Interpretable Regime Trajectories via Generative Graph State-Space Models

Anonymous Author(s)
Affiliation
Address
email

Abstract

Forecasting the behavior of real-world spatiotemporal systems often requires not only accurate predictions but also interpretable regime trajectories, i.e. discrete states that describe how dynamics change over time. However, existing approaches often entangle space and time, obscuring regime structure or trading interpretability for scale. We introduce ReGraSS, a unified framework that learns discrete, interpretable latent regimes from spatiotemporal data, represented as dynamic graphs, combining variational training with strictly time-ordered state-space inference. Predictions are produced by a mixture-of-experts modulated by the inferred regime probabilities, enforcing regime-specific specialization and supporting interpretability. Trained with self-supervised one-step prediction, the model learns in label-scarce settings and provides calibrated uncertainty by estimating a distribution over discrete regimes. ReGraSS matches or surpasses state-of-the-art spatiotemporal baselines in one-step forecasting. It shows the smallest error spike at regime changes and the fastest recovery thereafter, indicating regime-level interpretability and reliable trajectory tracking without compromising accuracy. We believe our interpretable, uncertainty-aware framework for regime-aware forecasting on dynamic graphs has direct application in healthcare, finance, and epidemiology.

1 Introduction

Recent advances in modern sensing and data acquisition reveal how real-world systems evolve across space and time [1, 2, 3]. In these settings, accurate forecasting of the system's trajectory is necessary but often not sufficient: interpretable regime trajectories, i.e. discrete states governing the system's evolution, may also be required. They reveal to be critical in high-stakes domains such as healthcare (e.g., disease progression staging [4, 5, 6]), finance (e.g., market regimes [7, 8]) or epidemiology (e.g., transmission phases [9, 10]), where decisions rely on understanding when and why a regime shifts, not only on accurate forecasting of future events. Yet, current learning systems lack regime-aware, time-ordered explanations alongside forecasts, leaving a critical gap for models that jointly learn spatial structure, temporal evolution, and discrete interpretable regimes.

Graph Neural Networks (GNNs) [11, 12] are powerful tools to model spatial relationships through relational inductive biases, providing a unified framework for domains with hierarchical structure and rich spatial interactions. Extending them to spatiotemporal settings is challenging because both the graph topology and the node signals may evolve over time. Dynamic variants such as EvolveGCN [13] and ROLAND [14] address part of this challenge, yet they still interleave spatial aggregation with temporal updates via recursive message passing. As topology evolves, this coupling obscures what changed from when it changed, hindering interpretable identification of discrete regimes.

On the other hand, State Space Models (SSMs) provide a well-established framework for modeling temporal dynamics via latent state representations and structured transitions. Recent work on learnable SSMs (e.g., S4 [15], Mamba [16]), achieve strong performance on sequence modeling tasks, overcoming several limitations of classical SSMs such as adaptation to regime shifts and multi-scale dynamics. However, they typically require large amounts of data and tend to sacrifice latent-state interpretability, limiting their applicability where understanding the underlying dynamics is essential.

Recent efforts integrate SSMs with graphs by decoupling spatial and temporal reasoning, applying state-space updates at the node level and mixing via GNN layers (e.g., GrassNet [17], Graph Mamba [18]). Dynamic variants further interleave Mamba-based sequence modules and spatiotemporal graph blocks to handle evolving topologies (STG-Mamba [19], DG-Mamba [20]). However, these models typically use SSMs as feature extractors rather than for interpretable regime tracking through state representation.

In this work, we propose **ReGraSS**, an unified framework that models discrete, interpretable regimes as latent states on dynamic graphs. A dynamic GNN encodes evolving structure and features, while temporal regime dynamics are decoupled and captured via a variational distribution over latent states. Predictions are state-conditioned via a mixture-of-experts weighted by regime probabilities, so the inferred state actively governs emissions, yielding trajectory-level explanations and calibrated uncertainty without degrading forecast accuracy. During training, a categorical VAE [21] with a learnable transition prior supports uncertainty-aware regime discovery; at inference, we switch to a strictly time-ordered state-space rollout that conditions only on past and present, enabling transparent trajectory analysis without future leakage. To function in label-scarce settings common in high-stakes applications, we adopt an autoregressive one-step forecasting objective that forces the model to internalize graph-coupled dynamics by predicting next-step node features and produces regime trajectories consistent with predictive performance. On controlled synthetic tests with induced non-stationarity, the framework captures regime transitions, supports uncertainty-aware trajectories, and matches or surpasses strong spatio-temporal and graph-SSM baselines, demonstrating that interpretable regime tracking can be achieved without a trade-off in accuracy.

2 Proposed Approach

2.1 Problem Statement

We consider a discrete-time sequence of graph snapshots $\mathcal{G} = (G_t)_{t=0}^{T}$, where each $G_t = (V_t, E_t, X_t)$ consists of a vertex set V_t, an edge set E_t, and node features $X_t \in \mathbb{R}^{N_t \times D}$ with $N_t = |V_t|$ and feature dimension D. We assume that the topology and vertex set may vary over time (vertices may appear or disappear).

