Sequence Modeling with Spectral Mean Flows

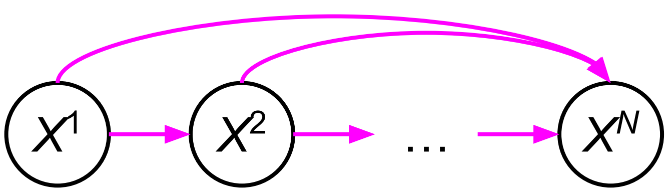
Jinwoo Kim[1] Max Beier[2] Petar Bevanda[2] Nayun Kim[1] Seunghoon Hong[1]

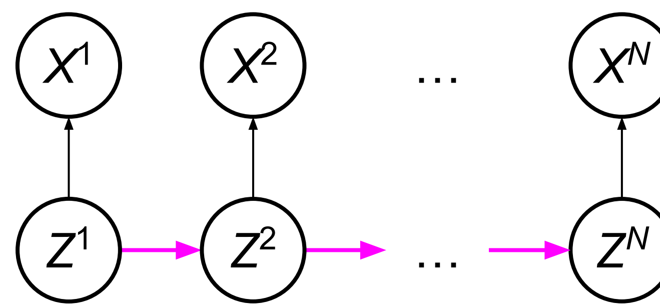
[1]KAIST [2]TU Munich

NEURAL INFORMATION PROCESSING SYSTEMS

Sequence Modeling

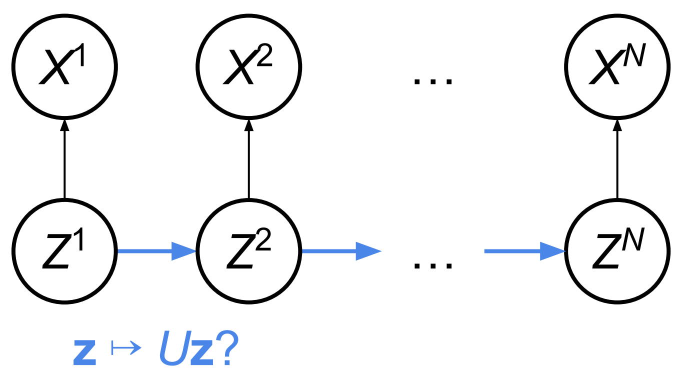
- How to represent and learn highly nonlinear and probabilistic dynamics?
- The usual way: Model each step by a stochastic nonlinear neural network.
  - Issue: Generation is serialized across sequence length.





Sequence Modeling

- How to represent and learn highly nonlinear and probabilistic dynamics?
- Linear recurrence offers parallelizability.
  - Issue: Deterministic in nature. Uncertainty often handled post hoc.



Sequence Modeling

- Can we make a sequence model that is…
  - Computationally linear over sequence length,
  - And natively handles nonlinear and probabilistic dynamics?

- How can we characterize its generative process?
- How can we parameterize and train it?

Operator Theory

- Operator theory lifts the notion of vector spaces to functions and probability distributions.

- This allows one to work with probability distributions and their evolutions using linear algebraic tools.

Operator Theory

- A probability distribution is represented as a vector (in a Hilbert space)

$$X \sim \rho \qquad \Rightarrow \qquad \mu_\rho = \mathbb{E}[\phi(X)]$$

  - The vector is called the mean embedding via the feature map ϕ. Under some conditions, it encodes the full information of embedded distribution.

- Maximum mean discrepancy (MMD) measures distance between distributions

$$\mathrm{MMD}(\rho, \pi) = ||\mu_\rho - \mu_\pi||$$

Operator Theory

- A conditional P[X|Z] is represented by a linear map (between Hilbert spaces)

$$\mu_{X|Z=\mathbf{z}} = \mathbb{E}[\phi(X) \mid Z=\mathbf{z}] = U\varphi(\mathbf{z}) \qquad \Rightarrow \qquad \mu_X = U\,\mu_Z$$

  - The map U is called the conditional mean embedding (CME) operator.

- For a linear operator, we may compute a spectral decomposition

$$U = \sum_i \lambda_i \, (h_i \otimes g_i)$$

Operator Theory

- A joint distribution is represented as a tensor (in a product Hilbert space)

$$X^1 \ldots X^N \sim \rho \qquad \Rightarrow \qquad \mu_\rho = \mathbb{E}[\ \phi(X^1) \otimes \phi(X^2) \otimes \ldots \otimes \phi(X^N)\ ]$$

  - Under some conditions, it encodes the full information of the distribution.

