HyperNSDE: Personalized Neural SDEs for Joint Static–Longitudinal Clinical Data Generation

Perrine Chassat
Inria, Université Paris Cité, Inserm, HeKA
F-75015 Paris, France
perrine.chassat@inria.fr

Agathe Guilloux
Inria, Université Paris Cité, Inserm, HeKA
F-75015 Paris, France
agathe.guilloux@inria.fr

Abstract

Synthetic patient data generation is a promising solution to the dual challenge of data scarcity and privacy constraints in healthcare machine learning. Realistic synthesis of patient-level clinical data requires jointly modeling heterogeneous static covariates, irregularly sampled longitudinal trajectories, and informative observation times — three tightly coupled components in practice yet rarely addressed together. We propose *HyperNSDE*, a continuous-time generative model that conditions a latent Neural SDE on static patient representations through a hypernetwork, allowing baseline characteristics to shape trajectory evolution beyond the initial condition without requiring a trajectory encoder, while stochastic latent dynamics capture realistic variability in generated paths. Observation times are modeled jointly through a latent-state-dependent intensity process, and training on irregular stochastic paths is stabilized via a deterministic–stochastic path decomposition with a non-adversarial signature-kernel objective. Experiments on simulated and real clinical datasets show improved observation-time fidelity and competitive performance, while matched-grid analyses reveal that forecasting and correlation metrics are affected by observation-grid regularity and trajectory smoothness.

1 Introduction

Clinical cohorts are expensive to collect, difficult to share due to privacy regulations, and often too small to support reliable machine learning models [9]. Yet many healthcare applications require realistic patient-level data at scale: benchmarking predictive methods, simulating trial populations, studying rare subgroups, or augmenting underrepresented cohorts [58, 9, 47]. Synthetic data generation is therefore a promising approach to this concrete clinical challenge, provided that generated samples preserve the multimodal structure, heterogeneity, and variability of real patient records.

The structure of clinical data makes this a fundamentally challenging generative modeling problem, going beyond the generic time-series generation. Clinical records combine heterogeneous static baseline covariates — demographics, comorbidities, treatment history — with longitudinal measurements collected over irregular, patient-specific time grids. These components are not independent: baseline characteristics influence disease progression, observation times are often informative of the underlying health state, and part of the trajectory variability can reflect latent or unmeasured factors that are not fully explained by observed covariates. A realistic generative model should therefore capture the joint distribution of static features, longitudinal trajectories, and observation patterns, while accounting for both cross-subject heterogeneity and residual temporal variability.

Existing approaches only partially address this setting. Most generative models for time series operate in discrete time and assume regularly sampled sequences [66, 17, 32, 31], making them ill-suited to irregular clinical trajectories. Continuous-time models [52, 10] provide a natural framework for irregular data, but generally do not address joint static–longitudinal generation. The closest prior

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

work in this setting, MultiNODEs [63], combines a variational model for static covariates with a latent Neural ODE. However, static information affects the model mainly through the initial latent state rather than the full dynamics, latent initialization depends on the full observed trajectory rather than only baseline information, the latent dynamics remain deterministic, and informative observation times are not modeled. These limitations motivate a different design for realistic synthetic clinical trajectory generation.

We propose **HyperNSDE** (**Hyper**network **N**eural **SDE**), a unified continuous-time generative model that addresses these limitations jointly, by combining a generative model for static covariates with a personalized Neural SDE whose dynamics are conditioned on the patient's static latent representation through a hypernetwork. Observation times are modeled through a latent-state-dependent intensity process, and training is stabilized via a deterministic–stochastic path decomposition with a signature-kernel objective. Across simulated and real clinical datasets, HyperNSDE improves observation-time fidelity and several distributional and privacy metrics, while forecasting and correlation performance is more heterogeneous; controlled matched-grid analyses investigate these differences.

Contributions

- **Unified generative model for clinical data:** HyperNSDE is a holistic continuous-time model for the joint generation of heterogeneous static covariates, irregular longitudinal trajectories, and informative observation times, without requiring trajectory encoding or pre-imputation.
- **Stochastic latent dynamics for realistic trajectory variability:** We use stochastic latent dynamics, capturing an unmeasured source of variability not explained by baseline covariates alone and yielding richer and more realistic trajectory diversity than deterministic latent dynamics. For stable training on irregular paths, we introduce a deterministic–stochastic path decomposition optimized via a non-adversarial signature-kernel-based training objective.
- **Hypernetwork-based personalization of latent dynamics:** We propose a hypernetwork-conditioned latent Neural SDE that allows baseline covariates to modulate the full temporal vector field rather than only the initial condition. This explicitly links cross-subject heterogeneity in baseline profiles to variability in longitudinal evolution.

2 Background and related work

Table 1 compares representative generative models along five properties central to our setting: generation of static covariates, support for irregular time series, continuous-time latent dynamics, joint modeling of static and longitudinal variables, and explicit modeling of the observation intensity.

Table 1: Comparison of generative models for joint static and irregular longitudinal data. [1] Irregular timing handled externally. [2] Static covariates conditioned on, not generated. [3] Static variables represented as time-series channels.

Model	Static data	Irregular time-series	Continuous latent	Joint modeling static/TS	Modeled intensity
TimeGAN [66]	×	×	×	×	×
EHR-M-GAN [32]	×	×	×	×	×
TimeDiff [59]	×[3]	×	×	×[3]	×
EHR-Safe [67]	✓	✓	×	✓	×
LatentODE [52]	×	✓	✓	×	✓
VAMBN [30]	✓	✓[1]	×	✓	×
HL-VAE [43]	×[2]	✓	✓	×	×
DGBFGP [6]	×[2]	✓	✓	×	×
RTSGAN [46]	✓	✓	×	✓	✓
MultiNODEs [63]	✓	✓	✓	✓	×
Ours	✓	✓	✓	✓	✓

Time-series and joint generative models Most generative models for time series target regularly sampled sequences in discrete time [66, 17, 32, 31, 68]. While effective on standard benchmarks, such models are ill-suited to clinical trajectories without resampling or imputation. Methods designed for irregular time series [59, 28, 22] better address the temporal structure, but typically do not jointly

model static covariates and observation intensities. Among joint models, RTSGAN [46] incorporates static covariates and visit-time modeling in an adversarial framework, while HL-VAE [43], VAMBN [30], and EHR-Safe [67] combine heterogeneous static data with longitudinal modeling in discrete-time settings. MultiNODEs [63] is the closest prior work in the continuous-time setting.

Continuous-time latent variable models and Neural SDEs Neural ODEs [10] parameterize deterministic latent dynamics through a neural vector field, and Latent ODEs [52] embed this idea in a generative framework for irregularly sampled data. Neural SDEs [26] extend these models by introducing stochastic latent dynamics through learned drift and diffusion terms, allowing the latent process to capture intrinsic temporal uncertainty and to generate richer trajectory variability through the sampled Brownian motion. Training Neural SDEs remains challenging; proposed objectives include variational formulations [34], adversarial Wasserstein objectives [27], scoring rules [69], and signature-kernel scores on path space [21, 53, 8].

Hypernetworks Hypernetworks [20] generate the parameters of a target network as a function of a conditioning input, providing a flexible individualization mechanism beyond concatenation or feature-wise modulation. They have been applied to personalized modeling in scientific and clinical settings [15, 7].

3 Model

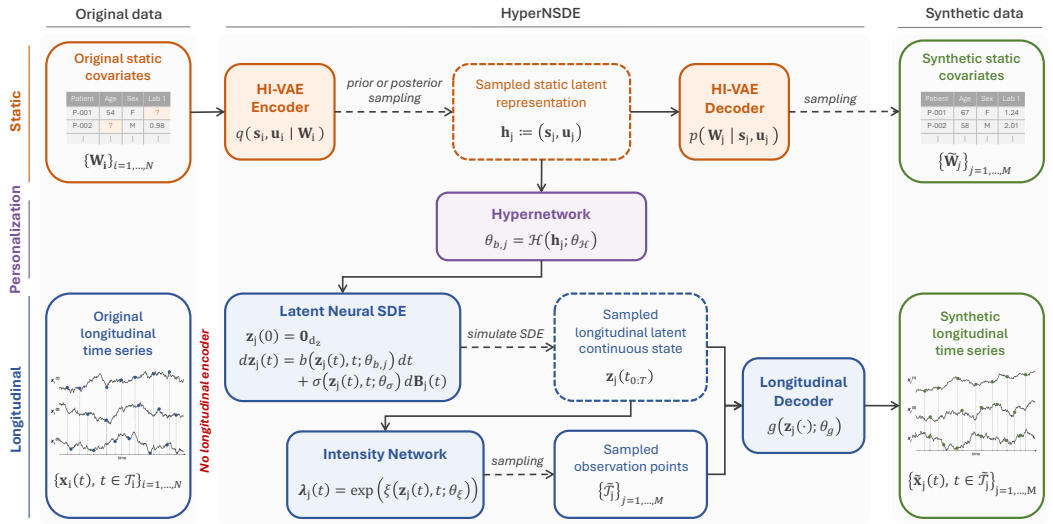

Figure 1: HyperNSDE generation pipeline. Generating a synthetic subject follows a hierarchical process (Appendix A.3). Static latent representations are sampled from the HI-VAE learned distributions, decoded into synthetic covariates, and passed through the hypernetwork to generate patient-specific SDE parameters. A latent path is then simulated from the personalized Neural SDE and decoded at observation times sampled from the intensity network to produce synthetic irregular longitudinal trajectories. Original longitudinal data are used only at training time via the loss terms, without relying on a longitudinal encoder. Solid arrows denote deterministic computations; dashed arrows denote stochastic steps.

Problem formulation We consider a cohort of N subjects. For each subject i, we observe a vector of mixed-type static covariates $\mathbf{W}_i \in \mathbb{R}^{d_W}$ measured at baseline ($t = 0$), encoding time-invariant characteristics (such as age, sex...). These covariates may be continuous, positive-valued, categorical, or ordinal. We also observe d_{x} irregularly sampled and partially observed longitudinal variables,

$$\mathbf{x}_i = \left\{\mathbf{x}_i^{(1)}, \ldots, \mathbf{x}_i^{(d_{\mathrm{x}})}\right\},$$

where each variable $\mathbf{x}_i^{(\ell)}$ is recorded on a subject and variable-specific set of time points $\mathcal{T}_i^{(\ell)} \subset [0, T_i]$. Here, $T_i \leq T$ denotes the subject-specific last observation time, and T the overall study horizon.

3.1 Static latent representation with HI-VAE

To model heterogeneous and incomplete static covariates $\mathbf{W}_i$, we use a Heterogeneous-Incomplete VAE (HI-VAE) [42], following the joint modeling strategy of VAMBN [30] and MultiNODEs [63]. In our framework, this component both provides a generative model for static covariates and a compact latent representation used to condition the longitudinal dynamics. For each individual i, the HI-VAE introduces a discrete latent variable $\mathbf{s}_i \in \{0,1\}^L$ and a continuous latent variable $\mathbf{u}_i \in \mathbb{R}^{d_u}$, with priors $p(\mathbf{s}_i) = \mathrm{Cat}(\mathbf{s}_i \mid \boldsymbol{\pi})$ and $p(\mathbf{u}_i \mid \mathbf{s}_i) = \mathcal{N}\big(\mathbf{u}_i \mid \boldsymbol{\mu}_p(\mathbf{s}_i), \mathbf{I}\big)$. The decoder uses covariate type-specific likelihoods (e.g., Gaussian, Poisson, or Categorical) to handle heterogeneous variables and missing values. We denote the resulting static latent representation by

$$\mathbf{h}_i := (\mathbf{s}_i, \mathbf{u}_i),$$

which is used to condition the longitudinal model. Additional details are provided in Appendix A.1.

3.2 Personalized stochastic latent dynamics

We model the irregular longitudinal trajectories $\{\mathbf{x}_i(t)\}_{t\in[0,T]}$ conditionally on subject-specific static information summarized by the latent representation $\mathbf{h}_i = (\mathbf{s}_i, \mathbf{u}_i)$.

Latent stochastic dynamics Following a standard latent-variable formulation for continuous-time generative modeling [29, 41], we assume that the observed trajectories $\mathbf{x}_i(t) \in \mathbb{R}^{d_\mathrm{x}}$ arise from a continuous-time latent process $\mathbf{z}_i(t) \in \mathbb{R}^{d_z}$, decoded through

$$\hat{\mathbf{x}}_i(t) = g\big(\mathbf{z}_i(t); \theta_g\big), \tag{1}$$

where g is a feed-forward neural network. Since $\mathbf{z}_i(t)$ is defined in continuous time, the decoder can be evaluated at any observation time t, which allows both reconstruction and generation on arbitrary irregular time grids. We model the latent dynamics of each subject i with a Neural SDE [60, 23, 35, 27]:

$$\begin{cases} d\mathbf{z}_i(t) = b_i\big(\mathbf{z}_i(t), t\big)\, dt + \sigma\big(\mathbf{z}_i(t), t; \theta_\sigma\big)\, d\mathbf{B}_i(t), \\ \mathbf{z}_i(0) = \mathbf{0}_{d_z}, \end{cases} \tag{2}$$

where $\mathbf{B}_i$ is a standard Brownian motion. The drift b_i and diffusion σ are neural networks that model, respectively, subject-specific deterministic dynamics and stochastic variability shared across subjects. We assume standard global Lipschitz conditions to ensure well-posedness [45]. Equation (2) provides the conceptual latent-SDE formulation that motivates our model, but it is not trained directly. Instead, we use the deterministic–stochastic decomposition described in Section 4 and Appendix B.2 to separate subject-specific trends from residual stochastic variability.

Personalization via hypernetworks To personalize the dynamics, we let the static representation $\mathbf{h}_i$ parameterize the drift via a hypernetwork $\mathcal{H}$ [20]:

$$\theta_{b,i} = \mathcal{H}(\mathbf{h}_i; \theta_\mathcal{H}), \qquad b_i(\cdot,\cdot) = b(\cdot,\cdot;\theta_{b,i}). \tag{3}$$

The hypernetwork maps each patient's static representation to a subject-specific drift parameter $\theta_{b,i}$, yielding individualized vector fields while sharing the hypernetwork weights $\theta_\mathcal{H}$ across subjects. The diffusion coefficient σ is kept population-shared, separating subject-specific trajectory trends from population-level stochasticity.

Unlike standard latent SDE formulations that infer a distribution over the initial latent state, we fix $\mathbf{z}_i(0) = \mathbf{0}$ for all subjects. Since baseline measurements are included in the static input $\mathbf{W}_i$ (see Appendix D.1) and encoded in $\mathbf{h}_i$, absolute signal levels are absorbed by the static representation, and $\mathbf{z}_i(t)$ models deviations from baseline driven by the personalized drift b_i .

3.3 Event intensity from the latent path

In many longitudinal settings, observation times are informative of the underlying state: for instance, patients may be monitored more frequently during periods of deterioration. Following [52], we model the observation process as a latent-state-dependent multivariate point process, and we additionally introduce an explicit end-of-sequence (EOS) event. For subject i, given latent trajectory $\mathbf{z}_i(t)$, we define

$$\boldsymbol{\lambda}_i(t) = \big(\lambda_i^{(1)}(t), \ldots, \lambda_i^{(d_\mathrm{x})}(t), \lambda_i^{\mathrm{EOS}}(t)\big) \in \mathbb{R}_+^{d_\mathrm{x}+1}, \qquad \boldsymbol{\lambda}_i(t) = \exp\big(\xi(\mathbf{z}_i(t), t; \theta_\xi)\big), \tag{4}$$

where ξ is a neural network with parameters θ_ξ, and the exponential is applied componentwise. Here, $\lambda_i^{(\ell)}(t)$ denotes the intensity of observation times for variable $\ell \in \{1, \dots, d_{\mathrm{x}}\}$, while $\lambda_i^{\mathrm{EOS}}(t)$ denotes the hazard that the observation process terminates at time t.

4 Training procedures

Training HyperNSDE requires jointly optimizing all interconnected components of the models — the HI-VAE, hypernetwork, latent dynamics, decoder, and intensity model. Joint training is essential: the static latent representation $\mathbf{h}_i$ must align with the requirements of the hypernetwork $\mathcal{H}$, and independent pretraining of the HI-VAE yields embeddings that fail to capture the features necessary to parameterize subject-specific drift functions, leading to suboptimal personalized dynamics.

