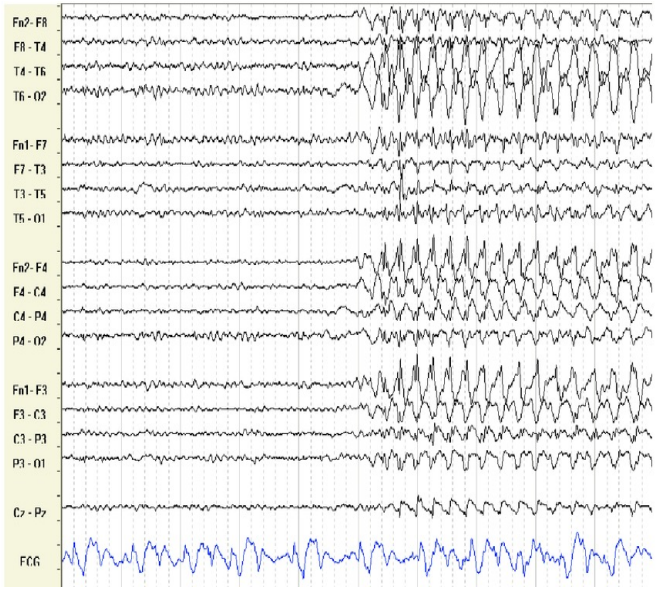
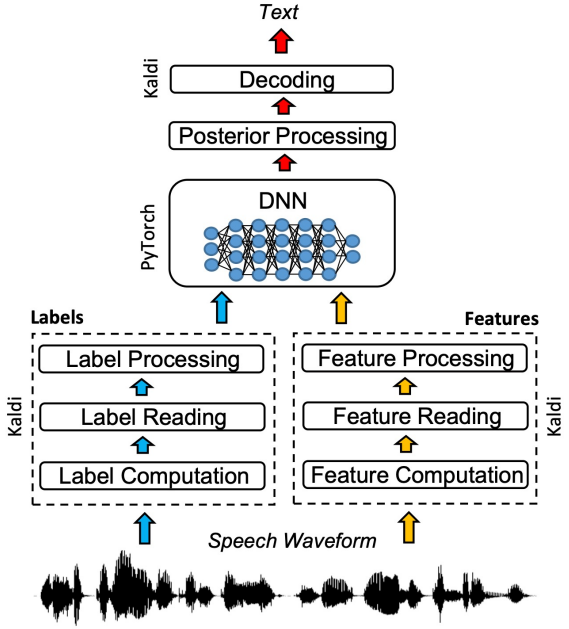
Efficiently Modeling Long Sequences with Structured State Spaces

Albert Gu

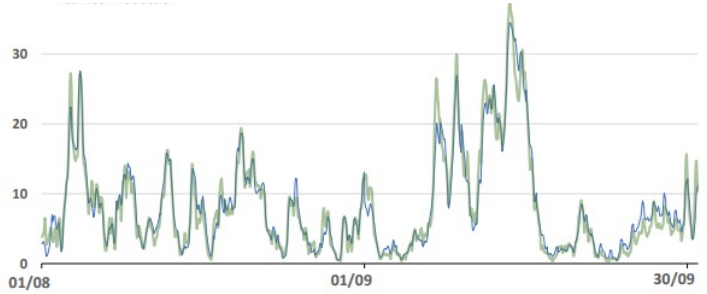
Long "Continuous" Time Series



EEG/ECG



Speech

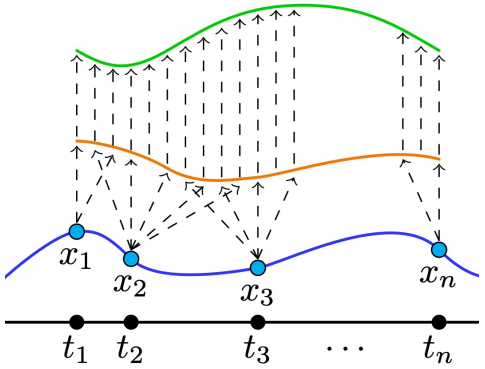


Energy Forecasting

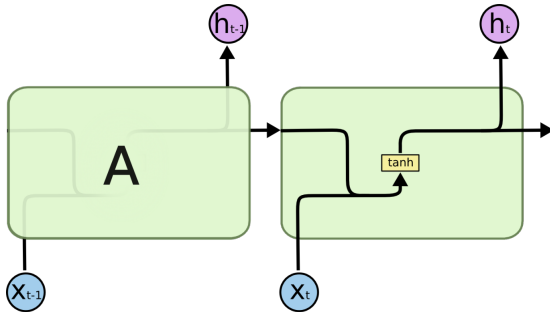
Time series sampled from an underlying (continuous) physical process

Sequence Modeling Paradigms

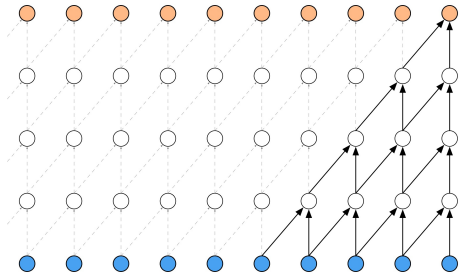
Continuous-time Model



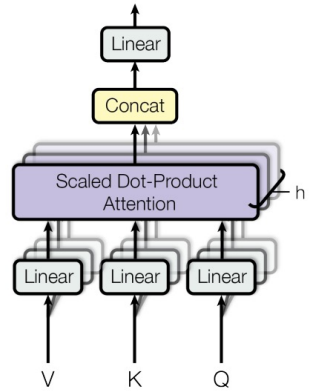
Recurrent Neural Net.



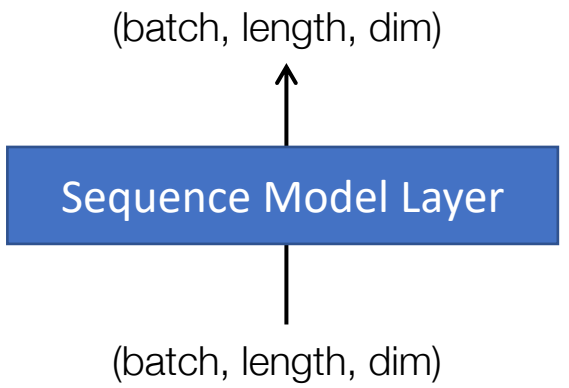
Convolutional Neural Net.



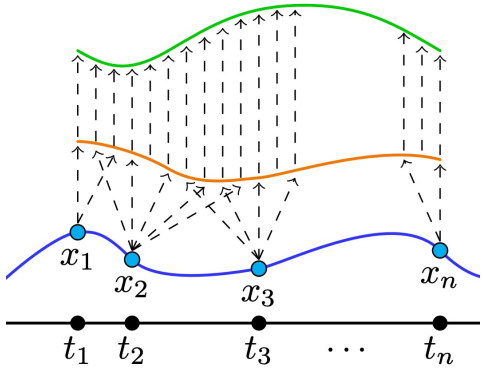
Transformer



Deep Sequence Model
Sequence-to-sequence map



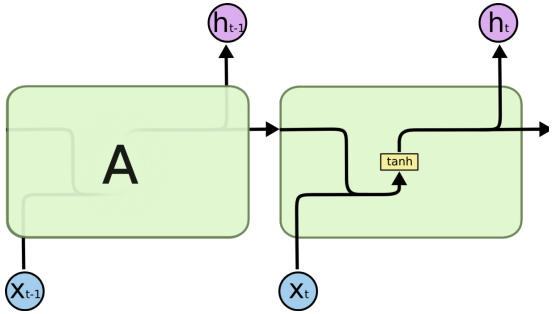
Paradigms for Long Time Series



✓ Continuous data
Irregular sampling

✗ Complex, very inefficient
Vanishing gradients

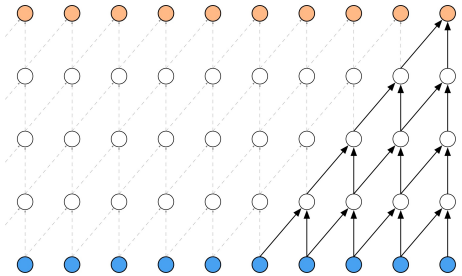
Continuous-time (CTM)



✓ Unbounded context
Stateful inference

✗ Inefficient training
Vanishing gradients

Recurrent (RNN)



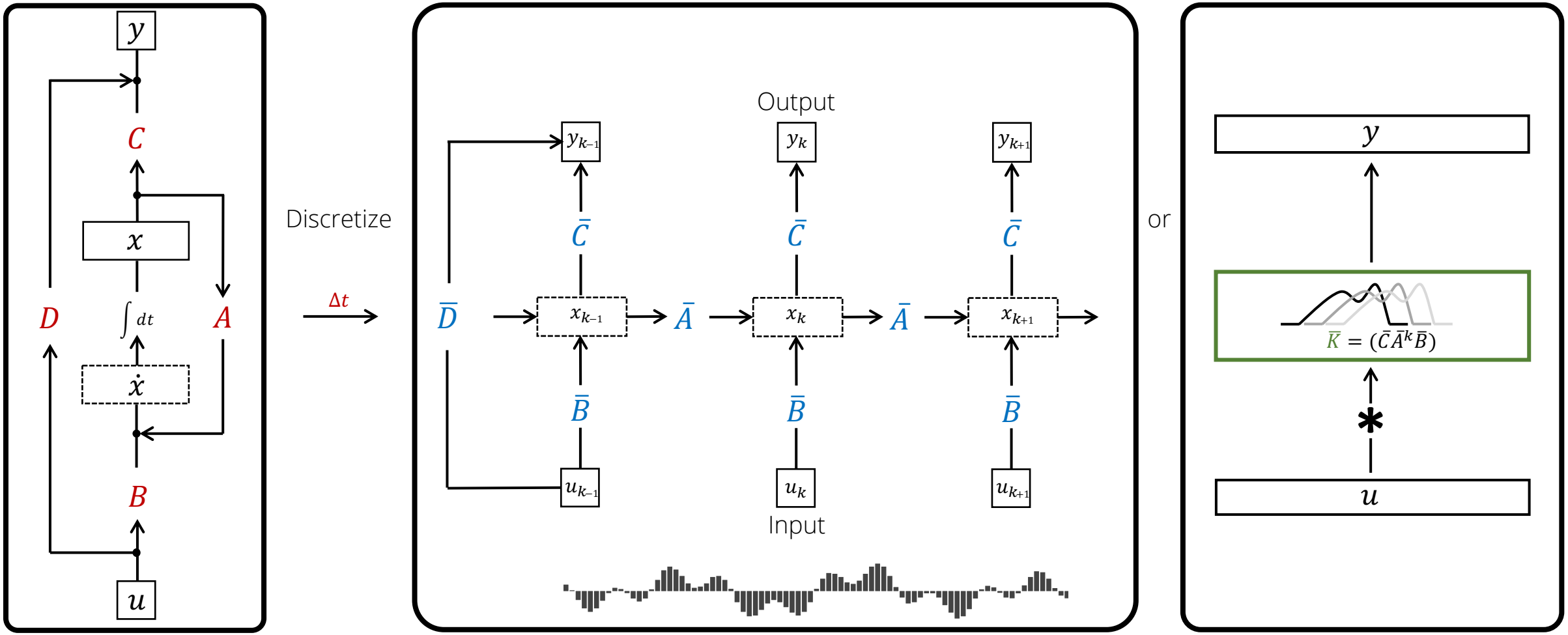
✓ Easy optimization
Parallelizable training

✗ Inefficient inference
Bounded context

Convolutional (CNN)

Existing model families have clear tradeoffs
All struggle with long-range dependencies (LRD)

Structured State Spaces (S4)



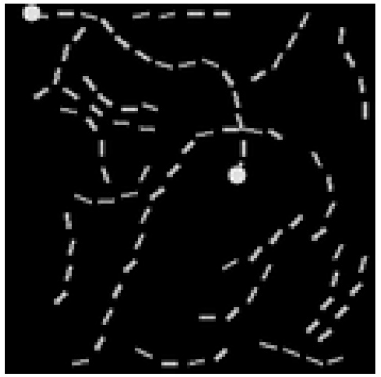
Continuous-time Recurrent Convolutional

Long Range Arena

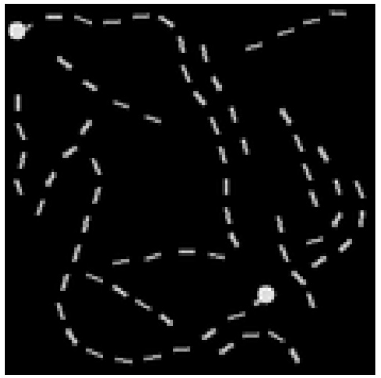
Benchmark spanning text, images, symbolic reasoning (length 1K-16K)

Model	ListOps	Text	Retrieval	Image	Pathfinder	Path-X	Avg
Random	10.00	50.00	50.00	10.00	50.00	50.00	36.67
Transformer	36.37	64.27	57.46	42.44	71.40	✗	53.66
Local Attention	15.82	52.98	53.39	41.46	66.63	✗	46.71
Sparse Trans.	17.07	63.58	59.59	44.24	71.71	✗	51.03
Longformer	35.63	62.85	56.89	42.22	69.71	✗	52.88
Linformer	35.70	53.94	52.27	38.56	76.34	✗	51.14
Reformer	37.27	56.10	53.40	38.07	68.50	✗	50.56
Sinkhorn Trans.	33.67	61.20	53.83	41.23	67.45	✗	51.23
Synthesizer	36.99	61.68	54.67	41.61	69.45	✗	52.40
BigBird	36.05	64.02	59.29	40.83	74.87	✗	54.17
Linear Trans.	16.13	65.90	53.09	42.34	75.30	✗	50.46
Performer	18.01	65.40	53.82	42.77	77.05	✗	51.18
FNet	35.33	65.11	59.61	38.67	77.80	✗	54.42
Nyströmformer	37.15	65.52	79.56	41.58	70.94	✗	57.46
Luna-256	37.25	64.57	79.29	47.38	77.72	✗	59.37
S4	**58.35**	**76.02**	**87.09**	**87.26**	**86.05**	**88.10**	**80.48**

Path-X



(a) A positive example.



