Recurrent Neural Network, Tensorization, and Chaotic Time Series Forecasting

Xiangyi Meng

Department of Physics, Boston University

Motivation

Why time series forecasting?

Mechanical analysis, fluid dynamics, traffic, weather, finance, …

all **relevant**.

How?

Classical:

1) Auto-regressive moving average (ARMA)
2) Hidden Markov models (HMM)

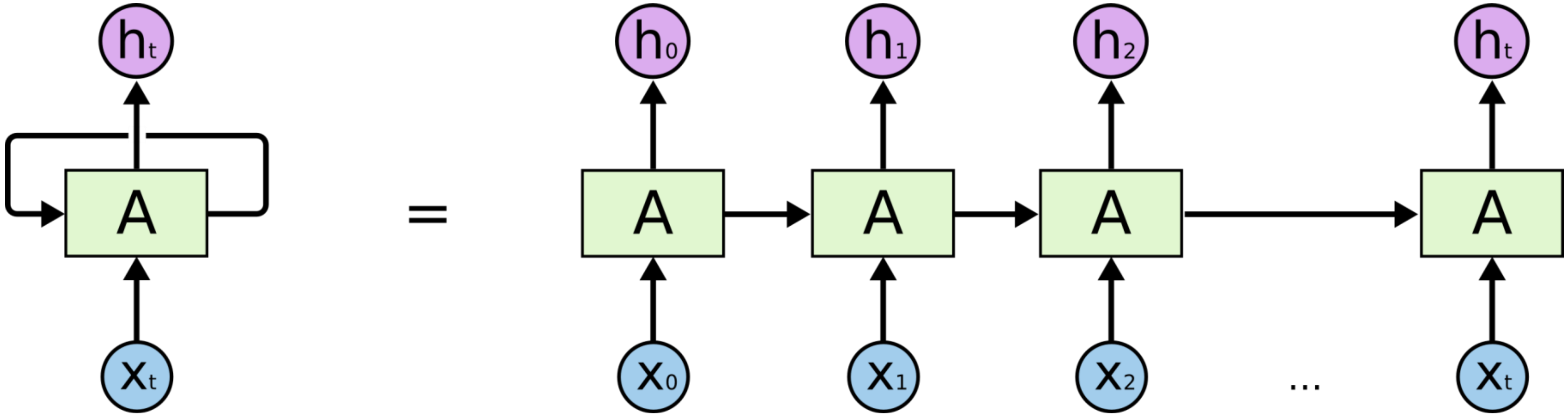
Machine Learning (ML):

1) Gradient boosted trees (GBT)
2) **Neural networks (NN)**

Motivation

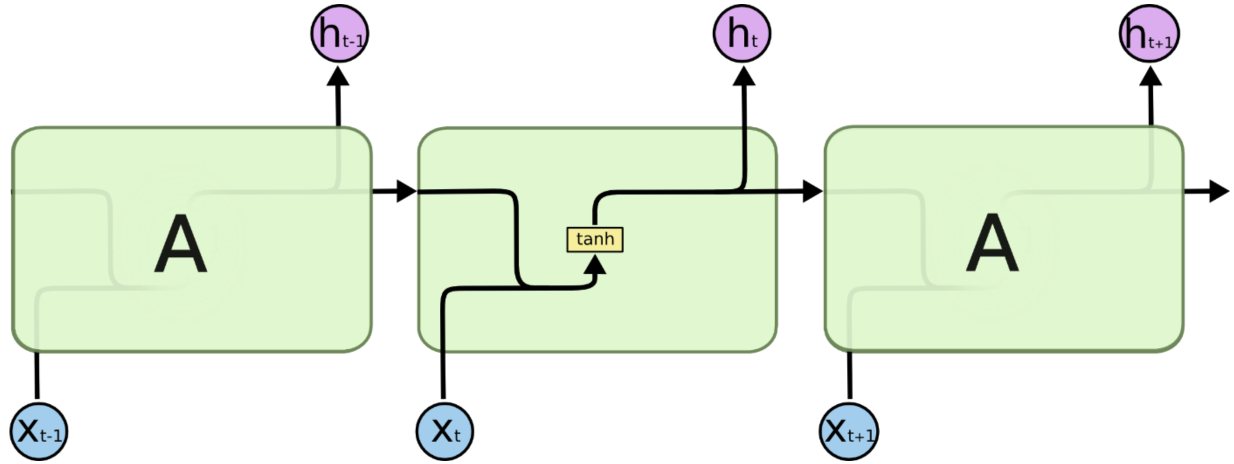
Time series data is best learned by a **recurrent NN architecture.**

The recurrent behavior of time series is **fundamental**.

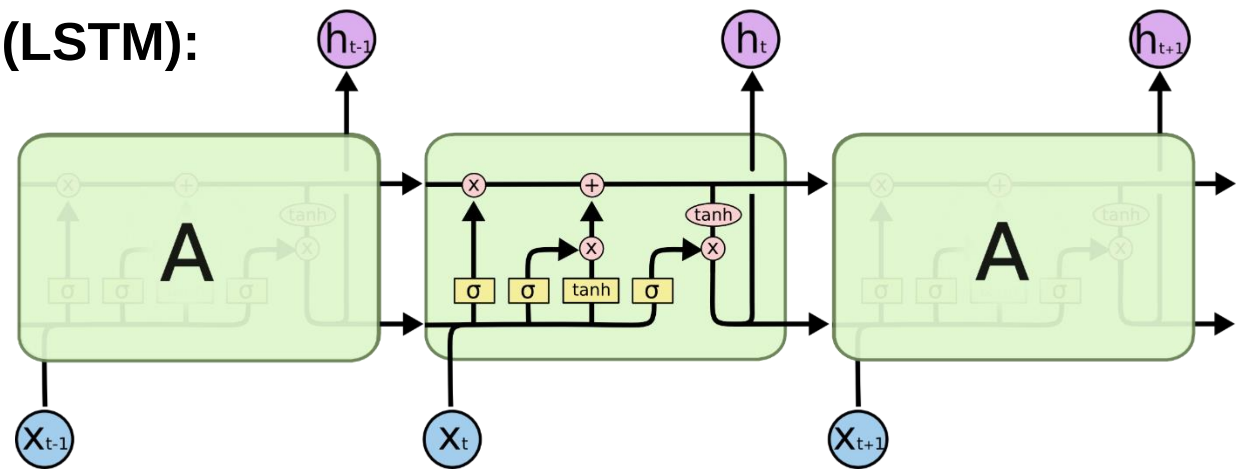


Recurrent NN Architectures

Vanilla RNN:



Long Short-Term Memory (LSTM):

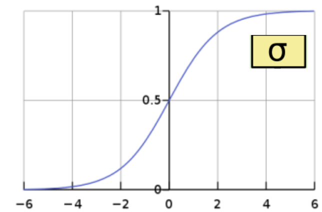


LSTM is the **gold standard**: speech recognition, video tasks, e-sports…

LSTM

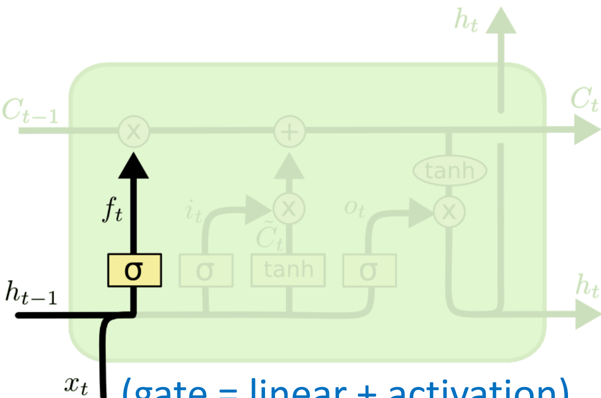
Definition

Why is LSTM useful?