Stable training via deterministic–stochastic path decomposition In our setting with irregular and sparse longitudinal data, directly training a single latent Neural SDE to capture both smooth trajectory trends and stochastic fluctuations proved unstable. We therefore introduce the following structured latent decomposition:

$$\mathbf{z}_i(t) = \mathbf{z}_i^{\mathrm{det}}(t) + \mathbf{z}_i^{\mathrm{sto}}(t), \qquad \mathbf{z}_i^{\mathrm{det}}(0) = \mathbf{z}_i^{\mathrm{sto}}(0) = \mathbf{0}. \tag{5}$$

The deterministic component $\mathbf{z}_i^{\mathrm{det}}$ captures the subject-specific smooth trend and solves a personalized Neural ODE with drift b_i^{det}, whose parameters are generated by the hypernetwork, $\theta_{b_i^{\mathrm{det}}} = \mathcal{H}(\mathbf{h}_i; \theta_{\mathcal{H}})$. The stochastic component $\mathbf{z}_i^{\mathrm{sto}}$ captures residual temporal variability through a population-shared Neural SDE with drift b^{sto} and diffusion σ^{sto}. For additional training stability, the intensity network $\xi(\cdot, t; \theta_\xi)$ is conditioned only on $\mathbf{z}_i^{\mathrm{det}}(t)$ rather than on the full decomposed latent state $\mathbf{z}_i(t)$. This decomposition is a modeling choice: it provides a structured separation between smooth dynamics and residual stochasticity, but is not, in general, equivalent to the single latent SDE in Eq. (2). Empirically, this parameterization improves training stability and the identification of static–longitudinal dependencies, as confirmed by the ablation studies in Table 4 and Figure 6. Further discussion is provided in Appendix B.1.

Loss terms The model is trained by minimizing a weighted combination of the four loss terms defined below. Since they differ substantially in scale and nature, we use a convex combination with weights tuned jointly with the other hyperparameters. Details on the weighting strategy and Pareto-based hyperparameter optimization are provided in Appendix B.3.

- **Static ELBO:** The HI-VAE is trained with the standard evidence lower bound [29, 42], computed on observed static entries only:

$$\begin{aligned}\mathcal{L}_{\mathrm{static}} = \sum_{i=1}^{N} & \mathbb{E}_{q(\mathbf{u}_i, \mathbf{s}_i \mid \mathbf{W}_i)} \left[\log p(\mathbf{W}_i \mid \mathbf{u}_i, \mathbf{s}_i)\right] \\ & - \mathbb{E}_{q(\mathbf{s}_i \mid \mathbf{W}_i)}[\mathrm{KL}(q(\mathbf{u}_i \mid \mathbf{W}_i, \mathbf{s}_i) \,\|\, p(\mathbf{u}_i \mid \mathbf{s}_i))] - \mathrm{KL}(q(\mathbf{s}_i \mid \mathbf{W}_i) \,\|\, p(\mathbf{s}_i)) \,.\end{aligned} \tag{6}$$

- **Point-process likelihood:** Conditioned on the latent path (Eq. 4), the event loss consists of two log-likelihood terms: one for observation times and one for the end-of-sequence event at T_i:

$$\mathcal{L}_{\mathrm{event}} = -\frac{1}{N} \sum_{i=1}^{N} \left[\sum_{\ell=1}^{d_x} \left(\sum_{t \in \mathcal{T}_i^{(\ell)}} \log \lambda_i^{(\ell)}(t) - \int_0^{T_i} \lambda_i^{(\ell)}(s) ds \right) + \log \lambda_i^{\mathrm{EOS}}(T_i) - \int_0^{T_i} \lambda_i^{\mathrm{EOS}}(s) ds \right]. \tag{7}$$

  In practice, the integral terms are approximated numerically on the discretization grid used to solve the latent dynamics (see Appendix A.2) [14, 52].

- **Deterministic reconstruction loss:** The deterministic latent part is trained using the mean squared error to reconstruct observed longitudinal values:

$$\mathcal{L}_{\mathrm{long,det}} = \frac{1}{N} \sum_{i=1}^{N} \sum_{\ell=1}^{d_x} \sum_{t \in \mathcal{T}_i^{(\ell)}} \left\| \mathbf{x}_i^{(\ell)}(t) - \hat{\mathbf{x}}_{i,\mathrm{det}}^{(\ell)}(t) \right\|_2^2, \qquad \hat{\mathbf{x}}_{i,\mathrm{det}}(t) = g\big(\mathbf{z}_i^{\mathrm{det}}(t); \theta_g\big). \tag{8}$$

- **Stochastic residual loss:** To train the residual stochastic component, we match reconstructed residual trajectories to empirical residuals using the *signature kernel score* [21, 53]. We denote the empirical residual trajectory and the reconstructed stochastic residual by

$$\mathbf{r}_i(t) = \mathbf{x}_i(t) - g\big(\mathbf{z}_i^{\mathrm{det}}(t); \theta_g\big), \qquad \hat{\mathbf{r}}_i(t) = g\big(\mathbf{z}_i^{\mathrm{sto}}(t); \theta_g\big).$$

We then define

$$\mathcal{L}_{\mathrm{long,sto}} = \widehat{\phi}_{\mathrm{sig}}\Big(\{\hat{\mathbf{r}}_i\}_{i=1}^N, \{\mathbf{r}_i\}_{i=1}^N\Big), \tag{9}$$

where $\widehat{\phi}_{\mathrm{sig}}$ denotes the empirical signature kernel score associated with a (RBF-)signature kernel on path space. This score is well-suited to irregularly sampled trajectories and is evaluated in practice through its mini-batch estimator. Further details are given in Appendix B.2.

5 Evaluation methodology

Evaluation metrics No universally accepted evaluation protocol exists for irregularly sampled longitudinal data with missingness [57, 56, 1, 28, 36], and most existing benchmarks target regularly sampled or tabular data, primarily assess fidelity and utility while overlooking privacy, and rely on interpolation that can distort temporal structure and bias evaluation. We assess performance along three complementary axes — *fidelity*, *utility*, and *privacy* — using metrics adapted to our multimodal and irregular setting. Full definitions, extended discussion, and limitations are provided in Appendix C and F.2.

- **Fidelity** assesses whether synthetic samples reproduce the statistical structure of real data while maintaining diversity. We report modality-adapted *MMD* [18] and *correlation RMSE (Corr. RMSE)* for both static and longitudinal data, using signature-kernel MMD [12, 53] and masked correlation matrices for longitudinal observations without interpolation. We additionally report the *KL divergence on the observation intensities (KL λ)* and a *discriminative score (Discrim.)* [66] from real-versus-synthetic classification.
- **Utility** measures whether synthetic data can substitute for real data in downstream tasks. We evaluate it through a forecasting task under a Train-on-Synthetic-Test-on-Real (TSTR) protocol [17]: a predictor trained on static covariates and the first part of each longitudinal trajectory forecasts the remaining time points, and performance is reported as the *prediction score (Pred.)* corresponding to the MSE on real test data, and compared against a Train-on-Real-Test-on-Real (TRTR) reference.
- **Privacy** evaluation assesses whether synthetic samples reveal information about the training data beyond what is expected from the underlying population distribution. We report *nearest-neighbor adversarial accuracy (NNAA)* [64], *membership inference risk (MIR)* [55], and a *dependence score (Dep.)* [56] measuring residual dependence between generated samples and training individuals, computed on joint multimodal representations.

Simulated datasets To evaluate whether models recover controlled multimodal dependencies, we construct simulated datasets in which patient-specific static covariates drive longitudinal dynamics. For each individual i, we generate static variables $\mathbf{W}_i = [\mathrm{W}_{i,\mathrm{num}}, \mathrm{W}_{i,\mathrm{bin}}] \in \mathbb{R}^{d_W}$, with one numerical feature sampled from a Gaussian distribution, and one binary feature generated through a probit model. Conditioned on $\mathbf{W}_i$, the longitudinal process $\mathbf{x}_i(t) \in \mathbb{R}^{d_{\mathrm{x}}}$ follows independent fractional Ornstein–Uhlenbeck dynamics,

$$d\mathbf{x}_i^{(\ell)}(t) = -a_{i,\ell}\big(\mathbf{x}_i^{(\ell)}(t) - b_{i,\ell}\big)\, dt + \zeta\, dB_{i,\ell}^H(t),$$

where $B_{i,\ell}^H(t)$ is a fractional Brownian motion, $\zeta > 0$ is a global diffusion scale, and the subject-specific parameters $a_{i,\ell}$ and $b_{i,\ell}$ are linear functions of the static covariates. Observation times are sampled from a non-homogeneous Poisson process to mimic irregular clinical follow-up. Each dataset contains $N = 1000$ individuals with $d_W = 2$ static variables, $d_{\mathrm{x}} = 3$ longitudinal variables, and trajectories generated over a horizon $T = 7$; the reference setting yields on average $n_{\mathrm{obs}} = 50$ visits per patient. Full simulation details are provided in Appendix D.2.

Real-world datasets We evaluate HyperNSDE on two real clinical datasets covering diverse medical domains and observation regimes (see Table 6 in Appendix D.3 for summary statistics). VELOUR [25, 48] is a phase III randomized trial of aflibercept plus FOLFIRI in metastatic colorectal cancer accessed via Project DataSphere, with dense longitudinal laboratory and vital sign measurements (23 longitudinal variables, 26 static covariates). PPMI [38] is a longitudinal Parkinson's disease cohort with clinical scores and vital signs collected over multi-year follow-up (15 longitudinal variables, 20 static covariates). Full preprocessing details are provided in Appendix D.3.

Baselines and reference We compare our method against three baselines: *MultiNODEs* [63], *RTSGAN* [46], and *DGBFGP* [6]. These models were selected for their relevance to the joint static and irregular longitudinal setting and the availability of public, sufficiently well-documented implementations, unlike several other models listed in Table 1. For *DGBFGP*, which does not natively provide a generative model for static covariates, we augment it with an HI-VAE [42] for static data generation (*DGBFGP+HI-VAE*). As *DGBFGP* requires substantial dataset-specific manual configuration, this comparison is restricted to simulated data. In all tables, the *Reference* row is not a generative model: it is obtained by randomly splitting the real dataset in half and treating one half as synthetic, providing an empirical reference point that a well-fitted generative model should approach. Additional details are provided in Appendix E.2.

Experimental setup For simulated data, results are reported over $N_{\text{MC}} = 30$ independent Monte Carlo runs, each based on a newly generated dataset with a different random seed. In all experiments, original data are split into training and test sets with an 80/20 ratio. Hyperparameters are selected via a preliminary tuning phase (Appendix B.3). To robustly assess generative performance, each trained model is used to generate $N_{\text{gen}} = 50$ independent synthetic test sets, each matching the real test set size N_{test}; reported results are averaged over these repeated generations. All experiments run on NVIDIA V100 GPUs on the Jean Zay cluster (Appendix E.1). Our implementation is publicly available at `https://github.com/perrinechassat/HyperNSDE`; signature-kernel computations are performed using `pySigLib` [54].

6 Results

Table 2 and Figure 2 report the results in simulated and real-world datasets. Because longitudinal Corr. RMSE depends on the time points jointly observed across variables, and TSTR prediction may be affected by both sampling-grid regularity and trajectory smoothness, we complement the native-grid comparison with controlled reevaluations in Table 3. MultiNODEs and HyperNSDE can be evaluated on common time grids, and a drift-only HyperNSDE variant further isolates the effect of trajectory smoothness.

Simulated data Table 2 and Figure 2 show that HyperNSDE performs competitively across the complementary evaluation axes in the reference simulated setting, averaged over $N_{\text{MC}} = 30$ Monte Carlo runs and $N_{\text{gen}} = 50$ generated datasets per trained model. It achieves the best discriminative, dependence, KL λ, and predictive mean scores, along with the lowest longitudinal MMD ($16\times$ lower than RTSGAN). Qualitatively, it is also the only method that simultaneously reproduces (i) the color stratification induced by W_0 and W_1, (ii) stochastic variability across trajectories, and (iii) irregular observation patterns. By contrast, MultiNODEs captures the W_1-driven separation but produces overly smooth deterministic trajectories; RTSGAN shows partial color stratification but collapses near $t = 0$; and DGBFGP + HI-VAE generates flat trajectories with low variance (Figure 4). MultiNODEs achieves a lower native-grid Corr. RMSE, but matched-grid analysis in Table 3 shows that this advantage is strongly affected by grid regularity: on the real observation grid, its Corr. RMSE increases from 0.084 to 0.494, whereas HyperNSDE changes from 0.118 to 0.115. Prediction scores are likewise grid-dependent, while the drift-only results show that the MSE objective also favors smoother trajectories.

The discriminative-score decomposition in Appendix F.2 further shows that static covariates are difficult to distinguish from real data across all methods, whereas observation patterns strongly distinguish the baselines; for HyperNSDE, the remaining gap primarily arises from trajectory values. Subgroup analyses restricted to $W_1 = 0$ and $W_1 = 1$ (Tables 17–18) further assess preservation of subpopulation-specific structure. Additional sensitivity analyses in Appendix F.3 examine limited training data, sparse observations, state-dependent observation sampling, prior versus posterior latent

Table 2: Evaluation metrics on simulated and real-world data. ↓ lower is better; → 0 closer to zero is better (NNAA). **Bold** is the best result excluding the reference, underlined is second best. Standard deviations across $N_{\text{MC}} = 30$ Monte Carlo runs and $N_{\text{gen}} = 50$ generated datasets from a single trained model are reported in Table 7 in Appendix F.1.

		Fidelity						Privacy			Utility
		Corr. RMSE ↓		Discrim. ↓	KL λ ↓	MMD ↓		Dep. ↓	MIR ↓	NNAA → 0	Pred. ↓
		Long.	Static			Long.	Static				
Simulated	Reference	0.061	0.060	0.499	0.139	0.021	0.007	0.001	0.033	0.021	3.759
	MultiNODEs	**0.084**	0.045	0.998	67.616	816.507	0.006	0.117	**0.034**	−0.008	11.328
	RTSGAN	0.103	0.073	0.932	2.525	18.427	0.013	0.020	0.037	−0.009	5.278
	DGBFGP + HIVAE	0.167	0.051	0.999	67.616	299.330	0.006	0.058	0.034	**−0.005**	11.603
	HyperNSDE	0.118	**0.044**	**0.926**	**0.687**	1.150	**0.004**	**0.014**	0.039	−0.021	**5.226**
VELOUR	Reference	0.091	0.132	0.472	0.844	0.001	0.019	0.063	0.073	−0.049	21.958
	MultiNODEs	**0.139**	0.154	**0.990**	139.716	0.144	0.019	0.628	**0.048**	0.011	**5.837**
	RTSGAN	0.842	**0.149**	1.000	**2.581**	121.596	0.019	0.118	0.050	0.084	6048.192
	HyperNSDE	0.239	0.164	0.998	3.513	**0.029**	**0.019**	**0.105**	0.063	**0.010**	53.514
PPMI	Reference	0.044	0.092	0.587	0.064	0.001	0.006	0.052	0.065	−0.032	39.137
	MultiNODEs	**0.108**	0.133	1.000	221.398	0.038	0.013	0.391	**0.017**	0.006	**19.948**
	RTSGAN	0.623	**0.114**	**0.983**	**1.042**	**0.009**	**0.012**	**0.057**	0.057	−0.007	52.954
	HyperNSDE	0.260	0.135	0.996	1.840	0.019	0.013	0.084	0.032	**0.002**	57.697

Table 3: Effect of observation-grid regularity and trajectory smoothness on longitudinal Corr. RMSE and utility metrics. We reevaluate MultiNODEs and HyperNSDE on HyperNSDE-generated and real observation grids and use drift-only HyperNSDE to assess the effect of trajectory smoothness. ↓ lower is better. **Bold** is best and underlined second best, excluding the reference.

		Simulated		VELOUR		PPMI	
Model	Setting (obs. grid)	Corr. RMSE ↓ Long.	Pred. ↓	Corr. RMSE ↓ Long.	Pred. ↓	Corr. RMSE ↓ Long.	Pred. ↓
Reference	real obs.	0.061	3.759	0.091	21.958	0.044	39.137
MultiNODEs	native (regular)	**0.084**	11.328	**0.139**	**5.837**	**0.108**	**19.948**
	HyperNSDE grid	0.502	5.815	0.940	39.275	0.915	57.621
	real obs.	0.494	6.067	0.940	7.357	0.914	21.841
HyperNSDE	native (generated)	0.118	5.226	0.239	53.514	0.260	57.697
	real obs.	0.115	4.379	0.284	10.633	0.261	36.173
	real obs., drift only	0.171	**3.636**	0.216	7.010	0.241	23.298

sampling, and the state dependence of the modeled observation intensity. Paired Wilcoxon tests across the 30 Monte Carlo runs confirm significant improvements in longitudinal and static MMD against all baselines, whereas longitudinal Corr. RMSE differences are not significant. For KL λ, the paired results are baseline-dependent, with significant improvements over MultiNODEs and DGBFGP+HI-VAE (Appendix F.1).

