Deep Learning (1470)

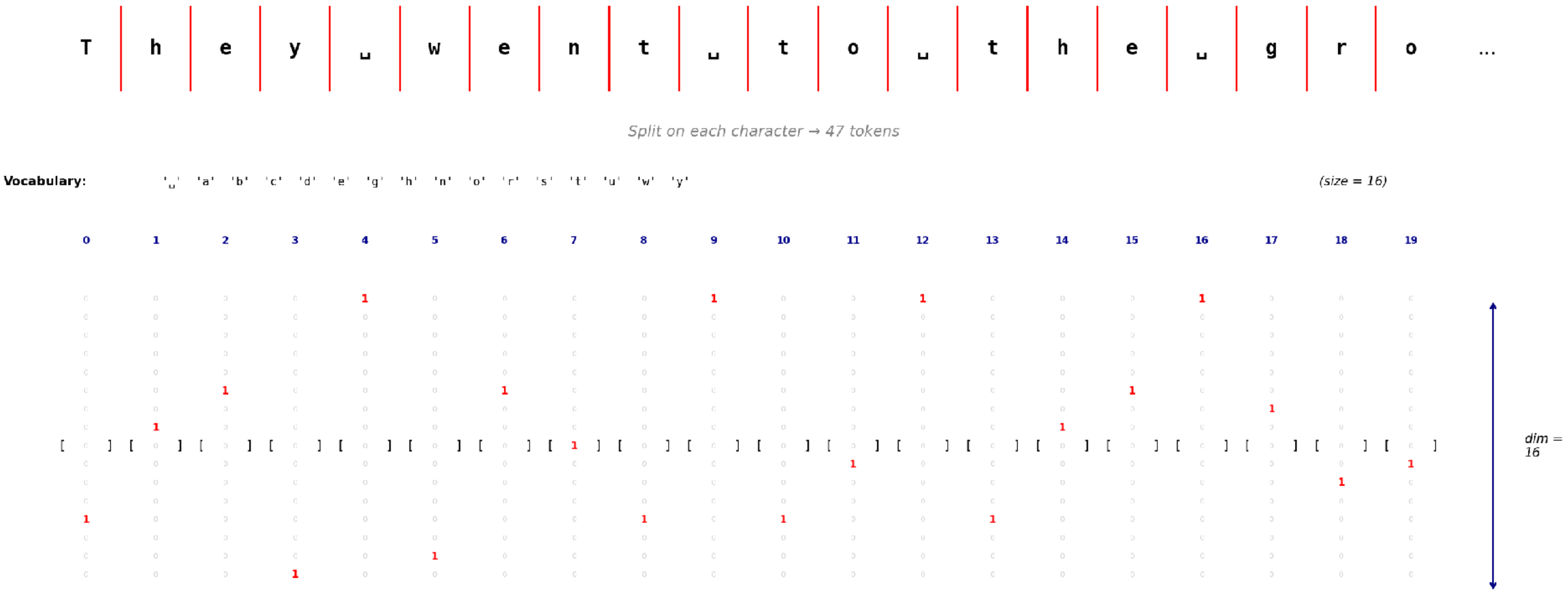
Randall Balestriero

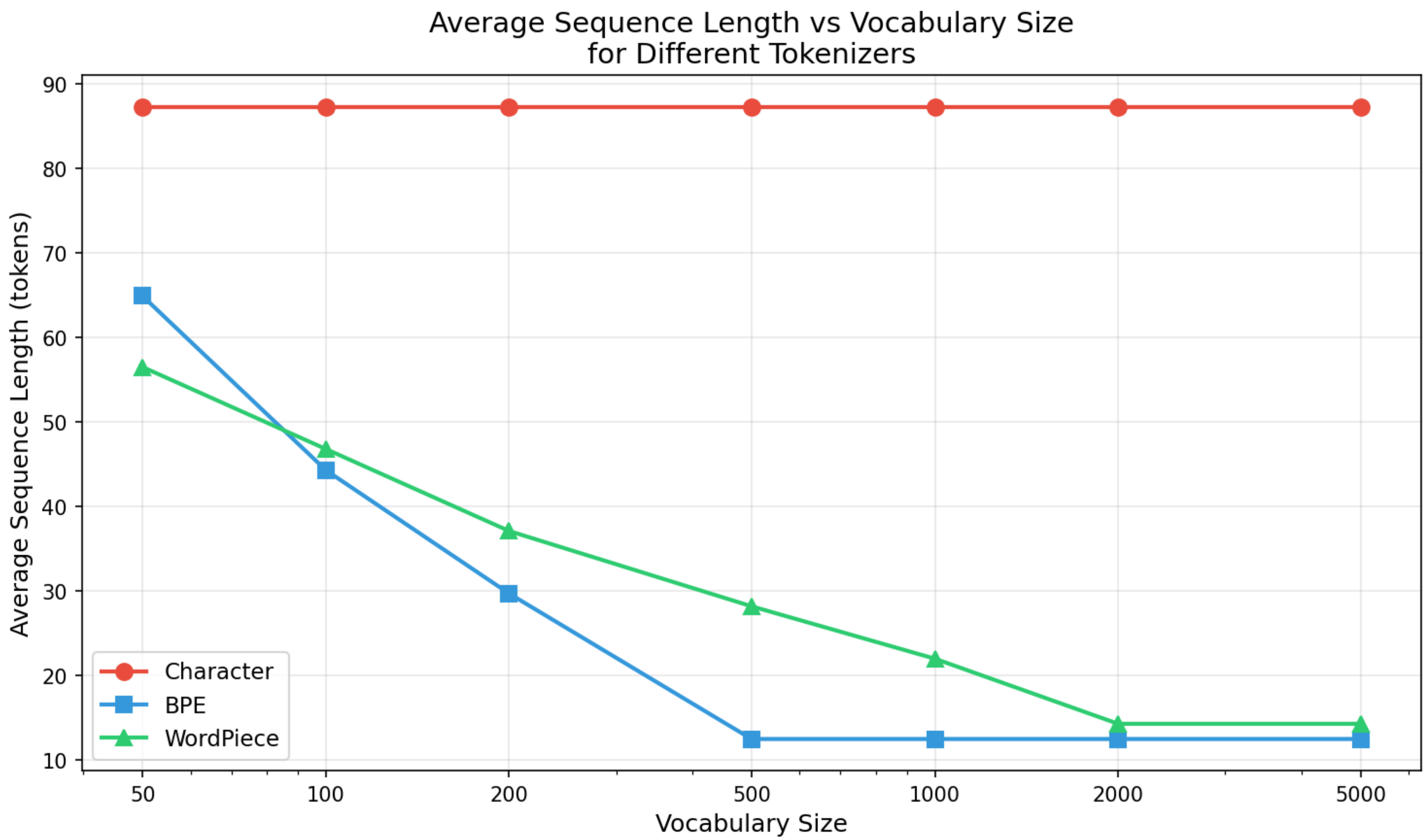
Class 11: Sequential Data and Language Modeling