Our main hypothesis is that a finite set of discrete regimes $\mathcal{R} = \{r_1, \ldots, r_K\}$ modulates the dynamics, with $R_t \in \mathcal{R}$ being the active regime at time t. Discrete regimes align with how practitioners typically characterize system progression (physiological stages, market states) even when these categories coarsen underlying continuous dynamics.

Focusing on node-feature dynamics, we assume that the next time point features X_{t+1} are generated from some probability distribution $\mathbb{P}\big(X_{t+1} \;\big|\; X_{0:t}, V_{0:t}, E_{0:t}, R_{0:t}\big)$ conditioned on past history. Thus our objective is to learn a model $\hat{f}$ that (i) approximates the probability distribution $\mathbb{P}\big(X_{t+1} \;\big|\; X_{0:t}, V_{0:t}, E_{0:t}, R_{0:t}\big)$ in an autoregressive manner, (ii) while inferring the active regime without being provided regime annotations. Formally, the model can be defined as $\hat{f}(X_{0:t}, V_{0:t}, E_{0:t}) = \big(\widehat{X}_{t+1},\ \widehat{R}_t\big)$. Extensions to topology prediction are straightforward.

2.2 Architecture

We introduce **ReGraSS** (Regime-aware Graph State Space model), an autoregressive generative framework for modeling spatio-temporal dynamics on graphs through an intepretable discrete latent state space, where the extracted states act as proxies for the underlying regimes governing the system evolution. ReGraSS follows a structured encoder-decoder design. The encoder approximates the posterior over discrete latent states and the decoder generates node features at future time steps, conditioned on both the latent state and observed inputs. The model operates differently during training and inference; we describe the training behavior here and defer the dual representation and inference details to Section 2.4. The architecture is illustrated in figure 2, and we describe its main building blocks below.

Figure 1: Visualization of the model's architecture and dual formulation.

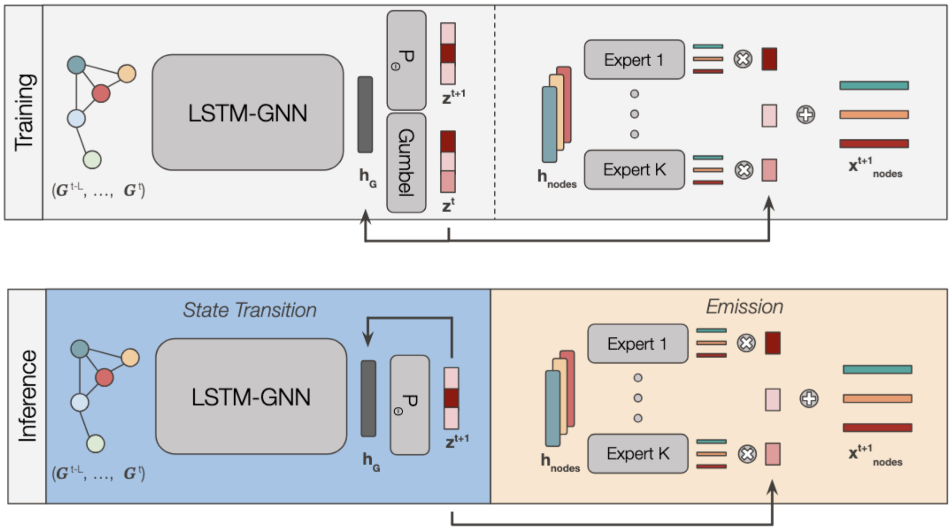

Figure 2: Visualization of the model architecture and dual-view formulation. **Top panel** (training phase): At each time step, graph snapshots are processed sequentially through the LSTM-GNN module to produce graph-level temporal embeddings. These embeddings are used to infer the current latent state z_t via the Gumbel-Softmax module, and to predict the next state z_{t+1} via the learnable prior module P_θ. Given the inferred states, the temporal node-level embeddings h_{nodes} are passed through a MoE module, where expert outputs are modulated by z_t, to generate the predicted node features at time $t+1$. **Bottom panel** (inference structure): The framework can be decomposed into two components: state transition and emission. This mirrors the classical state-space model (SSM) formulation, while extending it to a non-linear and graph-based setting.

Encoder. The encoder's first stage is a temporal GNN that aggregates information from past snapshots up to the current step t. We instantiate it with ROLAND ([14]), which maintains hierarchical node representations via GRU updates ([22]) and naturally supports evolving graph topology. We map the pooled temporal graph embedding $h_G^t \in \mathbb{R}^H$ to K unnormalized logits with a linear layer

$$\ell_t = W h_G^t + b, \qquad W \in \mathbb{R}^{K \times H},\ b \in \mathbb{R}^K,\ K = |\mathcal{R}|.$$

To obtain a posterior over the K regimes, we use the Gumbel–Softmax reparameterization [21]:

$$q_\phi(z_t \mid h_G^t) \ = \ \operatorname{softmax}\left(\frac{\ell_t + g_t}{\tau}\right), \qquad g_t \sim \operatorname{Gumbel}(0,1)^K,\ \ \tau > 0.$$

Sampling $z_t \sim q_\phi(\cdot \mid h_G^t)$ yields a differentiable, discrete latent vector that encodes the current regime R_t (approaching one-hot as $\tau \to 0$). Our probabilistic approach quantifies uncertainty in the current regime and offers a distributional view that bridges continuous dynamics and discrete regimes. Yet, the framework remains compatible with continuous latents if the underlying system is better described by continuous variables.