(b) A negative example.

Outline

- State space models (SSM) for deep sequence modeling
- Structured state spaces (S4) for long-term dependencies
- Solving LRDs in practice

Outline

- State space models (SSM) for deep sequence modeling
- Structured state spaces (S4) for long-term dependencies
- Solving LRDs in practice

State Space Models (SSM)

Input → State

$$x'(t) = \boldsymbol{A}x(t) + \boldsymbol{B}u(t)$$

$$y(t) = \boldsymbol{C}x(t) + \boldsymbol{D}u(t)$$

Parameters

$\boldsymbol{A} \in \mathbb{R}^{N \times N}$

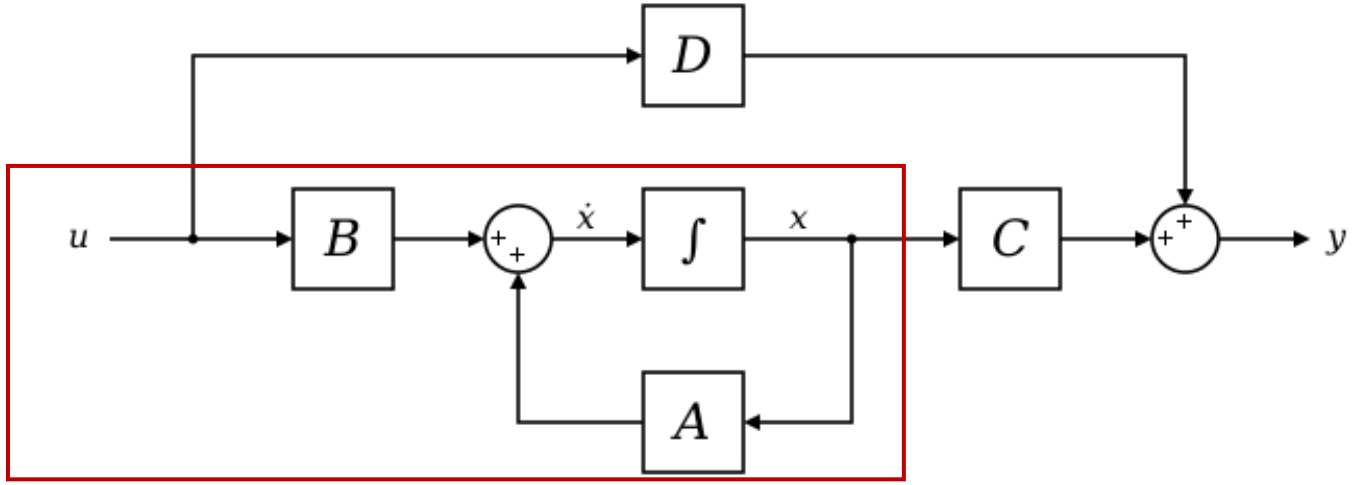
$\boldsymbol{B} \in \mathbb{R}^{N \times 1}$

$\boldsymbol{C} \in \mathbb{R}^{1 \times N}$

$\boldsymbol{D} \in \mathbb{R}^{1 \times 1}$

Function-to-function map $u(t) \mapsto y(t)$

1D input
u(t)



1D output
y(t)

State Space Models (SSM)

$$x'(t) = \boldsymbol{A}x(t) + \boldsymbol{B}u(t)$$

State → Output

$$y(t) = \boldsymbol{C}x(t) + \boldsymbol{D}u(t)$$

Parameters

$\boldsymbol{A} \in \mathbb{R}^{N \times N}$

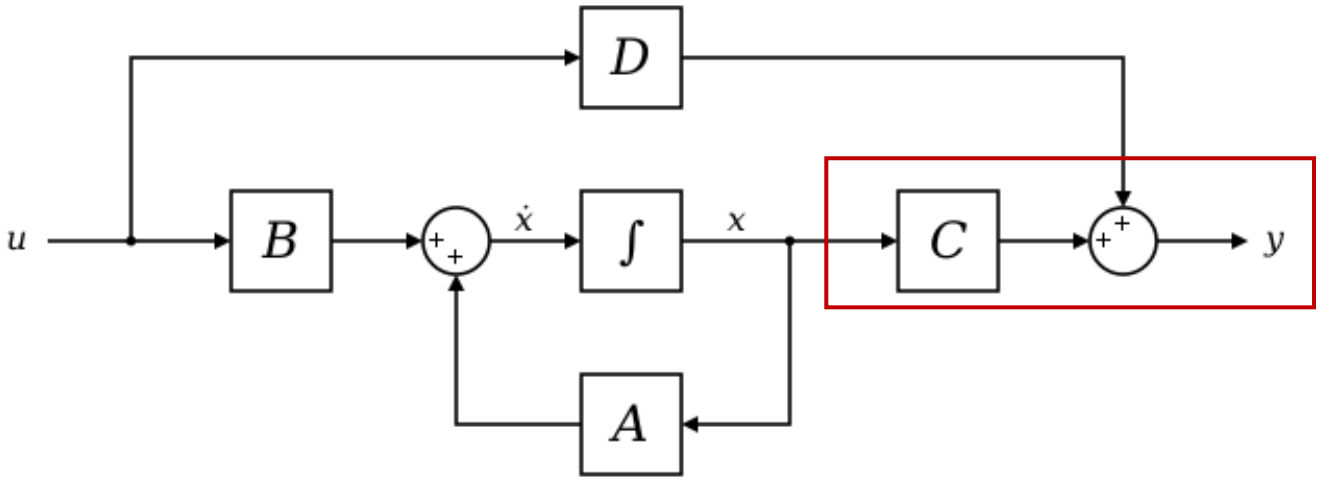
$\boldsymbol{B} \in \mathbb{R}^{N \times 1}$

$\boldsymbol{C} \in \mathbb{R}^{1 \times N}$

$\boldsymbol{D} \in \mathbb{R}^{1 \times 1}$

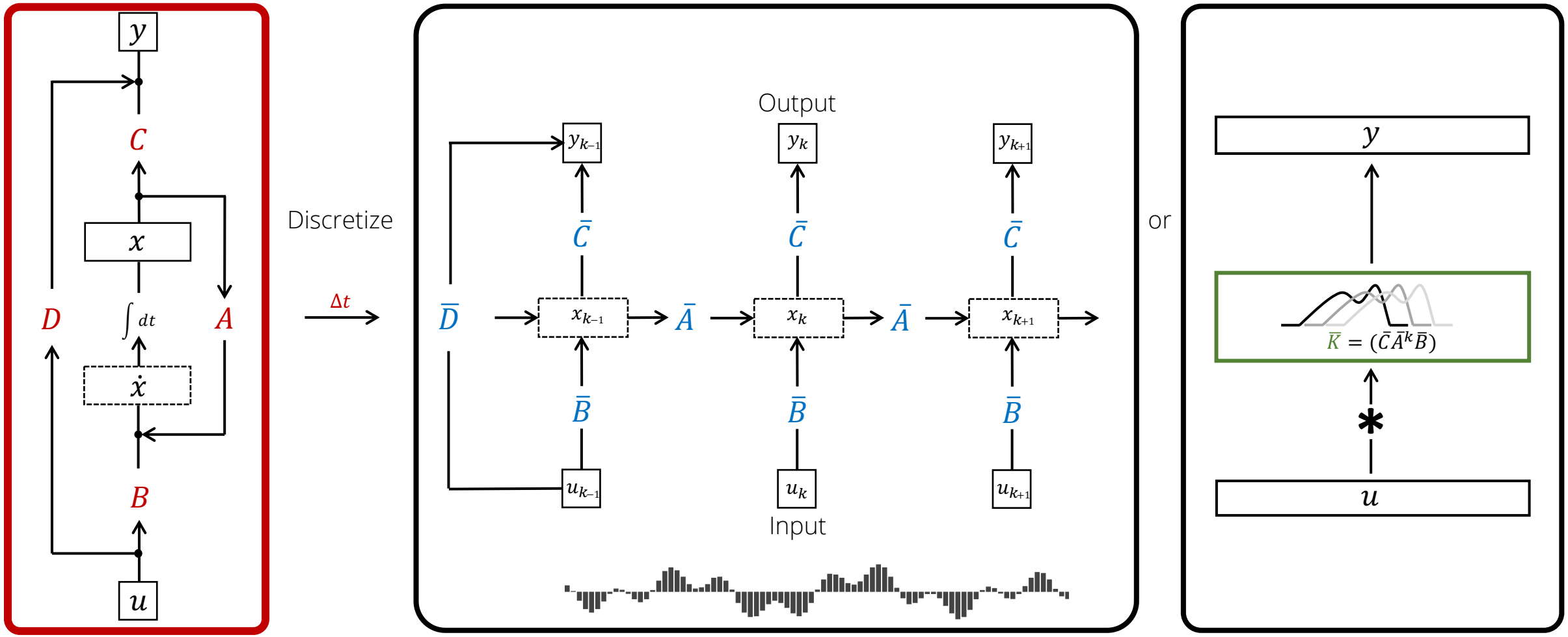
Function-to-function map $u(t) \mapsto y(t)$

1D input
u(t)



1D output
y(t)

SSM: Continuous Representation



Continuous-time

Recurrent

Convolutional

Computing with SSMs: Recurrent View

$$x'(t) = \boldsymbol{A}x(t) + \boldsymbol{B}u(t)$$
$$y(t) = \boldsymbol{C}x(t) + \boldsymbol{D}u(t)$$

Parameters

$\boldsymbol{A} \in \mathbb{R}^{N \times N}$

$\boldsymbol{B} \in \mathbb{R}^{N \times 1}$

$\boldsymbol{C} \in \mathbb{R}^{1 \times N}$

$\boldsymbol{D} \in \mathbb{R}^{1 \times 1}$

$\Delta \in \mathbb{R}$

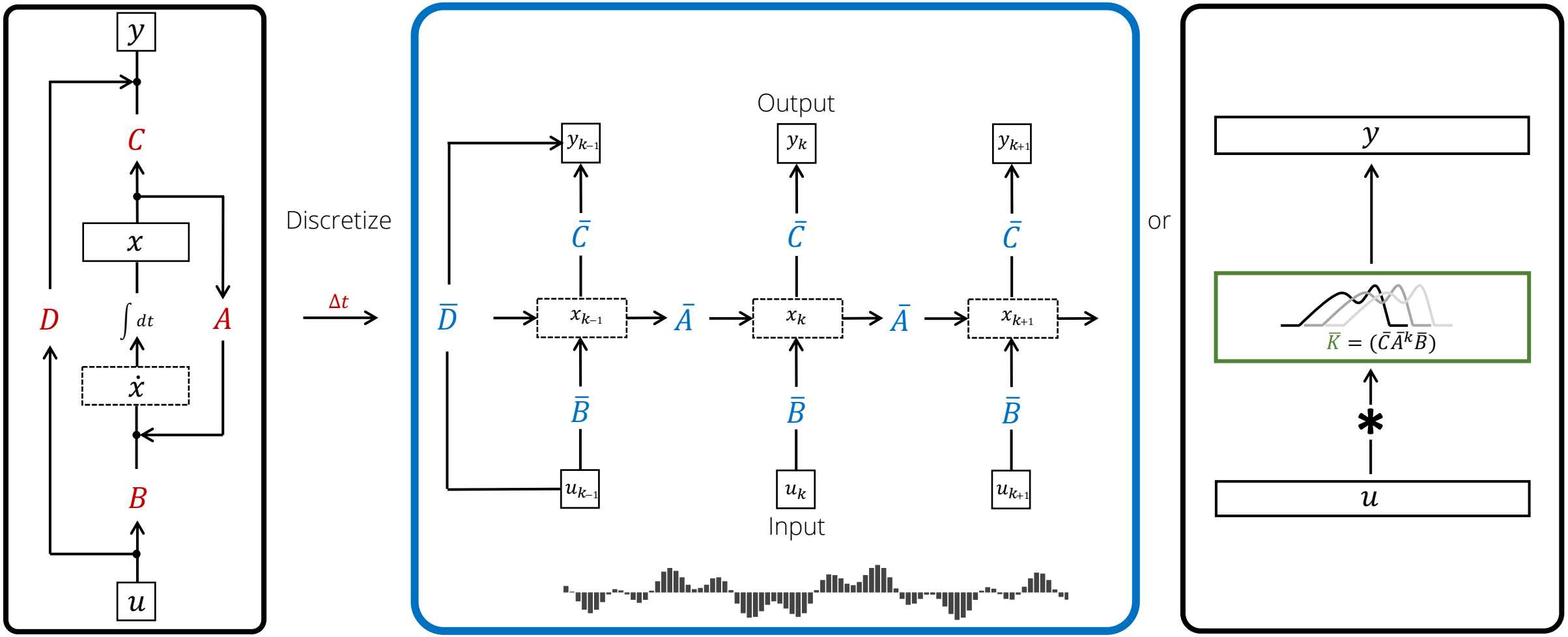
1. Discretize $\overline{\boldsymbol{A}} = \boldsymbol{I} + \Delta \boldsymbol{A}$