- Challenge 1: Issues with tractability due to the size

Operator Theory

- Suppose we have mean embedding μ_ρ. How do we generate a sample $\mathbf{x} \sim \rho$?
- MMD gradient flow defines a probability path $(p_t)_{t \geq 0}$ driven by vector field $(v_t)_{t \geq 0}$

$$v_t(\mathbf{x}) = -\nabla_{\mathbf{x}} \langle \phi(\mathbf{x}), \mu_{pt} - \mu_\rho \rangle$$

  - Continuity equation; $\partial_t p_t + \mathrm{div}(p_t v_t) = 0$
  - As $t \to \infty$, we have $p_t \to \rho$.
- Challenge 2: Sampling is slow as convergence is guaranteed in time limit

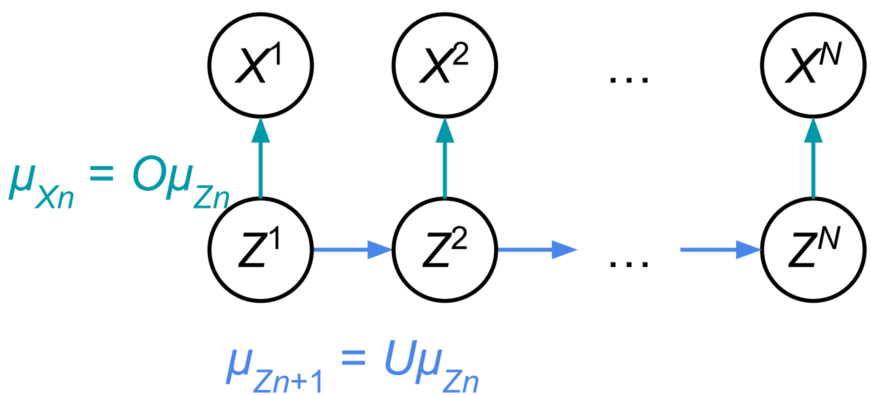
Hidden Markov Model

- Can express arbitrary nonlinear and probabilistic dynamics

- Described by conditionals P[Z^{n+1}|Z^n] and P[X^n|Z^n] and respective operators

  - Transition $\quad U\varphi(\mathbf{z}) = \mathbb{E}[\varphi(Z^{n+1}) \mid Z^n=\mathbf{z}]$

  - Observation $\quad O\varphi(\mathbf{z}) = \mathbb{E}[\phi(X^n) \mid Z^n=\mathbf{z}]$

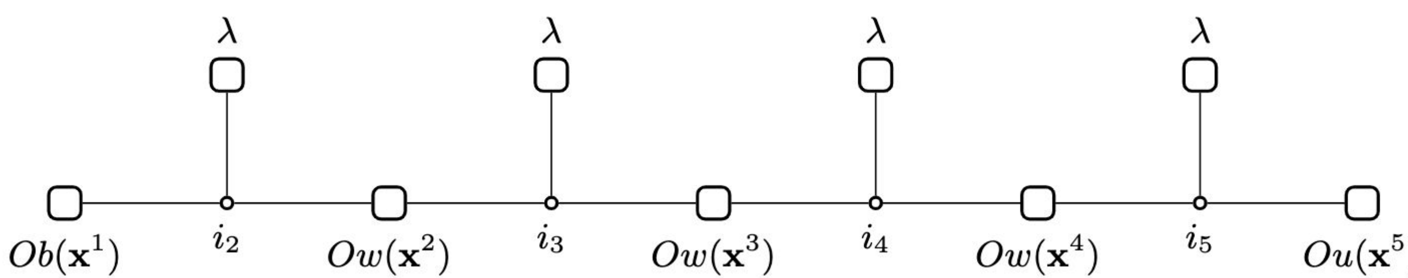


Hidden Markov Model

- From the linear operator structure, we derive a scalable decomposition of the full sequence mean embedding

$$\mu_\rho = O^{\otimes N}\, \mu_{Z1\ldots ZN}, \qquad \mu_{Z1\ldots ZN} = \textstyle\sum_{i2\ldots iN} b_{i2} \otimes \lambda_{i2}\, w_{i2,i3} \otimes \cdots \otimes \lambda_{iN-1}\, w_{iN-1,iN} \otimes \lambda_{iN}\, h_{iN}$$

- Tensor network structure; $\langle \phi(\mathbf{x}^1) \otimes \cdots \otimes \phi(\mathbf{x}^N), \mu_\rho \rangle$ is tractable and is linear in sequence length; resolves Challenge 1



Accelerating MMD Flow

- At each time point, the MMD flow follows the steepest direction of the MMD measured by a fixed feature map ϕ.