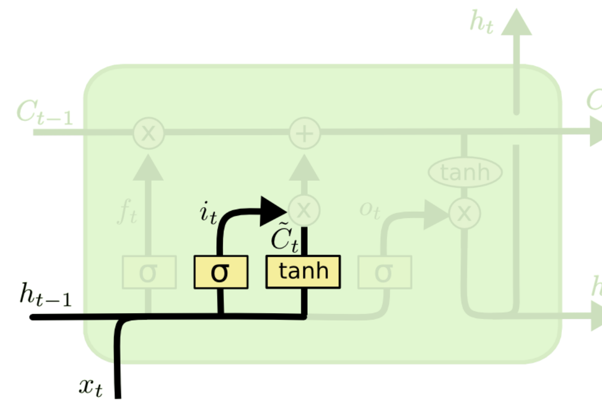
Cell state

State



(gate = linear + activation)

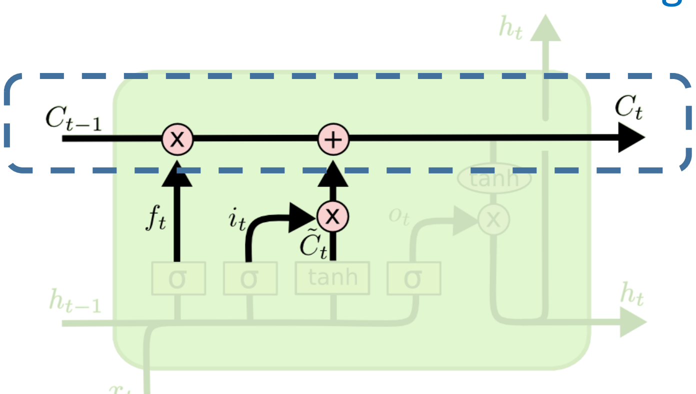
$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \ + \ b_f\right)$$



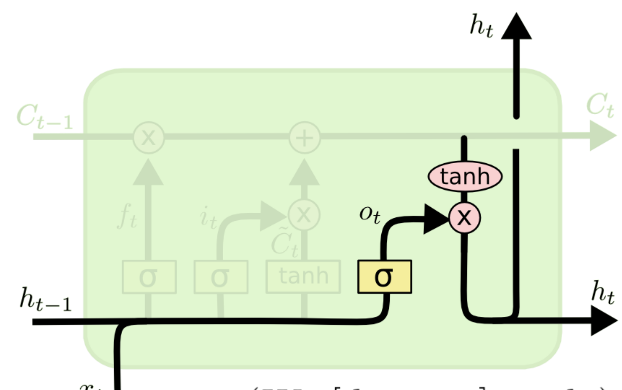
$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \ + \ b_i\right)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \ + \ b_C)$$

No activation function = nonvanishing gradient!



$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$



$$o_t = \sigma\left(W_o \ [h_{t-1}, x_t] \ + \ b_o\right)$$

$$h_t = o_t * \tanh\left(C_t\right)$$

Chaos Forecasting

Difficulties

LSTM is specialized in learning non-Markovian time series

(i.e., **long-term memory**)

What about chaos forecasting?

Chaos is governed by **short-term nonlinear** dynamics.

$$|\delta x_t| \approx \mathrm{e}^{\lambda t}|\delta x_0|$$

The difficulties are two-fold:

a) Small error propagates exponentially -> multiple-step-ahead predictions will be **exponentially worse** than one-step-ahead ones.

b) (more subtly) When Δt (time step of discretization) increases -> the minimum redundancy needed for smoothly descending to the global minima also **increases exponentially**.

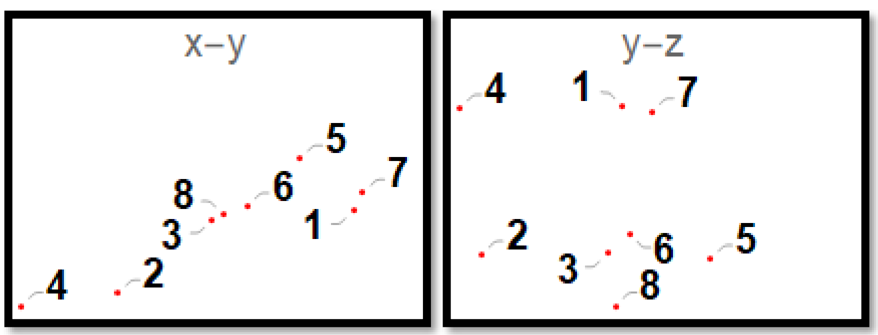
Chaos Forecasting

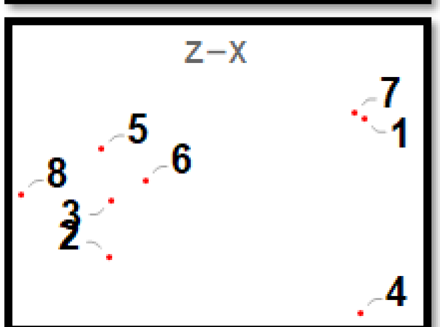
Difficulties

$$|\delta x_t| \approx \mathrm{e}^{\lambda t|\delta x_0|}$$

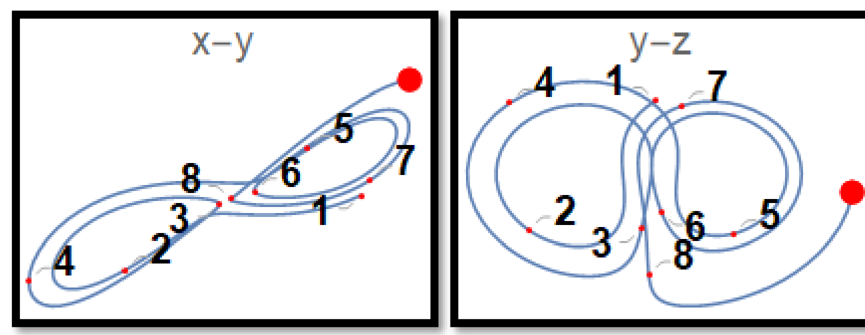
When Δt (or equivalently, λ) is large, difficulty (b) is more crucial -> usually **a trivial, ergodic local minimum** would most likely be reached instead.

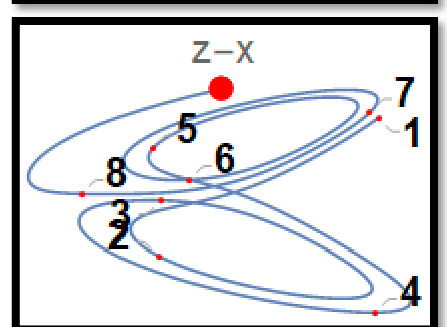
…which is what we focus on (unlike traditional studies)





Can you predict where the 9-th point is located?





Lorenz system!

Tensorization

Benefits

How to add nonlinearity into the NN architecture?

