Hierarchical Regime-Conditioned Dynamics for Spatiotemporal Graphs



Baffou
Jérémy

NeurIPS- 10/11/26

Relational systems

Entities that evolve jointly, with their interactions driving the system dynamics

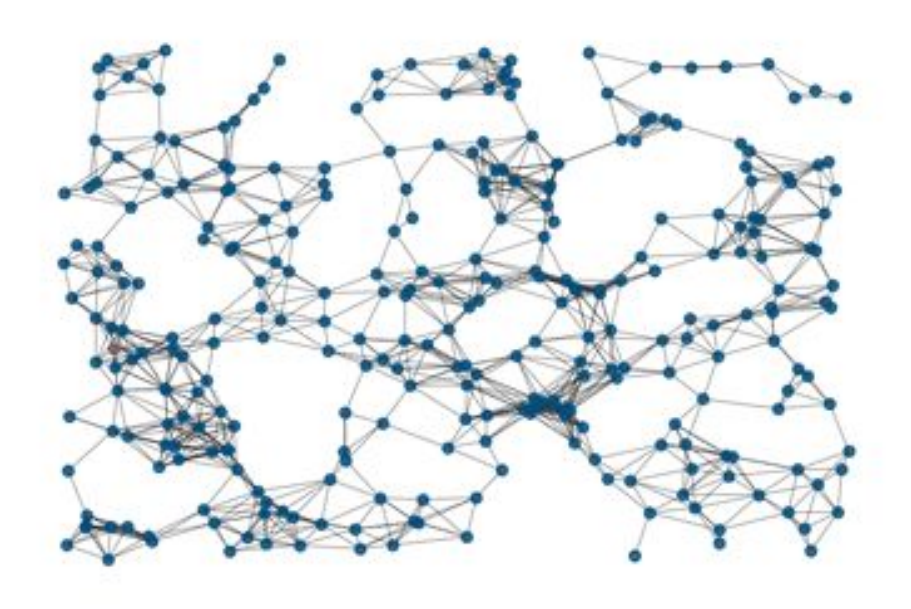
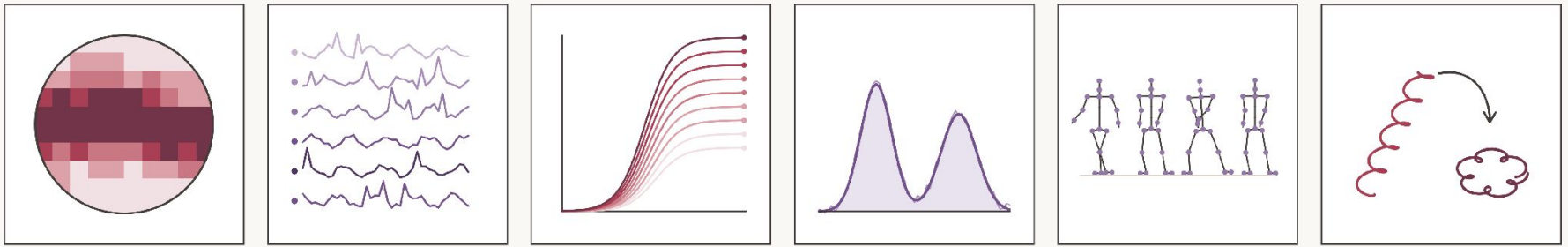


Figure 1: Continuous time SIS infection process, mimicking disease spread in an epidemiological network. Taken from MoSi Lab [1].