2. Recurrent "hidden state" $x_k = \overline{\boldsymbol{A}}x_{k-1} + \overline{\boldsymbol{B}}u_k$

3. Out projection $y_k = \overline{\boldsymbol{C}}x_k + \overline{\boldsymbol{D}}u_k$

Can be computed with **linear recurrence**, similar to RNNs

SSM: Recurrent Representation



Computing with SSMs: Convolution View

$$x_k = \overline{\boldsymbol{A}} x_{k-1} + \overline{\boldsymbol{B}} u_k$$

$$y_k = \overline{\boldsymbol{C}} x_k$$

Can explicitly unroll the linear recurrence in closed form

Computing with SSMs: Convolution View

$$x_k = \overline{\boldsymbol{A}} x_{k-1} + \overline{\boldsymbol{B}} u_k$$

$$y_k = \overline{\boldsymbol{C}} x_k$$

Can explicitly unroll the linear recurrence in closed form

$$x_0 = \overline{\boldsymbol{B}} u_0$$

Computing with SSMs: Convolution View

$$x_k = \overline{\boldsymbol{A}} x_{k-1} + \overline{\boldsymbol{B}} u_k$$

$$y_k = \overline{\boldsymbol{C}} x_k$$

Can explicitly unroll the linear recurrence in closed form

$$x_0 = \overline{\boldsymbol{B}} u_0 \qquad x_1 = \overline{\boldsymbol{AB}} u_0 + \overline{\boldsymbol{B}} u_1 \qquad x_2 = \overline{\boldsymbol{A}}^2 \overline{\boldsymbol{B}} u_0 + \overline{\boldsymbol{AB}} u_1 + \overline{\boldsymbol{B}} u_2 \qquad \dots$$

$$y_0 = \overline{\boldsymbol{CB}} u_0 \qquad y_1 = \overline{\boldsymbol{CAB}} u_0 + \overline{\boldsymbol{CB}} u_1 \qquad y_2 = \overline{\boldsymbol{CA}}^2 \overline{\boldsymbol{B}} u_0 + \overline{\boldsymbol{CAB}} u_1 + \overline{\boldsymbol{CB}} u_2 \qquad \dots$$

Computing with SSMs: Convolution View

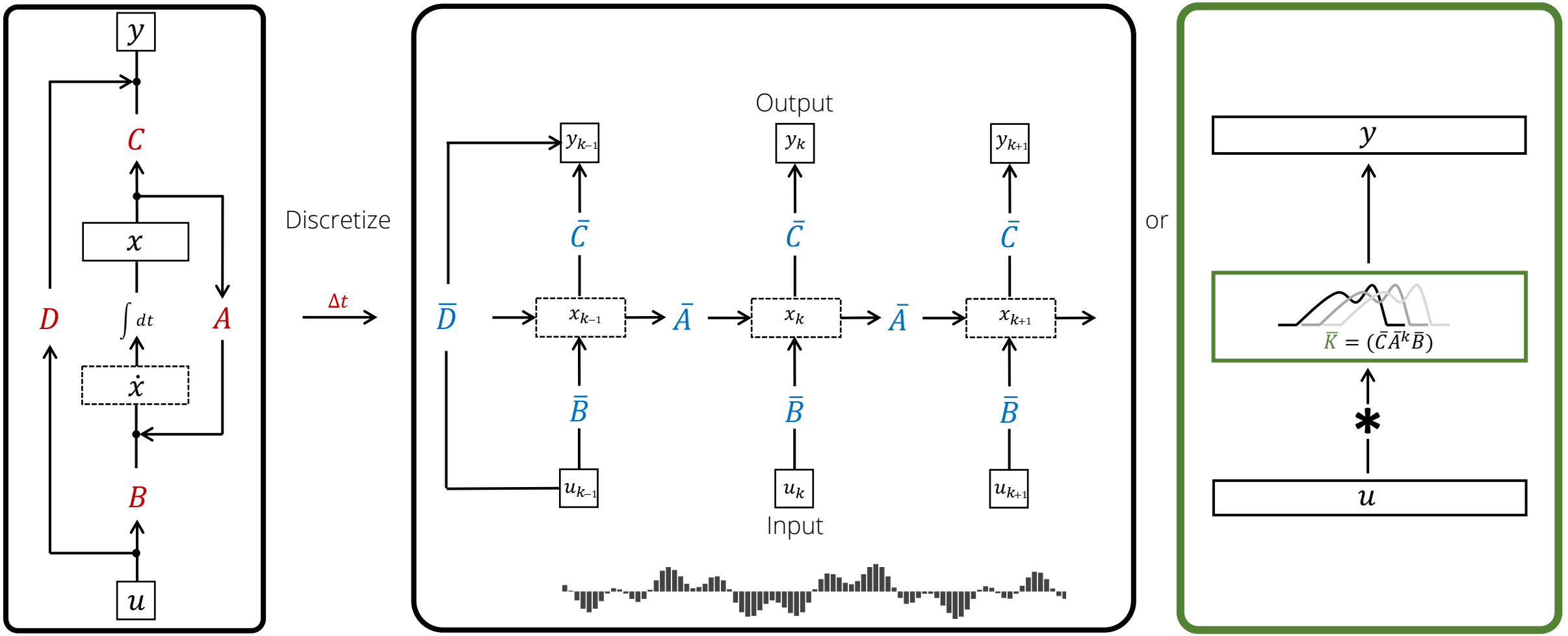
$$y_k = \overline{\boldsymbol{C}}\overline{\boldsymbol{A}}^k\overline{\boldsymbol{B}}u_0 + \overline{\boldsymbol{C}}\overline{\boldsymbol{A}}^{k-1}\overline{\boldsymbol{B}}u_1 + \cdots + \overline{\boldsymbol{C}\boldsymbol{A}\boldsymbol{B}}u_{k-1} + \overline{\boldsymbol{C}\boldsymbol{B}}u_k$$

$$\overline{\boldsymbol{K}} \in \mathbb{R}^L := (\overline{\boldsymbol{C}\boldsymbol{B}}, \overline{\boldsymbol{C}\boldsymbol{A}\boldsymbol{B}}, \ldots, \overline{\boldsymbol{C}}\overline{\boldsymbol{A}}^{L-1}\overline{\boldsymbol{B}})$$

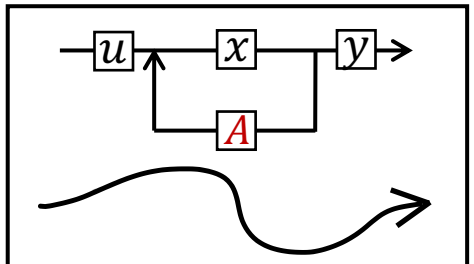
$$y = \overline{\boldsymbol{K}} * u$$

Can be computed with **convolutions**, similar to CNNs

SSM: Convolutional Representation



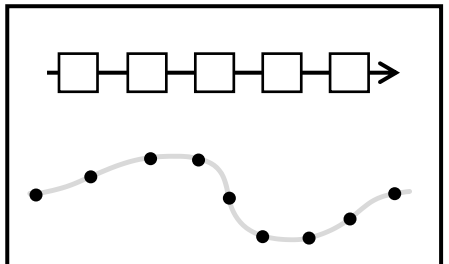
Summary: Properties of SSMs



$$\dot{x} = Ax + Bu$$
$$y = Cx + Du$$

Continuous Representation

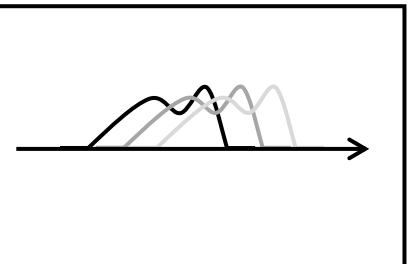
Discretize



$$x = \bar{A}x + \bar{B}u$$
$$y = \bar{C}x + \bar{D}u$$

Recurrent Representation

Unroll

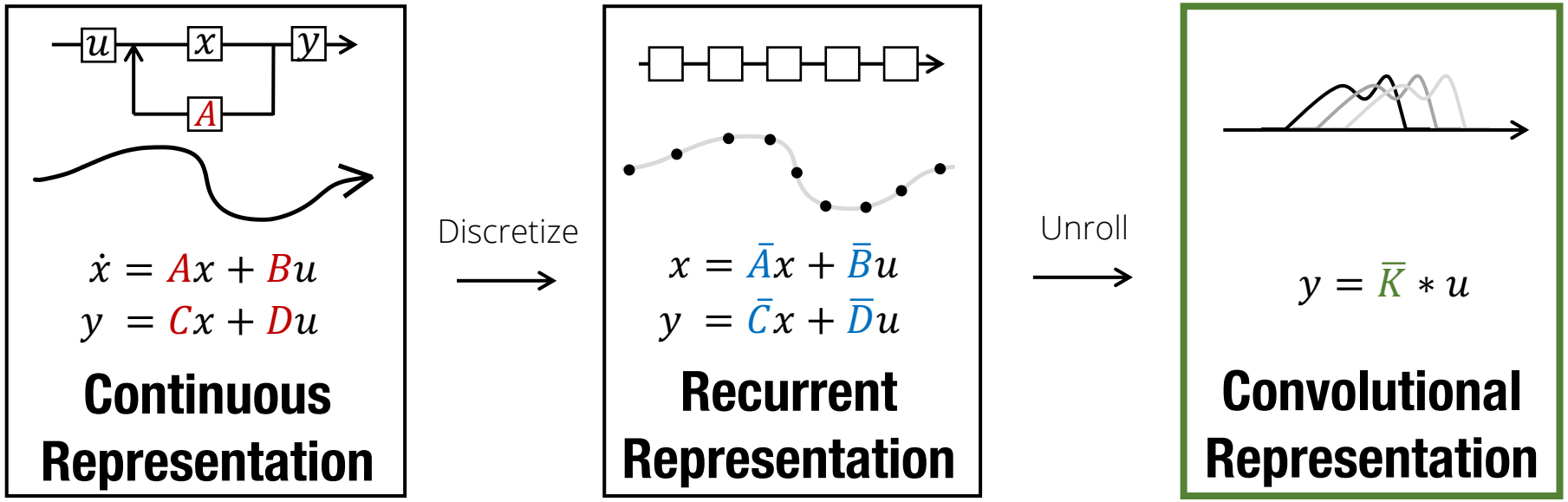


$$y = \bar{K} * u$$

Convolutional Representation

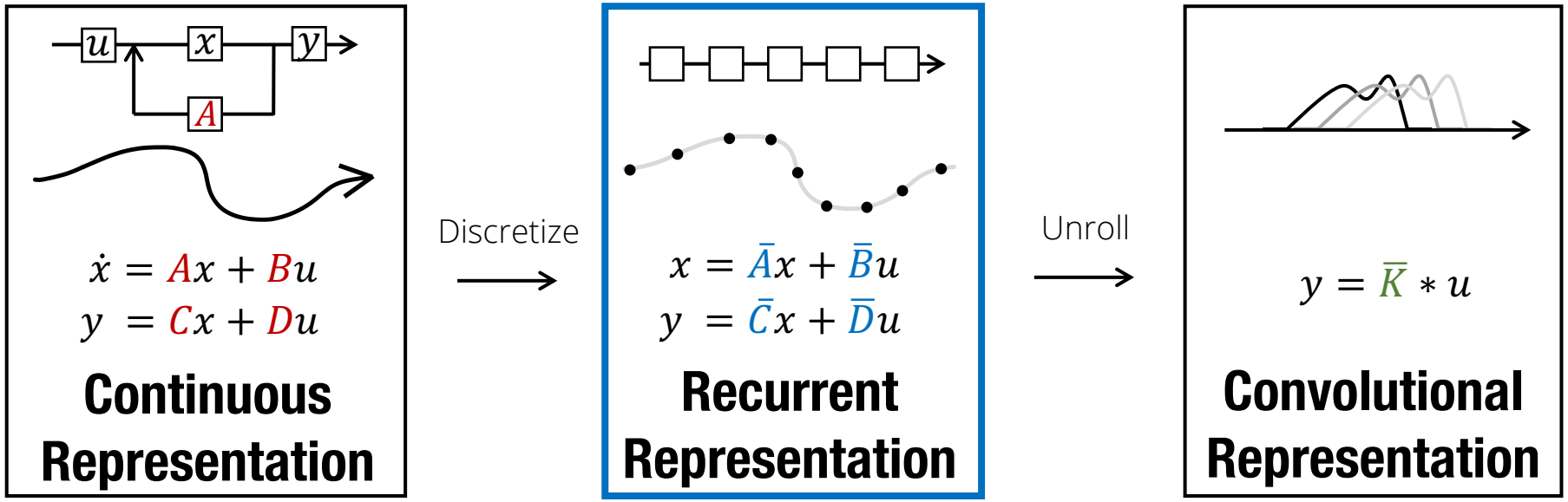
- ✓ *mathematically tractable*
- ✓ *handles irregular data*
- ✓ *unbounded context*
- ✓ *efficient inference*
- ✓ *easy optimization*
- ✓ *parallelizable training*