Real-world datasets Results on VELOUR and PPMI are more mixed than on simulated data, reflecting the added complexity of real clinical trajectories, including heterogeneous missingness, patient-specific follow-up, and informative observation times. HyperNSDE achieves the best NNAA, the best longitudinal MMD on VELOUR (about $5\times$ and $4000\times$ lower than MultiNODEs and RTSGAN), and a competitive longitudinal MMD on PPMI. For observation-intensity fidelity, HyperNSDE yields substantially lower KL λ than MultiNODEs on both datasets, although RTSGAN obtains the lowest values. MultiNODEs performs better on native-grid Corr. RMSE and prediction, but Table 3 shows that these comparisons are strongly affected by the evaluation setting. On VELOUR, moving to the real observation grid changes Corr. RMSE from 0.139 to 0.940 for MultiNODEs and from 0.239 to 0.284 for HyperNSDE, while prediction changes from 5.837 to 7.357 and from 53.514 to 10.633, respectively. HyperNSDE's drift-only prediction further reaches 7.010 on the same grid,

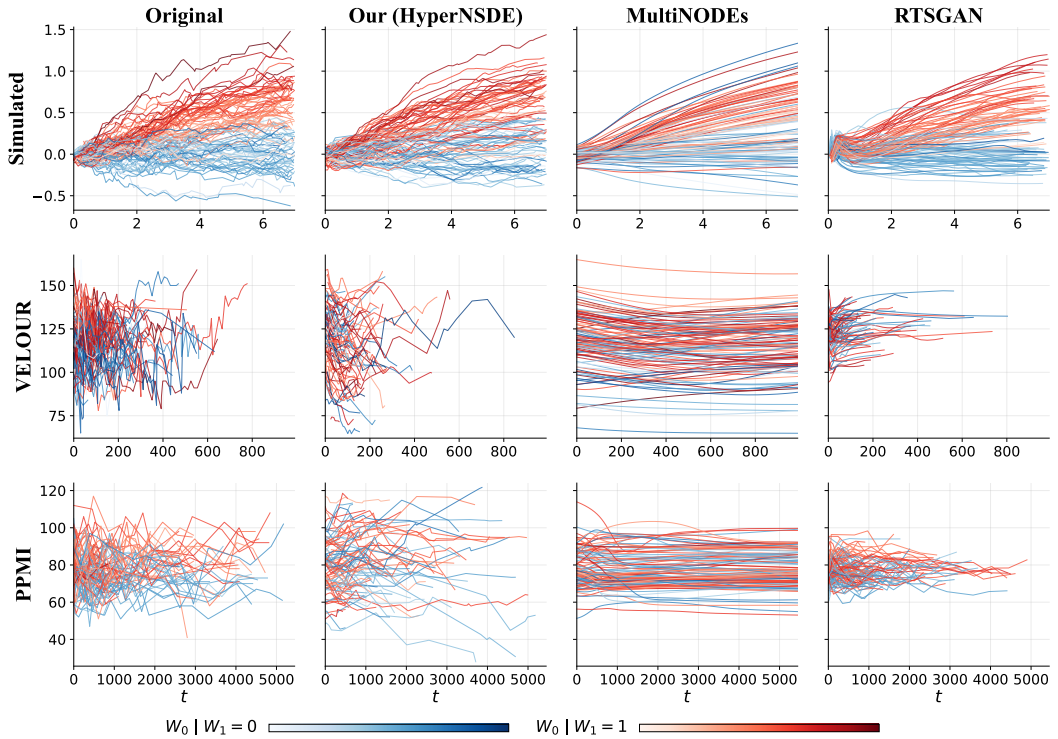

Figure 2: Qualitative comparison of real and generated longitudinal trajectories. Rows denote datasets and columns the real data and generative methods. Trajectory color represents the static covariates: blue trajectories have $W_1 = 0$ and red ones have $W_1 = 1$, with color intensity encoding the continuous covariate W_0. For the simulated data, $W_0 \in [-0.79, 0.73]$ and the plotted variable is x_0; for VELOUR, W_0 is age (22–79 years), W_1 is sex, and the plotted variable is hemoglobin; for PPMI, W_0 is years of education (0–30), W_1 is sex, and the plotted variable is supine diastolic blood pressure. Additional simulated-data results are in Appendix F.

close to MultiNODEs, indicating once again that the MSE forecasting objective favors smoother deterministic trajectories. Similar effects are observed on PPMI. This is consistent with Figure 2, where MultiNODEs produces smoother paths but does not reproduce the stochastic variability of the real trajectories. Finally, unlike the fixed follow-up used in simulation, real datasets exhibit highly variable patient-specific end-of-sequence times, further complicating evaluation on short heterogeneous trajectories.

Ablation studies Table 4 and Figure 6 in Appendix F.5 isolate the contribution of each component. Replacing the Neural SDE with an ODE (AS.1) preserves utility but degrades correlation RMSE and MMD, confirming that stochastic dynamics are necessary to capture trajectory variability beyond a single deterministic path. Removing the det–sto decomposition (AS.2) introduces excessive stochastic noise that overwhelms the trend signal, yielding chaotic trajectories with poor color stratification and a severe drop in correlation RMSE and utility, validating the importance of the proposed strategy for stabilizing training. Among conditioning alternatives (AS.3), initial-state conditioning (MNSDE), continuous concatenation (SNSDE), and FiLM modulation (FNSDE) are competitive on individual metrics, but none dominates HyperNSDE; the hypernetwork achieves the lowest longitudinal MMD and generates visually less noisy, better-separated trajectories while remaining competitive elsewhere. These results support hypernetwork-based personalization as an effective choice without establishing it as uniquely necessary. Finally, removing the observation model (AS.4) causes a large increase in KL λ (from 0.687 to 67.616) and MMD, underscoring the importance of explicitly modeling informative sampling patterns. Additional details on these ablations are provided in Appendix E.4.

Table 4: Ablation studies on simulated data. Standard deviations across $N_{\text{MC}} = 30$ Monte Carlo runs and $N_{\text{gen}} = 50$ generated datasets are reported in Table 8 in Appendix F.1.

	Fidelity						Privacy			Utility
	Corr. RMSE ↓		Discrim. ↓	KL λ ↓	MMD ↓		Dep. ↓	MIR ↓	NNAA → 0	Pred. ↓
	Long.	Static			Long.	Static				
Reference	0.061	0.060	0.499	0.139	0.021	0.007	0.001	0.033	0.021	3.734
AS.1 only ODE	0.160	0.040	0.992	0.773	92.621	0.004	0.095	0.046	−0.025	**4.515**
AS.2 training	0.233	0.052	0.964	0.971	1.209	0.006	0.042	**0.035**	−0.007	11.880
AS.3 MNSDE	0.154	**0.024**	0.954	**0.422**	1.555	**0.003**	0.023	0.045	−0.027	4.840
AS.3 SNSDE	0.110	0.026	**0.846**	0.582	1.201	0.003	0.013	0.047	−0.035	5.485
AS.3 FNSDE	0.149	0.073	0.926	0.584	1.673	0.010	**0.010**	0.035	**−0.001**	6.295
AS.4 w/o lbda	**0.101**	0.045	0.986	67.616	149.558	0.004	0.018	0.037	−0.016	11.563
HyperNSDE	0.118	0.044	0.926	0.687	**1.150**	0.004	0.014	0.039	−0.021	5.226

7 Conclusion

We introduced HyperNSDE, a continuous-time generative model for heterogeneous static covariates, irregular longitudinal trajectories, and informative observation times. By combining a generative model for static covariates with a hypernetwork-conditioned latent Neural SDE, HyperNSDE captures both inter-individual variability driven by baseline characteristics and intra-individual stochastic fluctuations. Experiments show improved observation-time fidelity and competitive performance across complementary metrics, while matched-grid analyses reveal important sensitivity of the metrics to sampling regularity and trajectory smoothness.

Several limitations remain. HyperNSDE learns the observational distribution of its training cohort rather than causal counterfactuals, and therefore cannot by itself correct selection bias or extrapolate to unsupported subgroups. It is also computationally more demanding than the baselines. More broadly, our experiments show that existing metrics can disagree with trajectory realism for sparse, irregular, stochastic data, while discriminative scores remain high for all methods, highlighting both the difficulty of the problem and the limits of current approaches. In their present form, such models may already support benchmarking use cases, but further progress is needed before they can be confidently used for cohort augmentation or clinical trial simulation.

These limitations suggest several directions for future work. On the modeling side, richer diffusion parameterizations [44], including covariates-dependent diffusion terms, could improve the representation of subject-specific stochasticity. Extending the framework to typed longitudinal variables (e.g., integer, positive, or bounded) through adapted output distributions would also improve realism for clinical biomarkers. Integrating differential privacy during training would provide formal guarantees to complement our empirical privacy evaluation. On the methodological side, a deeper theoretical and empirical study of the deterministic–stochastic decomposition would help clarify when and why it improves training in sparse irregular settings. Finally, improving evaluation protocols and metrics for synthetic irregular longitudinal data remains, in our view, the most important open problem, since current metrics are still the main limiting factor for reliably assessing generated-data quality and for making fair comparisons between models.

Acknowledgments and Disclosure of Funding

For this work, PC and AG have benefited from support from the National Agency for Research under the France 2030 program, reference ANR-22-PESN-0016. PC and AG received funding from the MEDITWIN Bpifrance i-Demo project as part of the France 2030 program. AG has also benefited from the Institut National du Cancer within INCa_CHAIRE IA-RC26-009.

References

[1] Jim L Achterberg, Marcel R Haas, and Marco R Spruit. On the evaluation of synthetic longitudinal electronic health records. *BMC Medical Research Methodology*, 24(1):181, 2024.

[2] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 2623–2631, 2019.

[3] Ahmed Alaa, Boris Van Breugel, Evgeny S Saveliev, and Mihaela Van Der Schaar. How faithful is your synthetic data? sample-level metrics for evaluating and auditing generative models. In *International conference on machine learning*, pages 290–306. PMLR, 2022.

[4] Yihao Ang, Qiang Huang, Yifan Bao, Anthony K. H. Tung, and Zhiyong Huang. TSGBench: Time series generation benchmark. *Proceedings of the VLDB Endowment*, 17(3):305–318, 2023.

[5] Ilze Amanda Auzina, Çağatay Yıldız, Sara Magliacane, Matthias Bethge, and Efstratios Gavves. Modulated neural odes. *Advances in Neural Information Processing Systems*, 36:44572–44594, 2023.

[6] Mehmet Yiğit Balık, Maksim Sinelnikov, Priscilla Ong, and Harri Lähdesmäki. Bayesian basis function approximation for scalable gaussian process priors in deep generative models. In *Forty-second International Conference on Machine Learning*, 2025.

[7] Stav Belogolovsky, Ido Greenberg, Danny Eytan, and Shie Mannor. Individualized dosing dynamics via neural eigen decomposition. *Advances in Neural Information Processing Systems*, 36:56211–56233, 2023.

[8] Patric Bonnier and Harald Oberhauser. Proper scoring rules, gradients, divergences, and entropies for paths and time series. *Bayesian Analysis*, 20(4):1615–1646, 2025.

[9] Richard J Chen, Ming Y Lu, Tiffany Y Chen, Drew FK Williamson, and Faisal Mahmood. Synthetic data in machine learning for medicine and healthcare. *Nature Biomedical Engineering*, 5(6):493–497, 2021.

[10] Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. *Advances in neural information processing systems*, 31, 2018.

[11] Zhao Chen, Vijay Badrinarayanan, Chen-Yu Lee, and Andrew Rabinovich. Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In *International conference on machine learning*, pages 794–803. PMLR, 2018.

[12] Ilya Chevyrev and Harald Oberhauser. Signature moments to characterize laws of stochastic processes. *Journal of Machine Learning Research*, 23(176):1–42, 2022.

[13] Fabienne Comte and Nicolas Marie. Nonparametric estimation for iid paths of fractional sde. *Statistical Inference for Stochastic Processes*, 24(3):669–705, 2021.

[14] Daryl J Daley and David Vere-Jones. *An introduction to the theory of point processes: volume I: elementary theory and methods*. Springer, 2003.

[15] Filipe de Avila Belbute-Peres, Yi-fan Chen, and Fei Sha. Hyperpinn: Learning parameterized differential equations with physics-informed hypernetworks. *The symbiosis of deep learning and differential equations*, 690, 2021.

[16] Yousef El-Laham, Zhongchang Sun, Haibei Zhu, Tucker Balch, and Svitlana Vyetrenko. Variational neural stochastic differential equations with change points. *Transactions on Machine Learning Research*, 2025.

[17] Cristóbal Esteban, Stephanie L Hyland, and Gunnar Rätsch. Real-valued (medical) time series generation with recurrent conditional gans. *arXiv preprint arXiv:1706.02633*, 2017.

[18] Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *The journal of machine learning research*, 13(1):723–773, 2012.

[19] Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured state spaces. In *The International Conference on Learning Representations (ICLR)*, 2022.

[20] David Ha, Andrew Dai, and Quoc V Le. Hypernetworks. *arXiv preprint arXiv:1609.09106*, 2016.

[21] Zacharia Issa, Blanka Horvath, Maud Lemercier, and Cristopher Salvi. Non-adversarial training of neural sdes with signature kernel scores. *Advances in Neural Information Processing Systems*, 36:11102–11126, 2023.

[22] Jinsung Jeon, Jeonghak Kim, Haryong Song, Seunghyeon Cho, and Noseong Park. Gt-gan: General purpose time series synthesis with generative adversarial networks. *Advances in Neural Information Processing Systems*, 35:36999–37010, 2022.

[23] Junteng Jia and Austin R Benson. Neural jump stochastic differential equations. *Advances in Neural Information Processing Systems*, 32, 2019.

[24] Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7482–7491, 2018.

[25] Ganessan Kichenadasse, John O. Miners, Arduino A. Mangoni, Christos S. Karapetis, Ashley M. Hopkins, and Michael J. Sorich. Proton pump inhibitors and survival in patients with colorectal cancer receiving fluoropyrimidine-based chemotherapy. *Journal of the National Comprehensive Cancer Network*, 19(9):1037 – 1044, 2021.

[26] Patrick Kidger, James Foster, Xuechen Li, and Terry J Lyons. Neural sdes as infinite-dimensional gans. In *International conference on machine learning*, pages 5453–5463. PMLR, 2021.

[27] Patrick Kidger, James Foster, Xuechen Chen Li, and Terry Lyons. Efficient and accurate gradients for neural sdes. *Advances in Neural Information Processing Systems*, 34:18747–18761, 2021.

[28] Jaewon Kim, Hyunwoo Choo, Soo-Yong Shin, and Kyoung Doo Song. Synthesis and quality assessment of combined time-series and static medical data using a real-world time-series generative adversarial network. *Scientific Reports*, 14(1):19064, 2024.

[29] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. *CoRR*, abs/1312.6114, 2013.

[30] Lisa Kühnel, Julian Schneider, Ines Perrar, Tim Adams, Sobhan Moazemi, Fabian Prasser, Ute Nöthlings, Holger Fröhlich, and Juliane Fluck. Synthetic data generation for a longitudinal cohort study–evaluation, method extension and reproduction of published data analysis results. *Scientific Reports*, 14(1):14412, 2024.

[31] Nicholas I Kuo, Louisa Jorm, Sebastiano Barbieri, et al. Synthetic health-related longitudinal data with mixed-type variables generated using diffusion models. *arXiv preprint arXiv:2303.12281*, 2023.

[32] Jin Li, Benjamin J Cairns, Jingsong Li, and Tingting Zhu. Generating synthetic mixed-type longitudinal electronic health records for artificial intelligent applications. *NPJ digital medicine*, 6(1):98, 2023.

[33] Xiaopeng Li, Zhourong Chen, Leonard K. M. Poon, and Nevin Lianwen Zhang. Learning latent superstructures in variational autoencoders for deep multidimensional clustering. In *International Conference on Learning Representations*, 2019.

[34] Xuechen Li, Ting-Kam Leonard Wong, Ricky TQ Chen, and David K Duvenaud. Scalable gradients and variational inference for stochastic differential equations. In *Symposium on Advances in Approximate Bayesian Inference*, pages 1–28. PMLR, 2020.

[35] Xuanqing Liu, Tesi Xiao, Si Si, Qin Cao, Sanjiv Kumar, and Cho-Jui Hsieh. Neural sde: Stabilizing neural ode networks with stochastic noise. *arXiv preprint arXiv:1906.02355*, 2019.

[36] Mohammad Loni, Fatemeh Poursalim, Mehdi Asadi, and Arash Gharehbaghi. A review on generative ai models for synthetic medical text, time series, and longitudinal data. *npj Digital Medicine*, 8(1):281, 2025.

[37] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2017.