Tensorization: vectors, matrices -> higher-order tensor products.

e.g., x_μ -> $\mathcal{T}_{\mu\nu\xi\eta\ldots} = x_\mu \otimes x_\nu \otimes x_\xi \otimes x_\eta \ldots$

Benefits of tensorization:

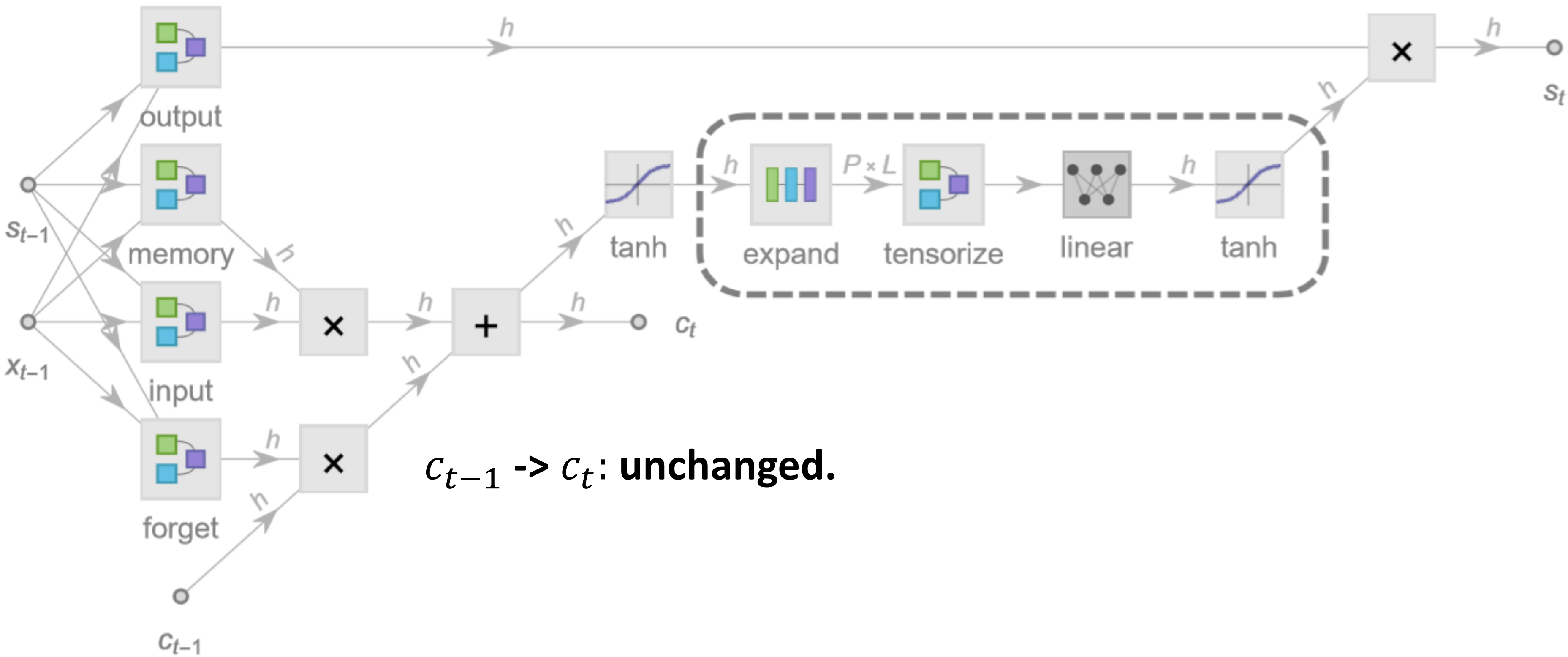
1) Introducing equally weighted nonlinear terms (by tensor products).
2) Theoretical analysis (polynomial functional space).
3) Experimentally tested in different recurrent NN architectures [1-2, etc.].

[1] Schlag, I. & Schmidhuber, J. Learning to Reason with Third Order Tensor Products. in *Proceedings of Neural Information Processing Systems 2018,* vol. 31 9981–9993.

[2] Yang, Y., Krompass, D. & Tresp, V. Tensor-Train Recurrent Neural Networks for Video Classification. in *Proceedings of the 34th International Conference on Machine Learning,* vol. 70 3891–3900 (PMLR, 2017).

Tensorization

On LSTM

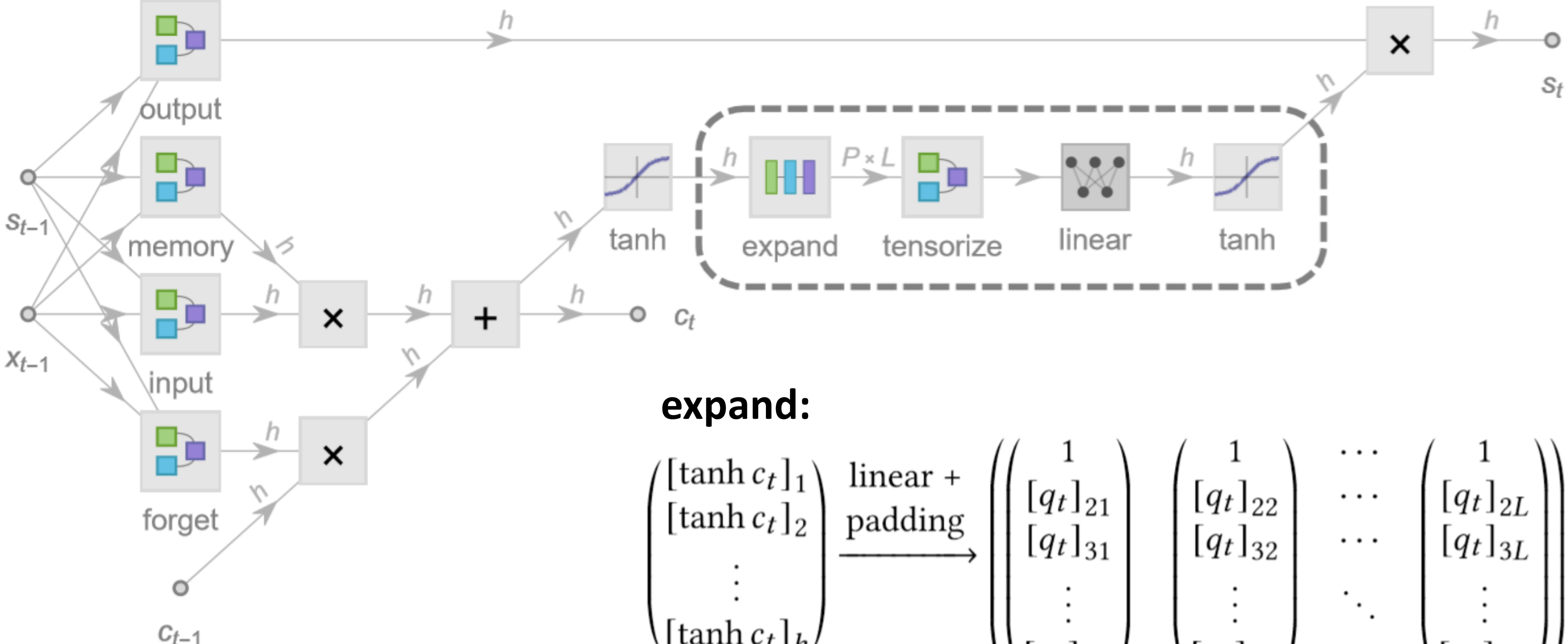


c_{t-1} **->** c_t**: unchanged.**

We introduce a new LSTM-based recurrent architecture, keeping the **long-term memory feature of LSTM** while simultaneously enhancing the learning of **short-term nonlinear complexity.**

Tensorization

On LSTM



expand:

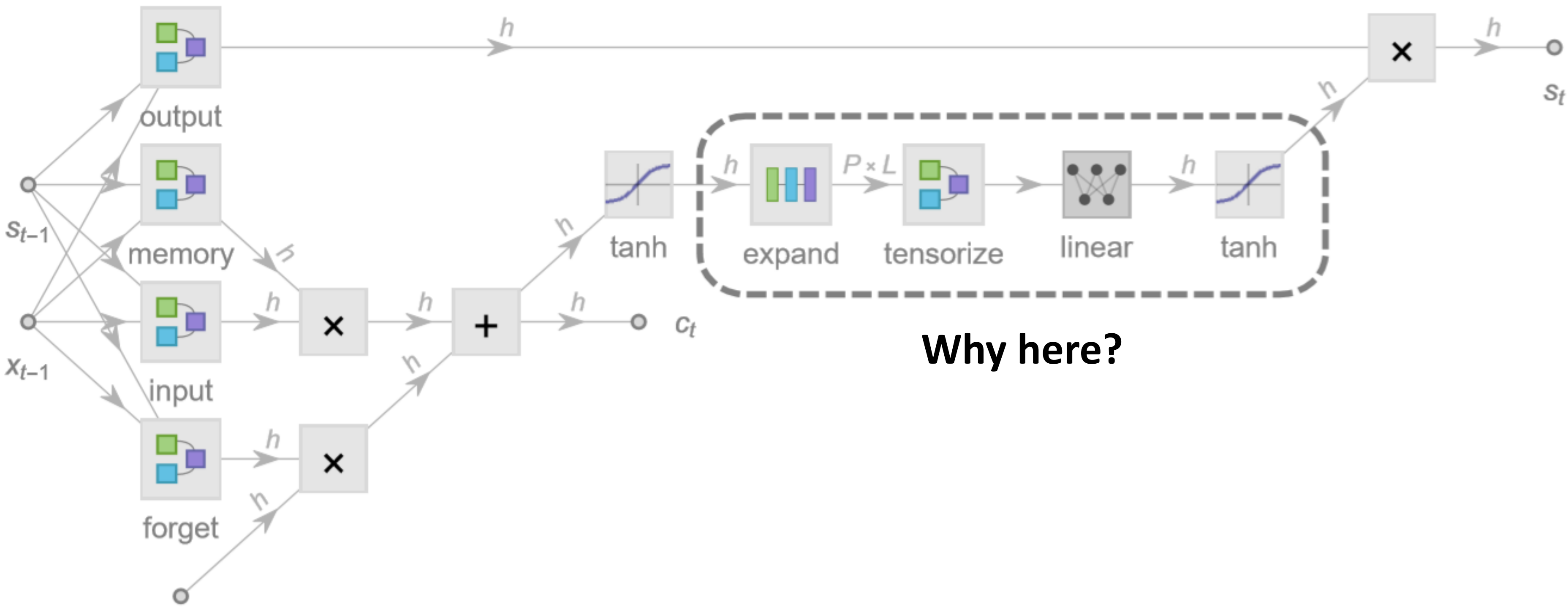
$$\begin{pmatrix} [\tanh c_t]_1 \\ [\tanh c_t]_2 \\ \vdots \\ [\tanh c_t]_h \end{pmatrix} \xrightarrow{\substack{\text{linear +} \\ \text{padding}}} \left(\begin{pmatrix} 1 \\ [q_t]_{21} \\ [q_t]_{31} \\ \vdots \\ [q_t]_{P1} \end{pmatrix} \begin{pmatrix} 1 \\ [q_t]_{22} \\ [q_t]_{32} \\ \vdots \\ [q_t]_{P2} \end{pmatrix} \begin{matrix} \cdots \\ \cdots \\ \cdots \\ \ddots \\ \cdots \end{matrix} \begin{pmatrix} 1 \\ [q_t]_{2L} \\ [q_t]_{3L} \\ \vdots \\ [q_t]_{PL} \end{pmatrix} \right),$$

tensorize: $\mathcal{T}(\tanh c_t) =$ tensor product of all column vectors above.
(Degrees of freedom = P^L. Here h, P, L are the dimensions.)

(When $L \to \infty$, $\mathcal{T}(\tanh c_t) \in \mathbb{T} = (1 \oplus q)^{\otimes L}$ which becomes a tensor algebra.)

Tensorization

Theoretical Analysis



Why here?

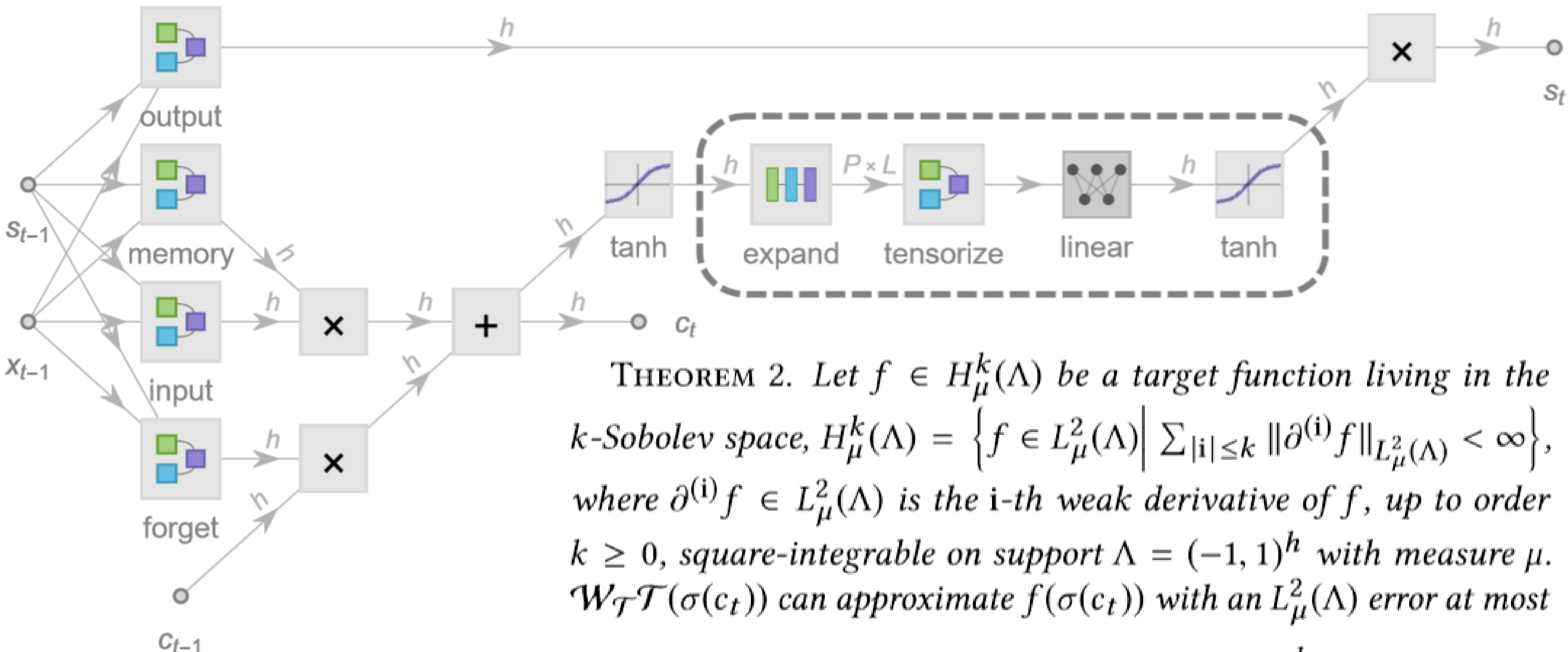
Lemma 1: *Given a chaotic dynamical system x_t characterized by a matrix λ of which the spectrum is the Lyapunov exponent(s), then, up to the first order (δx_{t-1}),*

$$|\delta s_t| \geq C e^{\lambda} |\delta c_t|.$$ Proof: *(derivative propagation)*

The c_t -> s_t path governs the chaotic behavior of x_t.