Learnable prior for causal transitions. We define a learnable prior $p_\theta(z_t \mid h_G^{t-1}, z_{t-1})$, parametrized by a 2-layer MLP, that receives the temporal embedding h_G^{t-1} and the previous latent state to predict z_t. During training, we align this prior with the variational posterior q_ϕ (see Section 2.3), yielding SSM-like transitions and enabling the dual representation in Section 2.4. Compared with a fixed Markov prior, this data-driven conditional prior better captures non-stationary regime dynamics on evolving graphs.

Decoder (mixture of experts). The decoder predicts the next-step node features X_{t+1} conditioned on the current features X_t and the latent state z_t. We implement it as a mixture-of-experts (MoE [23]): K experts $\{f_k\}_{k=1}^K$, each parametrized by an independent MLP, produce candidate outputs that are combined using the posterior mixing coefficients $\pi_t = q_\phi(z_t \mid h_G^t)$. Equivalently, $\widehat{X}_{t+1} = \sum_{k=1}^M \pi_{t,k}\, f_k(X_t)$. The choice of the number of experts is domain specific but is typically selected to match the number or regimes in the data ($K = |\mathcal{R}|$), thus encouraging a one-to-one correspondence between the variational-induced states z_t and the regimes r_k. This implementation induces state-conditioned output generation, mirroring classical SSM two-stage behavior, i.e., state transition then output emission [15]. It also encourages specialization: as the Gumbel–Softmax vector approaches one-hot, each expert learns the dynamics associated with a specific regime $r_k \in \mathcal{R}$. While our implementation uses MLPs, experts can be replaced with other modules such as GNNs when domain requires it, e.g. when regime transitions influence the diffusion process in the graph, which is better captured by GNN experts rather than MLPs.

2.3 Training Procedure

Training uses a variational objective derived from the categorical VAE ELBO ([21],[24]) with a forecasting likelihood. Formally, with $q_z^t = q_\phi(z_t \,|\, h_G^t)$ and $p_z^t = p_\theta(z_t \,|\, h_G^{t-1}, z_{t-1})$, the graph-level loss over a sequence $t = 0, \ldots, T-1$ is

$$\mathcal{L} \;=\; \sum_{t=0}^{T-1} \Big[\underbrace{\ell_{\text{forecast}}\big(\hat{X}_{t+1},\, X_{t+1}\big)}_{\text{one-step prediction}} \;+\beta\Big(\underbrace{\mathrm{KL}\big(q_z^t\big) \,\|\, \mathrm{sg}[\,p_z^t]\big)}_{\text{encoder-prior alignment}} \;+\gamma\;\; \underbrace{\mathrm{CE}\big(\mathrm{sg}[\,q_z^t],\, p_z^t\big)\big)}_{\text{teacher-forced prior fitting}}\;\Big)\Big], \tag{1}$$

where ℓ_{forecast} is a regression loss between the decoder prediction $\hat{X}_{t+1}$ (the MoE output) and the observed features X_{t+1}, and $\mathrm{sg}[\cdot]$ denotes the stop-gradient operator, i.e. that the gradients are not backpropagated further in the computation tree. The KL term updates the encoder so that the posterior q_z^t agrees with the prior p_z^t, while the cross-entropy (CE) term trains the transition module to match the encoder's next-time posterior q_z^{t+1}. This asymmetric pairing stabilises learning: the encoder does not chase a moving prior, and the prior learns from the encoder without backpropagating through its inputs. Detailed regularization and parameters schedules are provided in the Appendix 5.1.

By regressing X_{t+1} from information available at time t, the objective forces the model to internalize the system's transition mechanisms, remaining effective when regime annotations are missing, unreliable, or available only at endpoints. This, in turn, enables reconstruction of regime trajectories and stratification of sequences $(G_t)_{t=0}^T$ by regime and temporal evolution.

2.4 Dual Representation

Our framework couples variational training with state-space inference to bridge two limitations encountered in the literature. By learning spatial representations with a dynamic GNN and evolving them through discrete regimes, it disentangles space–time updates that obscure regime structure in temporal GNNs. At the same time, the inference-time state-space rollout restores interpretable state representation often lost in deep SSMs, while preserving forecasting accuracy through state-conditioned emissions. This dual formulation is robust to scarce or unreliable labels and preserves strict temporal causality. During **training**, a variational next-step regression objective learns a posterior over regime trajectories, enabling trajectory-level explanations even without ground-truth regime annotations. At **inference**, we replace the posterior, that benefits from future information via backpropagation, with the learned transition prior and roll forward using only past observations, eliminating future leakage. The probabilistic treatment yields calibrated uncertainty for the current regime and a distributional view of transitions, bridging continuous dynamics and discrete regimes without sacrificing predictive performance.