[38] Kenneth Marek, Danna Jennings, Shirley Lasch, Andrew Siderowf, Caroline Tanner, Tanya Simuni, Chris Coffey, Karl Kieburtz, Emily Flagg, Sohini Chowdhury, et al. The parkinson progression marker initiative (ppmi). *Progress in neurobiology*, 95(4):629–635, 2011.

[39] Marko Miletic and Murat Sariyar. Synthetic data generation methods for longitudinal and time series health data: a systematic review. *BMC Medical Informatics and Decision Making*, 2025.

[40] Neda Mohammadi, Leonardo V Santoro, and Victor M Panaretos. Nonparametric estimation for sde with sparsely sampled paths: An fda perspective. *Stochastic Processes and their Applications*, 167:104239, 2024.

[41] K.P. Murphy. *Machine Learning: A Probabilistic Perspective*. Adaptive Computation and Machine Learning series. MIT Press, 2012.

[42] Alfredo Nazabal, Pablo M Olmos, Zoubin Ghahramani, and Isabel Valera. Handling incomplete heterogeneous data using vaes. *Pattern Recognition*, 107:107501, 2020.

[43] Mine Öğretir, Siddharth Ramchandran, Dimitrios Papatheodorou, and Harri Lähdesmäki. A variational autoencoder for heterogeneous temporal and longitudinal data. In *2022 21st IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 1522–1529. IEEE, 2022.

[44] YongKyung Oh, Dong-Young Lim, and Sungil Kim. Stable neural stochastic differential equations in analyzing irregular time series data. *arXiv preprint arXiv:2402.14989*, 2024.

[45] Bernt Øksendal. Stochastic differential equations. In *Stochastic differential equations: an introduction with applications*, pages 38–50. Springer, 2003.

[46] Hengzhi Pei, Kan Ren, Yuqing Yang, Chang Liu, Tao Qin, and Dongsheng Li. Towards generating real-world time series data. In *2021 IEEE International Conference on Data Mining (ICDM)*, pages 469–478. IEEE, 2021.

[47] Jonas Peltner, Cornelia Becker, Julia Wicherski, Silja Wortberg, Mohamed Aborageh, Inês Costa, Vera Ehrenstein, Joana Fernandes, Steffen Heß, Erzsébet Horváth-Puhó, et al. The eu project real4reg: unlocking real-world data with ai. *Health research policy and systems*, 23(1):27, 2025.

[48] Project DataSphere LLC. Project datasphere. Dataset identifier: b1499ed4-5a32-4e6c-af61-7acdd6f7825c.

[49] Phillip E Protter. Stochastic integration and differential equations, 2004.

[50] Zhaozhi Qian, Rob Davis, and Mihaela Van Der Schaar. Synthcity: a benchmark framework for diverse use cases of tabular synthetic data. *Advances in neural information processing systems*, 36:3173–3188, 2023.

[51] James O Ramsay and Bernard W Silverman. *Functional data analysis*. Springer, 1997.

[52] Yulia Rubanova, Ricky TQ Chen, and David K Duvenaud. Latent ordinary differential equations for irregularly-sampled time series. *Advances in neural information processing systems*, 32, 2019.

[53] Cristopher Salvi, Thomas Cass, James Foster, Terry Lyons, and Weixin Yang. The signature kernel is the solution of a goursat pde. *SIAM Journal on Mathematics of Data Science*, 3(3):873–899, 2021.

[54] Daniil Shmelev and Cristopher Salvi. pysiglib-fast signature-based computations on cpu and gpu. *arXiv preprint arXiv:2509.10613*, 2025.

[55] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE symposium on security and privacy (SP)*, pages 3–18. IEEE, 2017.

[56] Michael Stenger, Robert Leppich, André Bauer, and Samuel Kounev. Steb: In search of the best evaluation approach for synthetic time series. *arXiv preprint arXiv:2505.21160*, 2025.

[57] Michael Stenger, Robert Leppich, Ian Foster, Samuel Kounev, and André Bauer. Evaluation is key: a survey on evaluation measures for synthetic time series. *Journal of Big Data*, 11(1):66, 2024.

[58] Kristian Thorlund, Louis Dron, Jay JH Park, and Edward J Mills. Synthetic and external controls in clinical trials–a primer for researchers. *Clinical epidemiology*, pages 457–467, 2020.

[59] Muhang Tian, Bernie Chen, Allan Guo, Shiyi Jiang, and Anru R Zhang. Reliable generation of privacy-preserving synthetic electronic health record time series via diffusion models. *Journal of the American Medical Informatics Association*, 31(11):2529–2539, 2024.

[60] Belinda Tzen and Maxim Raginsky. Neural stochastic differential equations: Deep latent gaussian models in the diffusion limit. *arXiv preprint arXiv:1905.09883*, 2019.

[61] Robin van Hoorn, Tom Bakkes, Zoi Tokoutsi, Ymke de Jong, R Arthur Bouwman, and Mykola Pechenizkiy. Generating privacy-preserving longitudinal synthetic data. In *NeurIPS 2023 Workshop on Synthetic Data Generation with Generative AI*, 2023.

[62] Simon Vandenhende, Stamatios Georgoulis, Wouter Van Gansbeke, Marc Proesmans, Dengxin Dai, and Luc Van Gool. Multi-task learning for dense prediction tasks: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 44(7):3614–3633, 2021.

[63] Philipp Wendland, Colin Birkenbihl, Marc Gomez-Freixa, Meemansa Sood, Maik Kschischo, and Holger Fröhlich. Generation of realistic synthetic data using multimodal neural ordinary differential equations. *NPJ Digital Medicine*, 5(1):122, 2022.

[64] Andrew Yale, Saloni Dash, Ritik Dutta, Isabelle Guyon, Adrien Pavao, and Kristin P Bennett. Generation and evaluation of privacy preserving synthetic health data. *Neurocomputing*, 416:244–255, 2020.

[65] Fang Yao, Hans-Georg Müller, and Jane-Ling Wang. Functional data analysis for sparse longitudinal data. *Journal of the American statistical association*, 100(470):577–590, 2005.

[66] Jinsung Yoon, Daniel Jarrett, and Mihaela Van der Schaar. Time-series generative adversarial networks. *Advances in neural information processing systems*, 32, 2019.

[67] Jinsung Yoon, Michel Mizrahi, Nahid Farhady Ghalaty, Thomas Jarvinen, Ashwin S Ravi, Peter Brune, Fanyu Kong, Dave Anderson, George Lee, Arie Meir, et al. Ehr-safe: generating high-fidelity and privacy-preserving synthetic electronic health records. *NPJ digital medicine*, 6(1):141, 2023.

[68] Xinyu Yuan and Yan Qiao. Diffusion-ts: Interpretable diffusion for general time series generation. *International Conference on Learning Representations*, 2024.

[69] Jianxin Zhang, Josh Viktorov, Doosan Jung, and Emily Pitler. Efficient training of neural stochastic differential equations by matching finite dimensional distributions. In *International Conference on Learning Representations*, volume 2025, pages 36730–36742, 2025.

[70] Yuzhen Zhao, Yating Liu, and Marc Hoffmann. Drift estimation for diffusion processes using neural networks based on discretely observed independent paths. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 28778–28785, 2026.

A Model details

A.1 Additional details on the HI-VAE

The HI-VAE [42] is designed to model static covariates with heterogeneous data types and missing values within a unified variational framework. In our setting, for each individual i, the observed static vector $\mathbf{W}_i$ is associated with a discrete latent variable $\mathbf{s}_i$ and a continuous latent variable $\mathbf{u}_i$, as introduced in Section 3.1. The decoder defines the conditional distribution

$$p(\mathbf{W}_i \mid \mathbf{s}_i, \mathbf{u}_i) = \prod_{k=1}^{d_{\mathrm{W}}} p\Big(\mathbf{W}_i^{(k)} \mid \gamma_k(\mathbf{y}_i, \mathbf{s}_i)\Big), \qquad \mathbf{y}_i = f(\mathbf{u}_i),$$

where each feature-specific decoder γ_k outputs the parameters of a likelihood adapted to the type of the k-th variable.

Feature-wise likelihoods Following [42], we use different observation models depending on the type of each static covariate:

- Continuous real-valued variables (Normal): $\gamma_k(\mathbf{y}_i, \mathbf{s}_i) = \big(\mu_k(\mathbf{y}_i, \mathbf{s}_i), \sigma_k^2(\mathbf{y}_i, \mathbf{s}_i)\big)$,
- Positive real-valued variables (Log-Normal): $\gamma_k(\mathbf{y}_i, \mathbf{s}_i) = \big(\mu_k(\mathbf{y}_i, \mathbf{s}_i), \sigma_k^2(\mathbf{y}_i, \mathbf{s}_i)\big)$,
- Count variables (Poisson): $\gamma_k(\mathbf{y}_i, \mathbf{s}_i) = \lambda_k(\mathbf{y}_i, \mathbf{s}_i)$,
- Categorical variables (Multinomial-logit): $\gamma_k(\mathbf{y}_i, \mathbf{s}_i) = \big(\gamma_{k0}(\mathbf{y}_i, \mathbf{s}_i), \ldots, \gamma_{k(R-1)}(\mathbf{y}_i, \mathbf{s}_i)\big)$.

Variational inference Inference is performed using variational encoders for both latent variables. The posterior factorization follows the HI-VAE construction:

$$q(\mathbf{s}_i, \mathbf{u}_i \mid \mathbf{W}_i) = q(\mathbf{s}_i \mid \mathbf{W}_i)\, q(\mathbf{u}_i \mid \mathbf{W}_i, \mathbf{s}_i).$$

The discrete latent variable $\mathbf{s}_i$ is handled using the Gumbel–Softmax reparameterization [33], while the continuous latent variable $\mathbf{u}_i \mid \mathbf{s}_i$ is sampled using the standard Gaussian reparameterization trick [29]. This enables end-to-end gradient-based optimization of the evidence lower bound.

A.2 Additional details on the event intensity model

The observation process is modeled as a latent-state-dependent multivariate point process with integrated intensity

$$\mathbf{\Lambda}_i(t) = \int_0^t \boldsymbol{\lambda}_i(s)\, ds = \int_0^t \exp\big(\xi(\mathbf{z}_i(s), s; \theta_\xi)\big) ds,$$

where the exponential is applied componentwise. In the original Latent ODE framework [52], the integrated intensity is introduced as an additional state variable within the ODE system and evolves jointly with the latent trajectory. In our setting, however, we found that augmenting the SDE system with $\mathbf{\Lambda}_i(t)$ increases both memory usage and computational cost, especially when backpropagating through long and irregular trajectories under stochastic dynamics. We therefore adopt a decoupled strategy. We first solve for the latent path $\mathbf{z}_i(t)$, then evaluate the intensity $\boldsymbol{\lambda}_i(t)$ at the required time points, and finally approximate the integrated intensity $\mathbf{\Lambda}_i(t)$ by numerical quadrature. This approach preserves the flexibility of the latent-state-dependent point-process model while remaining scalable to high-dimensional latent spaces and large patient cohorts.

A.3 Extended details on synthetic data generation

As outlined in Figure 1, generating a fully synthetic individual follows a hierarchical sampling path. This ensures that the generated longitudinal trajectories are intrinsically conditioned on the generated static baseline characteristics. The exact procedure for generating a synthetic subject j is as follows:

1. **Static covariate generation:** We sample the latent representations $\mathbf{h}_j = (\mathbf{s}_j, \mathbf{u}_j)$ from the HI-VAE using either *posterior* or *prior* sampling. Under posterior sampling, latent representations are drawn from individual approximate posteriors $q(\mathbf{s}_i, \mathbf{u}_i \mid \mathbf{W}_i)$, then pooled, shuffled, and

subsampled to form a synthetic cohort without preserving patient correspondence. Under prior sampling, they are drawn directly from the learned HI-VAE prior. We then decode these latents to obtain the synthetic static covariates $\tilde{\mathbf{W}}_j$. The two sampling strategies are compared in the sensitivity analysis Table 15.

2. **Conditioning the dynamics:** We map the static latent representation $\mathbf{h}_j$ through the hypernetwork $\mathcal{H}$ to generate the individualized drift parameters $\theta_{b,j} = \mathcal{H}(\tilde{\mathbf{h}}_j; \theta_{\mathcal{H}})$. This is the crucial step that preserves the static–longitudinal dependency.
3. **Latent trajectory simulation:** In practice, rather than simulating the full latent SDE of Eq. (2) directly, we use the deterministic–stochastic decomposition introduced in Section 4 and simulate the two components separately. The deterministic part $\mathbf{z}_j^{\mathrm{det}}(t)$ is obtained by solving the personalized Neural ODE with individualized drift parameters $\theta_{b,j}$, capturing the patient-specific smooth trend. The stochastic residual $\mathbf{z}_j^{\mathrm{sto}}(t)$ is simulated from the population-shared Neural SDE, capturing intra-individual stochastic fluctuations. The full latent path is $\mathbf{z}_j(t) = \mathbf{z}_j^{\mathrm{det}}(t) + \mathbf{z}_j^{\mathrm{sto}}(t)$, with both components initialized at $\mathbf{0}_{d_z}$.
4. **Irregular sampling simulation:** To replicate the irregular observation patterns of the real data, we compute the event intensities $\boldsymbol{\lambda}_j(t)$ based on the latent path via Equation (4), and sample a set of discrete observation times $\tilde{\mathcal{T}}_j$.
5. **Longitudinal decoding:** Finally, we evaluate each latent component at the sampled observation times and map these states through the decoder network to obtain the observable longitudinal values: $\tilde{\mathbf{x}}_j(t) = g(\mathbf{z}_j^{\mathrm{det}}(t); \theta_g) + g(\mathbf{z}_j^{\mathrm{sto}}(t); \theta_g)$.

This yields synthetic multimodal records preserving static–longitudinal dependence, temporal variability, and irregular observation patterns.

B Training details/hyperparameters

B.1 Deterministic–stochastic decomposition: further discussion

Full specification The deterministic component $\mathbf{z}_i^{\mathrm{det}}$ solves the personalized Neural ODE

$$d\mathbf{z}_i^{\mathrm{det}}(t) = b_i^{\mathrm{det}}\big(\mathbf{z}_i^{\mathrm{det}}(t), t;\, \theta_{b^{\mathrm{det}},i}\big)\, dt, \qquad \mathbf{z}_i^{\mathrm{det}}(0) = \mathbf{0},$$

where $\theta_{b^{\mathrm{det}},i} = \mathcal{H}(\mathbf{h}_i; \theta_{\mathcal{H}})$ are the hypernetwork-generated parameters, trained via the reconstruction loss $\mathcal{L}_{\mathrm{long,det}}$ (Eq. 8). The stochastic residual $\mathbf{z}_i^{\mathrm{sto}}$ solves the population-shared Neural SDE

$$d\mathbf{z}_i^{\mathrm{sto}}(t) = b^{\mathrm{sto}}\big(\mathbf{z}_i^{\mathrm{sto}}(t), t;\, \theta_{b^{\mathrm{sto}}}\big)\, dt + \sigma^{\mathrm{sto}}\big(\mathbf{z}_i^{\mathrm{sto}}(t), t;\, \theta_{\sigma^{\mathrm{sto}}}\big)\, d\mathbf{B}_i(t), \qquad \mathbf{z}_i^{\mathrm{sto}}(0) = \mathbf{0},$$

where b^{sto} is a low-capacity drift network and σ^{sto} models residual stochastic fluctuations, trained via the signature kernel score $\mathcal{L}_{\mathrm{long,sto}}$ (Eq. 9).

Then, the decomposition $\mathbf{z}_i = \mathbf{z}_i^{\mathrm{det}} + \mathbf{z}_i^{\mathrm{sto}}$ defines a different model from a single latent SDE on $\mathbf{z}_i$. The total process satisfies

$$d\mathbf{z}_i(t) = \big[b_i^{\mathrm{det}}(\mathbf{z}_i^{\mathrm{det}}, t) + b^{\mathrm{sto}}(\mathbf{z}_i^{\mathrm{sto}}, t)\big]\, dt + \sigma^{\mathrm{sto}}(\mathbf{z}_i^{\mathrm{sto}}, t)\, d\mathbf{B}_i(t),$$

which matches the form of a single latent SDE only when the drift decomposes additively across the two components — a condition satisfied exactly in the linear case and approximately otherwise. The decomposition is therefore a deliberate modeling choice. The low-capacity drift b^{sto} absorbs any residual trend not captured by $\mathbf{z}_i^{\mathrm{det}}$, ensuring the decomposition remains well-behaved even when the linear approximation breaks down.

Connection to related work Separately estimating drift and diffusion from discrete observations is a classical statistical problem [13, 40, 70], but existing methods typically require the number of observed time points per path to tend to infinity — an assumption incompatible with our sparse irregular setting. To the best of our knowledge, no prior work proposes a decomposition of the form (5) in the context of Neural SDEs. Our decomposition is instead a modeling choice inspired by the Doob–Meyer decomposition of a semimartingale into a predictable finite-variation process and a local martingale [49], and by the FDA practice of analyzing residual trajectories separately from a smooth trend [51, 65].