Summary: Properties of SSMs



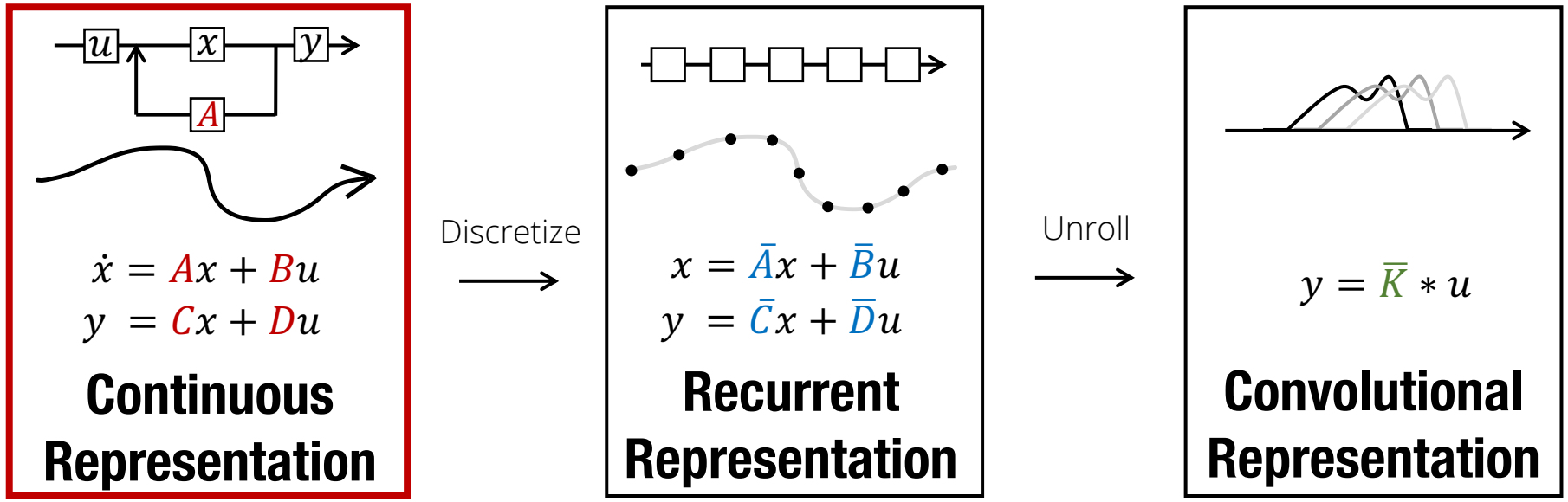
CNN: Efficient training, scalability

Summary: Properties of SSMs



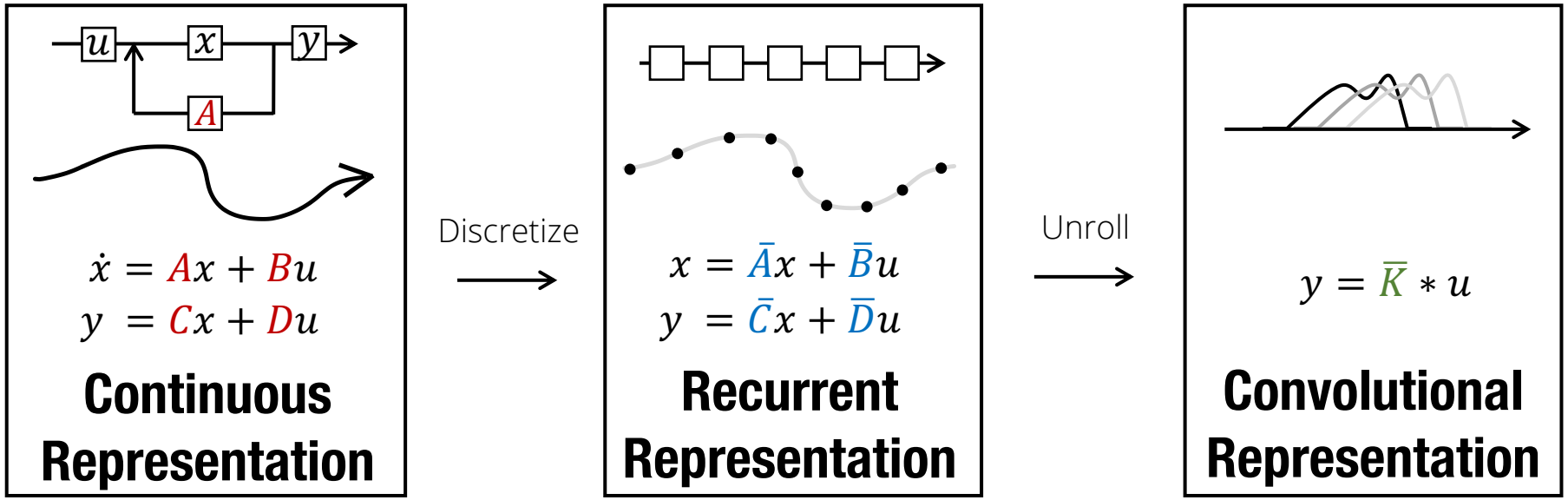
RNN: Efficient autoregressive inference

Summary: Properties of SSMs



CTM: Handle continuous time series

Summary: Properties of SSMs



Simple & principled... why haven't SSMs been used in deep learning?

Outline

- State space models (SSM) for deep sequence modeling
- Structured state spaces (S4) for long-term dependencies
- Solving LRDs in practice

Challenges of Deep SSMs

1 SSMs inherit properties of CTMs, RNNs, CNNs... including problems with LRDs

Modeling Challenge

$$x'(t) = \boldsymbol{A}x(t) + \boldsymbol{B}u(t)$$
$$y(t) = \boldsymbol{C}x(t) + \boldsymbol{D}u(t)$$

with random matrix A: 50% on MNIST

2 SSMs have nice properties ***provided that*** representations $\bar{A}$ and $\bar{K}$ are known

Computation Challenge

...but computing them is extremely hard!

Challenges of Deep SSMs

$$\overline{\boldsymbol{K}} = (\overline{\boldsymbol{CB}}, \overline{\boldsymbol{CAB}}, \ldots, \overline{\boldsymbol{CA}}^{L-1}\overline{\boldsymbol{B}})$$

1 Long-range Dependencies

Powering up $\boldsymbol{A} \rightarrow$ vanishing gradients?

2 Computation

$N \approx 100$

Powering up $\boldsymbol{A} \rightarrow O(N^2 L)$ computation

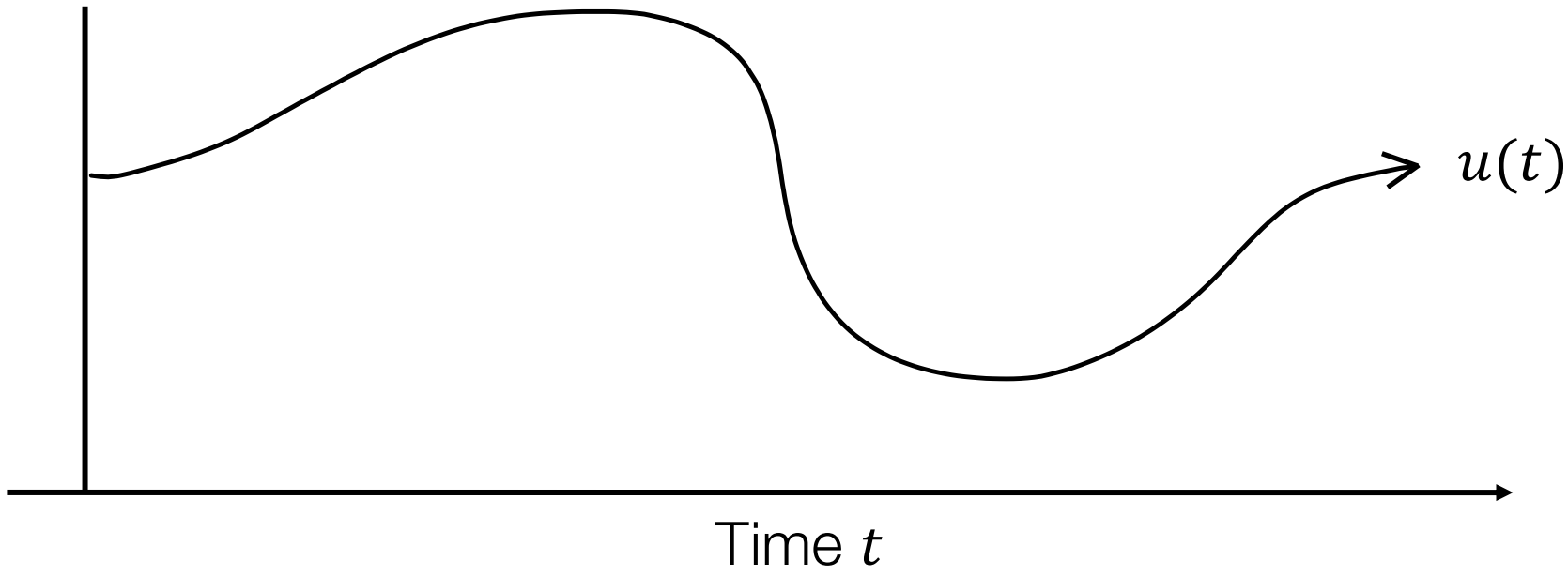
Ideal: $O(L)$ computation

Outline

- State space models (SSM) for deep sequence modeling
- Structured state spaces (S4) for long-term dependencies
  - Continuous-time memory with HiPPO
  - S4: new parameterization and algorithms
- Solving LRDs in practice