3 Experiments and Results

3.1 Dataset

We generate 150 spatiotemporal graph sequences with $T = 10$ snapshots (TP1–TP10). At TP1, we sample $C \sim \mathrm{Unif}\{3,4,5\}$ Gaussian clusters in $\mathbb{R}^d$ ($d = 8$), with centers $\mu_c \sim \mathrm{Unif}([-10,10]^d)$, isotropic covariance $0.6^2 I_d$, and sizes $M_c \sim \mathrm{Unif}\{5,\ldots,100\}$; node features are the sampled coordinates. We build an undirected k-NN graph at TP1 and keep edges fixed thereafter (translation-invariant under our dynamics). Each sequence follows a discrete regime $r_t \in \{r_1, r_2, r_3\}$ that induces a constant drift $v(r_t) \in \{-2\,\mathbf{1}_d,\, +2\,\mathbf{1}_d,\, \mathbf{0}_d\}$, with i.i.d. Gaussian noise $\varepsilon_i^{(t)} \sim \mathcal{N}(0, 0.6^2 I_d)$ at each step. The regime is resampled once between TP4 and TP5 to test models ability to remain robust to a mid-sequence nonstationarity (TP4→TP5). Full details are in Appendix 5.2.

3.2 Baseline Methods

We compare against baselines spanning complementary assumptions: (i) no space/no time, (ii) time-only naïve dynamics, (iii) spatio-temporal without latent regimes, and (iv) spatio-temporal with deep state-space modules, to ensure gains are not attributable to unstructured aggregation or trivial autocorrelation. **MLP** (no space, no time) concatenates all node features into a single embedding, testing whether simple global aggregation suffices. **Persistence** (time only) is a parameter-free baseline that predicts $X_{t+1} = X_t$ to assess whether autocorrelation alone explains performance. **LSTM–GNN** (spatio-temporal) uses the GNN–LSTM encoder (adapted from ROLAND [14]) as a standalone predictor, isolating the contribution of discrete regimes and mixture-of-experts decoding in our method. **STG-MAMBA ([19])** (spatio-temporal) integrates dynamic graph filtering with a Mamba block for multi-scale temporal modeling, providing a benchmark against state-space/dynamic-graph hybrids. These baselines rule out unstructured aggregation, trivial autocorrelation, and generic spatio-temporal encodings. Additional implementations details appear in the Appendix 5.3.

3.3 One-Step Prediction under Changing Regimes

First, we evaluate each model in a one-step regression setup to test whether it captures system evolution and adapts to regime changes. Given the observed history up to time t, $(X_{0:t}, V_{0:t}, E_{0:t})$, each model predicts the next features X_{t+1}; we apply this procedure iteratively across time points on the dataset in Section 3.1. We pay particular attention to the induced shift between TP4 and TP5 as a stress test for non-stationarity. Performance is quantified by mean squared error (MSE) between $\widehat{X}_{t+1}$ and X_{t+1}, and we additionally report the mean absolute feature value at each time point to contextualize error magnitude (Table 1).

Across the synthetic dataset, ReGraSS attains the lowest mean error over the sequence (2.79 MSE vs. 3.05 for LSTM–GNN; Table 1) and leads both before the induced shift (pre-TP5 average 1.33) and after it (post-TP5 average 3.96). The Persistence baseline ($x_{t+1} = x_t$) performs worst throughout (6.77 MSE), confirming that temporal autocorrelation alone does not explain performance. A structure-free MLP is competitive early but breaks at the shift (TP5), indicating that unstructured aggregation cannot adapt to non-stationarity. The LSTM–GNN (ROLAND-based [14]) is a strong spatio-temporal encoder without latent regimes; it matches or narrowly beats our method at isolated time points (TP4 and TP9), yet falls behind on average and recovers more slowly after the shift. STG-Mamba ([19]) underperforms on this setting, especially near the regime change, suggesting limited robustness to non-stationary dynamics.

Two observations highlight the intended advantages of discrete regimes with state-conditioned emissions. First, the performance drop at the regime change is the smallest for our method (TP5-TP4 jump 4.72 vs. 4.99 for LSTM–GNN, 5.36 for MLP, 5.02 for STG-Mamba), indicating better alignment to the new dynamics. Second, our method shows the fastest one-step recovery (TP5-TP6 drop -3.71 vs. -3.40 for LSTM-GNN) and post regime changes performances, consistent with rapid state reassignment and expert specialization once the system switches regimes. A residual limitation is a mild degradation within long single-regime segments (e.g., TP4 and TP9), which we attribute to occasional regime misassignment due to insufficient penalty on remaining in an incorrect state (see Figure 3). This suggests a simple mitigation with a calibrated self-transition regularizer without altering the overall architecture.

Table 1: Validation performance on synthetic data at different time-points (TP1–TP9) of the sequence $(G_t)_{t=1}^{9}$. For each method and time step we compute the MSE between $\widehat{X}_{t+1}$ and X_{t+1}.