B.2 Additional details on signature-kernel loss for training Neural SDEs

B.2.1 Signature-kernel score

To train stochastic path-valued generative models without adversarial optimization, we use the *signature kernel score* introduced in [53, 21]. This score compares probability distributions on paths through a positive-definite kernel defined from path signatures.

Path signature Let $X : [0, T] \to \mathbb{R}^d$ be a continuous path of bounded variation. Its signature is the sequence of iterated integrals

$$S(X) = \Big(1,\ \int_0^T dX_{t_1},\ \int_{0<t_1<t_2<T} dX_{t_1} \otimes dX_{t_2},\ \dots\Big).$$

The signature provides a nonparametric representation of the path and characterizes it up to mild equivalence relations [53]. Importantly, signature-based representations do not require regularly sampled trajectories: they are naturally defined for continuous paths and can be computed from discretely observed paths on irregular time grids.

Signature kernel For two paths X and Y, the signature kernel is defined as the inner product of their signatures in the tensor algebra,

$$k_{\mathrm{sig}}(X, Y) = \big\langle S(X), S(Y)\big\rangle.$$

In practice, normalized or RBF-transformed versions are often used for numerical stability; in our experiments, we use an RBF-signature kernel as in [21]. This defines a positive-definite kernel on path space and therefore an associated reproducing kernel Hilbert space.

Signature kernel score Let $\mathbb{P}_\theta$ denote the law of trajectories generated by the model and let $X \sim \mathbb{P}_{\mathrm{data}}$ denote a data path. The signature kernel score of $\mathbb{P}_\theta$ at X is

$$\phi_{\mathrm{sig}}(\mathbb{P}_\theta, X) = \mathbb{E}_{\hat{X},\hat{X}'\sim\mathbb{P}_\theta}\big[k_{\mathrm{sig}}(\hat{X}, \hat{X}')\big] - 2\,\mathbb{E}_{\hat{X}\sim\mathbb{P}_\theta}\big[k_{\mathrm{sig}}(\hat{X}, X)\big],$$

where $\hat{X}, \hat{X}'$ are independent samples from the model. The expected training objective is then

$$\mathcal{L}_{\mathrm{sig}} = \mathbb{E}_{X\sim\mathbb{P}_{\mathrm{data}}}\big[\phi_{\mathrm{sig}}(\mathbb{P}_\theta, X)\big].$$

This score differs from the full kernel MMD only by a term depending solely on the data distribution, and is therefore equivalent for optimization with respect to the model parameters.

Empirical estimator Given two mini-batches of trajectories

$$\mathcal{B}_{\hat{X}} = \{\hat{X}^i\}_{i=1}^B \sim \mathbb{P}_\theta, \qquad \mathcal{B}_X = \{X^j\}_{j=1}^B \sim \mathbb{P}_{\mathrm{data}},$$

a consistent empirical estimator of the signature kernel score is

$$\widehat{\mathcal{L}}_{\mathrm{sig}} = \frac{1}{B(B-1)}\sum_{i\neq j} k_{\mathrm{sig}}(\hat{X}^i, \hat{X}^j) - \frac{2}{B^2}\sum_{i=1}^B\sum_{j=1}^B k_{\mathrm{sig}}(\hat{X}^i, X^j).$$

This is the form used in practice for training.

B.3 Hyperparameters optimization method

Aggregation strategies Aggregating the four loss terms defined in Section 4 is non-trivial as they differ in scale and sign (e.g., MSE vs. log-likelihood). We compared four standard multi-objective weighting strategies [62] to combine the individual losses into $\mathcal{L}_{\mathrm{total}}$ (Eq. 10): *uncertainty-based weighting* [24], *GradNorm* [11], *initial-loss scaling*, and a *linear weighted sum*. In our experiments, uncertainty weighting and GradNorm were unstable, likely due to the heterogeneous nature of the objectives. Initial-loss scaling gave reasonable results but was too sensitive to initialization. We therefore use a linear weighted combination

$$\mathcal{L}_{\mathrm{total}} = \alpha_1\mathcal{L}_{\mathrm{static}} + \alpha_2\mathcal{L}_{\mathrm{event}} + \alpha_3\mathcal{L}_{\mathrm{long,det}} + \alpha_4\mathcal{L}_{\mathrm{long,sto}}, \qquad \alpha_j \geq 0,\quad \sum_{j=1}^4 \alpha_j = 1, \tag{10}$$

which provided the most stable training and the best validation trade-off.

Pareto hyperparameter optimization The weights $\boldsymbol{\alpha}$ are tuned jointly with the other model hyperparameters using Pareto-front optimization with Optuna [2]: each trial k trains the model on $\mathcal{L}_{\text{total}}(\boldsymbol{\alpha}^{(k)})$ and records the unweighted validation losses $\{\mathcal{L}_j^{\text{val}}\}_j$ as separate objectives. After approximately 100 trials, we select a configuration from the Pareto front that achieves the best trade-off across all validation objectives. This prevents any single component from dominating optimization.

Trade-off analysis Figure 3 shows the Pareto front obtained from the hyperparameter optimization of our model on the reference simulated-data setting, in the $(\mathcal{L}_{\text{long,det}}, \mathcal{L}_{\text{static}})$ plane. It reveals a clear trade-off between the two objectives: configurations that improve longitudinal reconstruction tend to worsen static reconstruction, and vice versa. Its broad coverage of the objective space indicates that the model supports a diverse set of non-dominated solutions, rather than collapsing to a narrow region. We select an operating point on this front that provides a favorable compromise between the two objectives.

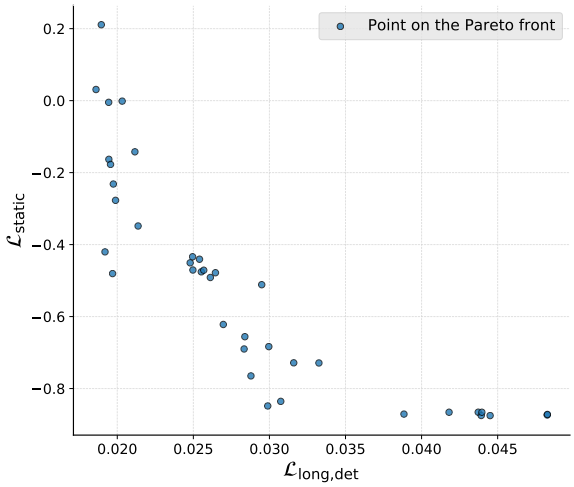

Figure 3: Pareto front between the longitudinal deterministic reconstruction loss $\mathcal{L}_{\text{long,det}}$ and the static reconstruction loss $\mathcal{L}_{\text{static}}$. Each point corresponds to a non-dominated hyperparameter configuration found by Optuna for our model on the reference simulated-data setting.

Hyperparameter values/search space Table 5 summarizes the search space explored by Optuna across approximately 100 trials per dataset. Other hyperparameters are fixed across all experiments based on preliminary exploration and kept constant during Optuna tuning. The ODE drift and decoder use `Tanh` activations; the hypernetwork uses `ReLU`; the diffusion and residual drift networks use `LipSwish` [26], which was found to improve stability of the SDE training. The ODE is solved with the `dopri5` adaptive-step solver (tolerances $\texttt{rtol} = 10^{-3}$, $\texttt{atol} = 10^{-4}$); the SDE uses Stratonovich integration with diagonal noise. Batch normalization is applied to static latent representations. Observation times are normalized to $[0, 1]$.

Table 5: Hyperparameter search space.

Hyperparameter	Range / Values	Type
Batch size	$\{128, 256\}$	categorical
Learning rate	$\{10^{-4}, 10^{-3}, 2 \cdot 10^{-3}, 5 \cdot 10^{-3}, 10^{-2}\}$	categorical
Static latent dims. d_{s}, d_{u}	$[2,\ 8]$	integer
Longitudinal latent dim. d_{z}	$[2,\ d_{\mathrm{z}}]$	integer
Hidden dim.	$\{16, 32, 64, 128\}$	categorical
ODE/SDE layers	$[1,\ 4]$	integer
Hypernetwork layers	$[1,\ 5]$	integer
Diffusion MLP layers	$[1,\ 3]$	integer
Dropout (decoder)	$\{0,\ 0.1,\ 0.2\}$	categorical
Weight decay	$[10^{-7},\ 10^{-2}]$	log-uniform
SDE training kernel $\sigma^{\dagger}$	$[0.01,\ 1.0]$	log-uniform
Loss weight α_{det}	$[0.1,\ 100]$	log-uniform
Loss weight α_{sto}	$[0.01,\ 100]$	log-uniform
Loss weight α_{event}	$[0.001,\ 1]$	log-uniform

† Kernel bandwidth used in the signature-kernel training loss $\mathcal{L}_{\mathrm{sto}}$, distinct from the evaluation MMD bandwidth.

C Evaluation metrics

C.1 Challenges and limitations of existing approaches

Evaluating synthetic clinical data remains an open problem, particularly for irregularly sampled longitudinal data. Most existing benchmarking frameworks and toolkits — including Synthcity [50] and TSGBench [4] — are designed primarily for regularly sampled time series or tabular data. When applied to irregular longitudinal data, they typically require an interpolation or imputation step as preprocessing, which can substantially distort the original temporal structure and introduce systematic biases into the evaluation results.

Beyond tooling, there is no universally accepted evaluation protocol for synthetic data, even in the simpler tabular setting [57, 1, 28, 36]. The selection of an appropriate set of measures is recognized as an open challenge, and recent work has shown that results can be highly sensitive to implementation choices — for instance, [56] demonstrated a notable influence of the embedding strategy on embedding-based metrics for synthetic time series.

In practice, a large proportion of papers on time-series generation report only fidelity and utility metrics, often relying on qualitative visual inspection rather than rigorous quantitative evaluation [3, 61]. Sampling regularity, event dependence, and cross-modal structure are almost never evaluated. Privacy, despite being a central requirement in healthcare, remains inconsistently assessed [39].

Our work requires evaluating jointly static covariates and irregular longitudinal trajectories, while accounting for their cross-modal dependencies, the event-driven observation process, and missingness. This combination falls outside the scope of most existing toolkits and metrics. Developing new metrics fully adapted to this setting is itself a research problem beyond the scope of this paper; however, we believe this discussion highlights a real and underappreciated gap in the literature that method papers in this space should explicitly address.

C.2 Our evaluation strategy

We evaluate synthetic data quality along three complementary axes — *fidelity*, *utility*, and *privacy* — using metrics that are adapted to the multimodal and irregular nature of our setting. When standard metrics are available, we use modality-aware variants that operate directly on irregular observations without interpolation or imputation. We describe each metric precisely below.

C.2.1 Fidelity metrics

Maximum Mean Discrepancy (MMD) We use the Maximum Mean Discrepancy [18] to compare the distributions of real and synthetic samples. Given two sample sets $X = \{x_i\}_{i=1}^{n}$ and $\tilde{X} =$

$\{\tilde{x}_j\}_{j=1}^m$, the empirical squared MMD is

$$\mathrm{MMD}^2(X, \tilde{X}) = \frac{1}{n(n-1)} \sum_{i \neq i'} k(x_i, x_{i'}) + \frac{1}{m(m-1)} \sum_{j \neq j'} k(\tilde{x}_j, \tilde{x}_{j'}) - \frac{2}{nm} \sum_{i,j} k(x_i, \tilde{x}_j),$$

where k is a positive-definite kernel. For static variables, we use a standard RBF kernel on the static covariates. For longitudinal data, we use a signature-kernel MMD [12, 53], which is well suited to irregularly sampled trajectories: paths are time-augmented and compressed according to the observation mask, enabling comparisons directly on irregular trajectories. Lower values indicate better agreement between real and synthetic distributions.

The signature-kernel MMD with an RBF static kernel, computed with `PySigLib` [54], depends critically on the bandwidth parameter σ. If σ is poorly matched to the scale of pairwise path discrepancies in signature space, the kernel values saturate, and the metric loses discriminative power. In this regime, the empirical estimator may also become numerically unstable; in particular, the unbiased estimate of MMD^2 can take negative values at finite sample size, and this effect may be exacerbated in practice by long, sparse, and highly irregular trajectories. To improve stability, we select σ using a median heuristic applied after time augmentation and path compression, and we linearly interpolate trajectories between observed points before computing the signature kernel. The latter ensures that the evaluated paths are piecewise linear, but sparse and irregular sampling may still make the interpolated paths only rough proxies for the underlying continuous trajectories. As a result, the signature-kernel MMD should be interpreted with caution, especially when comparing models that differ substantially in smoothness or in the induced observation patterns.

Correlation RMSE To assess whether synthetic data preserve pairwise dependency structure, we compare real and synthetic correlation matrices:

$$\text{Corr. RMSE} = \sqrt{\frac{1}{|\Omega|} \sum_{(p,q) \in \Omega} \left(C_{pq}^{\text{real}} - C_{pq}^{\text{syn}}\right)^2},$$

where Ω denotes the set of valid entries. For static variables, correlations are computed after type-aware preprocessing (standardization for continuous, rank encoding for ordinal, integer encoding for categorical). For longitudinal data, we flatten across individuals and time and compute a masked correlation matrix using only overlapping observed pairs, excluding entries with insufficient overlap. We also report a cross-modal Corr. RMSE computed from a joint masked correlation matrix augmented with static covariates. Lower values indicate better preservation of dependency structure.

KL divergence on the observation intensities (KL λ) To evaluate whether synthetic data reproduce the irregular and informative observation process, we compare the empirical event intensities of the real and synthetic datasets. For each observed variable and for the end-of-sequence (EOS) process, we estimate a piecewise-constant population intensity as the number of events divided by the corresponding person-time at risk,

$$D_k = \sum_b \frac{E_b^R}{N_R} \left[\lambda_{k,b}^R \log \frac{\lambda_{k,b}^R}{\lambda_{k,b}^S} - \lambda_{k,b}^R + \lambda_{k,b}^S \right]$$

where b indexes the time bins, E_b^R denotes the total real-data person-time exposure in bin b, N_R is the number of real subjects, and $\lambda_{k,b}^R$ and $\lambda_{k,b}^S$ are the corresponding empirical real and synthetic intensities. We report the component-averaged Poisson intensity divergence, $\frac{1}{d_x+1}\left(\sum_{\ell=1}^{d_x} D_\ell + D_{\text{EOS}}\right)$, as KL λ. It captures discrepancies in both the magnitude and temporal shape of the observation intensities while explicitly accounting for the EOS termination process. Lower values indicate better agreement between the real and synthetic observation processes.

Discriminative score As a joint fidelity measure, we use a classifier two-sample test (C2ST) [66]: a classifier is trained to distinguish real from synthetic samples, taking as input the longitudinal trajectories (with last-observation-carried-forward imputation) augmented with static covariates. The classifier is based on an S4 architecture [19] with masked global average pooling, ensuring predictions depend only on observed time steps. We report the test AUROC; values closer to 0.5 indicate that synthetic samples are harder to distinguish from real data, and hence better fidelity.

C.2.2 Utility metrics

Predictive score We assess downstream utility through a forecasting task under a Train-on-Synthetic-Test-on-Real (TSTR) protocol [17], compared with the Train-on-Real-Test-on-Real (TRTR) baseline. A predictor based on an S4 architecture [19, 16], is trained to predict m future time points ($\mathbf{x}_i(t_{L+1:L+m})$) from a observed prefix ($\mathbf{x}_i(t_{1:L})$) concatenated with static covariates. Missing values are imputed via last-observation-carried-forward. Performance is measured using Mean Squared Error (MSE) on real test data. Synthetic data yields better downstream utility when its TSTR MSE closely matches the Train-on-Real (TRTR) reference.

C.2.3 Privacy metrics

Nearest-Neighbor Adversarial Accuracy (NNAA) NNAA [64] measures whether the synthetic data distribution has systematically overfitted to the training data. For two datasets R (real) and S (synthetic), the adversarial accuracy (AA) is defined as the proportion of samples that are closer to a neighbor in their own set than to a neighbor in the opposite set:

$$\mathrm{AA}(R, S) = \frac{1}{2}\left(\frac{1}{|R|}\sum_{r\in R}\mathbb{I}(d_{RS}(r) > d_{RR}(r)) + \frac{1}{|S|}\sum_{s\in S}\mathbb{I}(d_{SR}(s) > d_{SS}(s))\right)$$

where $d_{AB}(x)$ is the distance from $x \in A$ to its nearest neighbor in B. If S simply memorizes R, AA drops below 0.5. We compute this quantity once using the training set ($\mathrm{AA}_{\text{train}}$) and once using a held-out test set ($\mathrm{AA}_{\text{test}}$). The final metric is $\mathrm{NNAA} = \mathrm{AA}_{\text{test}} - \mathrm{AA}_{\text{train}}$. Large positive values indicate that synthetic samples are systematically closer to training data than to unseen real data, suggesting higher memorization risk. Distances are computed separately for static and longitudinal modalities (using pairwise Euclidean and signature-based distances, respectively). For the multimodal NNAA score, the modality-specific distance matrices are normalized and combined via a weighted sum. Lower NNAA values indicate lower privacy risk.