- We can make it flexible using time-varying Hilbert space feature map $(\phi_t)_{t \geq 0}$

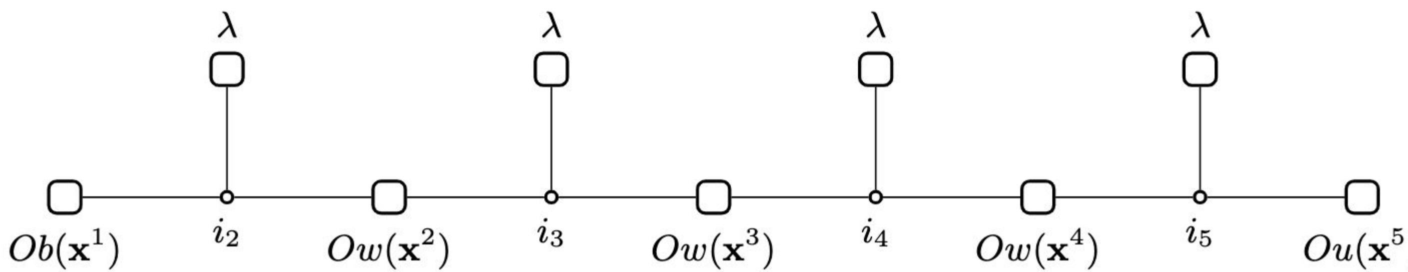
$$v_t(\mathbf{x}) = -\nabla_{\mathbf{x}} \langle \phi_t(\mathbf{x}), \mu_{pt,t} - \mu_{\rho,t} \rangle, \qquad \mu_{\rho,t} = \mathbb{E}[\phi_t(X)]$$

- But how can we identify ϕ_t that guarantees fast convergence?

  - By regressing a known vector field $(u_t)_{t \in [0,1]}$ inducing $(q_t)_{t \in [0,1]}$ with $q_1 \approx \rho$. A choice can be taken from flow matching literature; solves Challenge 2

Neural Network Parameterization

- Based on neural tangent kernel theory, we parameterize each Hilbert space element as a time-conditioned scalar-valued MLP. The MMD flow is defined through their gradients with respect to the input.



- All components are end-to-end trained with flow matching.

- (more engineering details in the paper)

Synthetic experiment

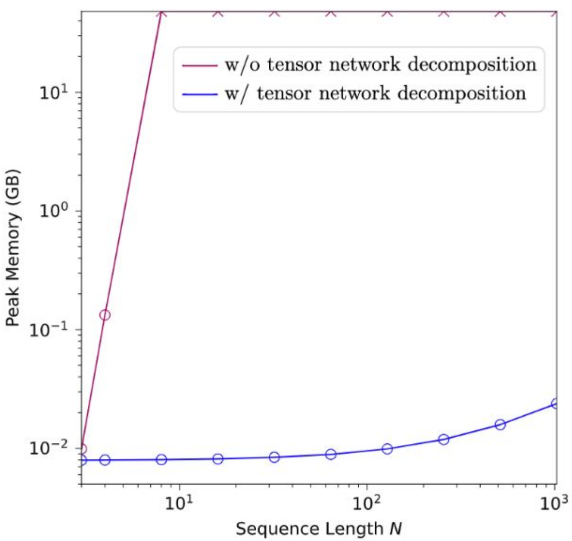


Figure 2: Peak GPU memory of inner product $\langle \mathbf{x}^1 \otimes \cdots \otimes \mathbf{x}^N, \mu \rangle$ depending on the use of tensor network decomposition (3.17).