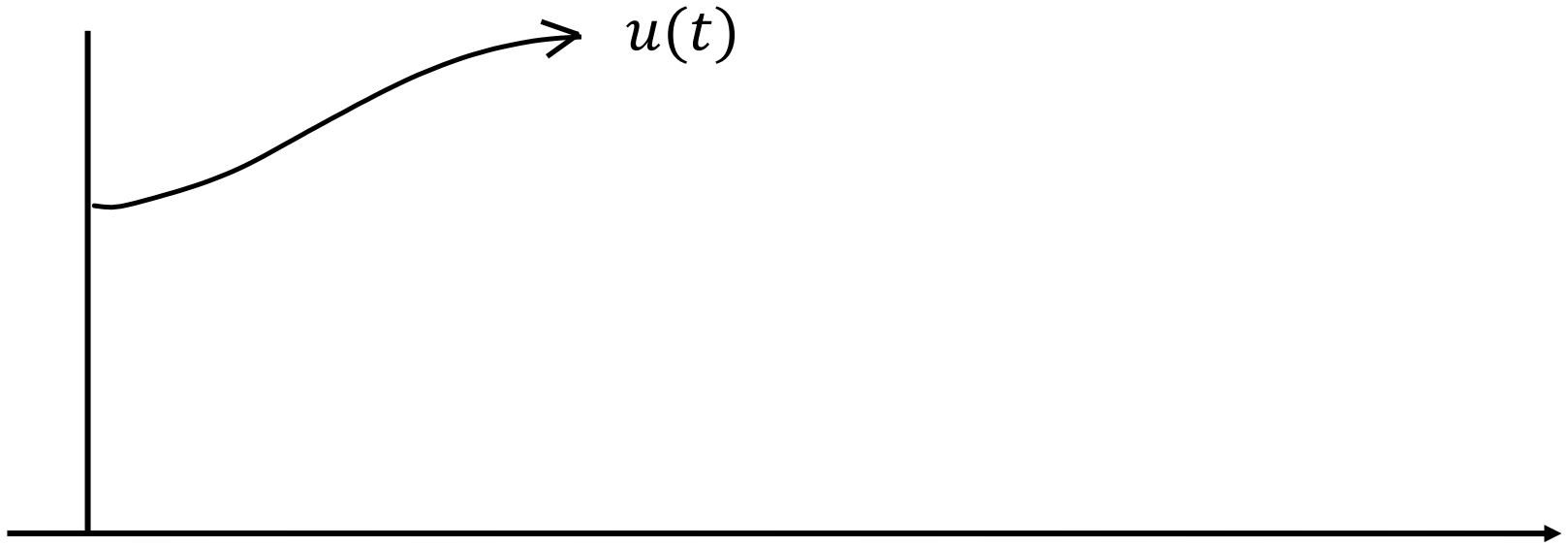
Online Function Approximation

Goal: Study **memorization** in the **continuous** setting



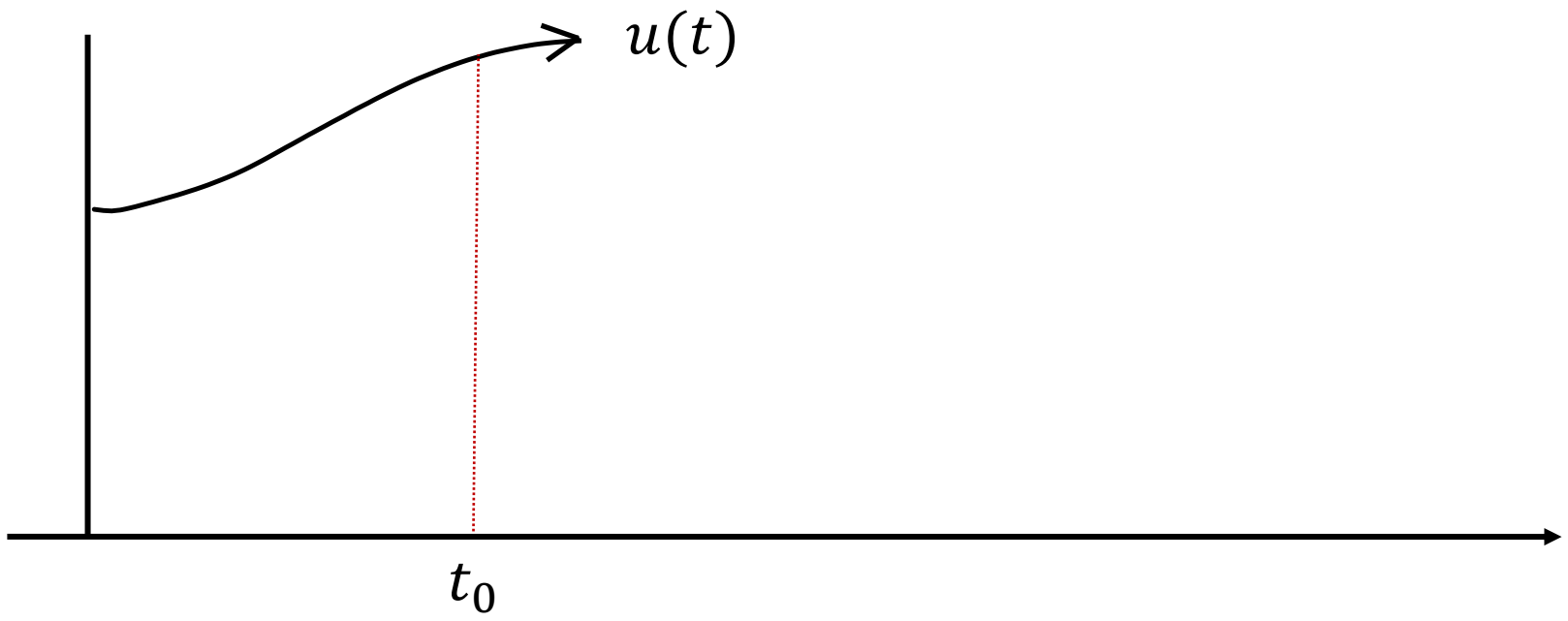
Online Function Approximation

Goal: Study **memorization** in the **continuous** setting



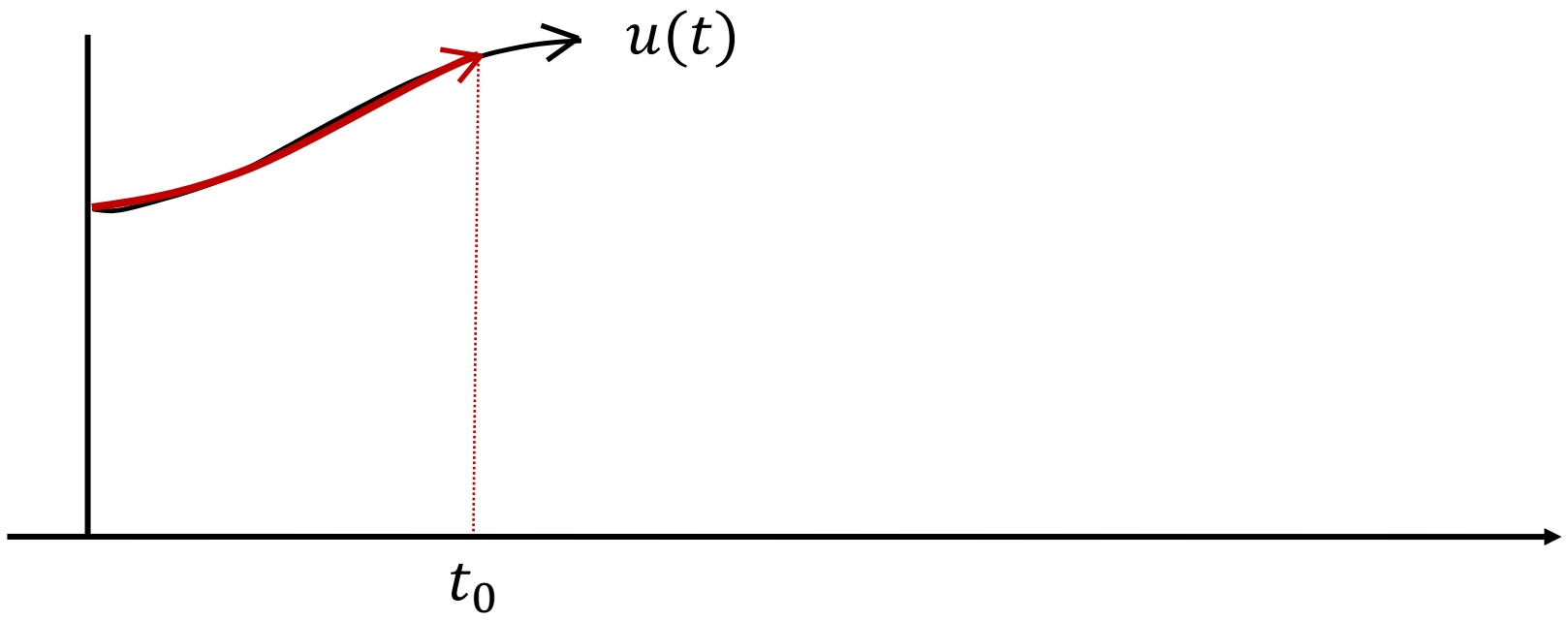
Online Function Approximation

Goal: Study **memorization** in the **continuous** setting



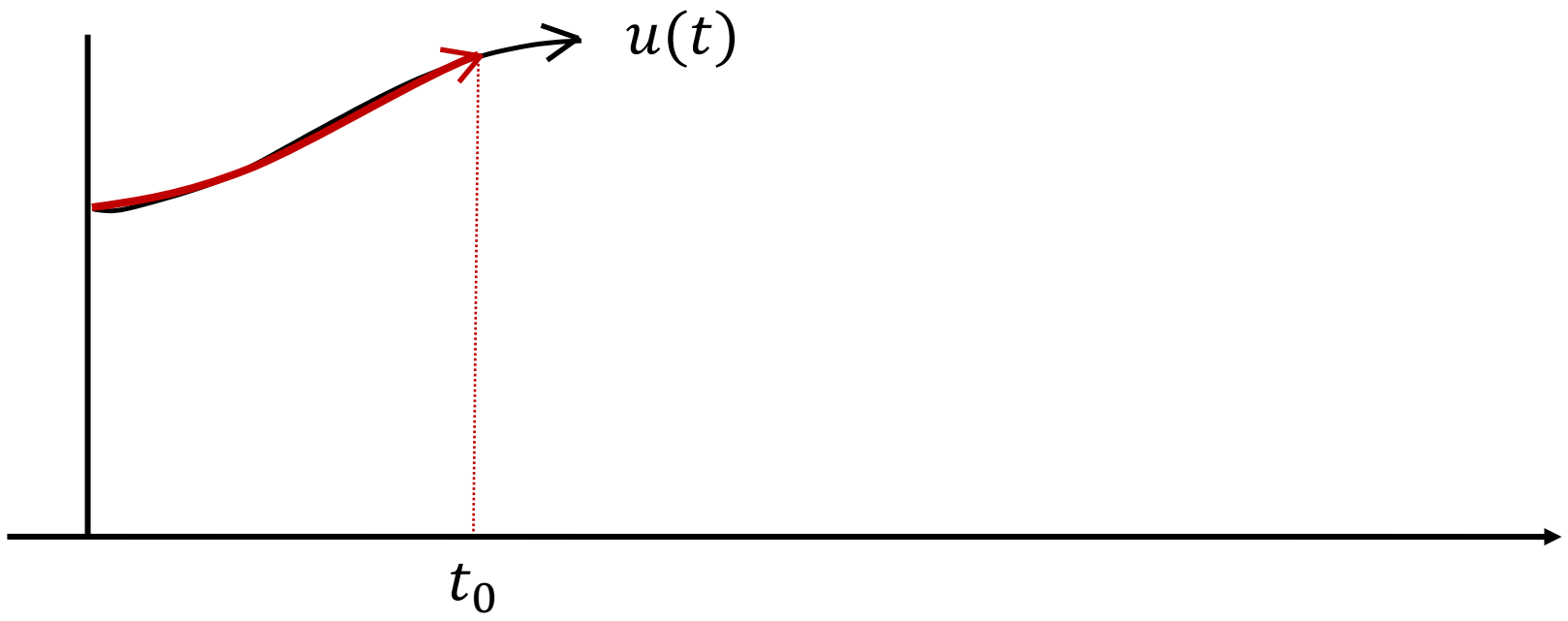
Online Function Approximation

Goal: Study **memorization** in the **continuous** setting



Online Function Approximation

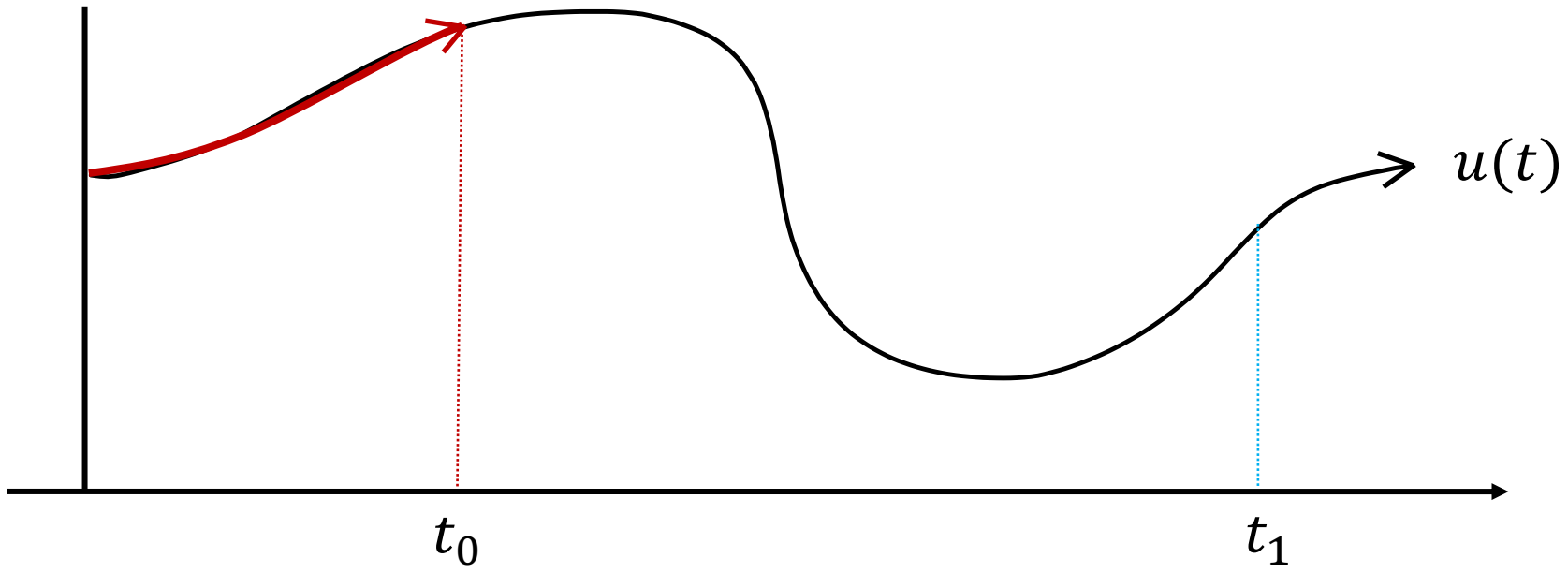
Goal: Study **memorization** in the **continuous** setting



$$x(t_0) = \begin{bmatrix} 0.1 \\ -1.1 \\ 3.7 \\ 2.5 \end{bmatrix}$$

Online Function Approximation

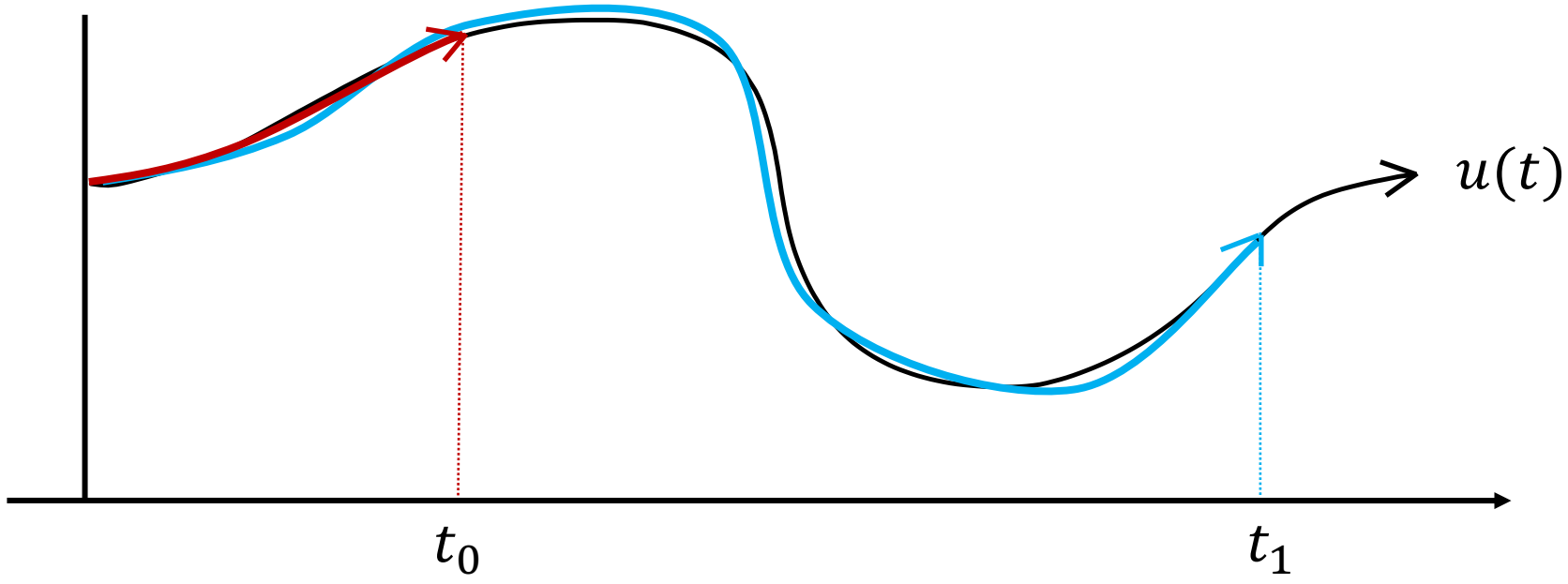
Goal: Study **memorization** in the **continuous** setting



$$x(t_0) = \begin{bmatrix} 0.1 \\ -1.1 \\ 3.7 \\ 2.5 \end{bmatrix}$$

Online Function Approximation

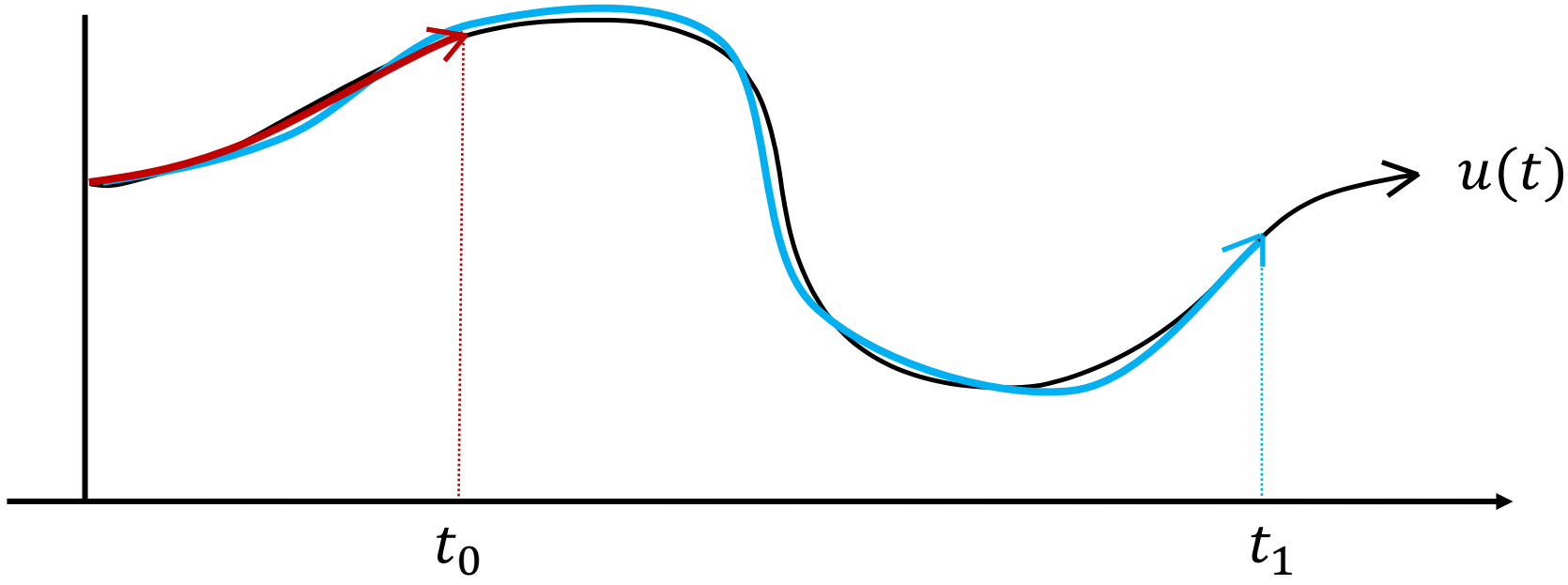
Goal: Study **memorization** in the **continuous** setting



