was used. Instead, we performed trajectory augmentation by subsampling. For a given sequence of M visits, we generated additional training samples by creating all possible contiguous sub-trajectories of shorter lengths, while strictly preserving the temporal order of the images.

We performed a subject-level partition for all datasets to prevent data leakage, ensuring all data from a single subject remained in the same set. For the larger ADNI (317 subjects) and GBM (91 subjects) cohorts, we used a %70%10%/20% split, resulting in approximately 220/30/67 (train/val/test) subjects for ADNI and 60/8/23 for GBM. For the smaller MS dataset (19 subjects), we performed a 5-fold cross-validation; the results reported correspond to a representative fold with 12 training and 7 test subjects. For each subject, their single 3D volume trajectory was expanded into 20 to 100 2D slice-based trajectories, significantly increasing the number of samples for training.

E.3 Training Details

Autoencoder Training. The first stage of our pipeline involves training an autoencoder to learn a compact latent representation of the 2D images. The architecture is based on the U-Net from (Dhariwal and Nichol, 2021), but with the long skip connections between the encoder and decoder removed to ensure a compressed latent bottleneck. The network incorporates residual layers with convolutions and multi-head self-attention layers. The latent space dimension was set to 4096 for clinical datasets and 256 for the Starmen dataset. The model was trained for 100 epochs using a hybrid loss function formulated as $\mathcal{L} = \text{MSE} + 0.5 \cdot (1 - \text{SSIM})$. Upon completion of training, the autoencoder weights were frozen.

Segmentation Network Training. For downstream tasks requiring biomarker quantification, we trained an auxiliary segmentation network. This network, based on a standard U-Net architecture (Isensee et al., 2021), was trained for 100 epochs using a binary cross-entropy loss on the ground truth annotations provided with each dataset. Crucially, the input images for this network were first passed through the complete, pre-trained autoencoder (encoder followed by decoder). This step ensures the segmentation model learns to operate on images that have the same distributional characteristics as our model's generated outputs.

Flow Matching Model Training. In the second stage, the IMMFM model was trained to learn the progression dynamics directly within the latent space. For each training step, input trajectories were first passed through the frozen encoder to obtain their corresponding latent representations. The IMMFM model was then trained for a longer duration, typically between 350 and 500 epochs. During inference, the trained IMMFM model operates on the latent vectors to produce a forecasted latent state, which is subsequently transformed back into the pixel domain by the pre-trained decoder. Key hyperparameters for all models are detailed in Table 2 and 3 for 2D and 3D models, respectively. Note that we use 8 ViT blocks for the MS dataset and 14 ViT blocks for the GBM and ADNI datasets.

Table 2: Model Hyperparameters for 2D version.

Hyperparameter	**Autoencoder**	**Segmentation Net**	**Flow Regressor**
Model Size	∼38M	∼3M	∼40-60M
Input Channels	1	1	-
Image Size	256×256	256×256	4096
Architecture	CNN + Attention	U-Net	ViT-based
Transformer Blocks	-	-	8-14 ViT Blocks
Skip Connections	-	Yes	Yes
ViT Inner Dimension	-	-	512
Channels	64	16	-
Channel Multiple	2,4,4,4,4,4	1,2,4,8,16	-
Residual Blocks per Down Block	1	2	-
Channels / Attention Heads	8	-	8
Attention Resolution	64,32,16,8,4,2,1	-	-
Dropout	0.0	0.0	0.0
Batch Size	24	24	24
Epochs	100	150	350–500
Warmup Epochs	25	25	25
Learning Rate	1×10^{-3}	1×10^{-4}	1×10^{-4}

E.4 Computing Infrastructure and Cost

All experiments were performed on Snellius, the Dutch national supercomputer. Each training job was allocated a node equipped with one NVIDIA A100 GPU (with 40GB VRAM) and 8 CPU cores. Typical training durations for our primary models on this configuration were as follows:

- Autoencoder: 12–16 hours.

Table 3: Model Hyperparameters for 3D version.