Tensorization

Theoretical Analysis



THEOREM 2. *Let $f \in H^k_\mu(\Lambda)$ be a target function living in the k-Sobolev space, $H^k_\mu(\Lambda) = \left\{ f \in L^2_\mu(\Lambda) \middle| \sum_{|\mathrm{i}| \le k} \|\partial^{(\mathrm{i})} f\|_{L^2_\mu(\Lambda)} < \infty \right\}$, where $\partial^{(\mathrm{i})} f \in L^2_\mu(\Lambda)$ is the* i*-th weak derivative of f, up to order $k \ge 0$, square-integrable on support $\Lambda = (-1, 1)^h$ with measure μ. $\mathcal{W}_{\mathcal{T}}\mathcal{T}(\sigma(c_t))$ can approximate $f(\sigma(c_t))$ with an $L^2_\mu(\Lambda)$ error at most*

$$\|f - \mathcal{W}_{\mathcal{T}}\mathcal{T}\|_{L^2_\mu(\Lambda)} \le C \min(L, \lfloor L(P-1)/h \rfloor)^{-k} \|f\|_{H^k_\mu(\Lambda)} \qquad (7)$$

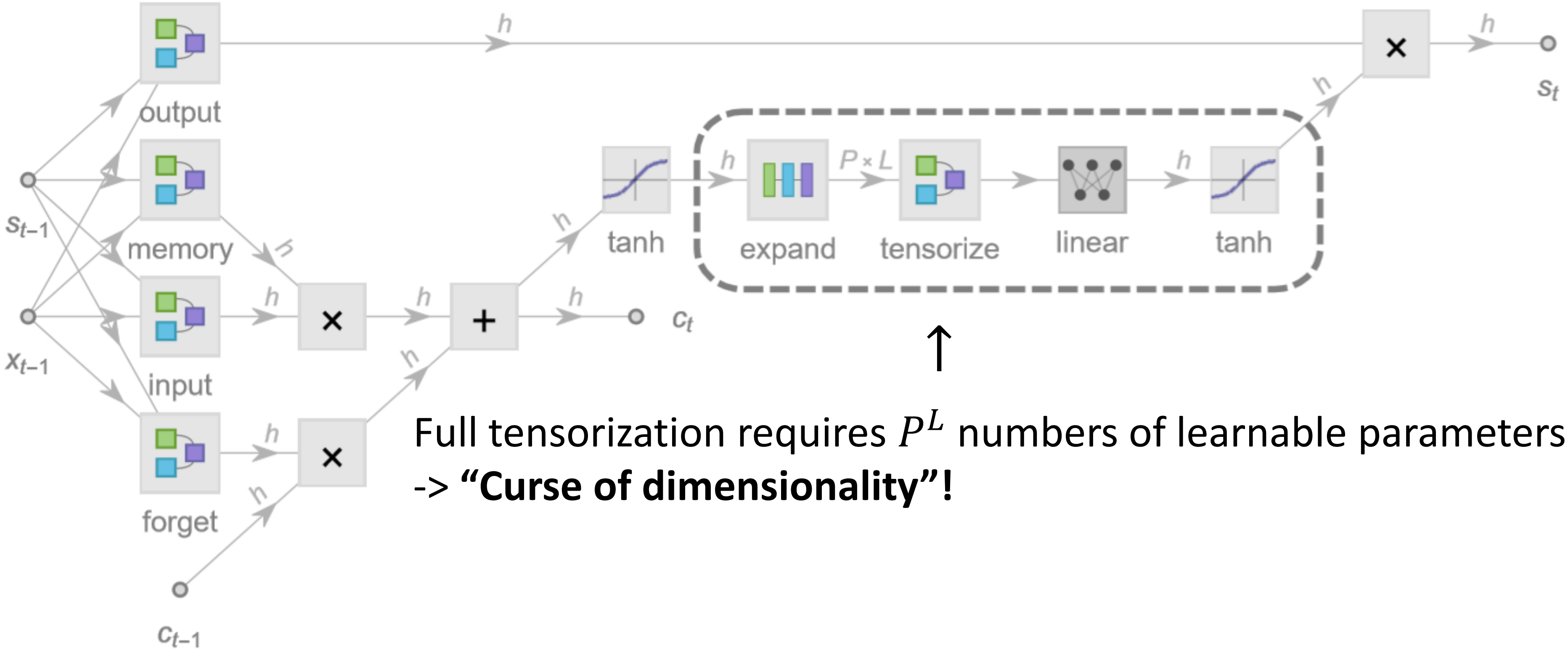
provided that $(h-1)hP^L \ge (h^{1+\min(L, \lfloor L(P-1)/h \rfloor)} - 1)$. $\|f\|_{H^k_\mu(\Lambda)} = \sum_{|\mathrm{i}| \le k} \|\partial^{(\mathrm{i})} f\|_{L^2_\mu(\Lambda)}$ is the Sobolev norm and C a finite constant.

Proof: *(The Hölder-continuous spectral convergence theorem for polynomial space).*

The expressive power of s_t (as a function of c_t) is guaranteed by the derivatives of s_t over c_t, which is of the order of $\sim e^{\lambda}$ (from Lemma 1).

Tensorization

Curse of Dimensionality



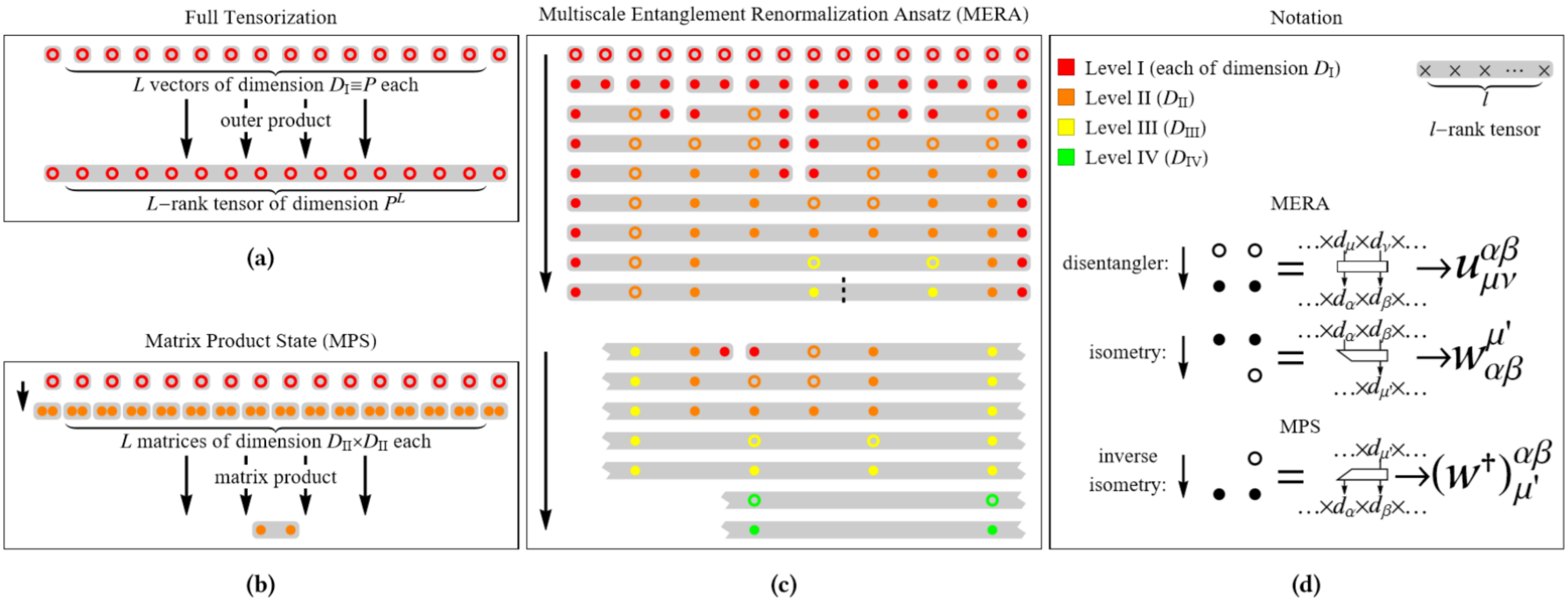
Full tensorization requires P^L numbers of learnable parameters.
-> **“Curse of dimensionality”!**

Luckily, two tensor representations have been widely used for reducing the degrees of freedom in the literature of condensed matter physics:

1) Matrix product state (MPS).
2) Multiscale entanglement renormalization ansatz (MERA).

Tensorization

MPS vs MERA



* The MERA algorithm is re-arranged so that the memory storage of arrays is minimally used -> essential for GPU training.