AI has transformed prediction

Diverse dynamics, learned from temporal data across widely different domains



Climate
temperature fields

Brain
neural activity

Cancer
tumor growth trajectories

Epidemics
daily case dynamics

Human motion
3D joint trajectories

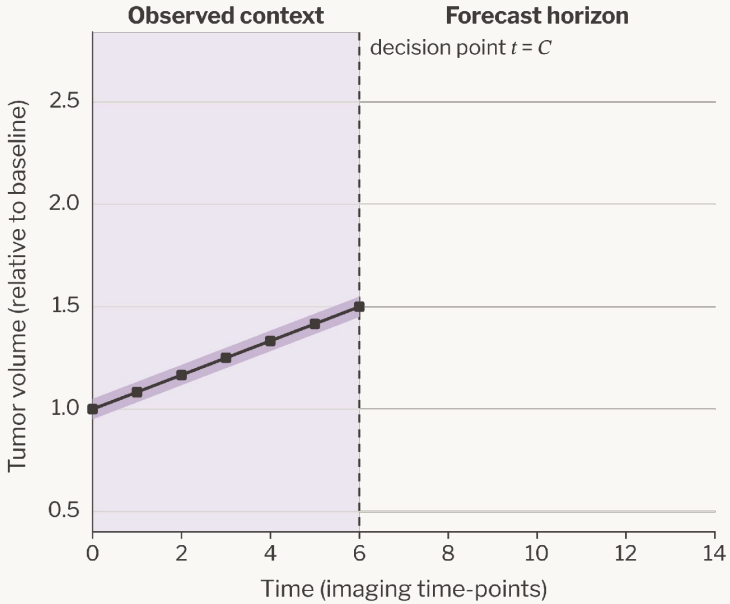
Proteins
folding dynamics

- Across domains, AI forecasts complex temporal dynamics with **state-of-the-art accuracy**.

Representative recent work: Climate: GraphCast (Lam et al., 2023) · Brain: CEBRA (Schneider et al., 2023) · Cancer: TDNODE (Laurie & Lu, 2023) · Epidemics: epidemic GNNs (Liu et al., 2024) · Motion: LaM-SLidE (Sestak et al., 2025) · Proteins: AlphaFold 3 (Abramson et al., 2024)