Hyperparameter	Autoencoder	Segmentation Net	Flow Regressor
Model Size	∼45M	∼5M	∼75M
Input Channels	1	1	-
Input Volume Size	$128 \times 128 \times 128$	$128 \times 128 \times 128$	131,072
Architecture	3D CNN + Attention	3D U-Net	ViT-based
Transformer Blocks	-	-	14 ViT Blocks
Skip Connections	-	Yes	Yes
ViT Inner Dimension	-	-	512
Channels	32	16	-
Channel Multiple	2,4,4,8,8	1,2,4,8,16	-
Residual Blocks per Down Block	1	2	-
Channels / Attention Heads	8	-	8
Attention Resolution	64,32,16	-	-
Dropout	0.0	0.0	0.0
Batch Size	4	4	2
Epochs	100	150	350–500
Warmup Epochs	25	25	25
Learning Rate	2×10^{-4}	2×10^{-4}	1×10^{-5}

- Segmentation Network: 6–8 hours.
- IMMFM: 12–18 hours.

In comparison to our proposed models, some baseline methods exhibited greater computational demands. Notably, the ImageFlowNet baseline consistently required a significantly longer training period, taking approximately 2x longer when executed on similar hardware. The total computational resources utilized for developing our models, conducting all experiments, and performing baseline comparisons in this study amounted to approximately 3000 GPU hours. Our implementation primarily relies on PyTorch. Note that both 2D and 3D experiments took similar time due to a reduction in dataset samples when treated as a 3D volume.

F Additional Experimental Results

F.1 Ablation Study of Proposed Components

To evaluate our methodological contributions, we conducted an ablation study to isolate and quantify the impact of each component of our proposed IMMFM framework. Table 4 shows the performance gains from the following components: the quadratic conditional path, the data-driven diffusion coefficient, and conditioning on the previous frame. Below, we describe the experimental setup for each ablation:

Previous-Frame Conditioning. To assess the importance of immediate temporal context, we trained our simplest model variant, O-IMMFM, without conditioning on the latent representation of the preceding frame. All other aspects of the experimental setup, including architecture and training hyperparameters, remained unchanged.

Quadratic Path. To measure the benefit of incorporating second-order temporal dynamics, we compared our ODE-based model, O-IMMFM (using the proposed quadratic path), against a baseline with a conventional linear path that connects consecutive marginals with straight lines. This isolates the performance gain attributable to our novel quadratic path construction.

Learned Diffusion Coefficient. To quantify the effect of a learned diffusion term, we compared the full SDE-based model SU-IMMFM against S-IMMFM, which uses a fixed, predefined diffusion schedule. Both variants share identical architectures, isolating the impact of using a data-driven approach to modeling stochasticity.

The results of our ablation study highlight the value of each proposed component. Introducing the quadratic path yields the most substantial improvements, increasing Dice Score by up to 3.7% and PSNR by over 2.0 dB. Similarly, incorporating a learned diffusion coefficient consistently improves performance on the most challenging GBM dataset, notably reducing Hausdorff Distance by over 6.2 pixels and increasing Dice Score by 1.5%. Finally, conditioning on the previous frame improves performance as well, boosting the Dice Score by up to 2.1% and PSNR by over 1.0 dB

Table 4: Ablation study results quantifying the contribution of each model component. The table reports the change in performance metrics (averaged over 3 seeds) from our ablation study. Each column represents the performance gain from a specific component.

Dataset	Metric	Prev. frame conditioning	Quadratic path	Data diffusion
ADNI	PSNR ↑	1.089	1.441	0.084
	SSIM ↑	0.021	0.012	0.001
	MSE ↓	0.001	>0.001	>0.001
	DSC ↑	0.021	0.027	0.002
	HD ↓	1.198	2.16	0.671
Brain MS	PSNR ↑	0.940	2.098	0.146
	SSIM ↑	0.032	0.015	0.000
	MSE ↓	0.000	>0.001	>0.001
	DSC ↑	0.007	0.030	0.004
	HD ↓	0.140	0.490	0.140
Brain GBM	PSNR ↑	1.092	1.696	0.467
	SSIM ↑	0.013	0.025	0.012
	MSE ↓	0.001	>0.001	>0.001
	DSC ↑	0.015	0.037	0.015
	HD ↓	1.030	1.523	6.290

While global metrics such as SSIM and MSE show minimal changes (e.g., SSIM improves by only 0.01–0.02), this is expected, as they are often saturated due to the already high image reconstruction quality and are less sensitive to localized changes in regions of interest (ROIs). For example, small volumetric changes in structures like the ventricles affect only a small fraction of the image, having a negligible influence on overall MSE or SSIM. Since our primary goal is to model clinically relevant regional evolution, improvements in ROI-specific metrics such as Dice and Hausdorff Distance provide stronger evidence of our model's effectiveness.