Tensorization

MPS vs MERA

How well can MPS/MERA approximate the full tensor? **It depends.**

We can characterize it by α-Rényi entropy in an analogous way:

THEOREM 3. *Given a tensor* $[W_{\mathcal{T}}]_{\mu_1 \cdots \mu_L}$ *and its tensor decomposition* $\overline{W}_{\mathcal{T}}$, *the worst-case* p*-norm* ($p \geq 1$) *approximation error is bounded from below by*

$$\min_{\{\overline{W}_{\mathcal{T}}\}} \max_{l \geq 1} \|W_{\mathcal{T}}(l) - \overline{W}_{\mathcal{T}}(l)\|_p$$

$$\geq \min_{\{\overline{W}_{\mathcal{T}}\}} \max_{l \geq 1} \left| e^{\frac{1-p}{p} S_p(W_{\mathcal{T}}(l))} \|W_{\mathcal{T}}(l)\|_1 - e^{\frac{1-p}{p} S_p(\overline{W}_{\mathcal{T}}(l))} \|\overline{W}_{\mathcal{T}}(l)\|_1 \right|, \tag{8}$$

where $S_{\alpha \equiv p}(W(l))$ *is the* α*-Rényi entropy [Eq. (5)].*

PROOF. Equation (8) is easily proved by noting the Minkowski inequality $\|A + B\|_p \leq \|A\|_p + \|B\|_p$ and that $(1 - \alpha) S_\alpha(l) = \alpha \log \|W_{\mathcal{T}}(l)\|_\alpha - \alpha \log \|W_{\mathcal{T}}(l)\|_1$ when $\alpha \equiv p \geq 1$ [Eq. (5)]. □

MPS is simpler when $S_p(W_{\mathcal{T}}(l))$ does not change with l;

MERA is better when $S_p(W_{\mathcal{T}}(l))$ scales with $\omega(\ln l)$.

Results

Settings

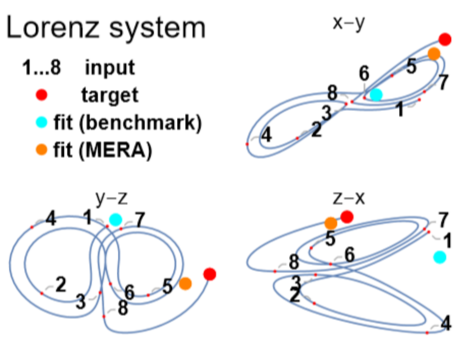
We investigate the accuracy of LSTM-MPS/MERA by evaluating the root mean squared error (RMSE) of its one-step-ahead predictions.

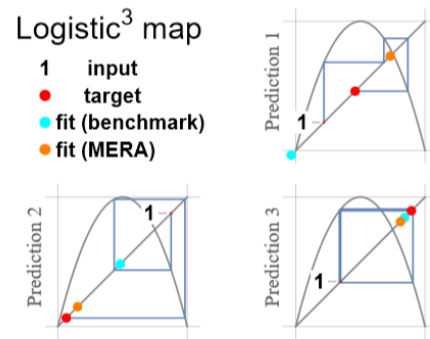
All models were (proudly) trained by Mathematica 12.0 on its NN infrastructure, Apache MXNet, using an ADAM optimizer.

Learning rate = 10^−2 and batch size = 64 were *a priori* chosen.

Results

Comparison of LSTM-Based Architectures





LSTM	# of param.	h	L	P	$\{D_{\rm I}, D_{\rm II}, \cdots\}$	RMSE
Benchmark	332	7	–	–	–	0.307
"Wider"	696	11	–	–	–	0.279
"Deeper"	640	7	–	–	–	0.105
MPS	663	7	2^3	2	$\{P, 4\}$	0.088
MERA	**640**	**7**	$\mathbf{2^3}$	**2**	$\{\boldsymbol{P}, \mathbf{2}, \mathbf{3}\}$	**0.066**

LSTM	# of param.	h	L	P	$\{D_{\rm I}, D_{\rm II}, \cdots\}$	RMSE
Benchmark	35	2	–	–	–	0.259
"Wider"	1169	16	–	–	–	0.187
"Deeper"	1156	11	–	–	–	0.204
MPS	1231	2	2^3	2	$\{P, 9\}$	0.181
MERA	**1053**	**2**	$\mathbf{2^3}$	**2**	$\{\boldsymbol{P}, \mathbf{4}, \mathbf{4}\}$	**0.010**

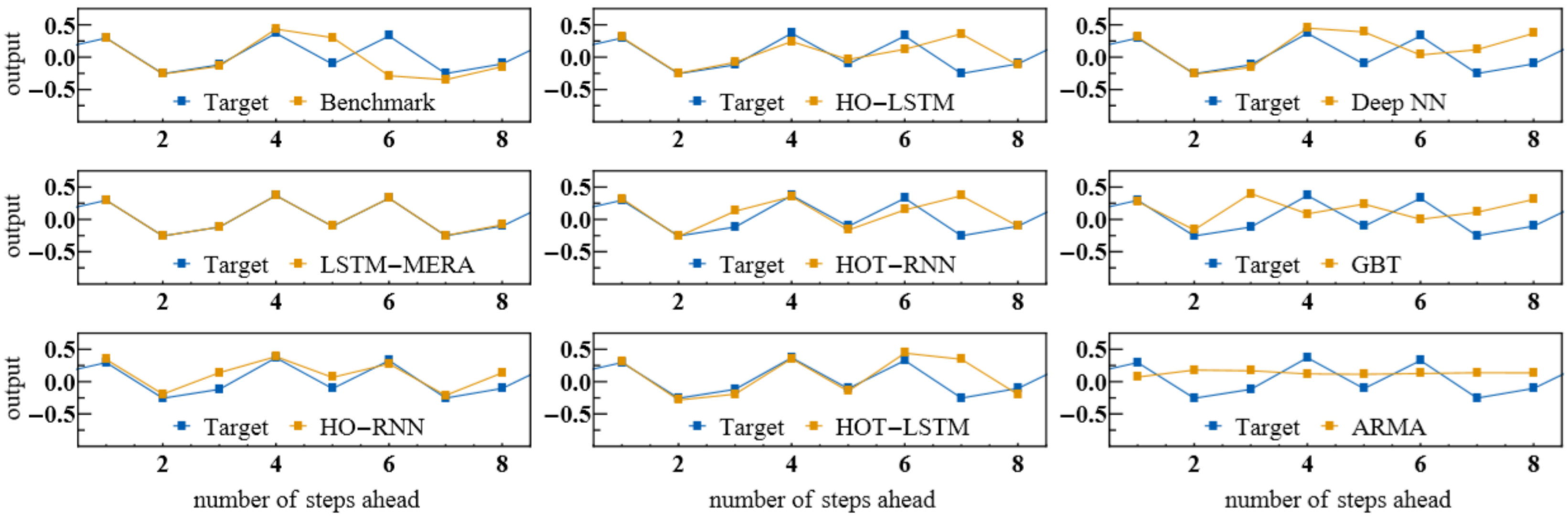
Tensorized LSTM performed better;

LSTM-MERA performed even better than LSTM-MPS.