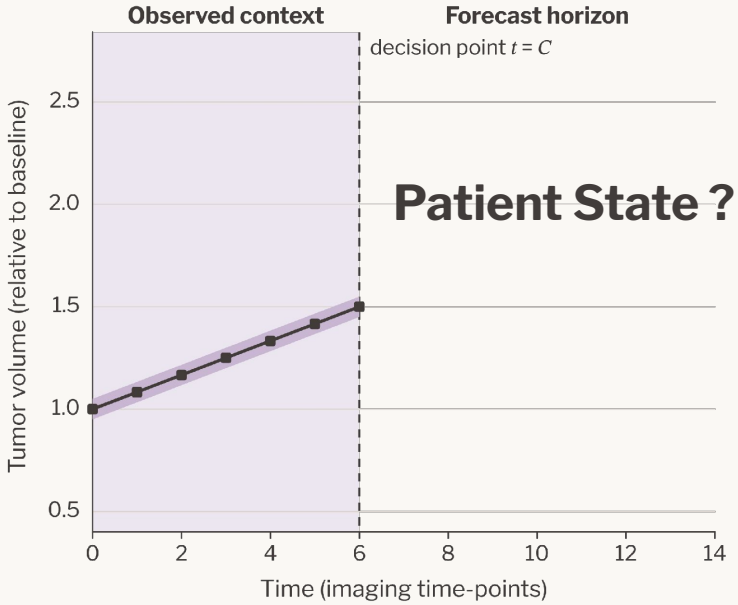
Forecasting alone is not enough

Similar observations can arise from different underlying mechanisms



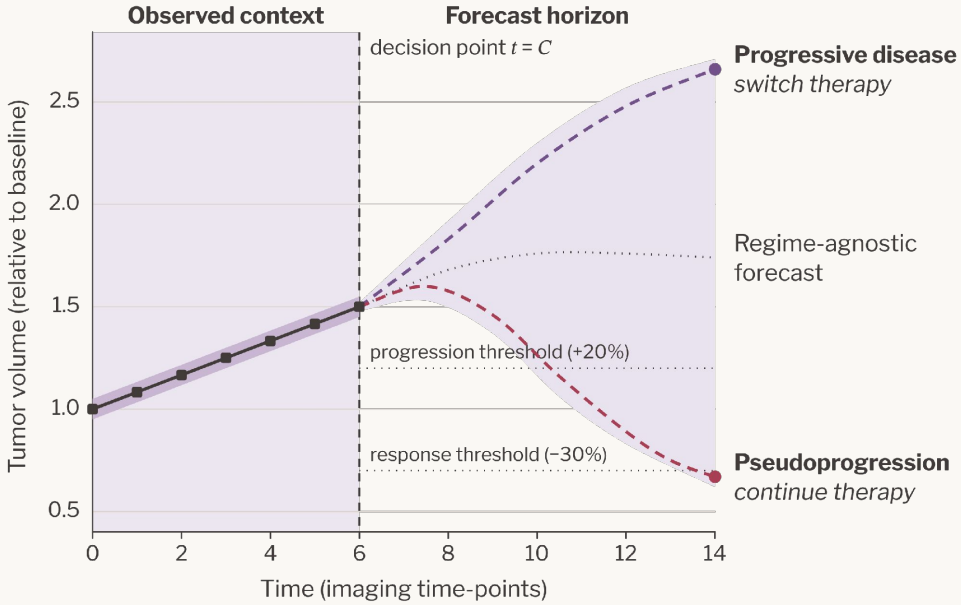
Forecasting alone is not enough

Similar observations can arise from different underlying mechanisms



Forecasting alone is not enough

Similar observations can arise from different underlying mechanisms

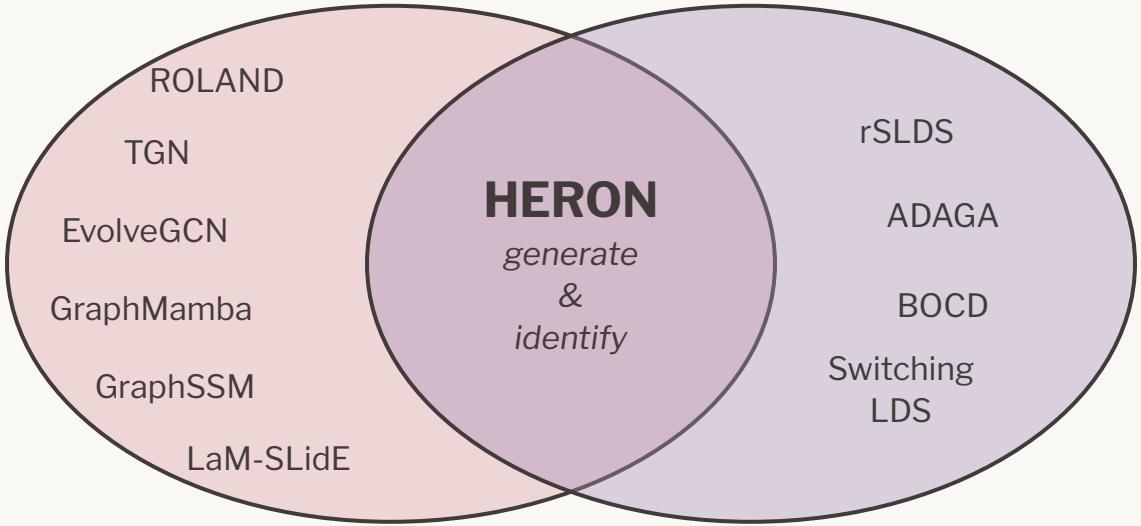


- **Forecast + Regime**
  - Decision making needs to understand both:
    - *What they will be* (**forecasts**)
    - *How they will change* (**regime**)

Two worlds, one gap

Strong graph forecasters keep regimes implicit; regime models can't forecast

Forecasting on dynamic graphs
regime implicit + entangled



Regime & mode inference
no features generation

- **HERON closes the gap** → discrete regimes inferred from graph-temporal context that condition graph forecasts.

HERON · HiErarchical Regime-cOnditioned dyNamics

A self-supervised, regime-aware generative framework for spatio-temporal graphs

- **Regime-aware framework**
  - Hierarchy can be plugged into existing backbones.
- **Regime-conditioned rollouts**
  - Track modes and simulate alternative futures.
- **Self-supervised forecasting**
  - No regime annotations required.