F.2 Analysis of Temporal Generalization

An important aspect of any forecasting model is understanding how its predictive accuracy changes as the forecast horizon increases. To assess the reliability of our model for both short-term and long-term performance, we evaluate its key metric at various future time points. Fig. 6 illustrates this temporal generalization by plotting Dice score, PSNR, and Hausdorff against the increasing prediction interval, quantifying the expected degradation in accuracy as the model predicts further into the future.

The results show an expected degradation in performance as the forecast horizon extends, with Dice Score and PSNR decreasing while Hausdorff Distance increases. The severity of this degradation differs across datasets; on the ADNI dataset, the Dice Score degrades by approximately ∼3% over three years, while the Brain MS dataset sees a similar drop of ∼4%. However, the Brain GBM dataset exhibits a much sharper decline in performance, with up to ∼ 13%. This is likely due to the difficult and sporadic nature of glioblastoma progression, which is highly unpredictable and challenging to model accurately, particularly with limited training data.

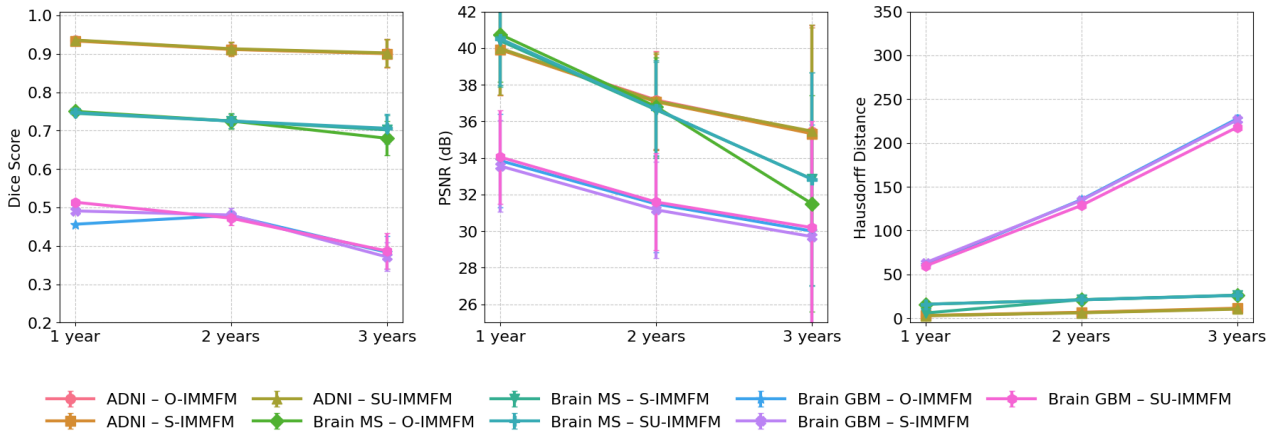

Figure 6: Performance for increasing forecast horizon for ADNI, MS, and GBM datasets. The error bar represents the average inter-subject variability of the datasets.

F.3 Generalization to Volumetric Data

To demonstrate the generalization capability of our framework to volumetric data, we conducted our primary experiments on the full 3D clinical datasets. The experimental setup, including dataset partitions and training protocols, remained identical to the 2D experiments. The core architecture of our flow regressor was also unchanged. To handle the volumetric data, we modified the autoencoder and the auxiliary segmentation network by replacing their 2D convolutional layers with their 3D counterparts. Following spatial alignment, the full 3D volumes were directly processed by these models to create and analyze the latent space trajectories. All the evaluation metrics that are defined in D were generalized to work on 3D data for computing the performance. More details can be found about data pre-processing and training in E.2 and E.3.

Table 5: Trajectory forecasting performance of the 3D models. Results are averaged over three runs and *all available timepoints*. Values are presented as Mean, with the subscript in green indicating inter-subject variability and the superscript in purple indicating inter-model variability.