Model	TP1	TP2	TP3	TP4	TP5	TP6	TP7	TP8	TP9
Persistence	6.57	6.55	6.69	6.66	6.90	6.92	6.92	6.97	6.74
MLP	2.26	1.98	2.17	1.87	7.23	3.79	3.46	4.12	3.92
LSTM-GNN	1.21	1.46	2.15	**1.65**	6.64	3.24	3.55	3.86	**3.72**
STG-Mamba	8.59	6.97	3.78	3.48	8.50	5.11	4.89	4.32	4.65
Our Method	**1.01**	**1.14**	**1.40**	1.75	**6.47**	**2.76**	**3.11**	**3.51**	3.96
Features Mean	3.52	3.98	4.76	5.77	5.96	6.27	6.80	7.56	7.26

3.4 Regime Trajectory Analysis

To evaluate the framework's ability to recover latent dynamics without supervision, we visualize the learned latent trajectories. We evaluate (i) unsupervised recovery of state trajectories and (ii) the speed of convergence to an identifiable latent-state distribution. Regime estimation was performed by sampling the learned posterior distribution 100 times per graph and time step, followed by majority voting to assign discrete state labels to regimes. This setup enables us to track how the inferred state distribution evolves over time, and how it aligns with ground-truth regimes.

In Figure 3, we visualize inferred state trajectories in our unsupervised setting. The model initially fails to recover the true state distribution. This is expected, as dynamic patterns must be inferred from sequential observations alone as no regime-predictive features are present. However, after a few time steps, the model converges to the true underlying regime distribution, demonstrating its capacity to infer system dynamics without regime-level supervision.

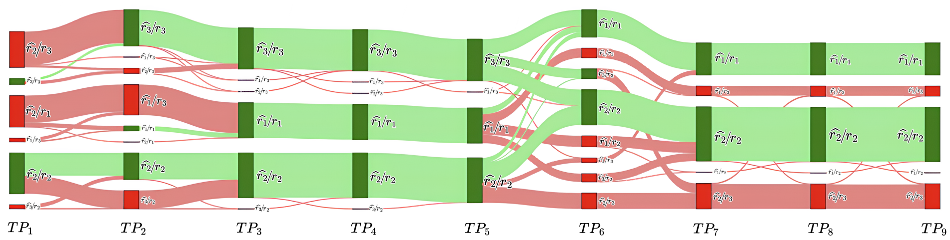

Figure 3: Unsupervised recovery of regime trajectories. The flow diagram shows one trained run across time points. Each block represents the distribution of samples (patients) by predicted/ground-truth regime pair, labeled $\widehat{r}_i/r_j$. Green indicates correct assignments ($\widehat{r}_i = r_j$); red indicates mismatches. Line widths encode the number of samples flowing between pairs over time. After a short transient the mass concentrates on correct pairing flows, showing that the model recovers the latent regimes and tracks their dynamics without supervision.

4 Conclusion

We proposed a generative framework that integrates dynamic graph neural networks with discrete state space modeling to capture interpretable spatio-temporal dynamics. By separating spatial reasoning (via GNNs) from temporal inference (via a discrete latent state and learnable transition prior), our approach addresses key limitations of prior DGNN and deep SSM models, namely limited interpretability, entangled updates, and challenges in modeling evolving graph structures through discrete regimes changes. The variational training procedure enables uncertainty-aware learning of state transitions and current regime estimation, while the inference-time state-space formulation supports forecasting without future leakage and trajectory analysis. Across synthetic experiments, ReGraSS achieves competitive predictive accuracy while exposing latent change in regimes aligned with system dynamics. Results show that our framework can recover temporal regimes from minimal supervision, highlighting its utility in settings with sparse labels.

Our study has limitations that point to potential next directions. First, the current objective emphasizes feature dynamics and may underweight structural change in the graph; incorporating topology-aware terms could better capture evolving connectivity, though care is needed to avoid prohibitive costs on large graphs. Second, the mixture-of-experts decoder scales with the number of discrete regimes, which can hinder efficiency in fine-grained settings; lighter parameter-sharing schemes may retain state-conditioned emissions with lower overhead. Third, a purely discrete latent space can be rigid when regimes overlap or evolve smoothly. Beyond methodology, our evaluation on controlled synthetic sequences should be complemented by real-world deployments that test its capabilities to maintain interpretable regimes tracking in setups with noisy samples and complex spatio-temporal dynamics.

References

[1] Longqi Liu, Ao Chen, Yuxiang Li, Jan Mulder, Holger Heyn, and Xun Xu. Spatiotemporal omics for biology and medicine. *Cell*, 187(17):4488–4519, 2024.

[2] Ali Hamdi, Khaled Shaban, Abdelkarim Erradi, Amr Mohamed, Shakila Khan Rumi, and Flora D Salim. Spatiotemporal data mining: a survey on challenges and open problems. *Artificial Intelligence Review*, 55(2):1441–1488, 2022.

[3] Mariana Belgiu and Alfred Stein. Spatiotemporal image fusion in remote sensing. *Remote sensing*, 11(7):818, 2019.