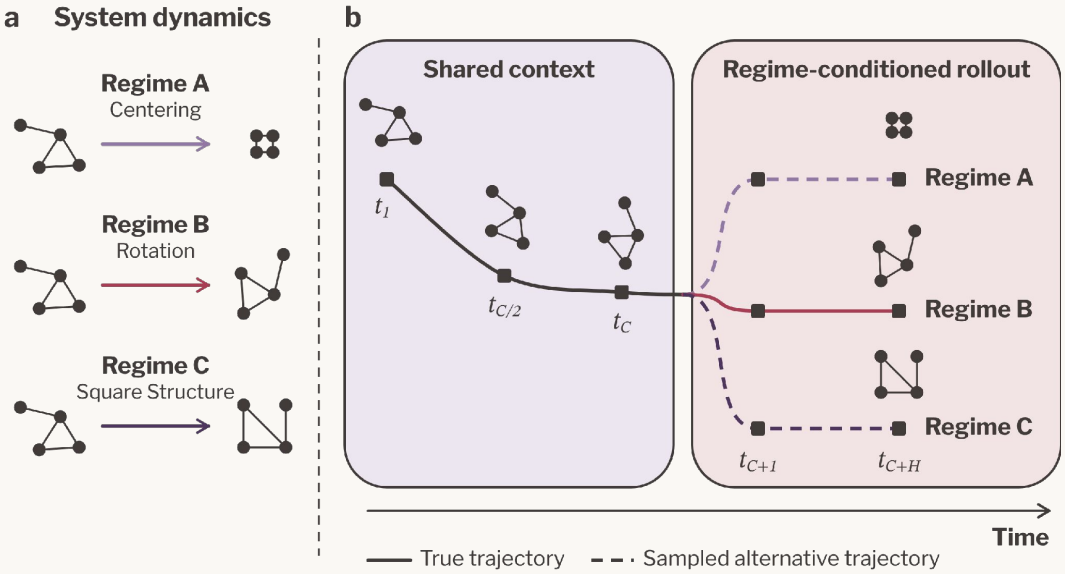


Figure 2: The system can evolve according to different dynamics named *regimes*. Given observations acting as *context*, we predict the feature evolution on a given topology over a *horizon*.

A hierarchical latent state

Disentangle global regimes from entity-level dynamics

- **System level**
  - A deterministic global context plus a discrete regime over K modes
- **Entity level**
  - A per-node deterministic state and a continuous latent variable

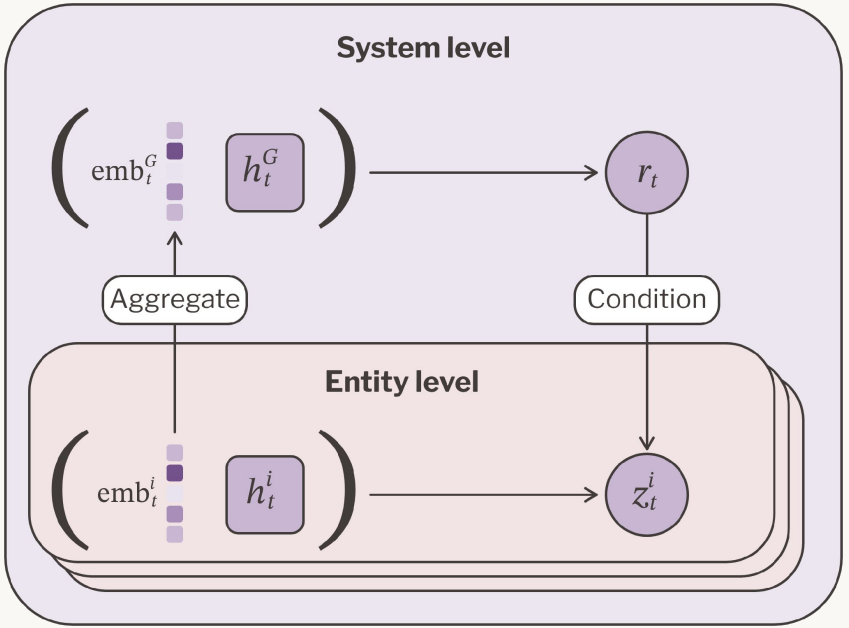


Figure 3: Aggregate entity embeddings to a regime; the regime conditions entity latents.