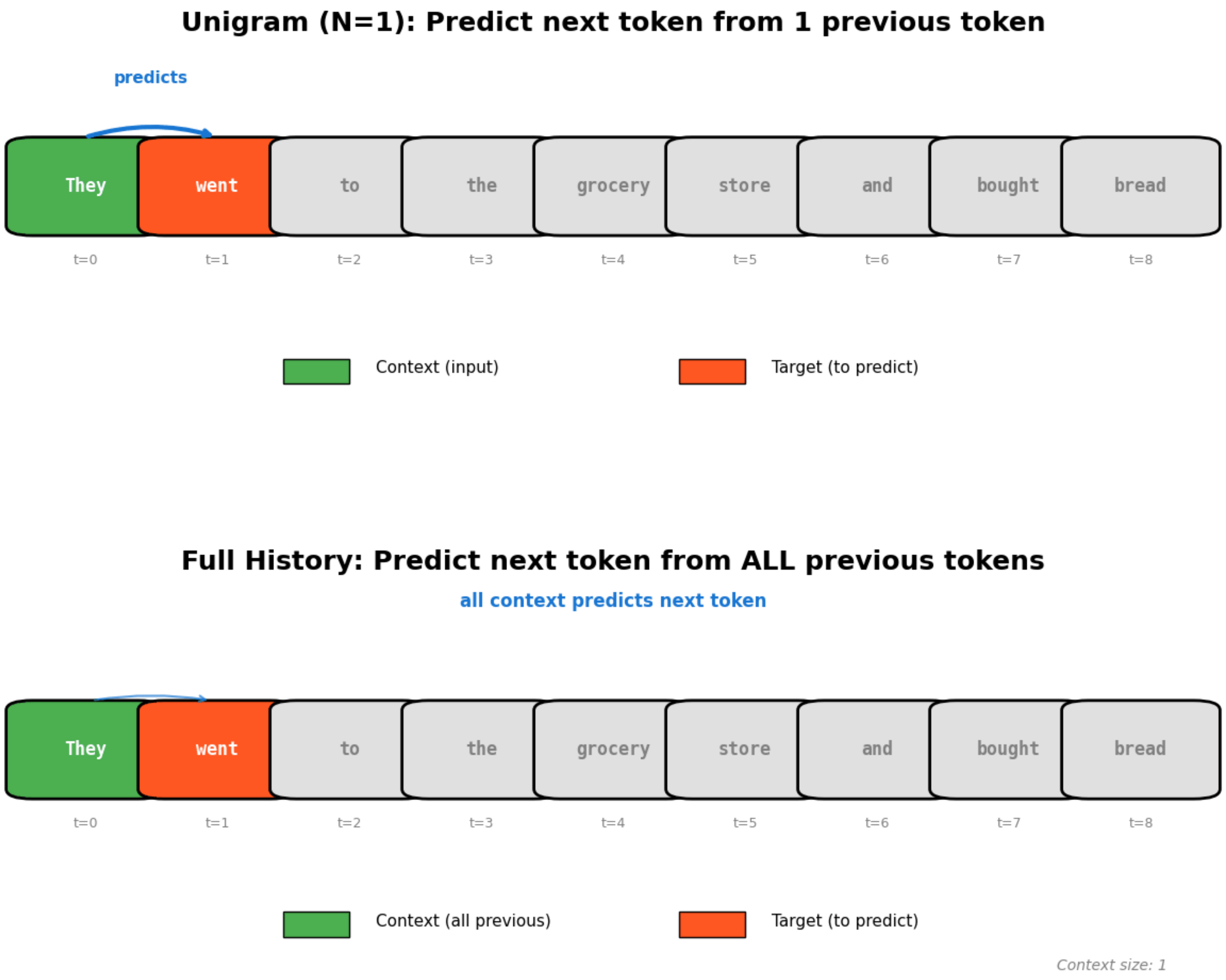
Recap!