Membership Inference Risk (MIR) MIR [55] quantifies the vulnerability of the synthetic dataset to a membership inference attack, assessing whether an adversary can distinguish training samples from held-out test samples based strictly on their distance to the synthetic data. For each real sample x, we compute an attack score $s(x) = -d_k(x, S)$, where d_k is the aggregated distance (e.g., the minimum or mean) to the k nearest synthetic neighbors. Treating training samples as the positive class and test samples as the negative class, we evaluate the discriminative power of $s(x)$ using the AUROC. We report the rescaled risk:

$$\mathrm{MIR} = 2\,|\mathrm{AUROC} - 0.5|$$

so that $\mathrm{MIR} \in [0, 1]$, where 0 corresponds to random guessing (no privacy leak). As with NNAA, MIR is computed per modality and then aggregated in the same manner. Lower MIR values indicate lower privacy risk.

Dependence Score We report a dependence score based on the temporal autocorrelation structure. For each longitudinal variable and lag k, we compute a dataset-level autocorrelation by averaging patient-level Pearson autocorrelations over jointly observed pairs, yielding a matrix $R \in \mathbb{R}^{V\times(K+1)}$. The dependence score is the Mean Squared Error (MSE) between the real and synthetic matrices:

$$\text{Dep. Score} = \frac{1}{|\Omega|}\sum_{(v,k)\in\Omega}\left(R^{\text{real}}_{v,k} - R^{\text{syn}}_{v,k}\right)^2$$

where Ω is the set of valid lag entries up to a maximum lag K. This metric serves as a critical dual-purpose diagnostic. Primarily, it acts as a fidelity measure: generative models that over-smooth the data fail to capture natural trajectory noise, resulting in an artificially high synthetic autocorrelation and, consequently, a large MSE. Conversely, recent benchmarking by Stenger et al. [56] empirically shows that autocorrelation error is also a top-performing proxy for privacy risk. If the error is suspiciously close to zero, it indicates the model is likely memorizing the exact trajectory noise of specific training samples rather than learning the underlying distribution. Thus, an optimal model must achieve a low autocorrelation error without exactly replicating the training data.

D Details on datasets

D.1 Formatting and preprocessing of the data

To enable batch processing while preserving irregular sampling patterns, we first define a common time grid as the union of all individual observation times:

$$\mathcal{T} = \bigcup_{i=1}^{N} \bigcup_{\ell=1}^{d_{\mathrm{x}}} \mathcal{T}_i^{(\ell)}.$$

Each individual trajectory is then represented on this grid, yielding a three-dimensional tensor

$$\mathbf{X} \in \mathbb{R}^{N \times |\mathcal{T}| \times d_{\mathrm{x}}},$$

where unobserved entries are padded with zeros. To explicitly encode missingness, we introduce a binary mask $\mathbf{M}$ of the same dimension, where $\mathbf{M}_{i,t,\ell} = 1$ if $\mathbf{x}_i^{(\ell)}(t)$ is observed and $\mathbf{M}_{i,t,\ell} = 0$ otherwise. Baseline measurements of the longitudinal variables, $\mathbf{x}_i^{(\ell)}\big(t_{i,0}^{(\ell)}\big)$, are extracted and appended to the static covariates, resulting in an augmented static feature vector of dimension $d_{\mathrm{W}} + d_{\mathrm{x}}$. These baseline values are then subtracted from the corresponding longitudinal trajectories, so that each time series represents deviations from baseline. Finally, each longitudinal variable is rescaled by its standard deviation computed on the training set. This normalization puts variables on comparable scales and improves numerical stability during training.

D.2 Simulated datasets

We simulate multimodal datasets in which static covariates explicitly control longitudinal dynamics.

Static variables For each individual i, we generate static variables

$$\mathbf{W}_i = [\mathbf{W}_{i,\mathrm{num}}, \mathbf{W}_{i,\mathrm{bin}}],$$

where $\mathbf{W}_{i,\mathrm{num}}$ denotes numerical variables and $\mathbf{W}_{i,\mathrm{bin}}$ binary variables. The numerical components are sampled as

$$\mathbf{W}_{i,\mathrm{num}} \sim \mathcal{N}(\mathbf{0}, \sigma_{\mathrm{num}}^2 \mathbf{I}).$$

The binary components are generated through a probit model. Let

$$\tilde{\mathbf{W}}_{i,\mathrm{bin}} \sim \mathcal{N}(\mathbf{0}, \sigma_{\mathrm{bin}}^2 \mathbf{I}), \qquad u_{i,j} \sim \mathcal{U}(0,1),$$

and let $c_j \sim \mathcal{N}(0,1)$ denote fixed coefficients. Then, for each binary dimension j,

$$\mathbf{W}_{i,\mathrm{bin}}^{(j)} = \mathbf{1}\Big[u_{i,j} < \Phi\Big(\tilde{\mathbf{W}}_{i,\mathrm{bin}}^{(j)} c_j\Big)\Big],$$

where Φ is the standard normal cumulative distribution function. In the experiments reported in the main paper, we use $d_W = 2$, corresponding to one numerical and one binary feature, $\sigma_{\mathrm{num}}^2 = 0.07$ and $\sigma_{\mathrm{bin}}^2 = 0.01$.

Longitudinal variables Conditioned on $\mathbf{W}_i$, the longitudinal process $\mathbf{x}_i(t) \in \mathbb{R}^{d_{\mathrm{x}}}$ follows independent fractional Ornstein–Uhlenbeck dynamics,

$$d\mathbf{x}_i^{(\ell)}(t) = -a_{i,\ell}\big(\mathbf{x}_i^{(\ell)}(t) - b_{i,\ell}\big)\, dt + \zeta\, dB_{i,\ell}^H(t), \qquad \ell \in \{1, \ldots, d_{\mathrm{x}}\},$$

where $B_{i,\ell}^H(t)$ is a fractional Brownian motion with Hurst parameter $H \in (0,1)$. The subject-specific parameters are obtained from the static covariates through linear maps:

$$a_{i,\ell} = s\,(\Omega^{(a)} \mathbf{W}_i)_\ell, \qquad b_{i,\ell} = s\,(\Omega^{(b)} \mathbf{W}_i)_\ell,$$

with $\Omega^{(a)}, \Omega^{(b)} \in \mathbb{R}^{d_{\mathrm{x}} \times d_W}$ random weight matrices and $s \in \mathbb{R}$ a fixed scaling factor. In our experiments, we set $d_{\mathrm{x}} = 3, \zeta = 0.1$ and $H = 0.4$.

Observation-time points To mimic irregular clinical monitoring, observation times are generated from an inhomogeneous Poisson process on $[0, T]$ with intensity $\lambda_i(t)$. We consider two observation schemes.

Time-dependent sampling. In the baseline setting, the observation process depends only on time:

$$\lambda_i(t) = \alpha + (1 - \alpha)\frac{1}{2}\sin\left(\frac{-2\pi t}{T}\right), \tag{11}$$

where $T = 7$ in our experiments, and the constant α is calibrated to obtain the desired average number of observations per individual.

State-dependent sampling. In the informative-sampling setting, the observation intensity depends on the underlying trajectory:

$$\lambda_i(t) = c\,\exp\left(-\sum_{\ell=1}^{d_{\mathrm{x}}} \mathbf{x}_i^{(\ell)}(t)\right), \tag{12}$$

where $c > 0$ is chosen so that the expected number of observations matches a prescribed target.

In both settings, visit times are generated by a discrete thinning algorithm applied on the regular grid, and latent trajectories are first simulated on a fine regular grid and then observed only at the sampled visit times.

In the main experiments, repeated Monte Carlo runs correspond to independently generated datasets obtained with different random seeds.

D.3 Real-world datasets

D.3.1 Preprocessing pipeline

All real-world datasets were preprocessed using the same general pipeline, with minor dataset-specific adaptations. We first separated baseline static covariates from longitudinal variables using dataset documentation and empirical missingness patterns. Static variables correspond to time-invariant or quasi-time-invariant patient characteristics, while longitudinal variables correspond to repeatedly measured clinical quantities.

We then applied the following steps. *(i) Static variables:* sentinel values were replaced by missing values; nominally static variables recorded multiple times were aggregated at the patient level using the median; variables observed in fewer than 50% of patients were discarded. *(ii) Longitudinal variables:* original observation times were preserved; duplicate patient-time entries were averaged; rows with no observed longitudinal variable were removed; variables with overall presence below a dataset-specific threshold were discarded (typically 30% or 50% depending on the dataset). *(iii) Patient filtering:* we retained only patients with both static and longitudinal information and removed patients with fewer than 5 longitudinal observation times after filtering. *(iv) Time normalization:* observation times were shifted so that each patient's first visit occurred at time zero (or study start for VELOUR), and a common fine temporal grid was defined as the union of all observed times. Longitudinal data were finally represented on this grid together with a binary observation mask. When needed, skewed positive variables were log-transformed before standardization.

The final numbers of retained patients, static variables, longitudinal variables, and grid points are reported in Table 6.

Table 6: Summary statistics of the real-world datasets after preprocessing.

Dataset	Patients	Long. variables	Static variables	Common grid points	Avg. observed visits/patient	Time span (days)
VELOUR	526	23	26	673	21.3	23–990
PPMI	1930	15	20	686	11.2	153–5448

D.3.2 Dataset-specific notes

VELOUR [25, 48] We use data from the VELOUR phase III randomized trial in metastatic colorectal cancer. Static covariates include demographics (sex, race, geographic region, age category),

baseline ECOG performance status, height, and binary indicators for the 10 most frequent adverse events and concomitant medications recorded at baseline. Retained longitudinal variables correspond to standard oncology laboratory panels (hepatic enzymes, blood counts, metabolic markers) and vital signs. Observations prior to treatment initiation were excluded, and highly right-skewed laboratory variables were log-transformed before standardization. VELOUR can be accessed via Project DataSphere (`https://data.projectdatasphere.org`), which requires registration and a data use agreement.

PPMI [38] We use participants from the Parkinson's Progression Markers Initiative, a multi-year observational cohort on Parkinson's disease progression. Static covariates include demographics (age, sex), enrollment cohort, genetic risk factors (LRRK2, SNCA status), and family history of Parkinson's disease. Retained longitudinal variables include vital signs (temperature, supine and standing blood pressure, and heart rate) and clinical rating scales (UPDRS parts 1–3, ESS, SCOPA-AUT, STAI). PPMI is available upon registration at `https://www.ppmi-info.org`.

E Experimental setup

E.1 Experiments setting and compute resources

All models are trained using AdamW [37]. HyperNSDE is trained with early stopping (patience 10, maximum 1000 epochs). All experiments were run on the Jean Zay cluster using single-GPU jobs on NVIDIA V100 GPU with 32 GB of memory, 10 CPU cores per task. and a wall-clock limit of 20 hours per job. On the reference simulated dataset, one epoch takes approximately 1 min 20 s, with training terminating around epoch 500 ($\approx$11 hours). Training time scales with the number of patients and observations for real-world datasets.

E.2 Baseline implementations and tuning

We used the public implementations of MultiNODEs [63], RTSGAN [46], and DGBFGP [6], with dataset-specific preprocessing adapted to our setting.[1]

Fair hyperparameter tuning To ensure a fair comparison with our model, we did not rely solely on the default hyperparameter settings provided in the public implementations. Whenever such a procedure was not already implemented in the original code, we performed an additional hyperparameter search using Optuna [2]. Each baseline was tuned using its native training objective, evaluated on the validation set, as a single optimization target. This additional tuning step was introduced to reduce the risk that performance differences arise from unequal optimization effort rather than from the underlying modeling assumptions.

DGBFGP + HI-VAE Since DGBFGP does not provide a generative model for static covariates, we coupled its longitudinal component with an HI-VAE trained separately on the static variables. As a result, static and longitudinal generation are not jointly learned in this baseline, unlike in our model and MultiNODEs. For this reason, and because DGBFGP requires substantial dataset-specific manual configuration, we restrict this baseline to the simulated-data experiments.

E.3 Details on the sensitivity analysis

We assess the robustness of all methods under several controlled modifications of the reference setting. We consider the following sensitivity scenarios.

(i) Limited training data: to evaluate sensitivity to training sample size, we reduce the training set to $N_{\text{train}} = 200$ while keeping the test set fixed at $N_{\text{test}} = 200$.

(ii) Sparse observation regime: to test robustness to increased irregularity and missingness, we keep the standard 80/20 split but reduce the average number of visits to $n_{\text{obs}} = 15$ per patient.

[1] Official repositories: MultiNODEs `https://github.com/SCAI-BIO/MultiNODEs/tree/main`, RTSGAN `https://github.com/acphile/RTSGAN`, and DGBFGP `https://github.com/YigitBalik/DGBFGP`.

(iii) State-dependent observation process: we replace the baseline time-dependent intensity by a more complex event model in which the visit intensity λ depends on the longitudinal state $\mathbf{x}$ defined in Equation 12, thereby introducing informative sampling. Unless otherwise stated, all other simulation parameters are kept identical to the reference setting.

(iv) Prior versus posterior sampling: to assess sensitivity to the static latent sampling strategy, we compare *posterior sampling (post.)* and *prior sampling (prior)*. Under posterior sampling, static latent representations are drawn from individual approximate posteriors, pooled, shuffled, and subsampled to form a synthetic cohort without preserving patient correspondence. Under prior sampling, they are drawn directly from the learned HI-VAE prior. We report posterior sampling in the main experiments for consistency with MultiNODEs and use this sensitivity analysis to assess whether the results depend on this choice.

(v) State dependence of the observation intensity: to assess the contribution of conditioning the observation process on the latent state, we compare the standard HyperNSDE intensity $\lambda_i(t) = \exp\{\xi(\mathbf{z}_i(t), t; \theta_\xi)\}$ with a state-independent variant in which the intensity depends only on time, $\lambda_i(t) = \exp\{\xi(t; \theta_\xi)\}$. For the simulated data case, we use the simulated data of the sensitivity analysis *(iii)*. All other model components and training procedures are unchanged.

E.4 Details on the ablation studies

All ablation studies are conducted on the reference simulated-data setting described in Section 5. Unless otherwise stated, all remaining architectural choices, hyperparameters, and evaluation protocols are kept unchanged.

(AS.1) Deterministic latent dynamics. In this variant, we replace the latent Neural SDE by a latent Neural ODE, thereby removing the stochastic component of the latent dynamics. The latent trajectory satisfies

$$d\mathbf{z}_i(t) = b_i\big(\mathbf{z}_i(t), t\big)\, dt, \qquad \mathbf{z}_i(0) = \mathbf{0}.$$

This ablation evaluates whether explicitly modeling stochasticity in latent space improves generative performance compared with a purely deterministic trajectory model. It is close in spirit to MultiNODEs, but adapted to our framework without an encoder and with the same event-intensity module.

(AS.2) Direct stochastic training without decomposition. In the proposed model, the latent path is decomposed into a deterministic component and a stochastic residual, each trained with a separate objective. In this ablation, we remove this decomposition and model the full latent path directly as a Neural SDE. The stochastic model is then trained directly from the signature-kernel discrepancy loss on the full latent trajectory. This ablation tests whether the proposed decomposition improves optimization stability and disentangles deterministic dynamics from stochastic fluctuations.

(AS.3) Alternative conditioning of longitudinal dynamics. We compare the proposed hypernetwork based personalization with two simpler strategies for incorporating the static latent representation $\mathbf{h}_i$ into the longitudinal dynamics.

- ***MNSDE:*** Inspired by MultiNODEs, we concatenate the static latent representation only to the initial latent state:

$$\mathbf{z}_{i,0}^{\mathrm{aug}} = \begin{pmatrix} \mathbf{z}_i(0) \\ \mathbf{h}_i \end{pmatrix} \in \mathbb{R}^{d_z + d_h}.$$

  The augmented process is then evolved according to

$$d\mathbf{z}_i^{\mathrm{aug}}(t) = b\big(\mathbf{z}_i^{\mathrm{aug}}(t), t; \theta_b\big)\, dt + \sigma\big(\mathbf{z}_i^{\mathrm{aug}}(t), t; \theta_\sigma\big)\, dB_t.$$

  In this formulation, static information is only injected at $t = 0$.
- ***SNSDE:*** Inspired by modulated dynamics models such as [5], we concatenate the static latent representation to the latent state throughout the trajectory:

$$d\mathbf{z}_i(t) = b\big(\mathbf{z}_i(t), t, \mathbf{h}_i; \theta_b\big)\, dt + \sigma\big(\mathbf{z}_i(t), t; \theta_\sigma\big)\, dB_t.$$

  In practice, this amounts to providing $\mathbf{h}_i$ as an input to the drift network at all time points. This ablation evaluates whether the proposed hypernetwork parameterization is more effective than direct concatenation-based conditioning.