Dataset	Metric	O-IMMFM	S-IMMFM	SU-IMMFM
ADNI	PSNR ↑	$41.07^{\pm 0.11}_{\pm 2.84}$	$40.98^{\pm 0.11}_{\pm 2.80}$	$41.15^{\pm 0.20}_{\pm 2.84}$
	SSIM ↑	$0.980^{\pm 0.001}_{\pm 0.039}$	$0.979^{\pm 0.002}_{\pm 0.039}$	$0.980^{\pm 0.001}_{\pm 0.039}$
	MSE ↓	$0.001^{\pm 0.001}_{\pm 0.004}$	$0.001^{\pm >0.001}_{\pm 0.004}$	$0.001^{\pm 0.000}_{\pm 0.004}$
	DSC ↑	$0.938^{\pm 0.009}_{\pm 0.077}$	$0.942^{\pm 0.004}_{\pm 0.081}$	$0.947^{\pm 0.005}_{\pm 0.070}$
	HD ↓	$2.889^{\pm 0.054}_{\pm 2.687}$	$3.021^{\pm 0.231}_{\pm 2.777}$	$2.877^{\pm 0.052}_{\pm 2.692}$
Brain GBM	PSNR ↑	$34.38^{\pm 0.31}_{\pm 5.13}$	$34.63^{\pm 0.26}_{\pm 4.90}$	$34.15^{\pm 0.61}_{\pm 4.03}$
	SSIM ↑	$0.923^{\pm 0.001}_{\pm 0.181}$	$0.923^{\pm 0.002}_{\pm 0.167}$	$0.935^{\pm 0.001}_{\pm 0.193}$
	MSE ↓	$0.001^{\pm >0.001}_{\pm 0.001}$	$0.001^{\pm >0.001}_{\pm 0.001}$	$0.001^{\pm >0.001}_{\pm 0.001}$
	DSC ↑	$0.478^{\pm 0.009}_{\pm 0.152}$	$0.480^{\pm 0.004}_{\pm 0.161}$	$0.489^{\pm 0.005}_{\pm 0.150}$
	HD ↓	$127.61^{\pm 2.054}_{\pm 28.19}$	$124.40^{\pm 1.231}_{\pm 29.91}$	$121.11^{\pm 1.520}_{\pm 27.89}$

Our results in Table 5 show that the 3D models achieve a notable improvement in forecasting accuracy when compared to their 2D counterparts in Table 1. For the ADNI dataset, the SU-IMMFM variant shows marked improvements across the board: the Dice Similarity Coefficient (DSC) increases from 0.920 to 0.947 (an *improvement of 2.9%*), the PSNR rises from 37.52 to 41.15, and the Hausdorff Distance drops significantly from 6.50 to 2.88.

A similar trend is observed for the Brain GBM dataset, where the SU-IMMFM model improves the DSC from 0.460 to 0.489, a more pronounced *increase of 6.3%*, and reduces the Hausdorff Distance from 135.08 to 121.11. Across both datasets, not only did the average performance improve, but the inter-subject variability (i.e., the standard deviation) also decreased across most metrics, indicating more consistent predictions.

This performance gain can be attributed to the 3D autoencoder's ability to leverage inter-slice spatial context. By processing the entire volume, it learns a richer latent representation that encodes the full 3D anatomical structure. This is crucial for accurately modeling volumetric, disease-related changes, such as ventricular enlargement in ADNI or the complex, infiltrative growth patterns of tumors in GBM, rather than treating them as disconnected 2D area changes. It should be noted that the Brain MS dataset was excluded from the 3D experiments, as its limited size was insufficient for effectively training the higher-capacity 3D models.

G Additional Methodological Details

G.1 ADNI Dataset

G.1.1 Measuring Overlap

To quantitatively assess the separation between our AD and CN populations, we fit Gaussians to the ventricular area estimates and measure the overlap between their respective distributions using the Overlap Coefficient (OVL) (Inman and Bradley Jr, 1989). For two Gaussian distributions with means μ_1, μ_2 and standard deviations σ_1, σ_2, when the variances are unequal, the calculation must account for the two intersection points where the probability density functions meet. The intersection points (c_1, c_2) are determined by:

$$(c_1, c_2) = \frac{\mu_1\sigma_2^2 - \mu_2\sigma_1^2 \pm \sigma_1\sigma_2\sqrt{(\mu_1 - \mu_2)^2 + (\sigma_2^2 - \sigma_1^2)\ln(\sigma_2^2/\sigma_1^2)}}{\sigma_2^2 - \sigma_1^2} \tag{58}$$

The overlap coefficient is then calculated as:

$$\text{OVL} = [1 - F_1(c_1) + F_2(c_1)] - [F_2(c_2) - F_1(c_2)] \tag{59}$$

where F_1 and F_2 are the cumulative distribution functions of the respective Gaussian distributions. When both distributions have equal variance, the OVL simplifies to $2\Phi(-|\mu_1 - \mu_2|/\sqrt{2\sigma^2})$, where Φ is the standard normal CDF. This value ranges from 0 (completely separated distributions) to 1 (identical distributions), providing an intuitive measure of classification difficulty (Reiser and Faraggi, 1999).