$$x(t_0) = \begin{bmatrix} 0.1 \\ -1.1 \\ 3.7 \\ 2.5 \end{bmatrix}$$

Online Function Approximation

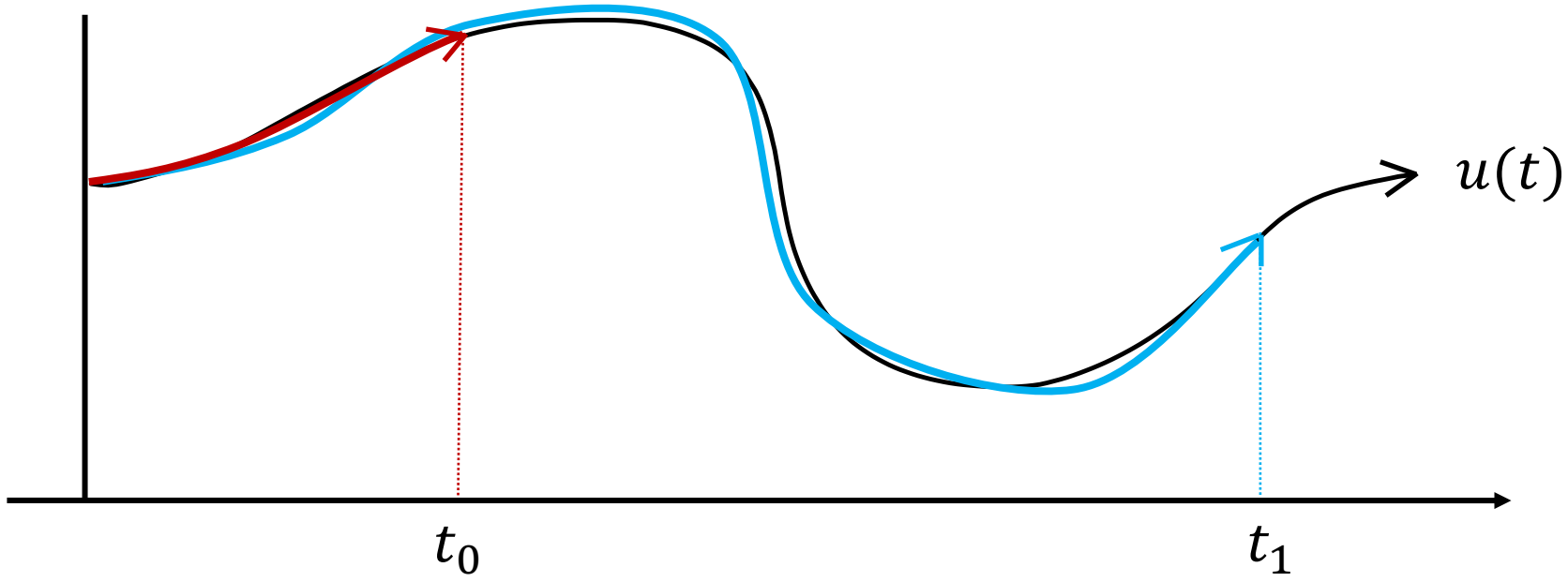
Goal: Study **memorization** in the **continuous** setting



$$x(t_0) = \begin{bmatrix} 0.1 \\ -1.1 \\ 3.7 \\ 2.5 \end{bmatrix} \xrightarrow{\text{update}} x(t_1) = \begin{bmatrix} 1.5 \\ 2.9 \\ -0.3 \\ 2.0 \end{bmatrix}$$

Online Function Approximation

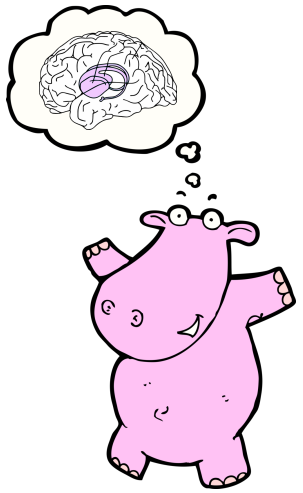
At every time t: maintain state $x(t) \in \mathbb{R}^N$
that encodes the history of u up to time t



$$x(t_0) = \begin{bmatrix} 0.1 \\ -1.1 \\ 3.7 \\ 2.5 \end{bmatrix} \xrightarrow{\text{update}} x(t_1) = \begin{bmatrix} 1.5 \\ 2.9 \\ -0.3 \\ 2.0 \end{bmatrix}$$

HiPPO: Continuous-time Memorization

$$x'(t) = \boldsymbol{A}x(t) + \boldsymbol{B}u(t)$$



Theorem (informal)

For any (suitably behaved) approximation measure ω there exists a **HiPPO operator** such that x optimally memorizes the history of u w.r.t. ω

$$A_{nk} = \begin{cases} (2n+1)^{1/2}(2k+1)^{1/2} & \text{if } n > k \\ n+1 & \text{if } n = k \\ 0 & \text{if } n < k \end{cases}, \qquad B_n = (2n+1)^{\frac{1}{2}}$$

G.*, Dao*, Ermon, Rudra, Ré. "HiPPO: Recurrent Memory with Optimal Polynomial Projections." *NeurIPS 2020*

G., Johnson, Goel, Saab, Dao, Rudra, Ré. "Combining Recurrent, Convolutional, and Continuous-time Models with Linear State Space Layers." *NeurIPS 2021*

Outline

- State space models (SSM) for deep sequence modeling
- Structured state spaces (S4) for long-term dependencies
  - Continuous-time memory with HiPPO
  - S4: new parameterization and algorithms
- Solving LRDs in practice

Structured State Spaces (S4)

$$\overline{\boldsymbol{K}} \in \mathbb{R}^{L} := (\overline{\boldsymbol{C}}\overline{\boldsymbol{B}}, \overline{\boldsymbol{C}}\overline{\boldsymbol{A}}\overline{\boldsymbol{B}}, \dots, \overline{\boldsymbol{C}}\overline{\boldsymbol{A}}^{L-1}\overline{\boldsymbol{B}})$$

Theorem (informal)

Class of HiPPO operators **A** are structured (e.g. quasiseparable)
Allow computation of SSM kernel in $\tilde{O}(N + L)$ operations

$$A_{nk} = \begin{cases} (2n+1)^{1/2}(2k+1)^{1/2} & \text{if } n > k \\ n+1 & \text{if } n = k \\ 0 & \text{if } n < k \end{cases}, \qquad B_n = (2n+1)^{\frac{1}{2}}$$

De sa, G., Puttagunta, Ré, Rudra. "A two-pronged progress in structured dense matrix vector multiplication." *SODA 2018*

G., Johnson, Goel, Saab, Dao, Rudra, Ré. "Combining Recurrent, Convolutional, and Continuous-time Models with Linear State Space Layers." *NeurIPS 2021*

Structured State Spaces (S4)

$$\overline{\boldsymbol{K}} \in \mathbb{R}^{L} := (\overline{\boldsymbol{C}}\overline{\boldsymbol{B}}, \overline{\boldsymbol{C}}\overline{\boldsymbol{A}}\overline{\boldsymbol{B}}, \ldots, \overline{\boldsymbol{C}}\overline{\boldsymbol{A}}^{L-1}\overline{\boldsymbol{B}})$$

Algorithm 1 S4 Convolution Kernel (Sketch)

Input: S4 parameters $\boldsymbol{\Lambda}, \boldsymbol{p}, \boldsymbol{q}, \boldsymbol{B}, \boldsymbol{C} \in \mathbb{C}^N$ and step size Δ
Output: SSM convolution kernel $\overline{\boldsymbol{K}} = \mathcal{K}_L(\overline{\boldsymbol{A}}, \overline{\boldsymbol{B}}, \overline{\boldsymbol{C}})$ for $\boldsymbol{A} = \boldsymbol{\Lambda} - \boldsymbol{p}\boldsymbol{q}^*$ (equation (5))

1: $\widetilde{\boldsymbol{C}} \leftarrow \left(\boldsymbol{I} - \overline{\boldsymbol{A}}^L\right)^* \overline{\boldsymbol{C}}$ ▷ Truncate SSM generating function (SSMGF) to length L

2: $\begin{bmatrix} k_{00}(\omega) & k_{01}(\omega) \\ k_{10}(\omega) & k_{11}(\omega) \end{bmatrix} \leftarrow \left[\widetilde{\boldsymbol{C}}\ \boldsymbol{q}\right]^* \left(\frac{2}{\Delta}\frac{1-\omega}{1+\omega} - \boldsymbol{\Lambda}\right)^{-1} [\boldsymbol{B}\ \boldsymbol{p}]$ ▷ Black-box Cauchy kernel

3: $\hat{\boldsymbol{K}}(\omega) \leftarrow \frac{2}{1+\omega}\left[k_{00}(\omega) - k_{01}(\omega)(1 + k_{11}(\omega))^{-1}k_{10}(\omega)\right]$ ▷ Woodbury Identity

4: $\hat{\boldsymbol{K}} = \{\hat{\boldsymbol{K}}(\omega) : \omega = \exp(2\pi i \frac{k}{L})\}$ ▷ Evaluate SSMGF at all roots of unity $\omega \in \Omega_L$

5: $\overline{\boldsymbol{K}} \leftarrow \mathsf{iFFT}(\hat{\boldsymbol{K}})$ ▷ Inverse Fourier Transform

G., Goel, Ré. "Efficiently Modeling Long Sequences with Structured State Spaces."

Challenges of Deep SSMs

1 SSMs inherit problems with LRDs

generic $\boldsymbol{A}$ → 50% on MNIST ✗

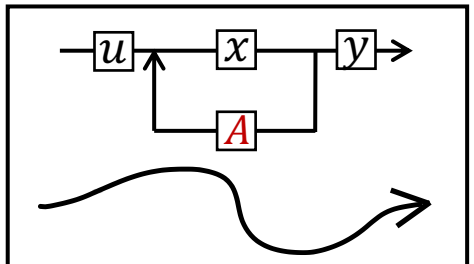
HiPPO $\boldsymbol{A}$ → 99% on MNIST ✓

2 SSMs have trouble computing $\overline{K}$

generic $\boldsymbol{A}$ → $O(N^2L)$ computation ✗

HiPPO $\boldsymbol{A}$ → $\tilde{O}(N+L)$ computation ✓

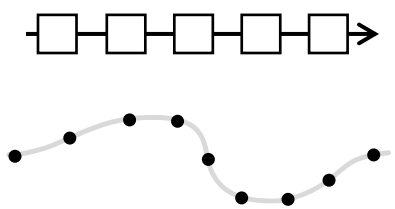
S4: Making Deep SSMs Practical



$$\dot{x} = Ax + Bu$$
$$y = Cx + Du$$

Continuous Representation

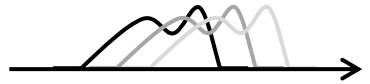
Discretize



$$x = \bar{A}x + \bar{B}u$$
$$y = \bar{C}x + \bar{D}u$$

Recurrent Representation

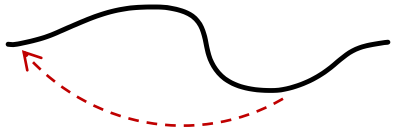
Unroll



$$y = \bar{K} * u$$

Convolutional Representation

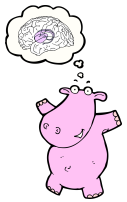
Computing conv. kernel



$$A = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 2 & 0 \\ 1 & 3 & 3 \end{bmatrix}$$

Continuous-time Memorization

Modeling LRD



Class of **structured state spaces** $\boldsymbol{A}$ that resolves LRD **and** computational challenges

Outline

- State space models (SSM) for deep sequence modeling

- Structured state spaces (S4) for long-term dependencies
  - Continuous-time memory with HiPPO
  - S4: new parameterization and algorithms