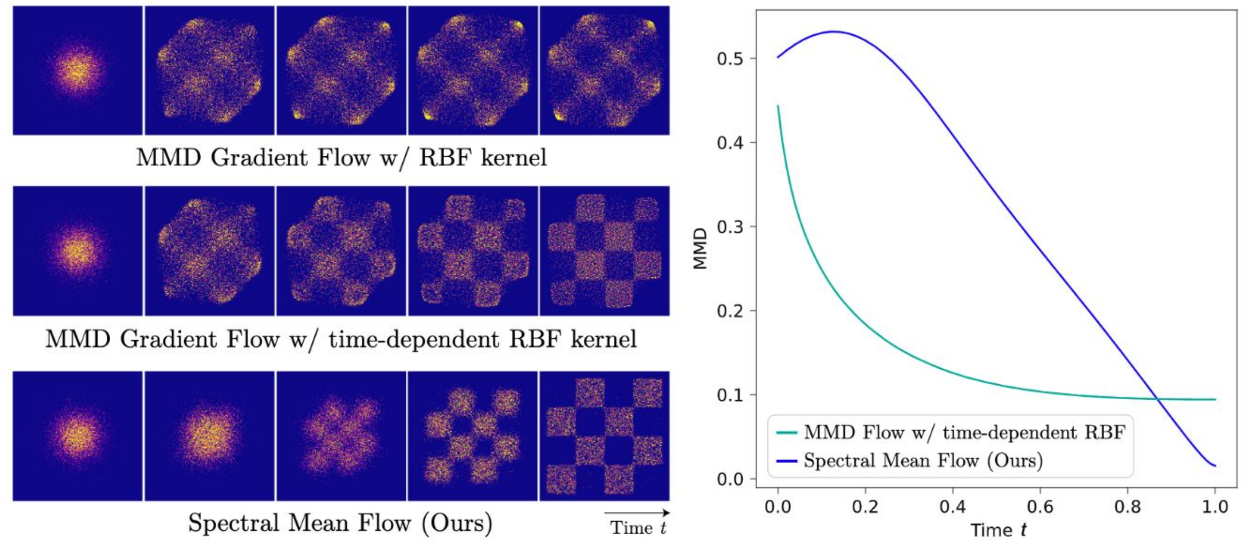


Figure 3: 2D checkerboard experiment. Left: Intermediate distributions over sampling timesteps (**zoom in** for a better view). Right: MMD between the intermediate and target distributions over sampling timesteps, measured with an RBF kernel of bandwidth 1.

Time-series Modeling

Table 1: Time-series generative modeling.

Metric	Methods	Sines	Stocks	ETTh	MuJoCo	Energy	fMRI
Context-FID Score ↓	Ours	**0.004±.001**	**0.008±.003**	**0.058±.007**	0.018±.002	**0.051±.009**	**0.116±.004**
	Diffusion-TS	0.013±.001	0.169±.021	0.126±.007	**0.015±.001**	0.113±.011	0.118±.007
	DiffTime	0.006±.001	0.236±.074	0.299±.044	0.188±.028	0.279±.045	0.340±.015
	Diffwave	0.014±.002	0.232±.032	0.873±.061	0.393±.041	1.031±.131	0.244±.018
	TimeGAN	0.101±.014	0.103±.013	0.300±.013	0.563±.052	0.767±.103	1.292±.218
	TimeVAE	0.307±.060	0.215±.035	0.805±.186	0.251±.015	1.631±.142	14.449±.969
	Cot-GAN	1.337±.068	0.408±.086	0.980±.071	1.094±.079	1.039±.028	7.813±.550
Correlational Score ↓	Ours	0.027±.012	0.010±.007	**0.040±.015**	**0.173±.016**	**0.732±.107**	**0.737±.021**
	Diffusion-TS	**0.016±.005**	0.010±.009	0.049±.013	0.188±.035	0.788±.075	1.252±.070
	DiffTime	0.017±.004	**0.006±.002**	0.067±.005	0.218±.031	1.158±.095	1.501±.048
	Diffwave	0.022±.005	0.030±.020	0.175±.006	0.579±.018	5.001±.154	3.927±.049
	TimeGAN	0.045±.010	0.063±.005	0.210±.006	0.886±.039	4.010±.104	23.502±.039
	TimeVAE	0.131±.010	0.095±.008	0.111±020	0.388±.041	1.688±.226	17.296±.526
	Cot-GAN	0.049±.010	0.087±.004	0.249±.009	1.042±.007	3.164±.061	26.824±.449
Discriminative Score ↓	Ours	**0.006±.006**	**0.022±.013**	**0.027±.010**	**0.005±.004**	0.161±.021	**0.136±.207**
	Diffusion-TS	0.030±.006	0.085±.026	0.075±.007	0.012±.006	**0.154±.012**	0.158±.020
	DiffTime	0.013±.006	0.097±.016	0.100±.007	0.154±.045	0.445±.004	0.245±.051
	Diffwave	0.017±.008	0.232±.061	0.190±.008	0.203±.096	0.493±.004	0.402±.029
	TimeGAN	0.011±.008	0.102±.021	0.114±.055	0.238±.068	0.236±.012	0.484±.042
	TimeVAE	0.041±.044	0.145±.120	0.209±.058	0.230±.102	0.499±.000	0.476±.044
	Cot-GAN	0.254±.137	0.230±.016	0.325±.099	0.426±.022	0.498±.002	0.492±.018
	RNN-AR	0.495±.001	0.226±.035	-	-	0.483±.004	-
Predictive Score ↓	Ours	**0.093±.000**	**0.037±.000**	0.123±.005	0.008±.001	**0.251±.000**	**0.100±.000**
	Diffusion-TS	0.095±.000	**0.037±.000**	**0.121±.002**	**0.007±.001**	**0.251±.000**	**0.100±.000**
	DiffTime	**0.093±.000**	0.038±.001	**0.121±.004**	0.010±.001	0.252±.000	**0.100±.000**
	Diffwave	**0.093±.000**	0.047±.000	0.130±.001	0.013±.000	**0.251±.000**	0.101±.000
	TimeGAN	**0.093±.019**	0.038±.001	0.124±.001	0.025±.003	0.273±.004	0.126±.002
	TimeVAE	**0.093±.000**	0.039±.000	0.126±.004	0.012±.002	0.292±.000	0.113±.003
	Cot-GAN	0.100±.000	0.047±.001	0.129±.000	0.068±.009	0.259±.000	0.185±.003
	RNN-AR	0.150±.022	0.038±.001	-	-	0.315±.005	-
	Original	0.094±.001	0.036±.001	0.121±.005	0.007±.001	0.250±.003	0.090±.001