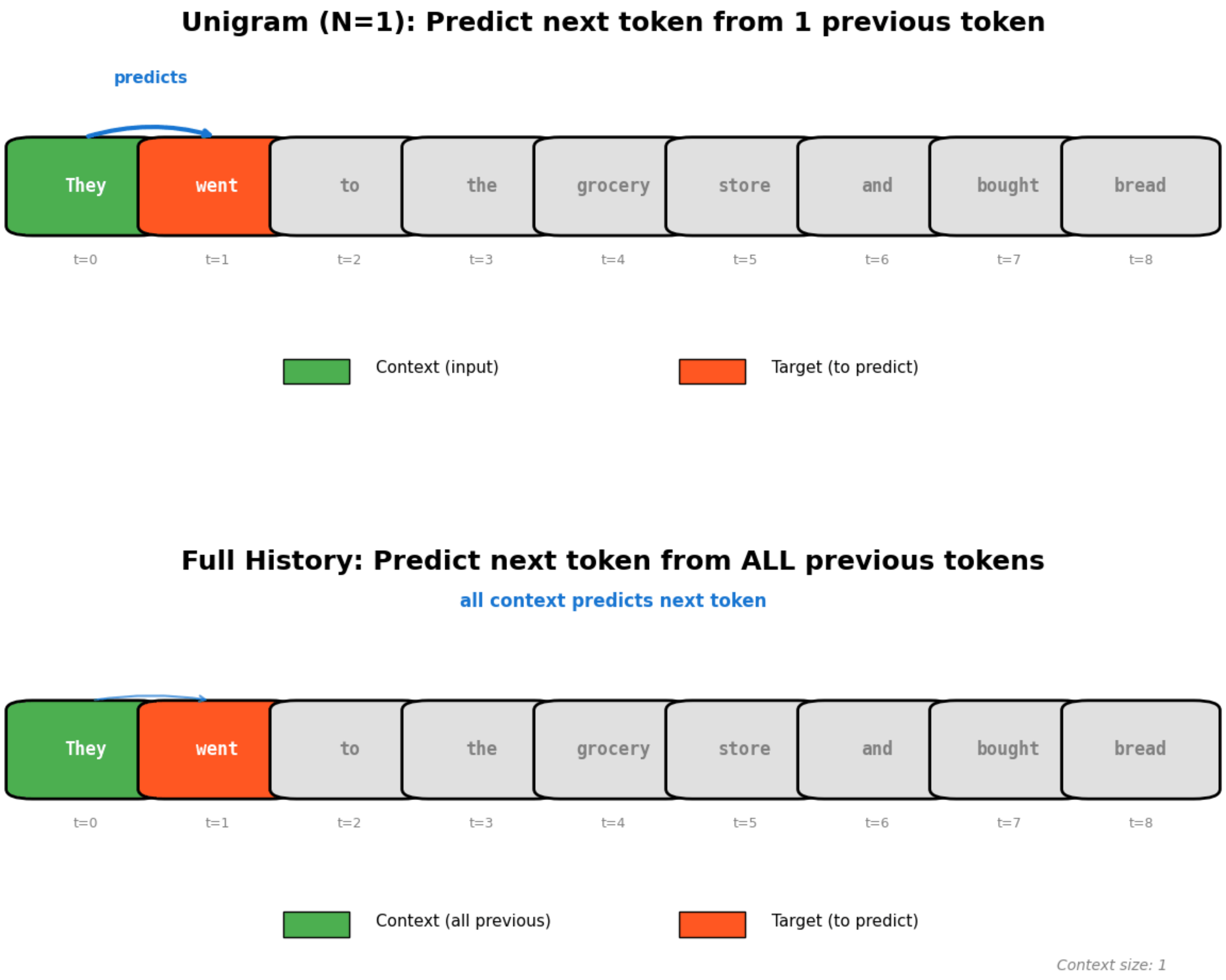
Word Tokenizer

Character Tokenizer



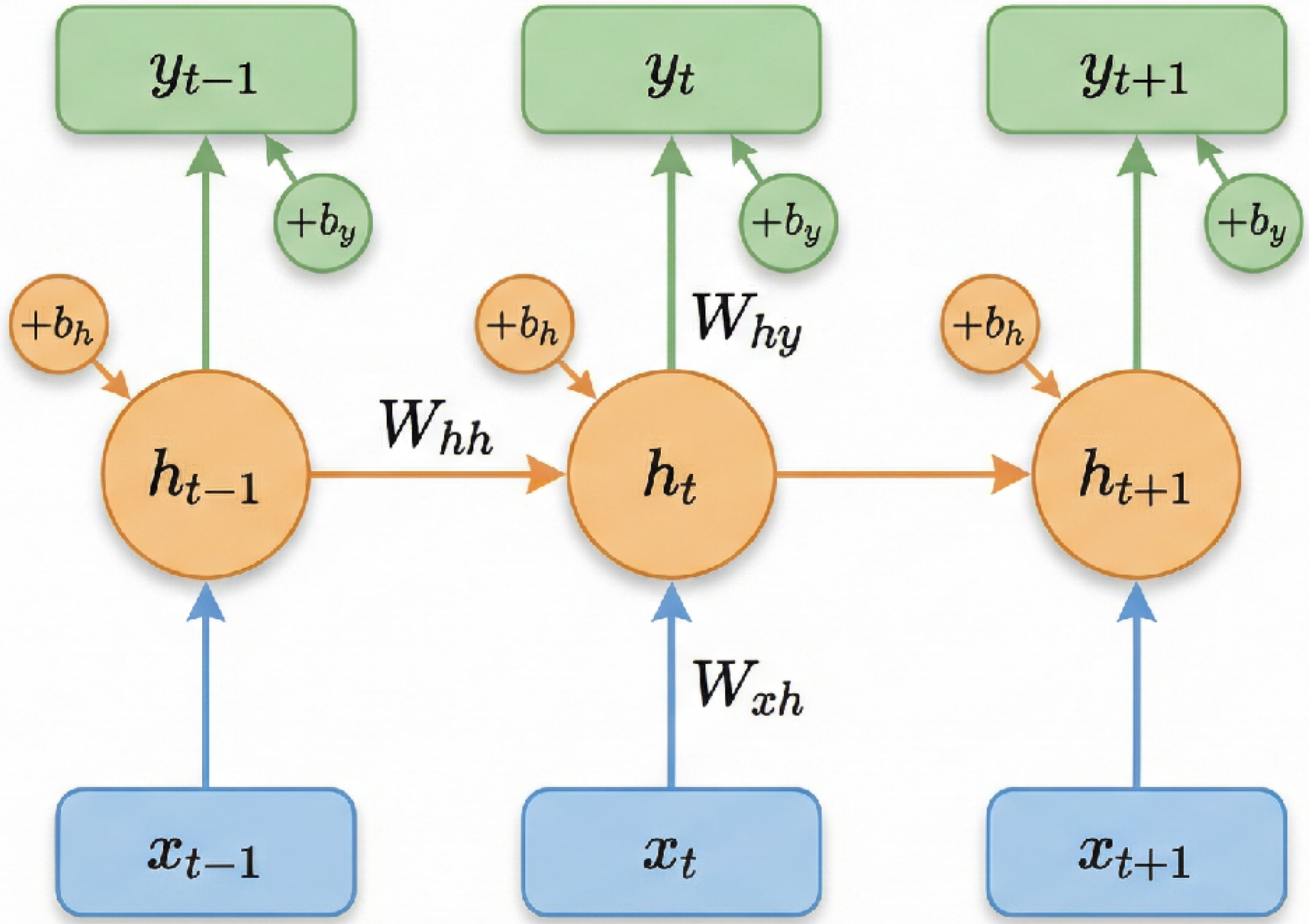

Average Sequence Length vs Vocabulary Size
for Different Tokenizers
Average Sequence Length (tokens)
Vocabulary Size
Character
BPE
WordPiece
90
80
70
60
50
40
30
20
10
50
100
200
500
1000
2000
5000


Unigram (N=1): Predict next token from 1 previous token
predicts
They
went
to
the
grocery
store
and
bought
bread
t=0
t=1
t=2
t=3
t=4
t=5
t=6
t=7
t=8
Context (input)
Target (to predict)
Full History: Predict next token from ALL previous tokens
all context predicts next token
They
went
to
the
grocery
store
and
bought
bread
t=0
t=1
t=2
t=3
t=4
t=5
t=6
t=7
t=8
Context (all previous)
Target (to predict)
Context size: 1



Unigram (N=1): Predict next token from 1 previous token
predicts
They
went
to
the
grocery
store
and
bought
bread
t=0
t=1
t=2
t=3
t=4
t=5
t=6
t=7
t=8
Context (input)
Target (to predict)
Full History: Predict next token from ALL previous tokens
all context predicts next token
They
went
to
the
grocery
store
and
bought
bread
t=0
t=1
t=2
t=3
t=4
t=5
t=6
t=7
t=8
Context (all previous)
Target (to predict)
Context size: 1