[4] Ronnachai Jaroensri, Ellery Wulczyn, Narayan Hegde, Trissia Brown, Isabelle Flament-Auvigne, Fraser Tan, Yuannan Cai, Kunal Nagpal, Emad A Rakha, David J Dabbs, et al. Deep learning models for histologic grading of breast cancer and association with disease prognosis. *NPJ breast cancer*, 8(1):113, 2022.

[5] Ling Dai, Liang Wu, Huating Li, Chun Cai, Qiang Wu, Hongyu Kong, Ruhan Liu, Xiangning Wang, Xuhong Hou, Yuexing Liu, et al. A deep learning system for detecting diabetic retinopathy across the disease spectrum. *Nature communications*, 12(1):3242, 2021.

[6] Shuai Niu, Jing Ma, Qing Yin, Liang Bai, Chen Li, and Xian Yang. A deep clustering-based state-space model for improved disease risk prediction in personalized healthcare. *Annals of Operations Research*, 341(1):647–672, 2024.

[7] Daniel Cunha Oliveira, Dylan Sandfelder, André Fujita, Xiaowen Dong, and Mihai Cucuringu. Tactical asset allocation with macroeconomic regime detection. *arXiv preprint arXiv:2503.11499*, 2025.

[8] Rongbo Chen, Mingxuan Sun, Kunpeng Xu, Jean-Marc Patenaude, and Shengrui Wang. Clustering-based cross-sectional regime identification for financial market forecasting. In *International Conference on Database and Expert Systems Applications*, pages 3–16. Springer, 2022.

[9] Jose MG Vilar and Leonor Saiz. Dynamics-informed deconvolutional neural networks for super-resolution identification of regime changes in epidemiological time series. *Science Advances*, 9 (28):eadf0673, 2023.

[10] Carolyn Augusta, Rob Deardon, and Graham Taylor. Deep learning for supervised classification of spatial epidemics. *Spatial and Spatio-temporal Epidemiology*, 29:187–198, 2019.

[11] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, Yoshua Bengio, et al. Graph attention networks. *stat*, 1050(20):10–48550, 2017.

[12] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *International Conference on Learning Representations*, 2019. URL `https://openreview.net/forum?id=ryGs6iA5Km`.

[13] Aldo Pareja, Giacomo Domeniconi, Jie Chen, Tengfei Ma, Toyotaro Suzumura, Hiroki Kanezashi, Tim Kaler, Tao Schardl, and Charles Leiserson. EvolveGCN: Evolving graph convolutional networks for dynamic graphs. *Proc. Conf. AAAI Artif. Intell.*, 34(04):5363–5370, April 2020.

[14] Jiaxuan You, Tianyu Du, and Jure Leskovec. ROLAND: Graph learning framework for dynamic graphs. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 2358–2366, New York, NY, USA, August 2022. ACM.

[15] Albert Gu, Karan Goel, and Christopher Re. Efficiently modeling long sequences with structured state spaces. In *International Conference on Learning Representations*, 2022. URL `https://openreview.net/forum?id=uYLFoz1vlAC`.

[16] Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. In *International Conference on Learning Representations*, 2024. URL `https://openreview.net/forum?id=AL1fq05o7H`.

[17] Gongpei Zhao, Tao Wang, Yi Jin, Congyan Lang, Yidong Li, and Haibin Ling. Grassnet: State space model meets graph neural network. *CoRR*, abs/2408.08583, 2024. doi: 10.48550/ARXIV.2408.08583. URL `https://doi.org/10.48550/arXiv.2408.08583`.

[18] Ali Behrouz and Farnoosh Hashemi. Graph mamba: Towards learning on graphs with state space models. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, volume 24, pages 119–130, New York, NY, USA, August 2024. ACM.

[19] Lincan Li, Hanchen Wang, Wenjie Zhang, and Adelle Coster. Stg-mamba: Spatial-temporal graph learning via selective state space model. *arXiv preprint arXiv:2403.12418*, 2024.

[20] Haonan Yuan, Qingyun Sun, Zhaonan Wang, Xingcheng Fu, Cheng Ji, Yongjian Wang, Bo Jin, and Jianxin Li. DG-Mamba: Robust and efficient dynamic graph structure learning with selective state space models. *Proc. Conf. AAAI Artif. Intell.*, 39(21):22272–22280, April 2025.

[21] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL `https://openreview.net/forum?id=rkE3y85ee`.

[22] Junyoung Chung, Çaglar Gülçehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *CoRR*, abs/1412.3555, 2014. URL `http://arxiv.org/abs/1412.3555`.

[23] Saeed Masoudnia and Reza Ebrahimpour. Mixture of experts: a literature survey. *Artificial Intelligence Review*, 42(2):275–293, 2014.

[24] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In Yoshua Bengio and Yann LeCun, editors, *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014. URL `http://arxiv.org/abs/1312.6114`.

[25] Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Russ R Salakhutdinov, and Alexander J Smola. Deep sets. *Advances in neural information processing systems*, 30, 2017.