Time-series Modeling

Table 2: Time-series modeling in larger model regime.

Metric	Methods	Sines	Stocks	MuJoCo
Context-FID Score ↓	Ours	**0.002±.000**	**0.004±.001**	0.013±.001
	SDFormer-AR	0.008±.001	0.006±.001	**0.008±.000**
	SDFormer-M	0.010±.002	0.034±.008	0.030±.003
	ImagenTime	0.009±.001	0.011±.002	0.017±.002
Discriminative Score ↓	Ours	**0.007±.008**	0.012±.013	**0.009±.009**
	SDFormer-AR	0.016±.010	**0.006±.006**	**0.009±.006**
	SDFormer-M	0.008±.004	0.020±.011	0.025±.007
	ImagenTime	0.016±.010	0.010±.007	0.011±.005
Predictive Score ↓	Ours	**0.093±.000**	**0.037±.000**	0.008±.001
	SDFormer-AR	**0.093±.000**	**0.037±.000**	0.008±.002
	SDFormer-M	**0.093±.000**	**0.037±.000**	**0.007±.001**
	ImagenTime	0.095±.000	**0.037±.000**	0.033±.002

Table 3: Long time-series modeling.

Metric	Methods	FRED-MD	NN5 Daily
Marginal Score ↓	Ours	**0.019±n.a.**	**0.006±n.a.**
	ImagenTime	0.022±n.a.	0.009±n.a.
	LS4	0.022±n.a.	0.007±n.a.
	SaShiMi-AR	0.048±n.a.	0.020±n.a.
Classification Score ↑	Ours	**1.338±.753**	**0.950±.257**
	ImagenTime	0.755±.343	0.560±.174
	LS4	0.544±n.a.	0.636±n.a.
	SaShiMi-AR	0.001±n.a.	0.045±n.a.
Predictive Score ↓	Ours	**0.030±.006**	0.539±.196
	ImagenTime	0.034±.020	0.584±.188
	LS4	0.037±n.a.	**0.241±n.a.**
	SaShiMi-AR	0.232±n.a.	0.849±n.a.

Table 4: Irregular time-series modeling based on Stocks dataset, evaluated with discriminative score ↓.

Task	Methods	0% Drop	30% Drop	50% Drop	70% Drop
Irregular → Regular	Ours	**0.009±.008**	**0.020±.011**	**0.019±.008**	**0.015±.007**
	Koopman VAE	0.021±.022	0.109±.051	0.067±.038	0.049±.052
	GT-GAN	0.077±.031	0.251±.097	0.265±.073	0.230±.053
	TimeGAN	0.102±.021	0.411±.040	0.477±.021	0.485±.022
	RCGAN	0.196±.027	0.436±.064	0.478±.049	0.381±.086
	C-RNN-GAN	0.399±.028	0.500±.000	0.500±.000	0.500±.000
	RNN-AR	0.226±.035	0.305±.002	0.308±.010	0.317±.019
Irregular → Irregular	Ours	**0.009±.008**	**0.049±.017**	**0.044±.017**	**0.138±.137**
	Koopman VAE	0.021±.022	0.227±.096	0.211±.078	0.187±.075

Table 5: Physics-informed modeling of a nonlinear pendulum.

Methods	Corr. Score ↓
Ours w/ stability loss	**0.0005±.0004**
KoVAE w/ stability loss	0.0030±.0004
Ours w/o stability loss	**0.0029±.0008**
KoVAE w/o stability loss	0.0040±.0005

PyTorch Implementation Available

github.com/jw9730/spectral-mean-flow