Results

Comparison with Statistical/ML Models

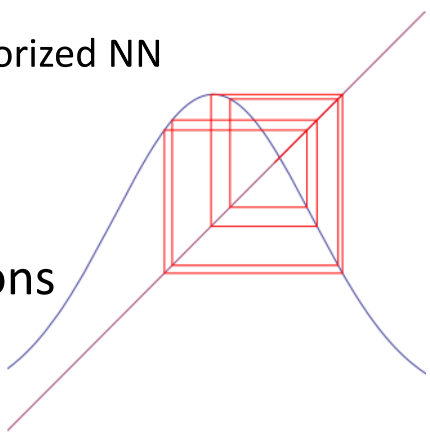
Gauss "cubed" map, i.e., a Gauss iterated map sampled every three steps



Model (n steps ahead)	# of param.	RMSE ($\times 10^{-2}$) 1 step	2 steps	4 steps
Benchmark	35	1.54	7.63	32.03
LSTM-MERA	**89**	**0.19**	**0.89**	13.77
HO-RNN	23	11.91	23.16	27.69
HO-LSTM	83	2.96	14.50	47.99
HOT-RNN	81	12.04	23.76	29.61
HOT-LSTM	315	1.39	6.40	26.83
Deep NN	17950	0.81	3.66	23.49
GBT	–	3.15	16.25	31.37
ARMA	–	24.95	24.93	23.54

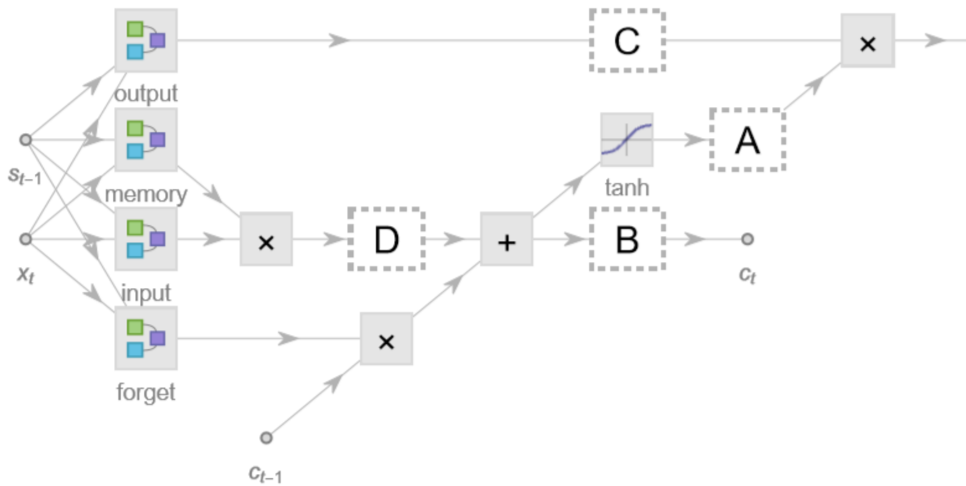
Benchmark: vanilla LSTM
HO/HOT-RNN/LSTM: traditional tensorized NN architectures.
GBT: gradient boosted trees.

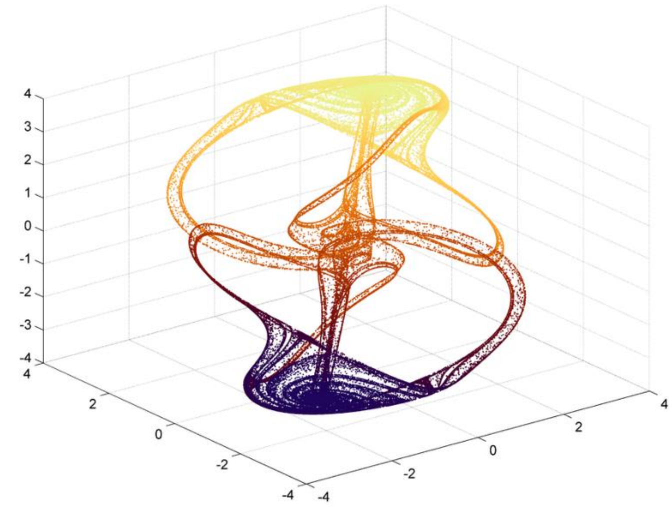
Multiple-step-ahead predictions are exponentially worse.



Results

Comparison with LSTM-MERA Alternatives



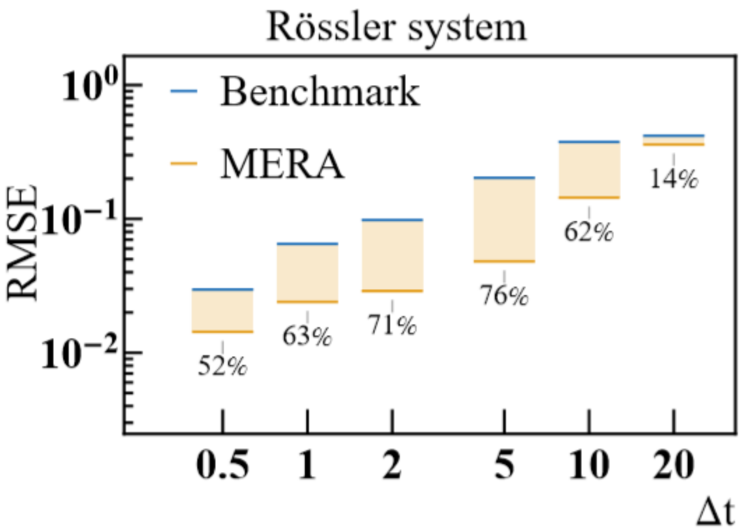


Thomas' cyclically symmetric dynamical system		
Model	Site	RMSE ($\times 10^{-1}$)
Benchmark		1.13
LSTM-MERA	**A**	**0.45**
Alternatives	B	1.12
	C	1.10
	D	0.73

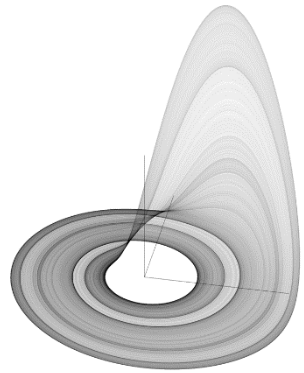
The c_t -> s_t path governs the chaotic behavior of x_t.

Results

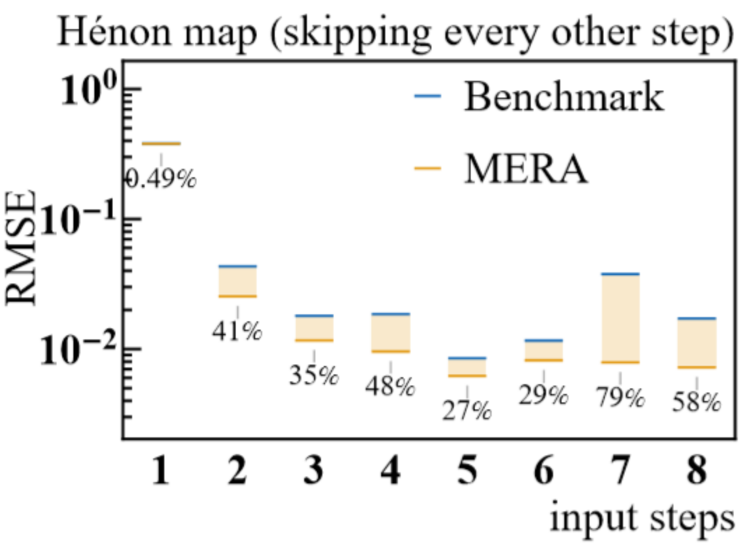
Generalization and Parameter Dependence of LSTM-MERA



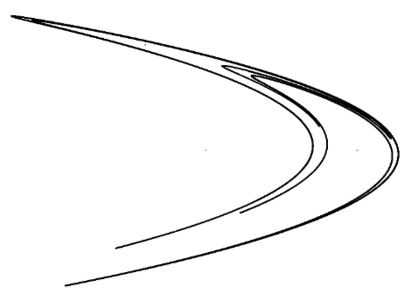
(a)



When Δt is too large, it's difficult for LSTM-MERA to reach the global minimum too.