5 Appendix

5.1 Training Procedure Details

Decoder routing and teacher forcing. During training the MoE is routed by a convex state mix

$$s_t \;=\; (1-\eta)\, p_\theta(z_t \,|\, h_G^{t-1}, z_{t-1}) \;+\; \eta\, q_\phi(z_t \,|\, h_G^{t}), \tag{2}$$

with $\eta \in [0,1]$. Early in training the decoder relies more on the posterior (teacher forcing, $\eta \approx 1/3$); as training progresses, η is annealed to 0 so emissions are governed by the learned prior, matching the causal rollout used at inference. This reduces exposure bias without sacrificing stability.

Temperature and alignment schedules. We parameterize q_ϕ with a Gumbel–Softmax at temperature τ. We anneal τ from 1.0 to a small floor (e.g., 0.2) over the first half of training to promote confident, non-degenerate state usage while avoiding premature hard assignments. The alignment weight β is linearly warmed from 0 to 1 over the first third of training so that forecasting stabilizes before the encoder–prior terms dominate. The prior-fitting weight γ is set to 1 and may be mildly reduced later (e.g., to 0.7 after 60% of training) if the learned prior becomes too reactive. The decoder mix η is annealed linearly to 0 over the first 40% of training to phase out teacher forcing. These schedules were chosen empirically to prevent posterior collapse, avoid chasing a moving prior, and align the training-time routing with the inference-time strictly time-ordered rollout.

Lightweight regularization. We add two small regularizers that do not alter the loss but improve state usage: (i) a *diversity* term that keeps the batch-average posterior close to uniform, $\mathrm{KL}(\bar{q} \,\|\, \mathrm{Unif})$ with $\bar{q} = \frac{1}{B}\sum_i q_\phi^{(i)}(z_t \,|\, h_G^t)$, to avoid dead states; and (ii) a *sharpness* term that lowers the entropy of per-graph posteriors, $\mathbb{E}[H(q_\phi(z_t \,|\, h_G^t))]$, ramped in after the temperature has decreased. Both are coefficients of small amplitudes (e.g., $\lambda_{\text{marg}} \approx 0.1$, $\lambda_{\text{sharp}} \leq 0.05$).

Implementation notes. The temporal encoder is instantiated with a ROLAND-style [14] dynamic GNN that maintains node memories via GRU updates and pools to h_G^t, but we discard the the live update and caching mechanisms that are not relevant in our setup. All training were performed using internal cluster GPUs. A couple of workers (2-4) are sufficient due to the small size of the dataset. The dataset was randomly split with label stratification (based on regime) following a 75%/25% split for training/validation. Hyperparameters of each method were selected using 4-fold cross-validation on the training set.

The MoE decoder comprises K independent MLP experts $\{f_k\}_{k=1}^K$ with outputs combined by s_t from Eq. (2). Experts can be replaced with domain-specific modules without changing the objective. We optimize with Adam, apply gradient clipping for stability, and select checkpoints on validation one-step error.

5.2 Dataset Generation Process

We generate 150 spatio-temporal graph sequences with $T = 10$ snapshots (TP1–TP10). Each sequence begins with a randomly sampled discrete regime $r_1 \in \{r_1, r_2, r_3\}$, and undergoes a single potential regime change before TP5, as detailed below.

Node clusters. Let $d = 8$ denote the feature dimension. We sample the number of clusters $C \sim \mathrm{Unif}\{3,4,5\}$. For each cluster $c \in \{1,\ldots,C\}$, we draw a center $\mu_c \sim \mathrm{Unif}([-10,10]^d)$ and use an isotropic covariance $\Sigma = 0.6^2 I_d$. We then sample the cluster size $M_c \sim \mathrm{Unif}\{5,\ldots,100\}$ and node coordinates

$$x_i^{(1)} \sim \mathcal{N}\big(\mu_{c(i)},\, \Sigma\big) \quad \text{for } i = 1,\ldots,\sum_{c=1}^{C} M_c.$$

Edges (spatial proximity). For each snapshot t, we build an undirected k-nearest neighbor graph on $\{x_i^{(t)}\}_i$ in $\mathbb{R}^d$ with Euclidean distance and

$$k \;=\; \max_{1 \leq c \leq C} M_c \;+\; 1$$

to assure bridges between clusters. We keep the same set of edges the graph at each t; since the dynamics below are global translations plus small noise, the topology is translation-invariant and empirically stable across t.

Regimes and dynamics. Regimes induce constant drifts along all coordinates:

$$v(r_1) = -2\,\mathbf{1}_d, \qquad v(r_2) = +2\,\mathbf{1}_d, \qquad v(r_3) = \mathbf{0}_d.$$

Let $\varepsilon_i^{(t)} \sim \mathcal{N}\big(0,\ 0.6^2 I_d\big)$ be i.i.d. perturbations. For $t = 1, \ldots, 9$, node positions evolve as

$$x_i^{(t+1)} = x_i^{(t)} + v(r_t) + \varepsilon_i^{(t)}.$$

Before applying the update to obtain TP5 (i.e., between TP4 and TP5), we resample the regime $r_k^{TP5} \sim \mathrm{Unif}\{r_1, r_2, r_3\}$ independently of r_k^{TP1}; consequently, some sequences keep their regime while others switch. The labels $\{r_k^{TPi}\}$ are not used for supervision.