Regimes gate a mixture of experts

The inferred regime actively conditions entity dynamics and not a passive label

- **Recurrent update**
  - GRUs propagate the deterministic states at both levels.

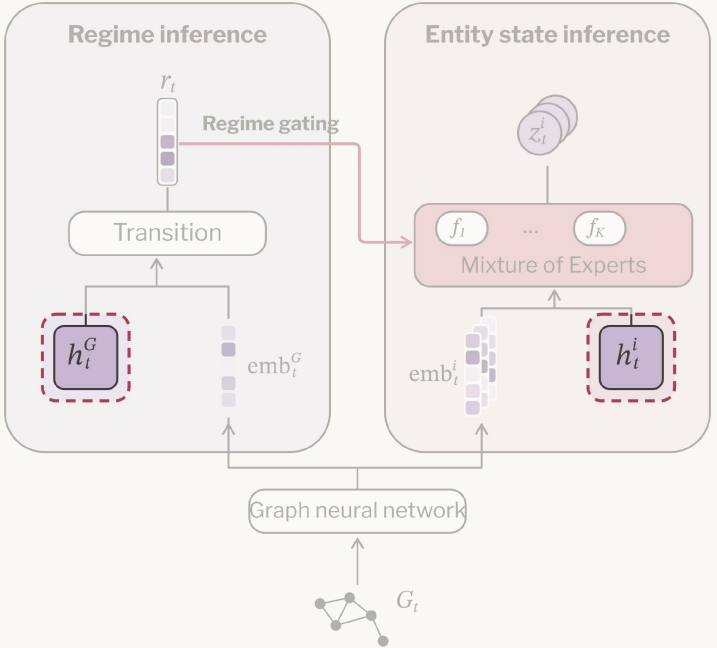


Figure 4: Regime gating drives a per-regime mixture of experts.

Regimes gate a mixture of experts

The inferred regime actively conditions entity dynamics and not a passive label

- **Recurrent update**
  - GRUs propagate the deterministic states at both levels.
- **Relational embeddings**
  - A GNN with pooling summarizes the current graph snapshot.

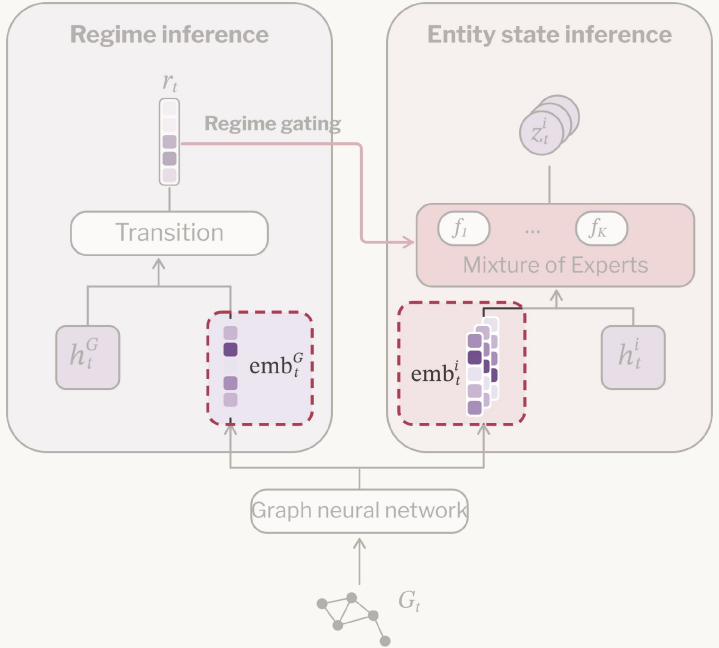


Figure 4: Regime gating drives a per-regime mixture of experts.

Regimes gate a mixture of experts

The inferred regime actively conditions entity dynamics and not a passive label

- **Recurrent update**
  - GRUs propagate the deterministic states at both levels.
- **Relational embeddings**
  - A GNN with pooling summarizes the current graph snapshot.
- **Hierarchical inference**
  - Infer the regime, then the regime-conditioned entity latents.