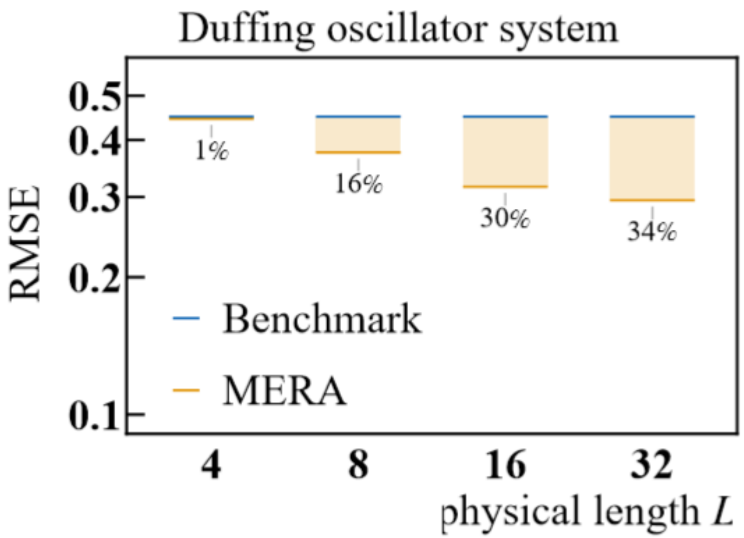
(b)



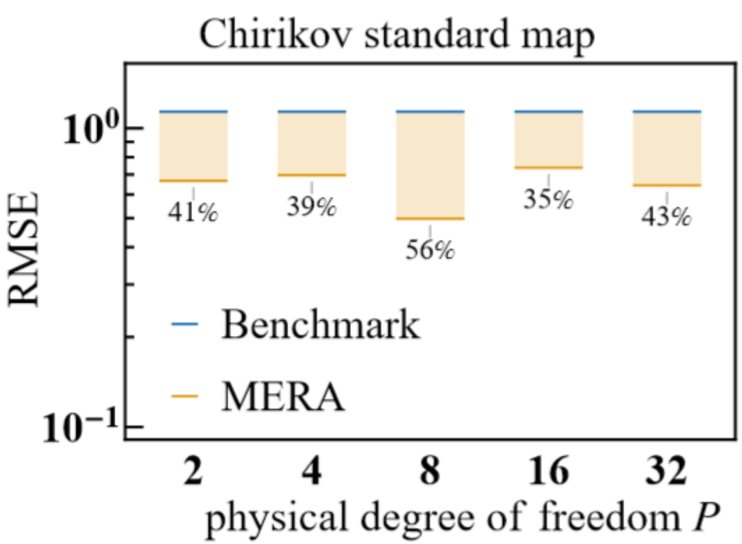
LSTM-MERA still performed better when partial historical information was given.

Results

Generalization and Parameter Dependence of LSTM-MERA



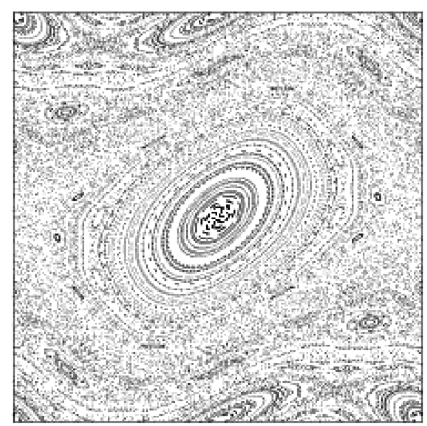
(c)



(d)

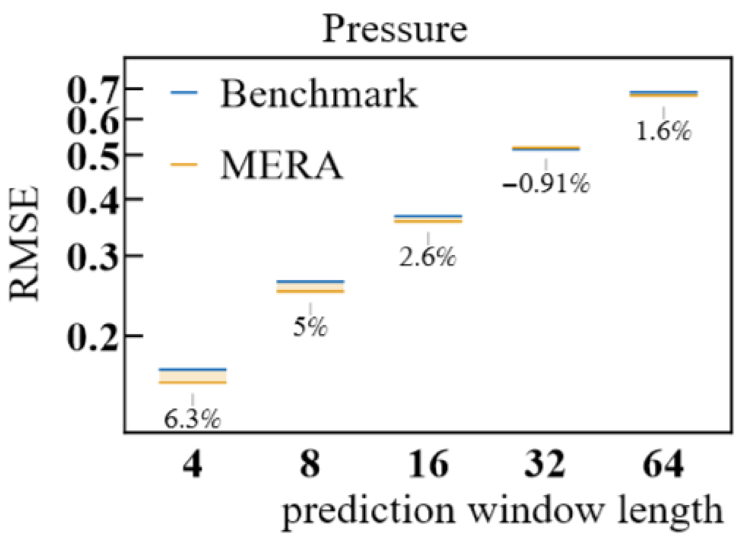
The performance increases with L;

No simple dependence on P.

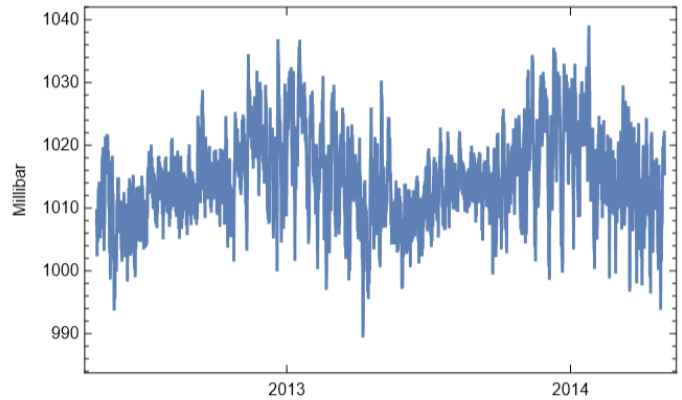


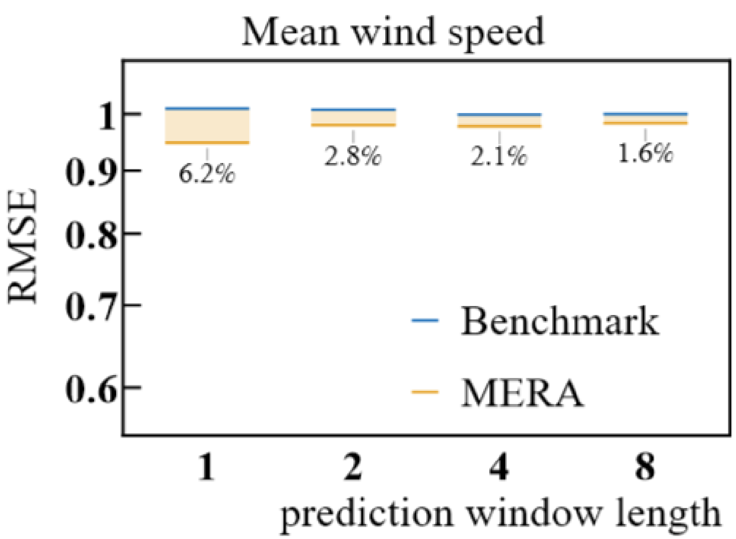
Results

Generalization and Parameter Dependence of LSTM-MERA

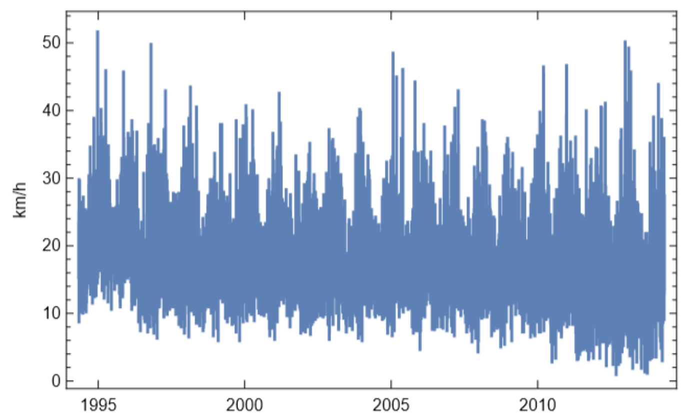


(e)





(f)



LSTM-MERA is less significant.

(Because LSTM itself is already good at learning long-term memory.)

Summary

Limitation:

□ Our model is only better than traditional LSTM at capturing short-term nonlinearity but not long-term non-Markovianity.

Advantage:

□ The LSTM long-term feature is preserved.

□ Tensorization introduces nonlinear terms, suitable for chaos forecasting.

□ Theoretical analysis is conductible.

□ Tensor decomposition techniques are available.

Thank you

References:

Some pictures are from Wikipedia and https://colah.github.io/posts/2015-08-Understanding-LSTMs/.