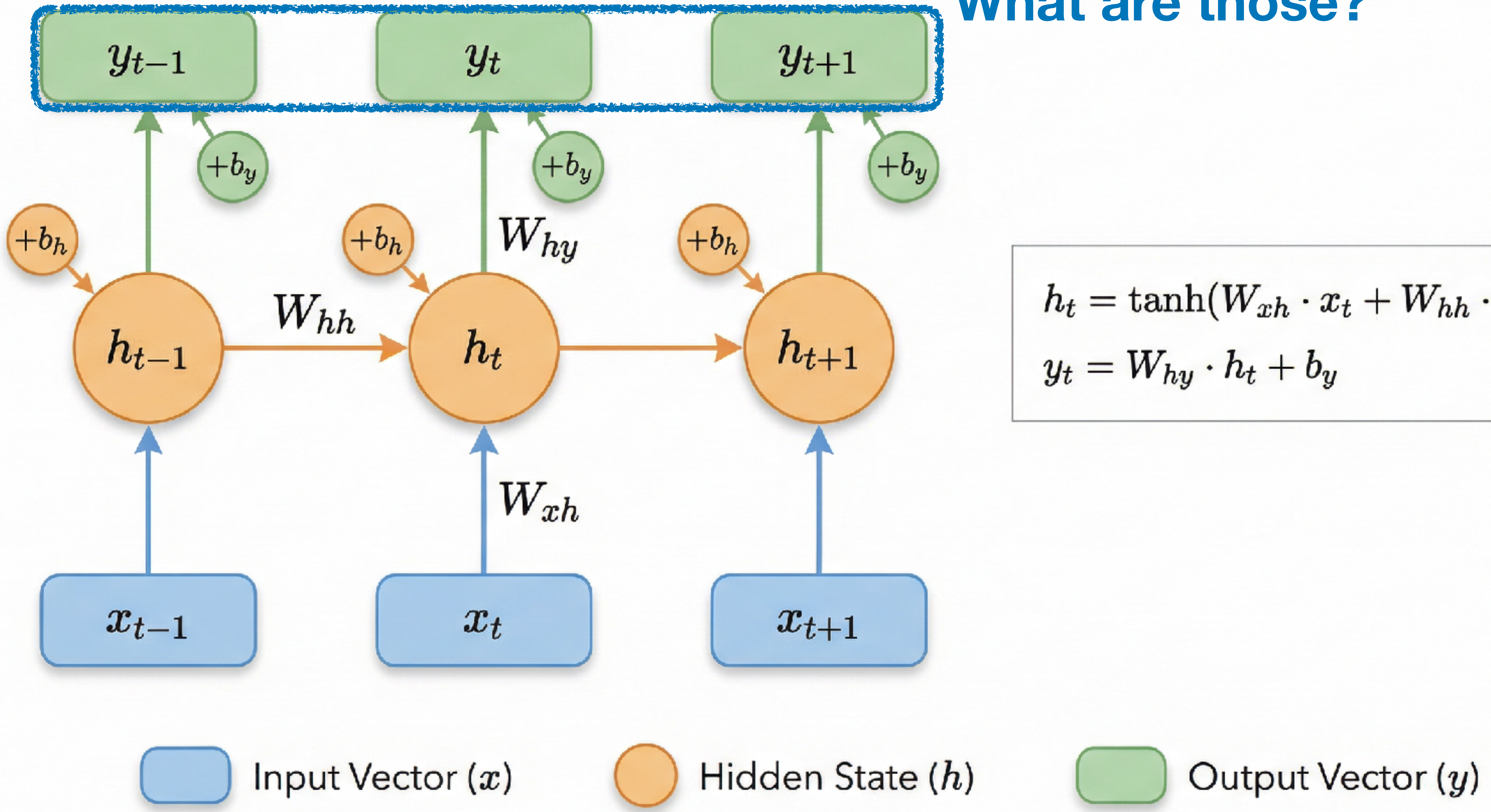
Vanilla RNN Layer



$$h_t = \tanh(W_{xh} \cdot x_t + W_{hh} \cdot h_{t-1} + b_h)$$
$$y_t = W_{hy} \cdot h_t + b_y$$