– ***FNSDE:*** We additionally consider FiLM-based personalization, where $\mathbf{h}_i$ generates layer-wise scale and shift parameters that modulate each hidden layer of the shared drift network. Unlike the hypernetwork, which generates the full drift parameters, FiLM only generates feature-wise affine modulations, providing a lower-dimensional personalization mechanism.

(AS.4) No observation-process modeling. In this variant, we remove the latent-state-dependent event-intensity model, including the explicit end-of-sequence (EOS) component. This ablation isolates the contribution of explicitly modeling the observation process.

F Additional results

F.1 Statistical variability and robustness

We complement the mean results reported in the main text with measures of variability, paired significance tests, and median-based aggregation. These analyses assess the stability of the reported comparisons across Monte Carlo runs and generated datasets.

Table 7: Evaluation metrics on simulated and real-world data — standard deviations. Standard deviations computed across $N_{\text{MC}} = 30$ Monte Carlo runs (for simulated data only) and $N_{\text{gen}} = 50$ generated datasets from a single trained model. Corresponding means are reported in Table 2.

		Fidelity						Privacy			Utility
		Corr. RMSE		Discrim.	KL λ	MMD		Dep.	MIR	NNAA	Pred.
		Long.	Static			Long.	Static				
	Reference	0.047	0.037	0.069	0.031	0.321	0.004	0.000	0.027	0.044	1.785
Simulated	MultiNODEs	0.070	0.038	0.006	0.413	1717.726	0.004	0.044	0.026	0.035	5.251
	RTSGAN	0.071	0.057	0.120	12.336	66.257	0.024	0.033	0.029	0.040	2.784
	DGBFGP + HIVAE	0.068	0.039	0.003	0.413	1150.507	0.004	0.023	0.027	0.033	5.430
	HyperNSDE	0.070	0.043	0.134	0.213	3.382	0.003	0.011	0.029	0.034	2.114
VELOUR	MultiNODEs	0.007	0.005	0.010	0.000	0.192	0.000	0.004	0.030	0.044	0.162
	RTSGAN	0.008	0.006	0.000	1.003	736.300	0.000	0.020	0.028	0.031	1427.366
	HyperNSDE	0.007	0.005	0.000	1.293	0.005	0.000	0.007	0.029	0.033	1.120
PPMI	MultiNODEs	0.006	0.003	0.000	0.000	0.004	0.000	0.001	0.013	0.011	0.957
	RTSGAN	0.004	0.009	0.009	0.132	0.002	0.001	0.003	0.012	0.019	3.747
	HyperNSDE	0.006	0.004	0.004	0.210	0.001	0.000	0.006	0.019	0.009	3.490

Table 8: Ablation studies on simulated data — standard deviations. Standard deviations computed across $N_{\text{MC}} = 30$ Monte Carlo runs and $N_{\text{gen}} = 50$ generated datasets from a single trained model. Corresponding means are reported in Table 4.

	Fidelity						Privacy			Utility
	Corr. RMSE		Discrim.	KL λ	MMD		Dep.	MIR	NNAA	Pred.
	Long.	Static			Long.	Static				
Reference	0.047	0.037	0.069	0.031	0.321	0.004	0.000	0.027	0.044	1.785
AS.1 only ODE	0.077	0.036	0.032	0.347	241.458	0.003	0.053	0.035	0.037	1.852
AS.2 training	0.156	0.039	0.053	0.361	3.329	0.004	0.056	0.026	0.036	5.339
AS.3 MNSDE	0.071	0.024	0.094	0.091	3.564	0.004	0.014	0.034	0.037	1.641
AS.3 SNSDE	0.069	0.024	0.189	0.137	3.647	0.003	0.009	0.035	0.039	2.197
AS.3 FNSDE	0.103	0.058	0.112	0.134	24.650	0.008	0.007	0.026	0.034	2.953
AS.4 w/o lbda	0.044	0.037	0.024	0.413	864.737	0.002	0.011	0.028	0.033	5.513
HyperNSDE	0.070	0.043	0.134	0.213	3.382	0.003	0.011	0.029	0.034	2.114

Table 9: Paired Wilcoxon signed-rank tests, HyperNSDE vs each baseline. $N_{\mathrm{MC}} = 30$ seeds (the 50 generations averaged within each seed before pairing), two-sided, Holm-corrected across all 30 tests. Cells show the median difference (HyperNSDE vs. baseline); the sign is oriented per metric so that ✓ marks where HyperNSDE is significantly better and × where significantly worse. $^{*}\ p < .05$, $^{**}\ p < .01$, $^{***}\ p < .001$; n.s. = not significant after correction.

Metric	vs MultiNODEs	vs RTSGAN	vs DGBFGP+HIVAE
MMD longi ↓	-2378.8^{***} ✓	-1.242^{***} ✓	-103.6^{***} ✓
MMD static ↓	-0.0026^{**} ✓	-0.0048^{***} ✓	-0.0026^{***} ✓
Corr. RMSE longi ↓	$+0.020$ n.s.	$+0.007$ n.s.	-0.064 n.s.
Corr. RMSE static ↓	-0.004 n.s.	-0.027^{*} ✓	-0.014 n.s.
Discrim. Score ↓	-0.035^{***} ✓	-0.002 n.s.	-0.037^{***} ✓
KL λ ↓	-66.86^{***} ✓	0.138^{*} ×	-66.86^{***} ✓
Dep. Score ↓	-0.107^{***} ✓	$+0.001$ n.s.	-0.041^{***} ✓
MIR ↓	$+0.0007$ n.s.	$+0.001$ n.s.	$+0.002$ n.s.
NNAA → 0	$+0.009$ n.s.	-0.004 n.s.	$+0.013^{*}$ ×
Pred. Score ↓	-5.229^{***} ✓	-0.098 n.s.	-5.656^{***} ✓

Table 10: Evaluation metrics on simulated and real-world data — medians. ↓ lower is better; → 0 closer to zero is better (NNAA). **Bold** is the best result excluding the reference, underlined is second best.

		Fidelity						Privacy			Utility
		Corr. RMSE ↓		Discrim. ↓	KL λ ↓	MMD ↓		Dep. ↓	MIR ↓	NNAA → 0	Pred. ↓
		Long.	Static			Long.	Static				
Simulated	Reference	0.062	0.062	0.493	0.133	0.000	0.006	0.001	0.021	0.047	3.449
	MultiNODEs	**0.053**	0.035	1.000	67.522	35.926	0.006	0.118	0.028	−0.014	9.527
	RTSGAN	0.079	0.060	**0.978**	**0.460**	1.175	0.008	**0.011**	0.030	−0.027	4.957
	DGBFGP + HIVAE	0.162	0.042	1.000	67.522	12.198	0.005	0.058	**0.028**	**−0.009**	9.730
	HyperNSDE	0.098	**0.031**	0.983	0.668	**0.145**	**0.003**	0.013	0.032	−0.028	**4.777**
VELOUR	Reference	0.091	0.132	0.300	0.845	0.001	0.019	0.063	0.073	−0.049	21.118
	MultiNODEs	**0.139**	0.154	1.000	139.716	0.140	**0.019**	0.628	0.052	**−0.003**	**5.846**
	RTSGAN	0.842	**0.149**	1.000	**2.533**	24.844	0.019	0.116	**0.048**	0.083	5821.113
	HyperNSDE	0.239	0.165	1.000	3.227	**0.029**	0.019	**0.106**	0.061	0.013	53.447
PPMI	Reference	0.044	0.092	0.587	0.064	0.000	0.006	0.052	0.065	−0.032	39.137
	MultiNODEs	**0.107**	0.133	1.000	221.398	0.037	0.013	0.391	**0.013**	0.007	**19.841**
	RTSGAN	0.623	**0.113**	**0.983**	**1.013**	**0.009**	**0.012**	**0.057**	0.057	−0.006	52.895
	HyperNSDE	0.260	0.136	0.997	1.796	0.019	0.013	0.083	0.028	**0.002**	58.470

F.2 Additional analyses of evaluation metrics

We complement the controlled analyses reported in the main text (Table 3) with additional diagnostics of the discriminative score.

Discriminative score decomposition To identify which components of the synthetic records drive the discriminative score, we repeat the real-versus-synthetic classification on simulated data using static covariates, observation patterns, and trajectory values separately. Static covariates are difficult to distinguish from real data across methods, whereas observation patterns strongly distinguish the baselines. For HyperNSDE, the remaining discriminability mainly arises from trajectory values.

Table 11: C2ST test AUROC by classifier input (simulated data). Values closer to 0.5 are better (harder to distinguish from real). Best non-reference value per column in **bold**.

Model	Static only	Pattern only	Values only	All (Table 2)
Reference	0.493	0.511	0.480	0.499
MultiNODEs	0.486	1.000	0.998	0.998
RTSGAN	**0.522**	1.000	0.947	0.932
DGBFGP + HIVAE	0.487	1.000	0.999	0.999
HyperNSDE	0.462	**0.760**	**0.942**	**0.926**

F.3 Sensitivity analysis

We further assess robustness under controlled variations described in Appendix E.3: limited training data, sparse observations, state-dependent sampling, and prior versus posterior latent sampling.

Table 12: Sensitivity analysis: *(i) Limited training data*. Performance comparison on simulated data when restricting the number of available training samples. ↓ lower is better; → 0 closer to zero is better (NNAA). **Bold** is the best result excluding the reference, underlined is second best.

	Fidelity						Privacy			Utility
	Corr. RMSE ↓		Discrim. ↓	KL λ ↓	MMD ↓		Dep. ↓	MIR ↓	NNAA → 0	Pred. ↓
	Long.	Static			Long.	Static				
Reference	0.061	0.060	0.499	0.139	0.021	0.007	0.001	0.033	0.021	3.759
MultiNODEs	0.190	0.085	0.998	67.616	897.653	0.010	0.128	0.044	**−0.000**	11.394
RTSGAN	0.115	0.070	0.973	1.754	282.023	0.013	0.016	**0.044**	0.004	5.680
DGBFGP + HIVAE	0.138	0.053	0.999	67.616	238.883	0.007	0.058	0.045	−0.000	11.663
HyperNSDE (post.)	**0.086**	**0.035**	**0.765**	**0.749**	**2.427**	**0.003**	**0.009**	0.048	−0.014	**4.764**
HyperNSDE (prior)	0.105	0.086	0.790	0.762	2.570	0.019	0.010	0.045	−0.001	5.019

Table 13: Sensitivity analysis: *(ii) Sparse observation regime*. Performance comparison on simulated data under reduced observation frequencies. ↓ lower is better; → 0 closer to zero is better (NNAA). **Bold** is the best result excluding the reference, underlined is second best.

	Fidelity						Privacy			Utility
	Corr. RMSE ↓		Discrim. ↓	KL λ ↓	MMD ↓		Dep. ↓	MIR ↓	NNAA → 0	Pred. ↓
	Long.	Static			Long.	Static				
Reference	0.065	0.060	0.504	0.209	−0.001	0.007	0.006	0.032	0.041	12.147
MultiNODEs	0.141	0.055	0.997	104.096	876.022	**0.006**	0.182	0.034	−0.008	32.967
RTSGAN	**0.077**	0.064	**0.851**	1.890	0.051	0.009	0.053	**0.033**	−0.045	**13.975**
DGBFGP + HIVAE	0.307	0.052	0.998	104.096	1187.980	0.006	0.115	0.033	**−0.007**	32.964
HyperNSDE (post.)	0.093	**0.046**	0.926	1.607	**0.013**	0.009	**0.033**	0.036	−0.054	15.265
HyperNSDE (prior)	0.114	0.072	0.929	**1.584**	0.016	0.021	0.037	0.034	−0.044	15.134

Table 14: Sensitivity analysis: ***(iii) State-dependent observation process*****.** Evaluation on simulated data where observation times are influenced by the underlying system state. ↓ lower is better; → 0 closer to zero is better (NNAA). **Bold** is the best result excluding the reference, underlined is second best.

	Fidelity						Privacy			Utility
	Corr. RMSE ↓		Discrim. ↓	KL λ ↓	MMD ↓		Dep. ↓	MIR ↓	NNAA → 0	Pred. ↓
	Long.	Static			Long.	Static				
Reference	0.067	0.060	0.494	0.187	0.079	0.007	0.001	0.037	0.030	3.714
MultiNODEs	**0.094**	0.044	0.993	67.708	1122.343	0.006	0.147	0.037	−0.008	9.828
RTSGAN	0.099	0.058	0.899	0.362	3.903	0.009	0.020	0.038	−0.014	**5.148**
DGBFGP + HIVAE	0.158	0.052	0.998	67.708	204.745	0.006	0.077	**0.036**	**−0.005**	10.001
AS.4 w/o lbda	0.096	0.049	0.946	66.857	287.777	0.004	0.035	0.037	−0.016	9.887
HyperNSDE (post.)	0.157	**0.036**	**0.896**	**0.722**	**0.275**	**0.004**	**0.017**	0.048	−0.028	5.726

Table 15: Sensitivity analysis: ***(iv) Prior versus posterior sampling*****.** Comparison of HyperNSDE under posterior and prior sampling of the static latent representation on simulated and real-world datasets. ↓ lower is better; → 0 closer to zero is better (NNAA). **Bold** is the best result excluding the reference.

		Fidelity						Privacy			Utility
		Corr. RMSE ↓		Discrim. ↓	KL λ ↓	MMD ↓		Dep. ↓	MIR ↓	NNAA → 0	Pred. ↓
		Long.	Static			Long.	Static				
Simulated	Reference	0.061	0.060	0.499	0.031	0.021	0.007	0.001	0.033	0.021	3.759
	HyperNSDE (post.)	**0.118**	**0.044**	**0.926**	**0.687**	1.150	**0.004**	**0.014**	0.039	−0.021	**5.226**
	HyperNSDE (prior)	0.123	0.090	0.932	0.705	**1.119**	0.018	0.015	**0.035**	**−0.007**	5.612
VELOUR	Reference	0.091	0.132	0.472	0.845	0.001	0.019	0.063	0.073	−0.049	21.958
	HyperNSDE (post.)	0.239	**0.164**	**0.998**	**3.513**	0.029	0.019	**0.105**	**0.063**	0.010	**53.514**
	HyperNSDE (prior)	0.239	0.165	1.000	3.708	**0.027**	0.019	0.107	0.065	**0.007**	53.745
PPMI	Reference	0.044	0.092	0.587	0.064	0.001	0.006	0.052	0.065	−0.032	39.137
	HyperNSDE (post.)	0.260	0.135	**0.996**	1.840	**0.019**	0.013	0.084	0.032	**0.002**	**57.697**
	HyperNSDE (prior)	**0.259**	0.135	0.997	**1.834**	0.020	0.013	**0.082**	**0.017**	0.011	65.961

Table 16: Sensitivity analysis: ***(v) State dependence of the observation intensity*****.** Comparison of HyperNSDE with a latent-state-dependent intensity and a state-independent intensity. ↓ lower is better; → 0 closer to zero is better (NNAA). **Bold** is the best result excluding the reference.

		Fidelity						Privacy			Utility
		Corr. RMSE ↓		Discrim. ↓	KL λ ↓	MMD ↓		Dep. ↓	MIR ↓	NNAA → 0	Pred. ↓
		Long.	Static			Long.	Static				
Simulated	Reference	0.067	0.060	0.494	0.187	0.079	0.007	0.001	0.037	0.030	3.714
	HyperNSDE (state-dep. λ)	**0.157**	**0.036**	**0.896**	0.722	**0.275**	**0.004**	**0.017**	0.048	−0.028	5.726
	HyperNSDE (state-indep. λ)	0.249	0.090	0.975	**0.427**	1.600	0.009	0.024	**0.036**	**−0.003**	**2.553**
VELOUR	Reference	0.091	0.132	0.472	0.845	0.001	0.019	0.063	0.073	−0.049	21.958
	HyperNSDE (state-dep. λ)	0.239	**0.164**	**0.998**	**3.513**	0.029	0.019	**0.105**	0.063	**0.010**	**53.514**
	HyperNSDE (state-indep. λ)	**0.221**	0.165	1.000	7.190	0.029	0.019	0.138	**0.057**	0.014	54.773
PPMI	Reference	0.044	0.092	0.587	0.064	0.001	0.006	0.052	0.065	−0.032	39.137
	HyperNSDE (state-dep. λ)	**0.260**	0.135	0.996	1.840	**0.019**	0.013	0.084	0.032	**0.002**	**57.697**
	HyperNSDE (state-indep. λ)	0.264	**0.134**	**0.963**	**1.339**	0.020	0.013	**0.079**	**0.021**	0.008	66.257

F.4 Subgroup and cohort analyses

We complement the aggregate evaluation with analyses of subgroup-specific fidelity on simulated data and source-cohort representativeness on VELOUR.