G.1.2 Classification Methodology

We performed a binary classification to differentiate Alzheimer's Disease (AD) from Cognitively Normal (CN) subjects, leveraging the normalized ventricular area as a key biomarker. The distinct distributional characteristics of this biomarker between AD and CN populations, which can be analyzed by fitting Gaussian models, motivate its use for this classification task.

The specific feature utilized for classification is the normalized ventricular area, evaluated for each subject in the main test set at two distinct timepoints:

- An early observed timepoint, t_{second} (corresponding to their second clinical visit).
- A future timepoint, t_{last} (normalized time $t = 1$, approximately 36 months from baseline), with the ventricular area forecasted from our IMMFM model's predictions.

For this specific classification experiment, the main test set was further randomly partitioned to create internal "threshold-training" and "threshold-evaluation" subsets. To assess the sensitivity of our classification results to the size of these internal partitions, we explored several split ratios. Specifically, we used proportions of 25%/75%, 50%/50%, and 75%/25% for allocating main test set subjects to the threshold-training versus threshold-evaluation subsets, respectively. For each of these configurations, this partitioning ensured that the determination

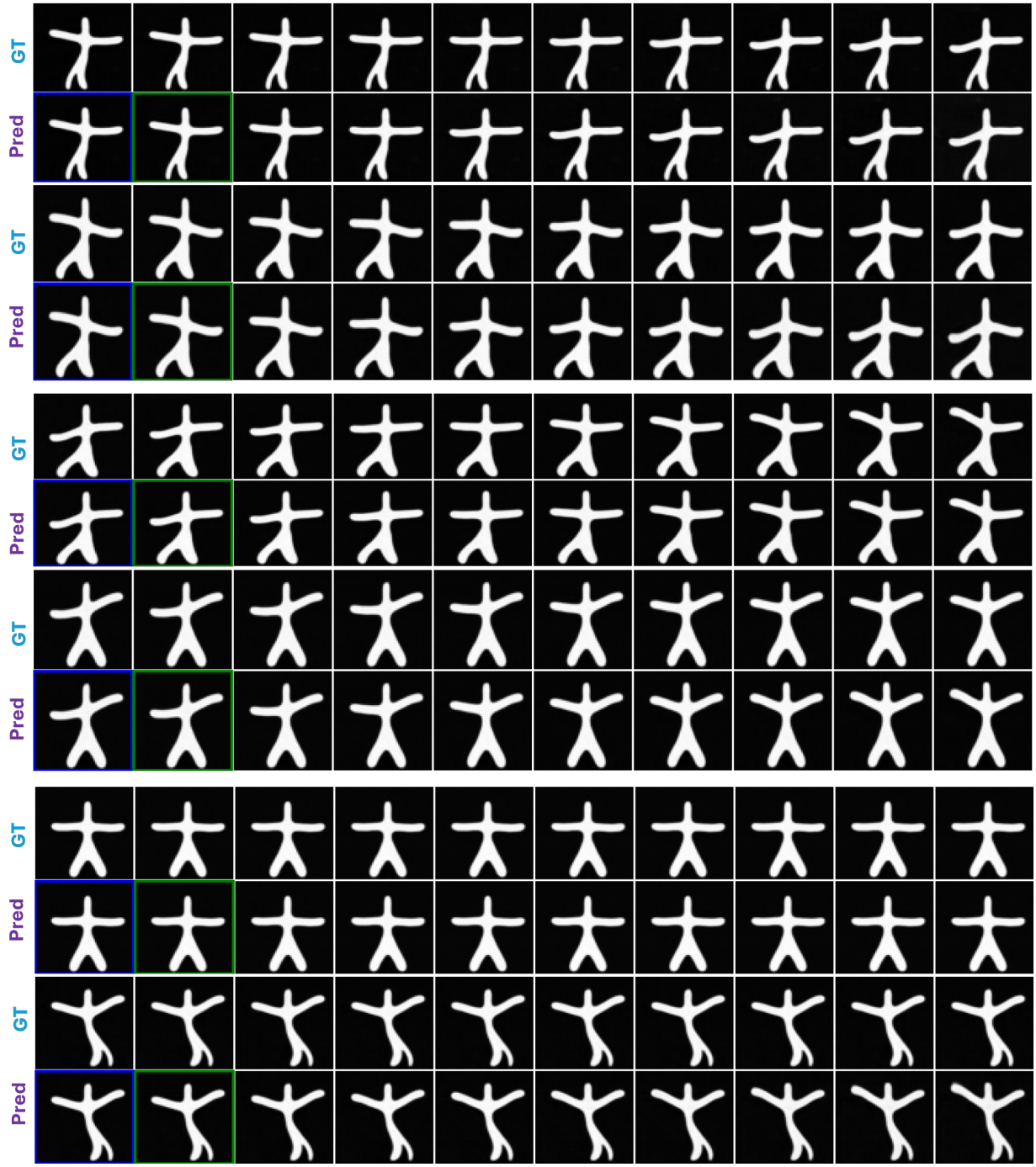

Figure 7: Trajectory simulation on Starmen dataset. The conditioning frame is marked with green, and the reference starting frame is marked with blue. From top to bottom blocks: *Hand-downward motion, Hand-upward motion, Static.*

of the optimal classification threshold and its subsequent performance assessment were conducted on entirely separate (non-overlapping) subsets.

While the analysis of distributional overlap (as detailed in Appendix G.1.1) involves identifying intersection points of fitted Gaussian distributions to understand theoretical separability, for the practical task of classifying individual subjects, we determined the decision threshold empirically to optimize predictive performance.

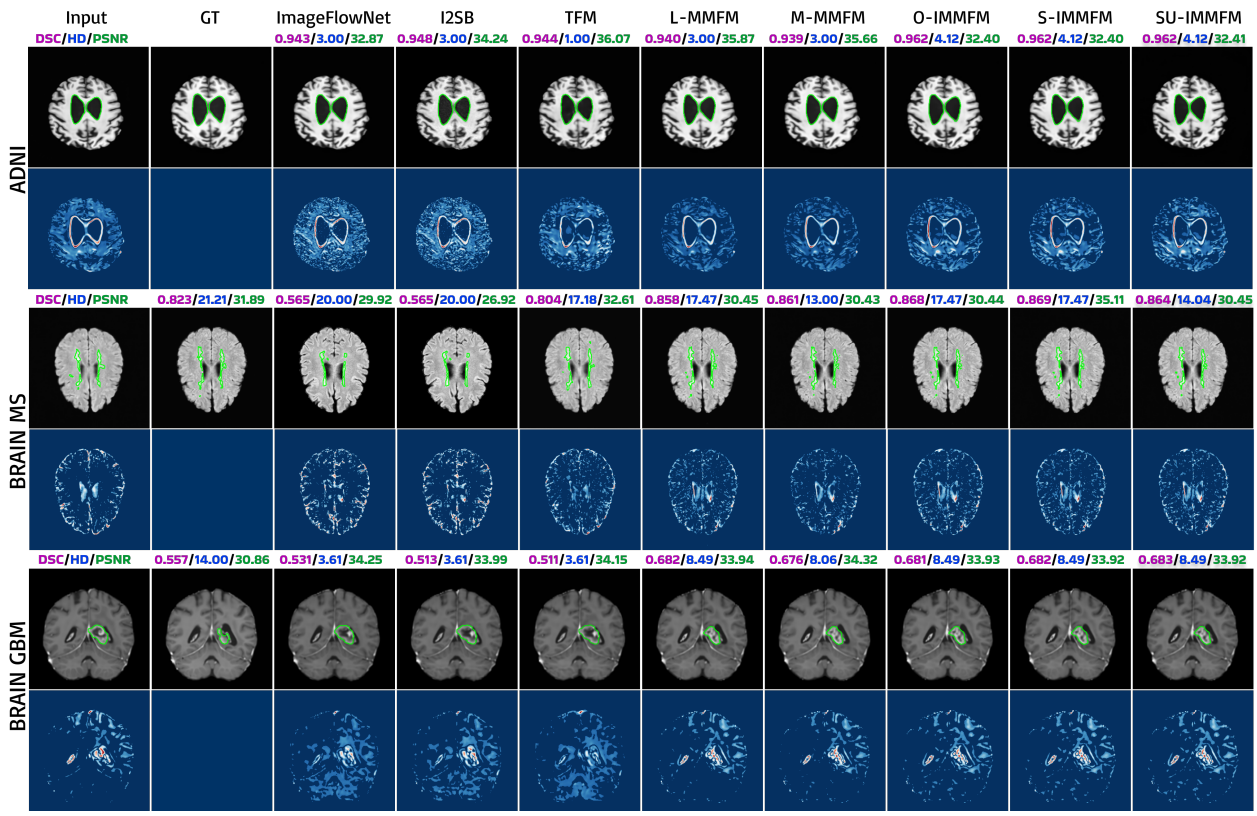

Figure 8: Additional Visual comparison of forecasting results on the ADNI (∼3yr), MS (∼3yr), and GBM (∼1yr) datasets. For each dataset, the first row displays our model's forecasted image. The second row shows the corresponding pixel-wise difference map between our forecast and the ground truth.