The dataset was randomly split with label stratification (based on regime) following a 75%/25% split for training/validation.

Rationale. This construction test models' ability to infer discrete regime trajectories from spatially structured observations and to remain robust to a mid-sequence non-stationarity (TP4→TP5).

5.3 Baselines Implementation Details

MLP (no space, no time) The MLP receives a concatenation of node features across the observed horizon, so the per-node input dimensionality grows linearly with time (e.g., with base feature size $d = 8$, inputs are 8 at TP1, 16 at TP2, etc.). To accommodate dynamic topology (variable node counts across graphs and time), we mimic message passing with *self-loops only*: a shared per-node MLP processes each node independently (no neighbor aggregation), producing per-node embeddings at time t. We then apply parametric pooling via a small pooling MLP (DeepSets-style [25]) to obtain a graph-level context vector. Final node-level predictions $\hat{X}_{t+1}$ are produced by another shared MLP that conditions on both the node's self-updated embedding and the pooled context. This design ignores explicit topology while still permitting information mixing through learnable pooling.

Persistence ($X_{t+1} = X_t$) A parameter-free, time-only baseline that copies the last observation to the next step. It has a slight advantage in regimes with near-constant dynamics (r_3 in the synthetic dataset, where the drift is absent) but remains weak overall, providing a lower bound that tests whether temporal autocorrelation alone explains performance.

LSTM–GNN (ROLAND-based) We instantiate a ROLAND-style dynamic GNN encoder ([14]) with GRU updates ([22]) for hierarchical node states. At each time step, node embeddings are updated by a graph layer and then temporally evolved via GRUs; the model natively supports dynamic topology. As in our main architecture, we discard caching and live-update mechanisms. For this baseline we directly project to node features with a linear head (no graph-level pooling), yielding a strong spatio-temporal encoder without discrete regimes or state-conditioned emissions.

STG-Mamba We follow STG-Mamba [19]: blocks interleave spatial mixing (graph filtering/propagation on the current adjacency) with temporal Mamba modules that implement selective state-space updates along time. Each block uses residual connections, normalization, and pointwise MLPs. Stacking several blocks yields multi-scale spatio-temporal modeling. Training minimizes next-step MSE. Other GNN-SSM hybrids were considered, but most lacked robustness to topological change (relevant for our real-world, public results not yet available) or had no public implementations, so we did not include them.

All baselines follow a similar training procedure as described in section 5.1 and 2.3. Notably all trainings were performed on internal cluster with GPUs. The hyperparameters of each baseline were selected using 4-fold cross-validation on the training set, later evaluated on the validation set as reported in table 1.

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: We claim that current methods often disregards regime detection and tracking in spatio-temporal systems. We show how our method leverage the regimes trajectory through the discrete state space in Section 3.4, and how it does not interfer with performances in section 3.3.

   Guidelines:

   - The answer NA 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 NA 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: Yes in the end of the conclusion section 4 we give some drawbacks and limitations of the methods together with some possible improvements.

   Guidelines:

   - The answer NA 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: [NA]

Justification: No theoretical theorem or results provided in the paper.

Guidelines:

- The answer NA 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: We extensively described the generation of the dataset together with implementation details of both our method and baselines in the Sections 3.1, 2.2, 3.2. We provide additional information on training procedure, scheduling and implementation in Section 5.2, 5.1, 5.3.

Guidelines:

- The answer NA 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: [No]

   Justification: The process for generating data is explained in extensive details, allowing replication. The code base is kept private until final publication.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.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://nips.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, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: All the details are present in the appendix (Section 5.1, 5.2, 5.3). We keep a couple of specific implementation details and hyperparameters search for the main publication.

   Guidelines:

   - The answer NA 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: [NA]

   Justification: The set of experiments reported in this paper is directly performed on the left-out test set after cross-validation on the training set. Thus no direct measure of statistical significance can be performed on the results reported in Table 1. We could report the detailed cross-validation results if the reviewers wish it.

   Guidelines:

- The answer NA 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: In the appendix (5.1), we provide information about annealing rate and general details on the training procedure.

   Guidelines:

   - The answer NA 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: We acknowledge the NeurIPS Code of Ethics and believe that the method described in this paper has no specific negative societal impact and potential harmful consequences.

   Guidelines:

   - The answer NA 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: We discussed how our work can be helpful in high-stake applications (Sections 1 and 4) where the interpretability of the spatio-temporal dynamics can be mapped to well established regimes, like in healthcare or finance.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA 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., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper poses no such risk.

    Guidelines:

    - The answer NA 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: We credited all previous work, especially previous published methods, that were used throughout the paper.

    Guidelines:

    - The answer NA 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 generated synthetic dataset has been extensively described in section 5.2.

    Guidelines:

    - The answer NA 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: [NA]

    Justification: No involvement of crowdsourcing or research with human subjects.

    Guidelines:

    - The answer NA 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: [NA]

    Justification: No involvement of crowdsourcing or research with human subjects.

Guidelines:

- The answer NA 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 rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: The core method development in not centered around LLMs.

    Guidelines:

    - The answer NA 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 (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.