- Solving LRDs in practice

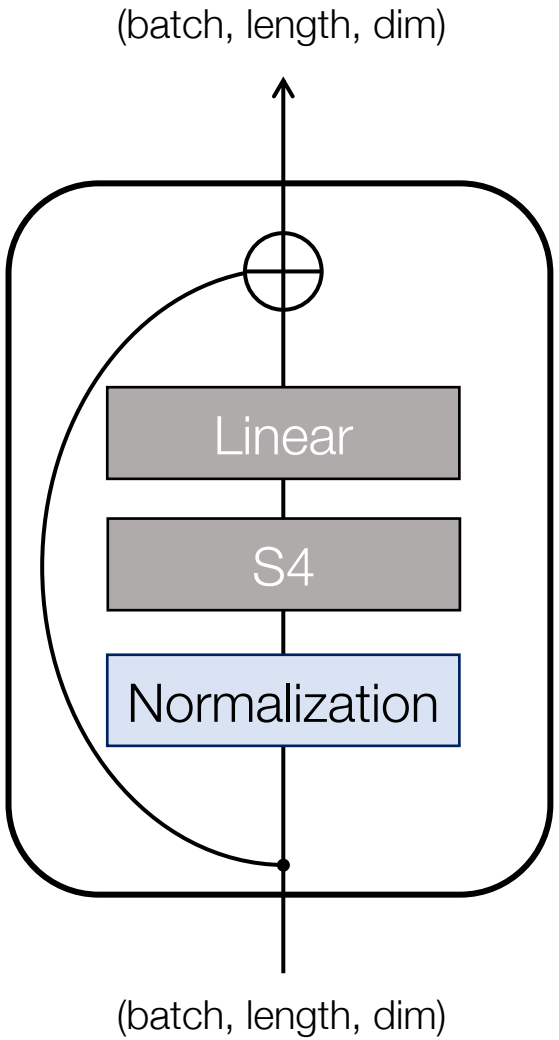
Deep S4 for General Sequence Modeling

Modalities

- Images
- Text
- Audio
- Time series

Tasks

- Classification
- Regression
- Generation
- Forecasting



Sequential Image Classification

	Model	sMNIST	pMNIST	sCIFAR
Transformers	Transformer (Vaswani et al., 2017; Trinh et al., 2018)	98.9	97.9	62.2
CNNs	CKConv (Romero et al., 2021)	99.32	<u>98.54</u>	63.74
	TrellisNet (Bai et al., 2019)	99.20	98.13	73.42
	TCN (Bai et al., 2018)	99.0	97.2	-
RNNs	LSTM (Hochreiter & Schmidhuber, 1997; Gu et al., 2020b)	98.9	95.11	63.01
	r-LSTM (Trinh et al., 2018)	98.4	95.2	72.2
	Dilated GRU (Chang et al., 2017)	99.0	94.6	-
	Dilated RNN (Chang et al., 2017)	98.0	96.1	-
	IndRNN (Li et al., 2018)	99.0	96.0	-
	expRNN (Lezcano-Casado & Martínez-Rubio, 2019)	98.7	96.6	-
	UR-LSTM	99.28	96.96	71.00
	UR-GRU (Gu et al., 2020b)	99.27	96.51	<u>74.4</u>
	LMU (Voelker et al., 2019)	-	97.15	-
	HiPPO-RNN (Gu et al., 2020a)	98.9	98.3	61.1
	UNIcoRNN (Rusch & Mishra, 2021)	-	98.4	-
	LMUFFT (Chilkuri & Eliasmith, 2021)	-	98.49	-
	LipschitzRNN (Erichson et al., 2021)	<u>99.4</u>	96.3	64.2
SSMs	**S4**	**99.63**	**98.70**	**91.13**

Setting	**S4**	ResNet
-Norm	**90.46**	79.52
Base	**91.12**	89.46
+Aug	93.16	**95.62**

Speech Classification

- Speech is difficult because LRD:

  1 second clip = length-16000

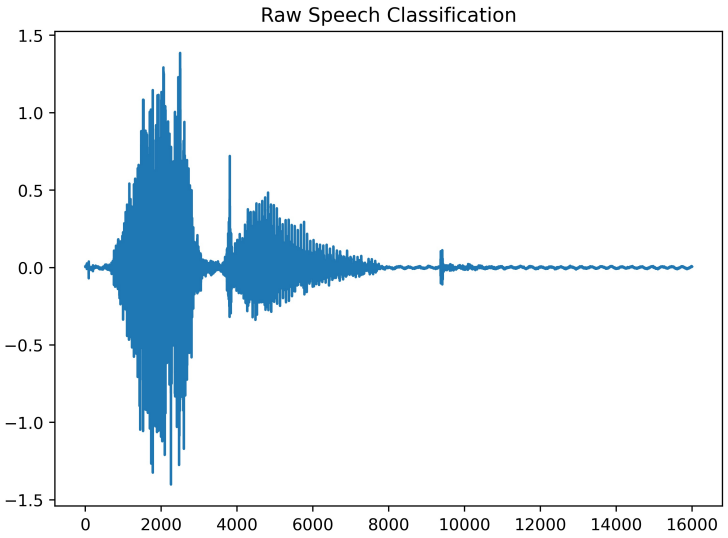
- Speech pipelines require feature engineering

- e.g. Mel-frequency Cepstrum Coefficients (MFCC) (length 160)

	MFCC	
Transformer	90.75	**Transformers**
Performer	80.85	
ODE-RNN	65.9	**CTMs**
NRDE	89.8	
ExpRNN	82.13	**RNNs**
LipschitzRNN	88.38	
CKConv	**95.3**	**CNNs**
WaveGAN-D	✗	
LSSL	93.58	**SSMs**
S4	<u>93.96</u>	

Speech Classification

Raw data requires specialized CNNs



	L=160	L=16000
	MFCC	Raw
Transformer	90.75	✗
Performer	80.85	30.77
ODE-RNN	65.9	✗
NRDE	89.8	16.49
ExpRNN	82.13	11.6
LipschitzRNN	88.38	✗
CKConv	**95.3**	71.66
WaveGAN-D	✗	96.25
LSSL	93.58	✗
S4	93.96	**98.32**

← **88x larger than S4**

Speech Classification

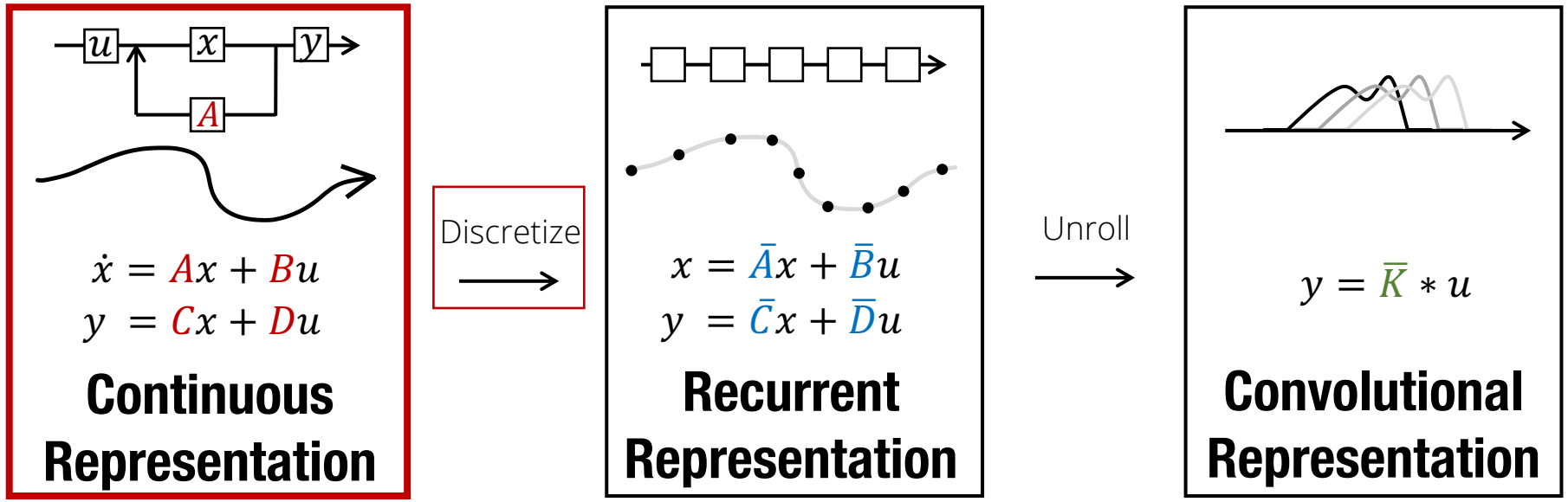
Irregular Continuous Data

Missing values

Varying sampling rates

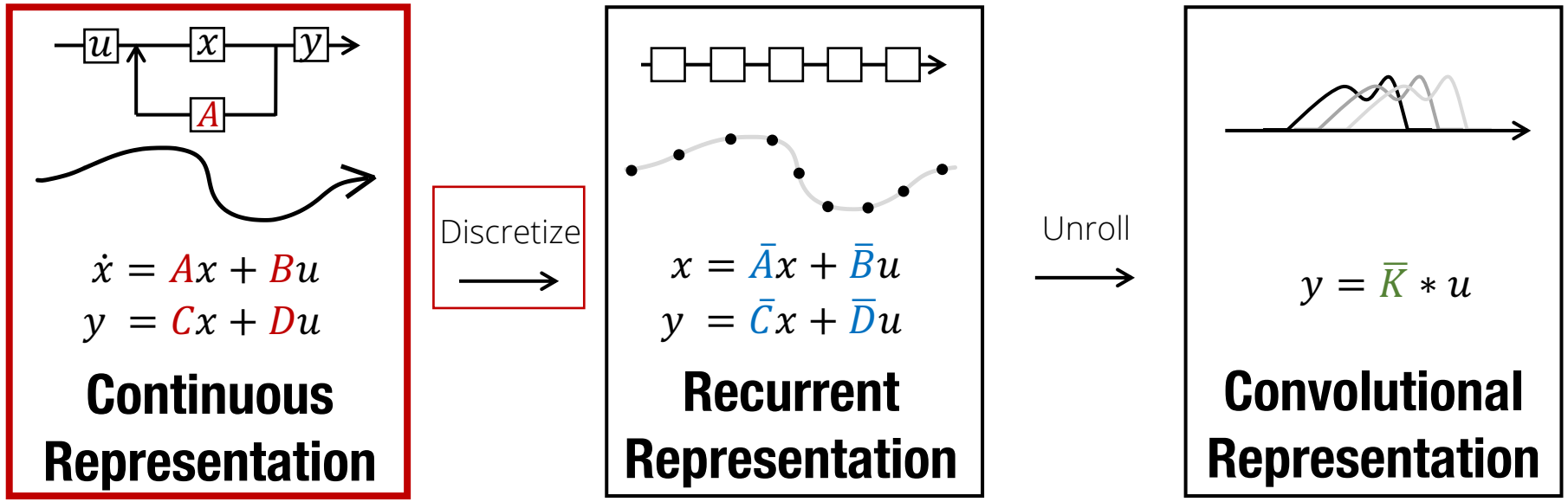
		Train: 16K Hz	Test: 8K Hz
	MFCC	Raw	0.5×
Transformer	90.75	✗	✗
Performer	80.85	30.77	30.68
ODE-RNN	65.9	✗	✗
NRDE	89.8	16.49	15.12
ExpRNN	82.13	11.6	10.8
LipschitzRNN	88.38	✗	✗
CKConv	**95.3**	71.66	<u>65.96</u>
WaveGAN-D	✗	<u>96.25</u>	✗
LSSL	93.58	✗	✗
S4	<u>93.96</u>	**98.32**	**96.30**

Using the Continuous View



Just choose a different step size Δ!

Questions for the Continuous View



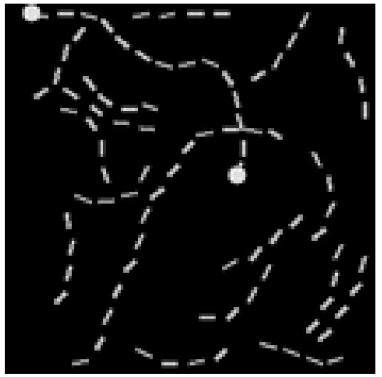
Self-supervision at multiple frequencies?
Zero-shot super-resolution?

Long Range Arena

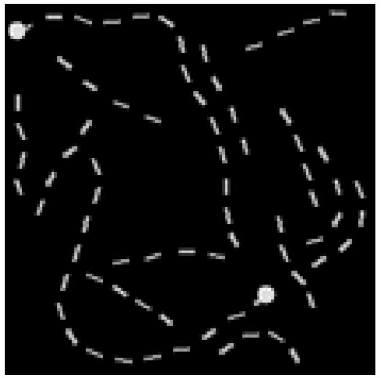
Benchmark spanning text, images, symbolic reasoning (length 1K-16K)

Model	ListOps	Text	Retrieval	Image	Pathfinder	Path-X	Avg
Random	10.00	50.00	50.00	10.00	50.00	50.00	36.67
Transformer	36.37	64.27	57.46	42.44	71.40	✗	53.66
Local Attention	15.82	52.98	53.39	41.46	66.63	✗	46.71
Sparse Trans.	17.07	63.58	59.59	44.24	71.71	✗	51.03
Longformer	35.63	62.85	56.89	42.22	69.71	✗	52.88
Linformer	35.70	53.94	52.27	38.56	76.34	✗	51.14
Reformer	37.27	56.10	53.40	38.07	68.50	✗	50.56
Sinkhorn Trans.	33.67	61.20	53.83	41.23	67.45	✗	51.23
Synthesizer	36.99	61.68	54.67	41.61	69.45	✗	52.40
BigBird	36.05	64.02	59.29	40.83	74.87	✗	54.17
Linear Trans.	16.13	65.90	53.09	42.34	75.30	✗	50.46
Performer	18.01	65.40	53.82	42.77	77.05	✗	51.18
FNet	35.33	65.11	59.61	38.67	77.80	✗	54.42
Nyströmformer	37.15	65.52	79.56	41.58	70.94	✗	57.46
Luna-256	37.25	64.57	79.29	47.38	77.72	✗	59.37
S4	**58.35**	**76.02**	**87.09**	**87.26**	**86.05**	**88.10**	**80.48**

Path-X



(a) A positive example.



(b) A negative example.