Table 6: Classification Results for ADNI on test set with varying split-ratio

Train/Test	Second Timepoint	Last Timepoint*	Acc. Gain
50/50 %	67.5 %	75.1%	8.6%
75/25 %	71.7%	80.8%	9.1%
Average	69.6%	78.0	8.4%

Using data solely from the "threshold-training" subset, an optimal decision threshold for the normalized ventricular area was identified. This was achieved by employing Receiver Operating Characteristic (ROC) curve analysis. The threshold selected was the one that maximized the accuracy in distinguishing AD from CN subjects. The classification threshold learned from the "threshold-training" subset was then applied to the "threshold-evaluation" subset to assign AD or CN labels to its subjects.

The detailed classification outcomes for different split ratios are presented in Table G.1.2.

G.2 On the Calibration of Learned Diffusion

In Section 3.2, we introduced a data-driven diffusion coefficient, $g_\theta(t, x, c)$, which is trained to match the squared error of the predictive construction. While traditional uncertainty quantification often evaluates calibration to yield explicit predictive variances or confidence intervals in the observation domain, the learned diffusion term in the IMMFM framework serves a structurally different purpose. Rather than acting as a standalone, classical probabilistic uncertainty estimate, g_θ is learned jointly with the drift to actively drive the SDE trajectory. It functions as a dynamic, directional corrective term that modulates the stochasticity of the latent-space state updates to guide the generated trajectory toward better reconstruction.

Furthermore, this diffusion coefficient operates entirely within a high-dimensional, compressed latent space (e.g.,

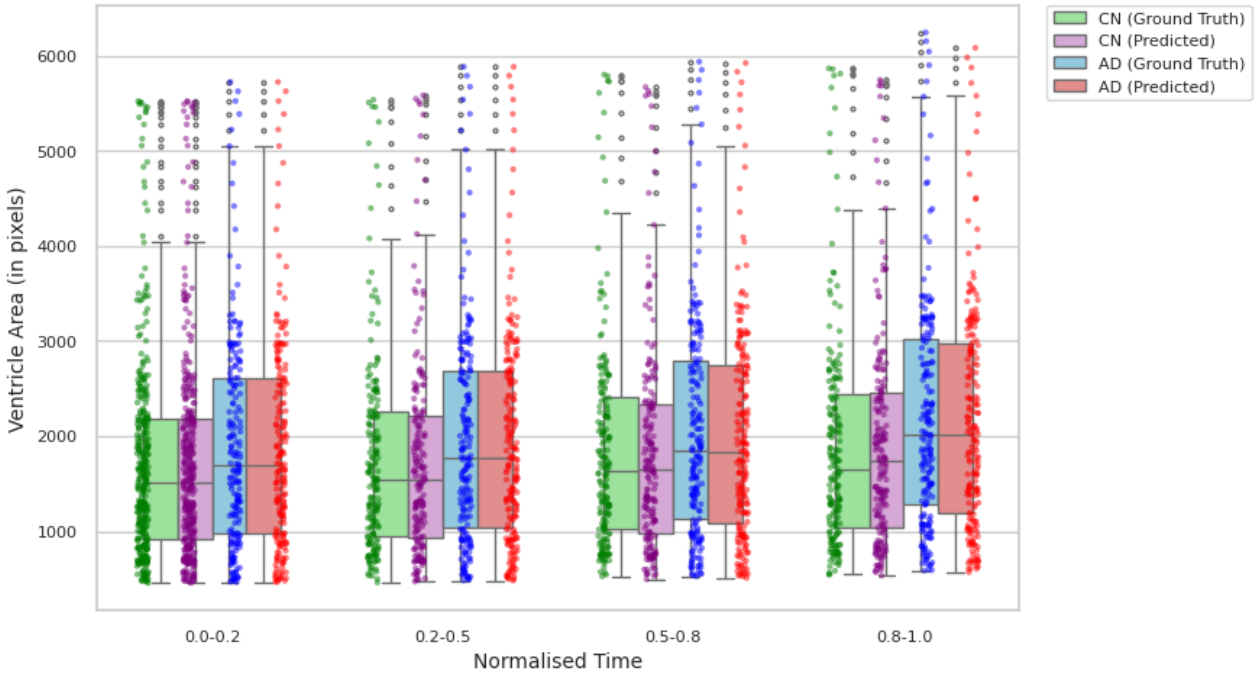

Figure 9: Distribution of ventricular region over time for Alzheimer's (AD) and Cognitively Normal (CN), binned into four discrete time points.