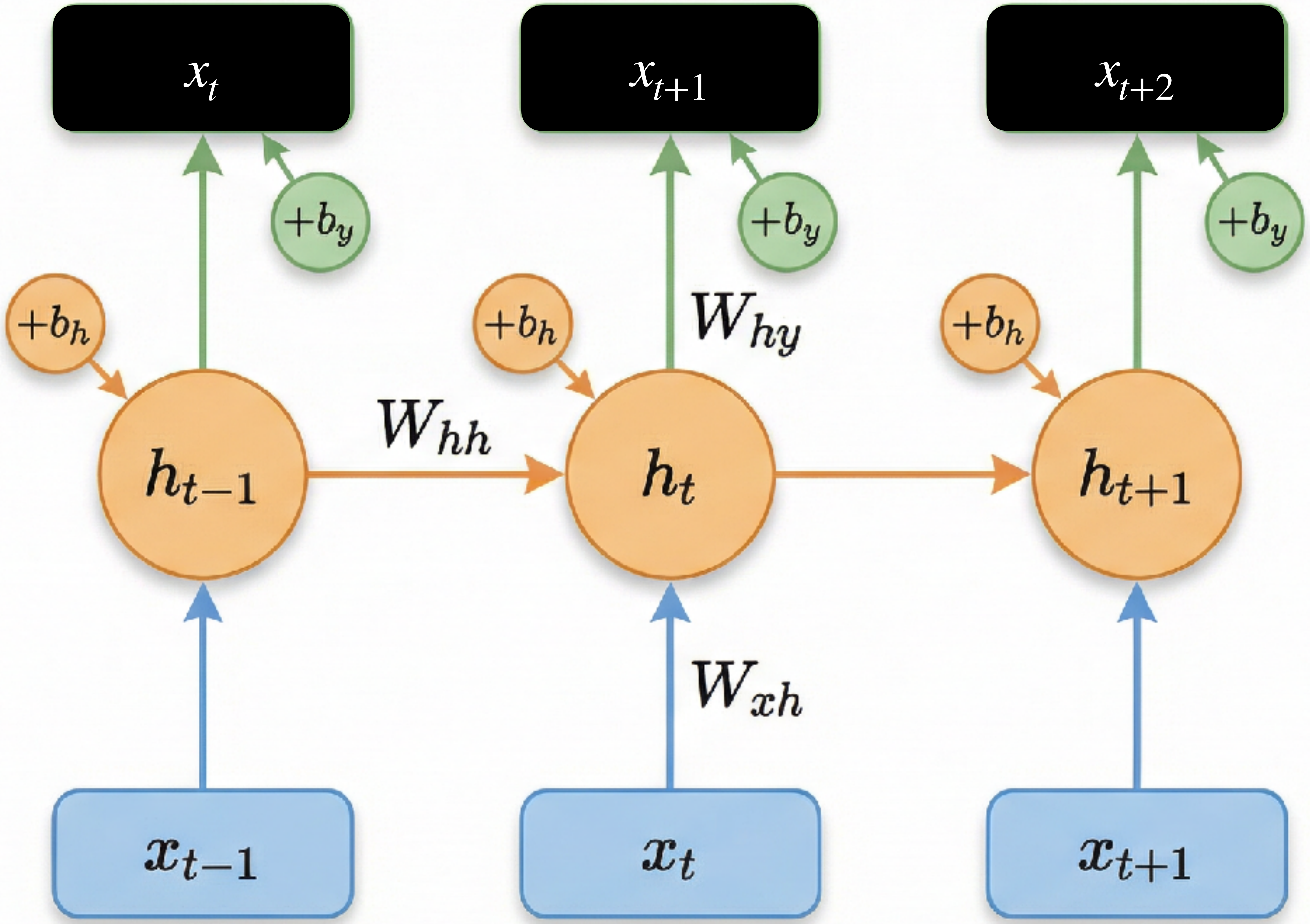
Vanilla RNN Layer

What are those?



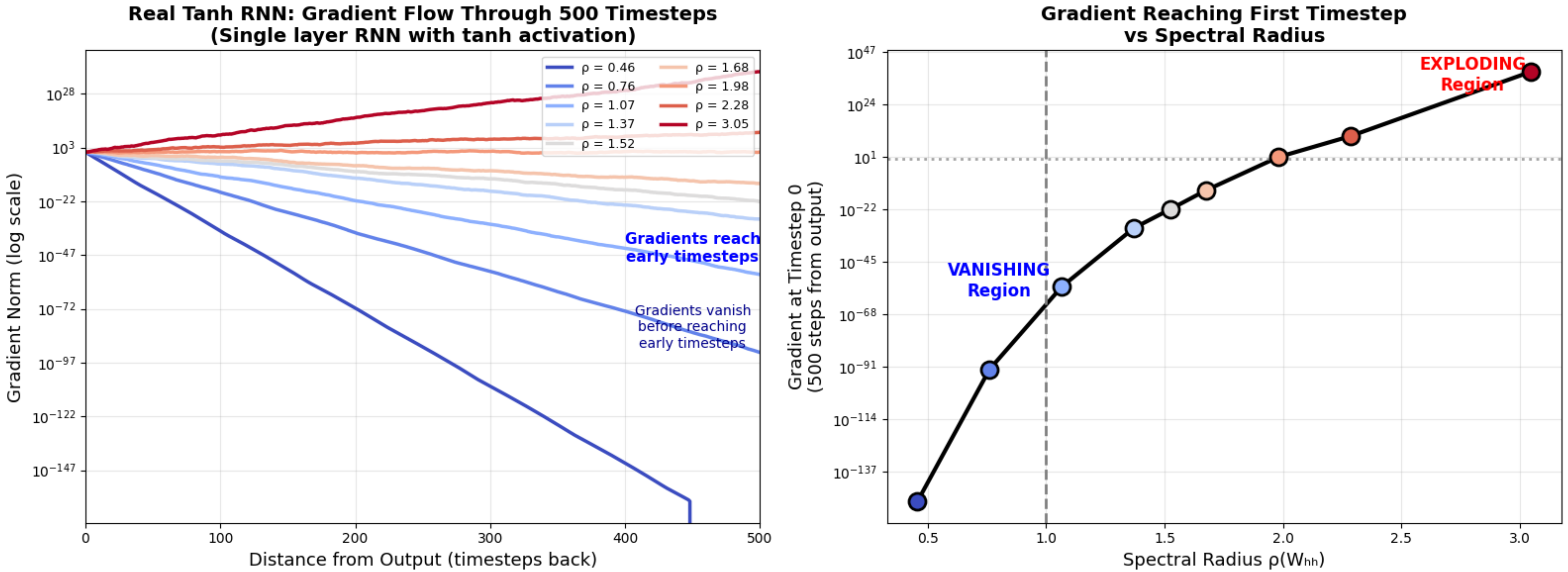
$$h_t = \tanh(W_{xh} \cdot x_t + W_{hh} \cdot h_{t-1} + b_h)$$
$$y_t = W_{hy} \cdot h_t + b_y$$

Vanilla RNN Layer



$$h_t = \tanh(W_{xh} \cdot x_t + W_{hh} \cdot h_{t-1} + b_h)$$
$$y_t = W_{hy} \cdot h_t + b_y$$

Input Vector (x) Hidden State (h) Output Vector (y)


Real Tanh RNN: Gradient Flow Through 500 Timesteps
(Single layer RNN with tanh activation)
Gradient Norm (log scale)
Distance from Output (timesteps back)
ρ = 0.46
ρ = 0.76
ρ = 1.07
ρ = 1.37
ρ = 1.52
ρ = 1.68
ρ = 1.98
ρ = 2.28
ρ = 3.05
Gradients reach early timesteps
Gradients vanish before reaching early timesteps
Gradient Reaching First Timestep vs Spectral Radius
Gradient at Timestep 0 (500 steps from output)
Spectral Radius ρ(Whh)
VANISHING Region
EXPLODING Region

Training Instability: One Gradient Step Destroys Careful Initialization

Mitigating Long Context Issues in RNNs

Any suggestions?