HyperNSDE learns the observational distribution represented in its training cohort. To assess how closely the generated population reproduces the source cohort, we compare real and synthetic static covariates on VELOUR using standardized mean differences (SMDs). Across the 26 static covariates, the median SMD is 0.092, indicating close agreement with the observed VELOUR cohort. This assesses fidelity to the source population and should not be interpreted as a correction of selection bias or representativeness with respect to a broader target population.

Table 17: Results on the subset $W_1 = 0$ of simulated data. Evaluation in the reference setting, restricted exclusively to the subpopulation of samples where $W_1 = 0$. $\downarrow$ lower is better; $\rightarrow 0$ closer to zero is better (NNAA). **Bold** is the best result excluding the reference, underlined is second best.

	Fidelity						Privacy			Utility
	Corr. RMSE ↓		Discrim. ↓	KL λ ↓	MMD ↓		Dep. ↓	MIR ↓	NNAA $\rightarrow 0$	Pred. ↓
	Long.	Static			Long.	Static				
Reference	0.091	0.000	0.482	0.288	−0.001	0.010	0.002	0.057	0.016	8.912
MultiNODEs	0.180	0.000	0.992	67.746	78.209	0.011	0.178	0.050	**−0.009**	**8.829**
RTSGAN	**0.150**	0.000	0.876	8.007	0.456	0.022	0.032	0.052	−0.010	15.541
DGBFGP + HIVAE	0.322	0.000	0.997	67.746	60.341	0.009	0.077	0.050	−0.010	8.862
AS.3 MNSDE	0.249	0.000	0.914	**0.605**	0.157	**0.006**	0.026	0.054	−0.024	53.848
AS.3 SNSDE	0.217	**0.000**	**0.796**	0.746	**0.087**	0.006	**0.014**	0.054	−0.029	166.064
AS.3 FNSDE	0.277	0.000	0.880	0.722	1.645	0.012	0.024	0.050	−0.010	129.664
HyperNSDE (post.)	0.244	0.000	0.890	0.918	0.101	0.011	0.017	0.052	−0.020	52.438
HyperNSDE (prior)	0.254	0.000	0.893	0.913	0.121	0.020	0.018	**0.049**	−0.014	54.613

Table 18: Results on the subset $W_1 = 1$ of simulated data. Evaluation in the reference setting, restricted exclusively to the subpopulation of samples where $W_1 = 1$. $\downarrow$ lower is better; $\rightarrow 0$ closer to zero is better (NNAA). **Bold** is the best result excluding the reference, underlined is second best.

	Fidelity						Privacy			Utility
	Corr. RMSE ↓		Discrim. ↓	KL λ ↓	MMD ↓		Dep. ↓	MIR ↓	NNAA $\rightarrow 0$	Pred. ↓
	Long.	Static			Long.	Static				
Reference	0.077	0.000	0.519	0.282	0.418	0.010	0.001	0.052	0.028	2.904
MultiNODEs	**0.105**	0.000	0.983	67.611	827.129	0.013	0.085	0.049	−0.008	11.267
RTSGAN	0.182	0.000	0.853	3.609	34.010	0.023	0.025	0.051	−0.013	4.479
DGBFGP + HIVAE	0.190	0.000	0.995	67.611	257.408	0.009	0.065	0.049	**−0.000**	11.738
AS.3 MNSDE	0.157	0.000	0.889	**0.567**	7.498	0.004	0.026	0.064	−0.034	**3.916**
AS.3 SNSDE	0.130	0.000	**0.789**	0.718	5.656	**0.004**	0.021	0.066	−0.043	4.280
AS.3 FNSDE	0.181	0.000	0.859	0.720	5.446	0.020	**0.017**	0.050	0.004	5.403
HyperNSDE (post.)	0.136	0.000	0.833	0.791	5.144	0.004	0.019	0.055	−0.026	4.414
HyperNSDE (prior)	0.142	0.000	0.840	0.809	**4.835**	0.025	0.020	**0.049**	−0.006	4.738

Table 19: Real-vs-synthetic static-covariate fidelity on VELOUR for HyperNSDE. For each covariate, the standardized mean difference (SMD; < 0.1 indicates negligible differences) and the real-vs-synthetic test p-value are computed against the real cohort and summarised across the 50 generated cohorts (median [Q1, Q3]). High p-values indicate the synthetic covariate is statistically indistinguishable from the real one.

Static covariate	SMD (median)	p-value (median [Q1, Q3])
SEX	0.152	0.334 [0.213, 0.582]
RACE	0.186	0.424 [0.184, 0.595]
REGION	0.221	0.635 [0.462, 0.889]
Age category	0.010	0.594 [0.343, 0.812]
ECOG Performance Status	0.136	0.625 [0.352, 0.790]
Height	0.013	0.560 [0.422, 0.855]
Adverse-event and concomitant-medication indicators (20 covariates)		
median across the 20	0.091	0.677 [0.41, 0.90]

Across all 26 covariates: median SMD 0.092, with 65% below 0.1; p-value quartiles [Q1, Q3] = [0.40, 0.87].

F.5 Qualitative results

The following figures complement the quantitative analyses with trajectory-level comparisons across methods, generation modes, and ablation variants.

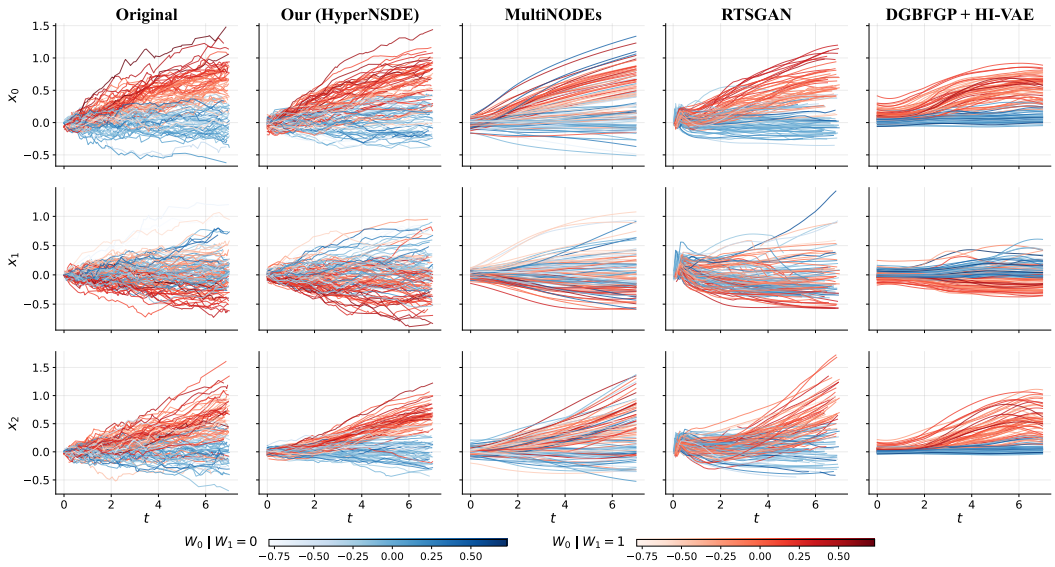

Figure 4: Qualitative comparison of HyperNSDE and baselines on simulated data. Rows correspond to the three longitudinal variables x_0, x_1, x_2; columns compare the original data against HyperNSDE and baselines. y-axis limits are shared across columns per row. The color encodes W_0 separately for subjects with $W_1 = 0$ (blue gradient) and $W_1 = 1$ (red gradient).

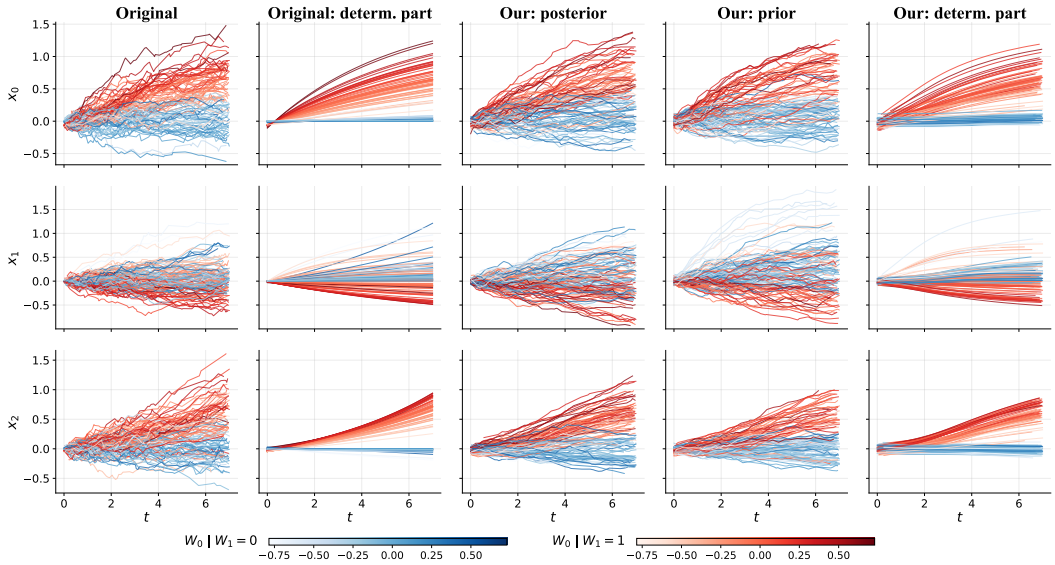

Figure 5: Qualitative comparison of HyperNSDE generation modes on simulated data. Rows correspond to the three longitudinal variables x_0, x_1, x_2; columns compare the original data and its deterministic component against HyperNSDE under posterior sampling, prior sampling, and estimated deterministic component. The color encodes W_0 separately for subjects with $W_1 = 0$ (blue gradient) and $W_1 = 1$ (red gradient).

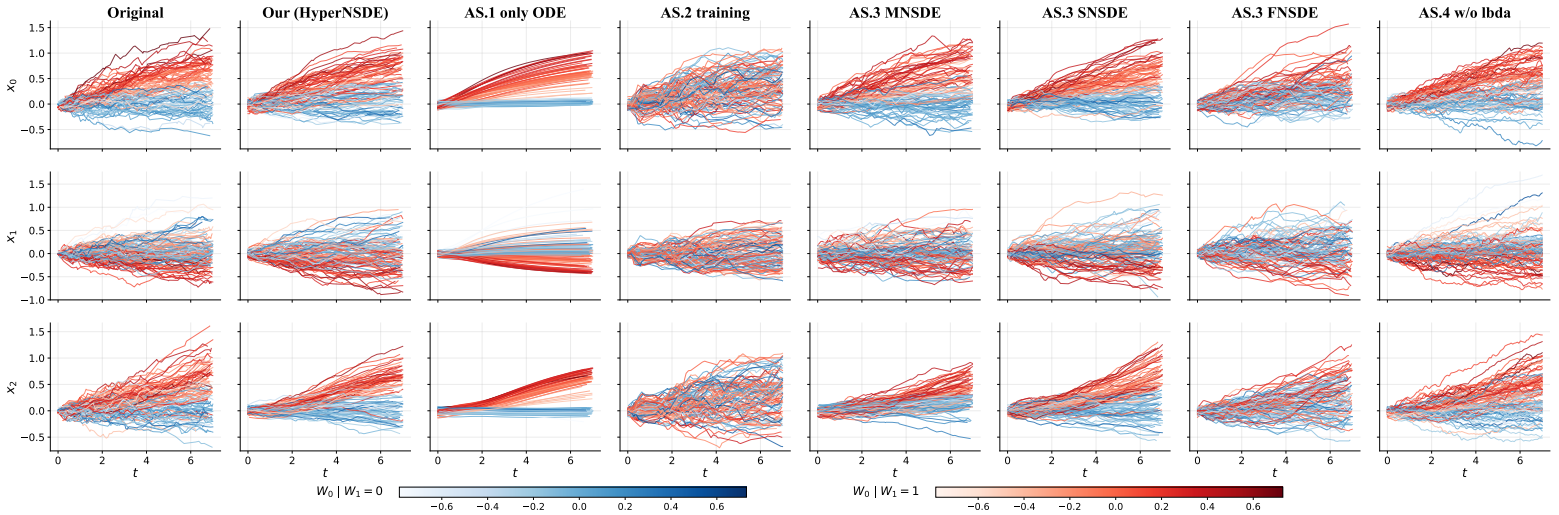

Figure 6: Qualitative comparison of ablation study variants on simulated data. Rows correspond to the three longitudinal variables x_0, x_1, x_2; columns compare the original data and HyperNSDE against the ablation variants described in Appendix E.4. The color encodes W_0 separately for subjects with $W_1 = 0$ (blue gradient) and $W_1 = 1$ (red gradient).

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction clearly state the three main contributions (unified generative model, hypernetwork-based personalization, tractable stochastic latent dynamics) and the evaluation scope across three dimensions: fidelity, utility, and privacy (simulated and real clinical datasets).

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Limitations are discussed in the Conclusion (Section 7), covering computational cost, scope of the empirical evaluation on real-world datasets, and the limitations of current evaluation metrics for irregular multimodal clinical data.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [N/A]

Justification: The paper does not include formal theoretical results. The deterministic–stochastic decomposition is discussed as a modeling choice with theoretical motivation provided in Appendix B.1, but no theorems or proofs are claimed.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: The model architecture, training procedure, evaluation protocol ($N_{\text{MC}} = 30$, $N_{\text{gen}} = 50$, 80/20 splits), hyperparameter tuning strategy, model of the simulated datasets, and dataset preprocessing pipelines are described in the main paper and Appendices B,D,E. Code is publicly available (Section 5).

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: Code is released at `https://github.com/perrinechassat/HyperNSDE`, with scripts to reproduce simulated experiments. Real datasets require individual data access agreements: VELOUR is available via Project DataSphere [48] and PPMI via `https://www.ppmi-info.org/`; preprocessing details are provided in Appendix D.3.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Data splits, the number of Monte Carlo runs, generation repetitions, optimizer, early stopping, and hyperparameter selection are described in Section 5 and Appendix B–E.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

Justification: Results are averaged over $N_{\text{MC}} = 30$ Monte Carlo runs (independently generated datasets with different random seeds) and $N_{\text{gen}} = 50$ independent synthetic datasets per trained model for simulated data, and over $N_{\text{gen}} = 50$ generations for real-world datasets. Standard deviations across these runs are reported in Tables 7 and 8 in Appendix F. No normality assumption is made; standard deviations are reported as a measure of variability across runs.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The Appendix E.1 reports the main compute resources used for the experiments, including single-GPU jobs on NVIDIA V100 32 GB GPUs on the Jean Zay cluster, with 10 CPU cores per task and a typical 20-hour wall-clock allocation. We also provide indicative training times on the reference simulated setting and note how runtime scales with the number of patients and observations. While we do not provide a full accounting of all preliminary or failed runs, the paper gives sufficient information to reproduce the reported experiments in practice.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

Justification: To the best of our knowledge, the research presented in this paper conforms to the NeurIPS Code of Ethics. The work addresses privacy-preserving synthetic data generation, aligned with responsible data sharing in healthcare. All datasets are accessed through official channels under their respective terms of use. No patient-level data is released.

Guidelines:

- The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: The paper discusses both positive and negative societal impacts in the conclusion (Section 7). On the positive side, synthetic clinical data may support privacy-aware data sharing, benchmarking, and method development in healthcare. On the negative side, the paper explicitly notes the current limits of realism and the risks of premature use for high-stakes applications such as cohort augmentation or clinical trial simulation. It also discusses privacy-related evaluation and the need for stronger safeguards such as differential privacy in future work.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: The contribution is methodological, and the released asset is limited to research code for reproducibility.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All datasets are properly cited with their respective access conditions. Baseline implementations are cited and used under their respective open-source licenses. The `pySigLib` package [54] is cited and used under its open-source license.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The paper introduces new research code implementing HyperNSDE. The manuscript and appendix describe the main modeling components, training setup, and implementation details needed to understand and reproduce the method.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: The paper does not involve crowdsourcing experiments or research with human subjects conducted by the authors.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: The paper does not involve crowdsourcing experiments or human-subject research conducted by the authors, and therefore no IRB approval or equivalent review was required for the work presented here.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: LLMs are not used as an important, original, or non-standard component of the proposed method, experiments, or scientific claims.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.