Long Range Arena

Benchmark spanning text, images, symbolic reasoning (length 1K-16K)

Model	ListOps	Text	Retrieval	Image	Pathfinder	Path-X	Avg
Random	10.00	50.00	50.00	10.00	50.00	50.00	36.67
Transformer	36.37	64.27	57.46	42.44	71.40	✗	53.66
Local Attention	15.82	52.98	53.39	41.46	66.63	✗	46.71
Sparse Trans.	17.07	63.58	59.59	44.24	71.71	✗	51.03
Longformer	35.63	62.85	56.89	42.22	69.71	✗	52.88
Linformer	35.70	53.94	52.27	38.56	76.34	✗	51.14
Reformer	37.27	56.10	53.40	38.07	68.50	✗	50.56
Sinkhorn Trans.	33.67	61.20	53.83	41.23	67.45	✗	51.23
Synthesizer	36.99	61.68	54.67	41.61	69.45	✗	52.40
BigBird	36.05	64.02	59.29	40.83	74.87	✗	54.17
Linear Trans.	16.13	65.90	53.09	42.34	75.30	✗	50.46
Performer	18.01	65.40	53.82	42.77	77.05	✗	51.18
FNet	35.33	65.11	59.61	38.67	77.80	✗	54.42
Nyströmformer	37.15	65.52	79.56	41.58	70.94	✗	57.46
Luna-256	37.25	64.57	79.29	47.38	77.72	✗	59.37
S4	**58.35**	**76.02**	**87.09**	**87.26**	**86.05**	**88.10**	**80.48**

As efficient as the best "efficient Transformers"

	Length 1024		Length 4096	
	Speed	Mem.	Speed	Mem.
Transformer	1×	1×	1×	1×
Performer	1.23×	0.43×	3.79×	0.086×
Linear Trans.	**1.58**×	**0.37**×	**5.35**×	**0.067**×
S4	**1.58**×	0.43×	5.19×	0.091×

Large-scale Generative Modeling: LM

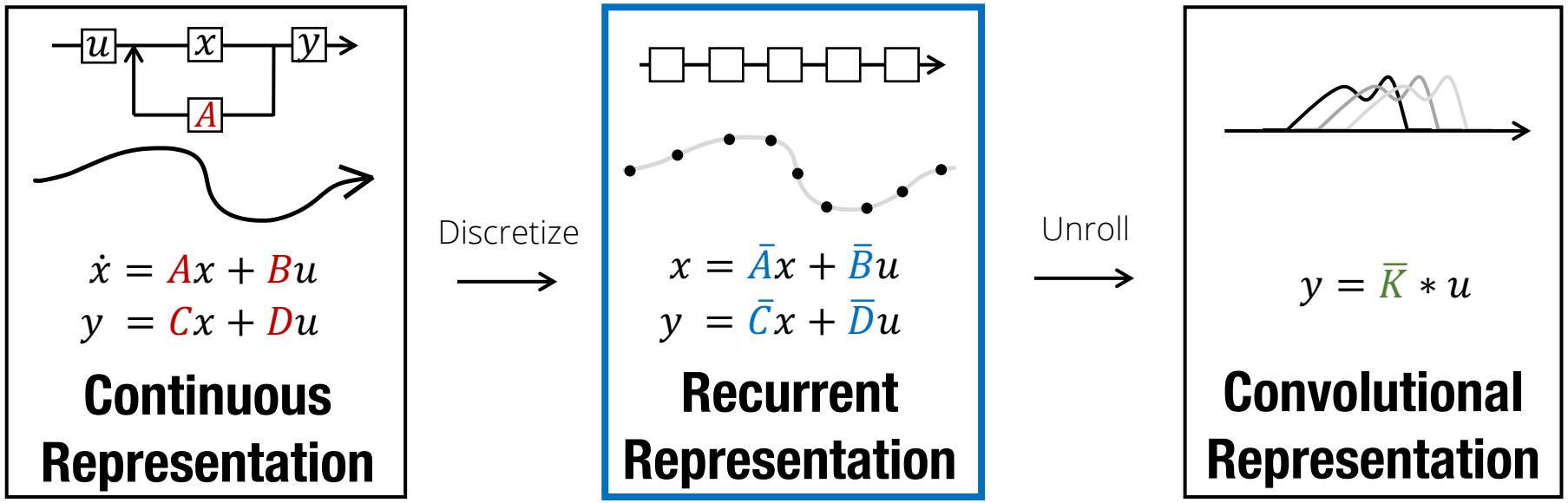
Same backbone

	Model	Params	Test ppl.	Tokens / sec
→	Transformer	247M	**20.51**	0.8K (1×)
	GLU CNN	229M	37.2	-
	AWD-QRNN	151M	33.0	-
	LSTM + Hebb.	-	29.2	-
	TrellisNet	180M	29.19	-
→	Dynamic Conv.	255M	25.0	-
→	TaLK Conv.	240M	23.3	-
→	**S4**	249M	**21.28**	**48K (60×)**

WikiText-103 (perplexity)

SotA for attention-free models – 60X faster generation

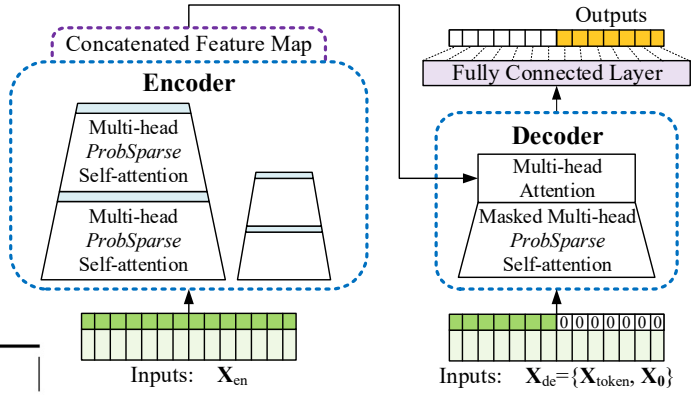
Questions for the Recurrent View



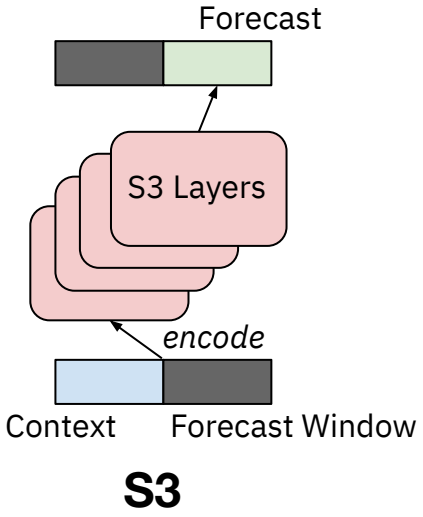
Natural autoregressive / stateful settings (RL, robotics)?
Beyond fixed windows: unbounded context?

Time Series Forecasting

Methods		S3		Informer		Informer†		LogTrans		Reformer		LSTMa		DeepAR		ARIMA		Prophet	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
$ETTh_1$	24	**0.061**	**0.191**	0.098	0.247	0.092	0.246	0.103	0.259	0.222	0.389	0.114	0.272	0.107	0.280	0.108	0.284	0.115	0.275
	48	**0.079**	**0.220**	0.158	0.319	0.161	0.322	0.167	0.328	0.284	0.445	0.193	0.358	0.162	0.327	0.175	0.424	0.168	0.330
	168	**0.104**	**0.258**	0.183	0.346	0.187	0.355	0.207	0.375	1.522	1.191	0.236	0.392	0.239	0.422	0.396	0.504	1.224	0.763
	336	**0.080**	**0.229**	0.222	0.387	0.215	0.369	0.230	0.398	1.860	1.124	0.590	0.698	0.445	0.552	0.468	0.593	1.549	1.820
	720	**0.116**	**0.271**	0.269	0.435	0.257	0.421	0.273	0.463	2.112	1.436	0.683	0.768	0.658	0.707	0.659	0.766	2.735	3.253
$ETTh_2$	24	0.095	0.234	**0.093**	**0.240**	0.099	0.241	0.102	0.255	0.263	0.437	0.155	0.307	0.098	0.263	3.554	0.445	0.199	0.381
	48	0.191	0.346	**0.155**	**0.314**	0.159	0.317	0.169	0.348	0.458	0.545	0.190	0.348	0.163	0.341	3.190	0.474	0.304	0.462
	168	**0.167**	**0.333**	0.232	0.389	0.235	0.390	0.246	0.422	1.029	0.879	0.385	0.514	0.255	0.414	2.800	0.595	2.145	1.068
	336	**0.189**	**0.361**	0.263	0.417	0.258	0.423	0.267	0.437	1.668	1.228	0.558	0.606	0.604	0.607	2.753	0.738	2.096	2.543
	720	**0.187**	**0.358**	0.277	0.431	0.285	0.442	0.303	0.493	2.030	1.721	0.640	0.681	0.429	0.580	2.878	1.044	3.355	4.664
$ETTm_1$	24	**0.024**	**0.117**	0.030	0.137	0.034	0.160	0.065	0.202	0.095	0.228	0.121	0.233	0.091	0.243	0.090	0.206	0.120	0.290
	48	**0.051**	**0.174**	0.069	0.203	0.066	0.194	0.078	0.220	0.249	0.390	0.305	0.411	0.219	0.362	0.179	0.306	0.133	0.305
	96	**0.086**	**0.229**	0.194	0.372	0.187	0.384	0.199	0.386	0.920	0.767	0.287	0.420	0.364	0.496	0.272	0.399	0.194	0.396
	288	**0.160**	**0.327**	0.401	0.554	0.409	0.548	0.411	0.572	1.108	1.245	0.524	0.584	0.948	0.795	0.462	0.558	0.452	0.574
	672	**0.292**	**0.466**	0.512	0.644	0.519	0.665	0.598	0.702	1.793	1.528	1.064	0.873	2.437	1.352	0.639	0.697	2.747	1.174
Weather	24	0.125	0.254	**0.117**	**0.251**	0.119	0.256	0.136	0.279	0.231	0.401	0.131	0.254	0.128	0.274	0.219	0.355	0.302	0.433
	48	0.181	**0.305**	**0.178**	0.318	0.185	0.316	0.206	0.356	0.328	0.423	0.190	0.334	0.203	0.353	0.273	0.409	0.445	0.536
	168	**0.198**	**0.333**	0.266	0.398	0.269	0.404	0.309	0.439	0.654	0.634	0.341	0.448	0.293	0.451	0.503	0.599	2.441	1.142
	336	0.300	**0.417**	**0.297**	**0.416**	0.310	0.422	0.359	0.484	1.792	1.093	0.456	0.554	0.585	0.644	0.728	0.730	1.987	2.468
	720	**0.245**	**0.375**	0.359	0.466	0.361	0.471	0.388	0.499	2.087	1.534	0.866	0.809	0.499	0.596	1.062	0.943	3.859	1.144
ECL	48	0.222	**0.350**	0.239	0.359	0.238	0.368	0.280	0.429	0.971	0.884	0.493	0.539	**0.204**	0.357	0.879	0.764	0.524	0.595
	168	0.331	**0.421**	0.447	0.503	0.442	0.514	0.454	0.529	1.671	1.587	0.723	0.655	**0.315**	0.436	1.032	0.833	2.725	1.273
	336	**0.328**	**0.422**	0.489	0.528	0.501	0.552	0.514	0.563	3.528	2.196	1.212	0.898	0.414	0.519	1.136	0.876	2.246	3.077
	720	**0.428**	**0.494**	0.540	0.571	0.543	0.578	0.558	0.609	4.891	4.047	1.511	0.966	0.563	0.595	1.251	0.933	4.243	1.415
	960	**0.432**	**0.497**	0.582	0.608	0.594	0.638	0.624	0.645	7.019	5.105	1.545	1.006	0.657	0.683	1.370	0.982	6.901	4.264
Count		22		5		0		0		0		0		2		0		0	

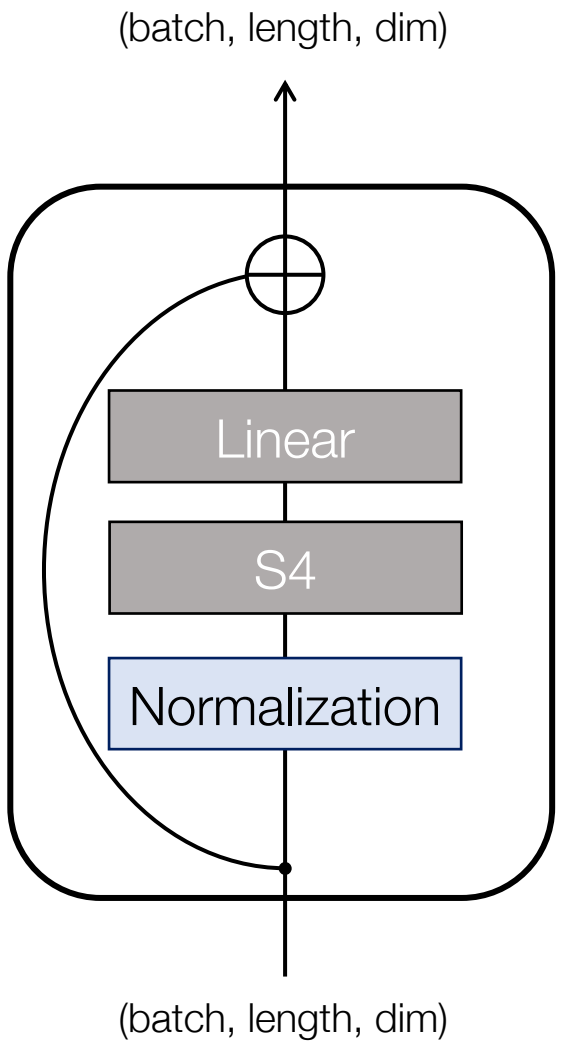


Informer



S3

Future Applications for S4



- Audio generation
- Large scale audio pre-training
- Applications to more time series
- S4 + Transformers for language modeling
- 2D + 3D versions of S4 (images, video)

Looking for more applications!

Thanks!

Papers

- "HiPPO: Recurrent Memory with Optimal Polynomial Projections" https://arxiv.org/abs/2008.07669
- "Combining Recurrent, Convolutional, and Continuous-time Models with Linear State Space Layers" https://arxiv.org/abs/2110.13985
- "Efficiently Modeling Long Sequences with Structured State Spaces" arxiv.org/abs/2111.00396

Code https://github.com/HazyResearch/state-spaces

Blogs https://hazyresearch.stanford.edu/blog/2022-01-14-s4-1
https://srush.github.io/annotated-s4/

Contact albertgu@stanford.edu
krng@stanford.edu

Karan Goel

Khaled Saab

Tri Dao

Chris Ré

University at Buffalo

Isys Johnson
Atri Rudra