$d = 4096$ for the clinical cohorts). Because this latent manifold is optimized for dense image representation rather than explicit feature disentanglement, the latent axes are not independently physically interpretable. Consequently, a unit of variance in this latent space does not translate to an interpretable spatial or anatomical confidence interval in the final pixel or voxel domain. Standard calibration analyses, which rely on mapping predictive variance to observable, physical prediction errors, are therefore not directly applicable to this latent corrective mechanism.

As such, we assess the validity and quality of the learned diffusion term through its downstream empirical impact rather than through classical calibration metrics. If the learned diffusion g_θ did not accurately capture the structural uncertainty of the trajectory, it would inject uncalibrated noise and degrade the generated synthesis. Instead, as demonstrated in our ablation study (Appendix F.1), incorporating the data-driven diffusion coefficient systematically improves predictive fidelity. This is most notable in highly stochastic datasets such as the Brain GBM cohort, where it significantly reduces the Hausdorff distance and increases the Dice score. This consistent reduction in downstream forecasting error confirms that the learned term is effectively aligned with the model's prediction error and successfully regularizes the learned dynamics.

H Limitations and Future Directions

Limitations. The use of a quadratic interpolation scheme introduces a specific inductive bias that can be characterized as a low-pass filtering effect. Because the conditional path is constructed as a quadratic segment with lookahead, the velocity evolution (Eq. 38) is linear, modeling local dynamics as smooth trajectories with stable curvature. While this construction inherently limits the model's ability to capture "jerky" transitions or high-frequency oscillations (non-zero third derivatives) between sparse observations, this behavior functions as a critical implicit regularization rather than a purely restrictive drawback. In chaotic or highly stochastic regimes, learning a continuous probability flow from sparse data is frequently ill-posed; by providing a smooth target path, we ensure the resulting vector field maintains the Lipschitz continuity required for stable flow matching and avoid the numerical instabilities or overfitting typically associated with the Runge phenomenon (Lipman et al., 2022). Furthermore, this smoothing does not imply that complex dynamical variations are discarded. Within the SU-IMMFM framework, we hypothesize a functional decomposition where the quadratic drift captures the

tractable mean trend, while the "missed" chaotic volatility is absorbed and represented by the learned diffusion component g_θ. In this sense, the deviation of the true chaotic path from the smooth quadratic target is explicitly accounted for by the stochastic layer, allowing the model to maintain global tractability without sacrificing the representation of local volatility.

While IMMFM demonstrates robust performance, its accuracy can be influenced by severe, systematic artifacts in input data, and its predictive scope is shaped by the diversity of trajectories within the training set. These considerations motivate several methodological extensions to enhance the framework's power and versatility.

Future Directions. A primary direction is to learn more informative latent representations. This can be achieved by developing *temporally aware autoencoders*, which move beyond processing snapshots independently and instead employ ordering-aware training objectives or architectural priors to ensure latent space continuity and coherence (Yang et al., 2023a; Blattmann et al., 2023). Such representations would provide a stronger foundation for several advanced applications. One is enriching the dynamics via *multi-modal conditioning*, allowing the model to integrate heterogeneous data like static covariates or external signals to learn more disentangled and explanatory trajectories (Shaik et al., 2024; Wu et al., 2024). Another is extending the framework's generative capabilities towards *causal and counterfactual reasoning*. By integrating principles of causal representation learning, the model could simulate trajectories under hypothetical interventions, transforming it from a prognostic tool into a system for decision support (von Kügelgen et al., 2024).

Finally, to improve plausibility and generalization in data-scarce settings, the model can be fortified with *domain-specific knowledge*. In domains with well-established governing equations, integrating frameworks from scientific machine learning, such as Physics-Informed Neural Networks (PINNs), can strictly constrain the learned dynamics (Qian et al., 2025; Cuomo et al., 2023). However, we note that in complex clinical contexts such as Alzheimer's disease progression, universally accepted physical laws might not exist. In such cases, this direction is more speculative; rather than enforcing strict physical laws, the framework could instead be adapted to incorporate approximate mechanistic priors, such as network diffusion models of pathology spread, to mildly regularize the data-driven trajectories.