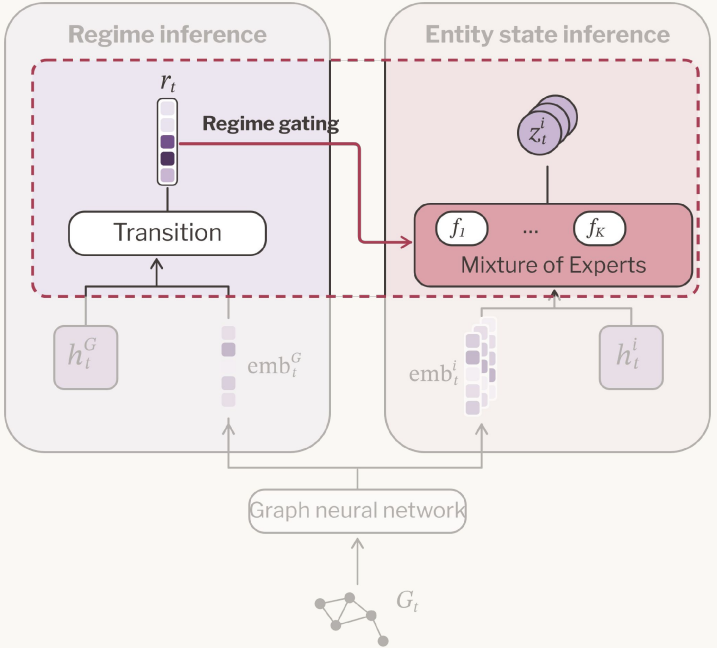


Figure 4: Regime gating drives a per-regime mixture of experts.

One hierarchical forward pass

Update memory → infer regime → condition entity latents → decode

- **Recurrent update**
  - GRUs propagate the deterministic states at both levels.
- **Relational embeddings**
  - A GNN with pooling summarizes the current graph snapshot.
- **Hierarchical inference**
  - Infer the regime, then the regime-conditioned entity latents.
- **Probabilistic decode**
  - An MLP maps the state to a Gaussian over next features.

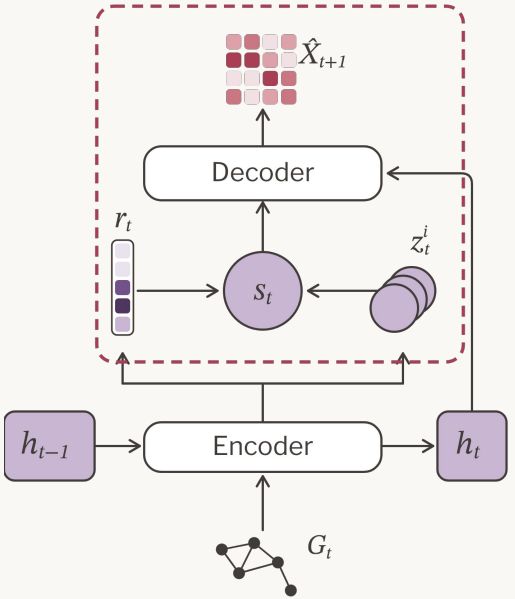
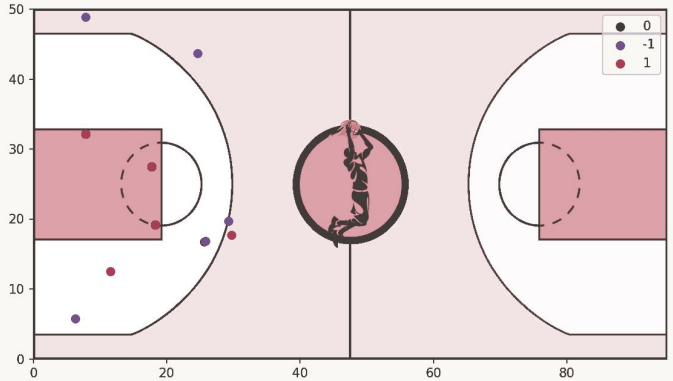


Figure 5: Highlight of a full forward pass of the model. Following the sequence encoder → regime → state → decoder.