Mitigating Long Context Issues in RNNs

Any suggestions?

Adam EMA will be too slow

Any update can lead to collapse/explosion

Constrain the parameters singular values!

Mitigating Long Context Issues in RNNs

Efficient Orthogonal Parametrisation of Recurrent Neural Networks Using Householder Reflections

Zakaria Mhammedi[1,2] Andrew Hellicar[2] Ashfaqur Rahman[2] James Bailey[1]

Unitary Evolution Recurrent Neural Networks

Martin Arjovsky * MARJOVSKY@DC.UBA.AR
Amar Shah * AS793@CAM.AC.UK
Yoshua Bengio
Universidad de Buenos Aires, University of Cambridge,
Université de Montréal. Yoshua Bengio is a CIFAR Senior Fellow.

projUNN: efficient method for training deep networks with unitary matrices

Bobak T. Kiani
MIT
bkiani@mit.edu

Randall Balestriero
Meta AI, FAIR
rbalestriero@fb.com

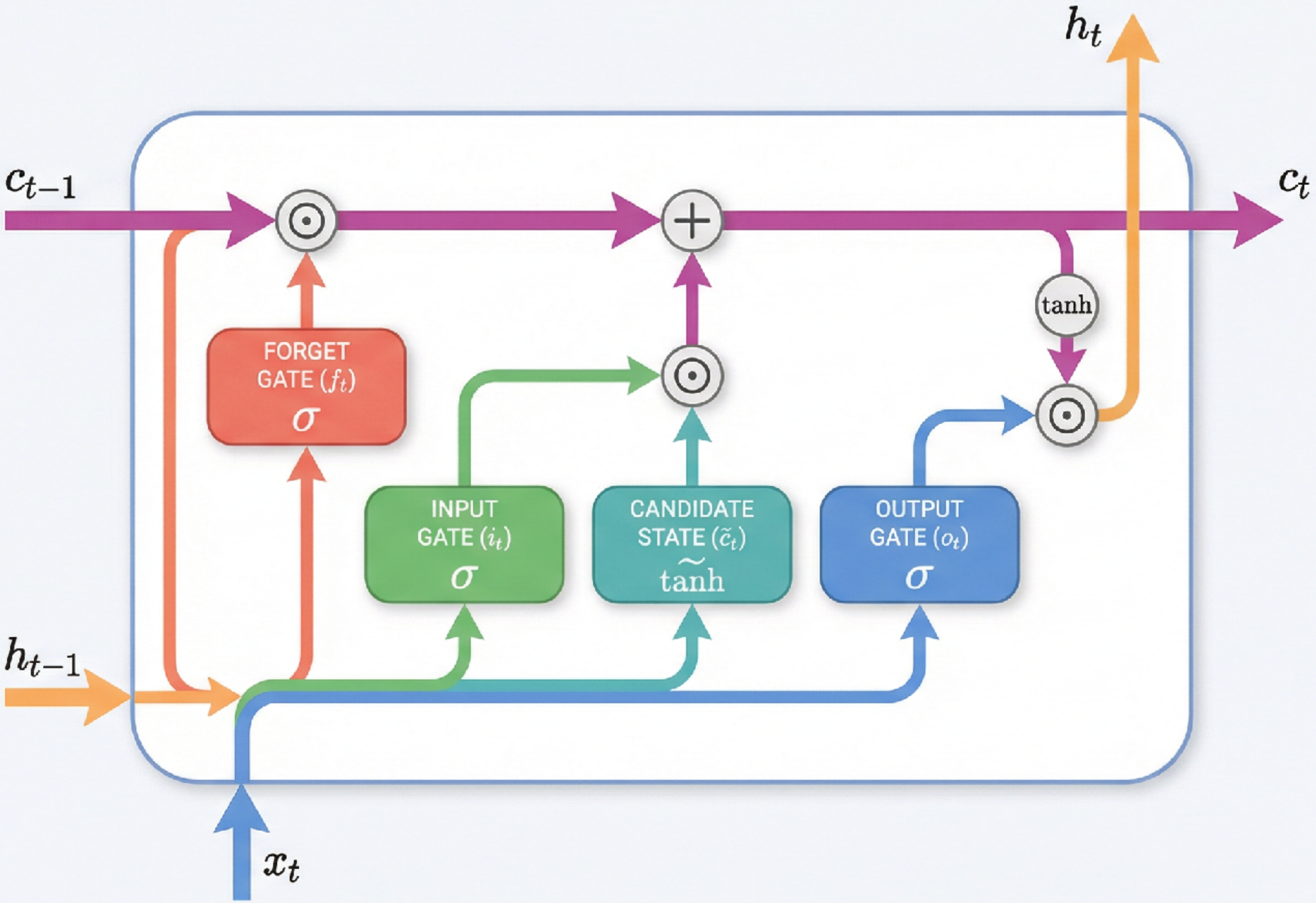
Yann LeCun
NYU & Meta AI, FAIR
yann@fb.com

Seth Lloyd
MIT & Turing Inc.
slloyd@mit.edu

Mitigating Long Context Issues in RNNs

Any idea on how to enforce orthogonal/unitary W?