Real-world benchmarks: NBA & AMASS

Multi-agent play (fixed regime) and human motion (switching regimes)

NBA SportVU · *multi-agent play*



- 10 players + ball · fully-connected graph
- Node = position + type; Edge = distance + team (±1)
- Only coordinates & edge-distance vary over time
- **2 regimes, fixed per play:** Scoring · Rebounding

AMASS / BABEL · *human motion*



- 22-joint SMPL skeleton · BMLmovi/rub, CMU, KIT
- Node = position + anatomy; Edge = limb + fixed distances
- Only joint positions vary over time
- **4 regimes, switching:** Walk · Stand · Turn · Lower

Competitive forecasting with regime conditioning

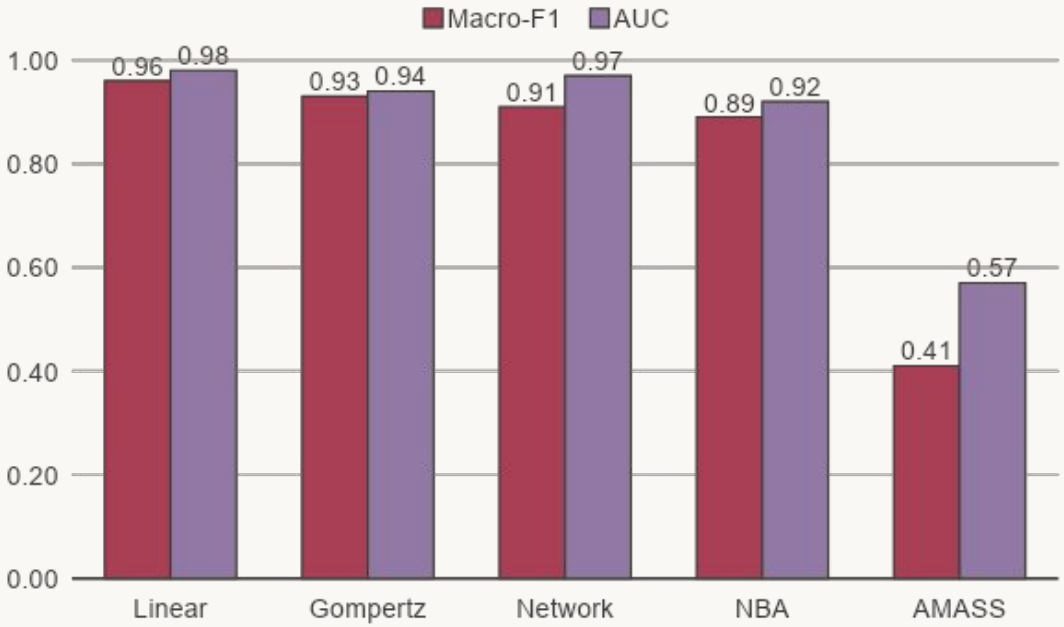
Test ADE (synthetic) and ADE / FDE (NBA, AMASS) — lower is better

Model	Linear	Gompertz	Network	NBA	AMASS
Persistence (Static)	5.595	2.521	0.105	4.48/7.86	.093/.135
Persistence (Velocity)	2.896	2.906	0.058	2.55/5.75	.129/.271
S4	4.072	4.307	0.162	2.09/5.11	.422/.616
EvolveGCN	3.454	2.623	0.057	1.72/4.92	.060/.088
LaM-SLidE	2.077	N/A	0.054	**0.73/1.32**	.061/.084
GraphSSM	2.537	2.178	0.049	1.65/4.59	.056/.082
GraphSSM + HERON	2.494	2.042	0.047	1.59/4.43	.055/.080
HERON	**1.937**	**1.938**	**0.038**	0.81/1.49	**.047/.066**

- Best ADE/FDE on 4 datasets while exposing trackable regimes
- Captures structured, propagation-driven dynamics
- Plugging HERON on top of another backbone improved performance by 5% on average.