LSTM Cell



Equations

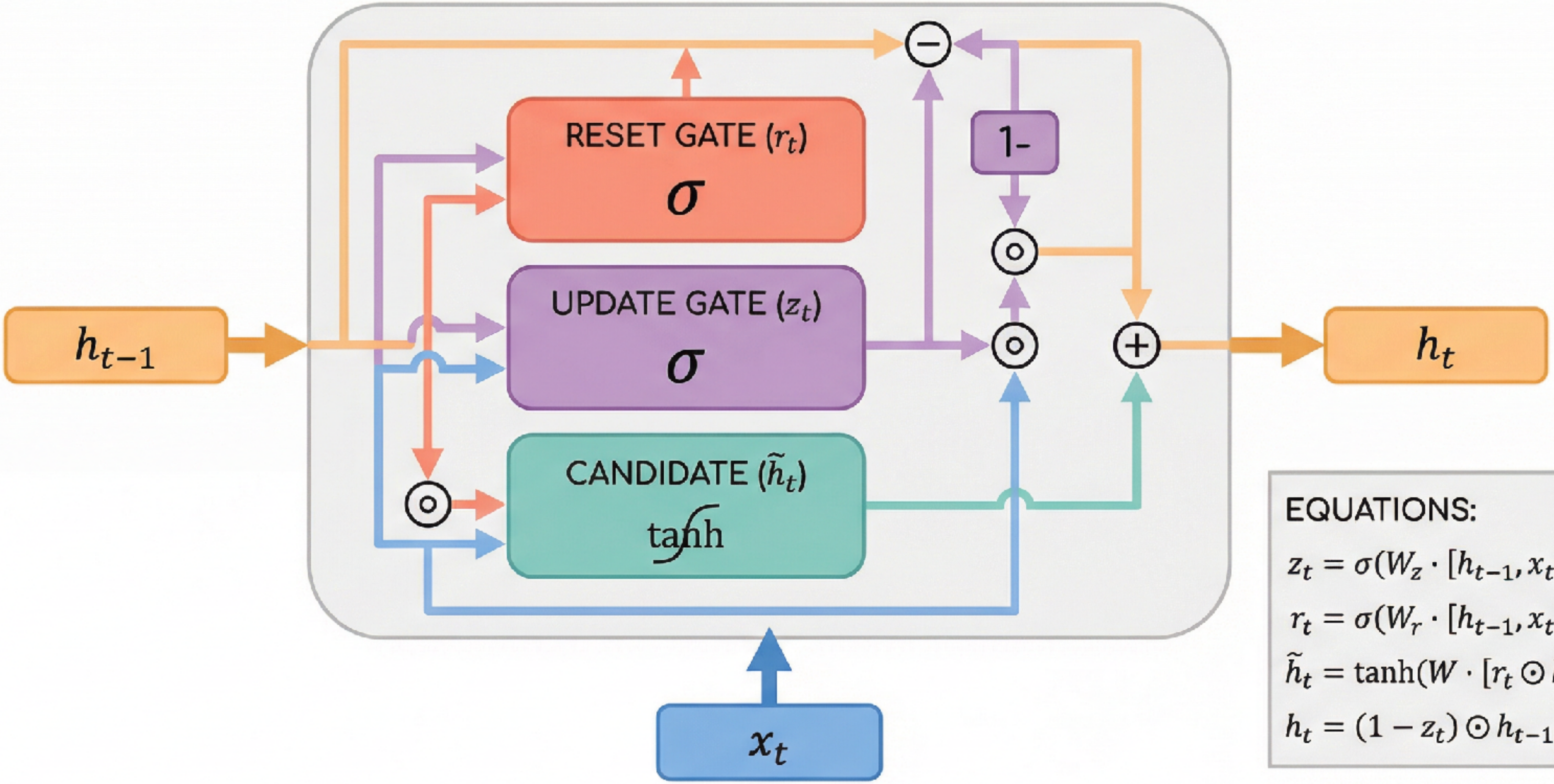
$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$
$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$
$$\tilde{c}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c)$$
$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t$$
$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$$
$$h_t = o_t \odot \tanh(c_t)$$

Legend

FORGET GATE (f_t)	Decides what to discard from cell state
INPUT GATE (i_t)	Decides what new info to store
CANDIDATE STATE ($\tilde{c}_t$)	Proposes new values
OUTPUT GATE (o_t)	Decides what to output
c_{t-1}/c_t (Cell State)	Memory Highway

σ	tanh	$\odot$	$+$
Sigmoid	Hyperbolic Tangent	Element-wise Multiplication	Addition

GRU Cell



EQUATIONS:

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t] + b_z)$$

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t] + b_r)$$

$$\tilde{h}_t = \tanh(W \cdot [r_t \odot h_{t-1}, x_t] + b)$$

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t$$

LEGEND:

- Reset gate: controls how much past to use for candidate - Update gate: controls interpolation between old and new

"GRU: 2 gates, no separate cell state"

Try it!

RNN

```
class torch.nn.RNN(input_size, hidden_size, num_layers=1, nonlinearity='tanh',
bias=True, batch_first=False, dropout=0.0, bidirectional=False, device=None,
dtype=None)
```

[source]

LSTM

```
class torch.nn.LSTM(input_size, hidden_size, num_layers=1, bias=True, batch_first=False,
dropout=0.0, bidirectional=False, proj_size=0, device=None, dtype=None) #
```

[source]

GRU

```
class torch.nn.GRU(input_size, hidden_size, num_layers=1, bias=True, batch_first=False,
dropout=0.0, bidirectional=False, device=None, dtype=None)
```

[source]

Questions?