HERON recovers meaningful regimes

Regime labels are used only for evaluation — never during training



- HERON identifies diffusion processes on Network dataset
- On NBA it distinguishes scoring vs. rebound play patterns
- Performance is reduced on AMASS due to the fine granularity of annotations and the quickly changing motion patterns

Regimes are actionable, not just descriptive

Clamp the regime during rollout → generate distinct, plausible futures

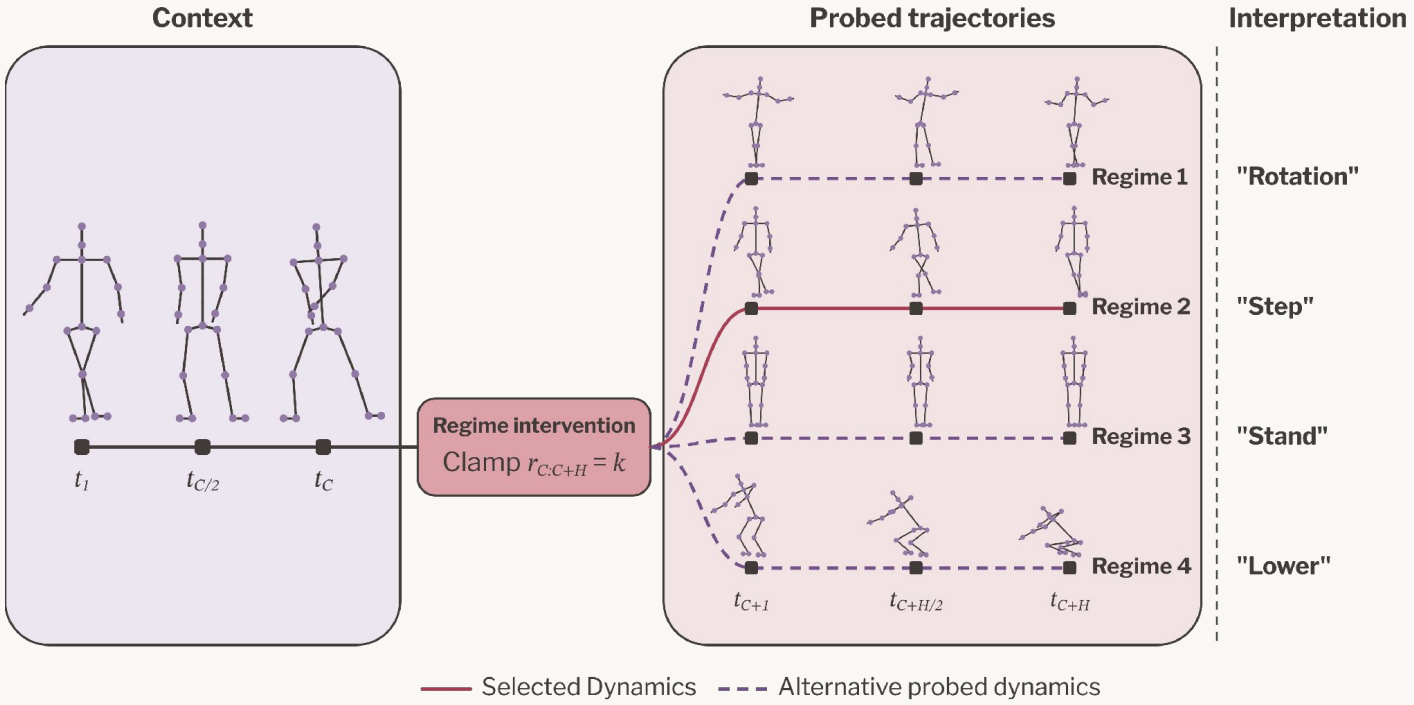


Figure 6: Clamping the regime during rollout exposes distinct, plausible futures.

Regimes that forecast, track, and control

A unified, self-supervised view of dynamics on graphs

- **Unify**
  - **Graph-structured** trajectory generation and **self-supervised regime identification** in one hierarchical model.
- **Condition**
  - **Discrete regimes are explicit conditioning variables**, not just post-hoc explanations.
- **Actionable**
  - Competitive forecasts, meaningful regimes, and **controllable alternative futures.**