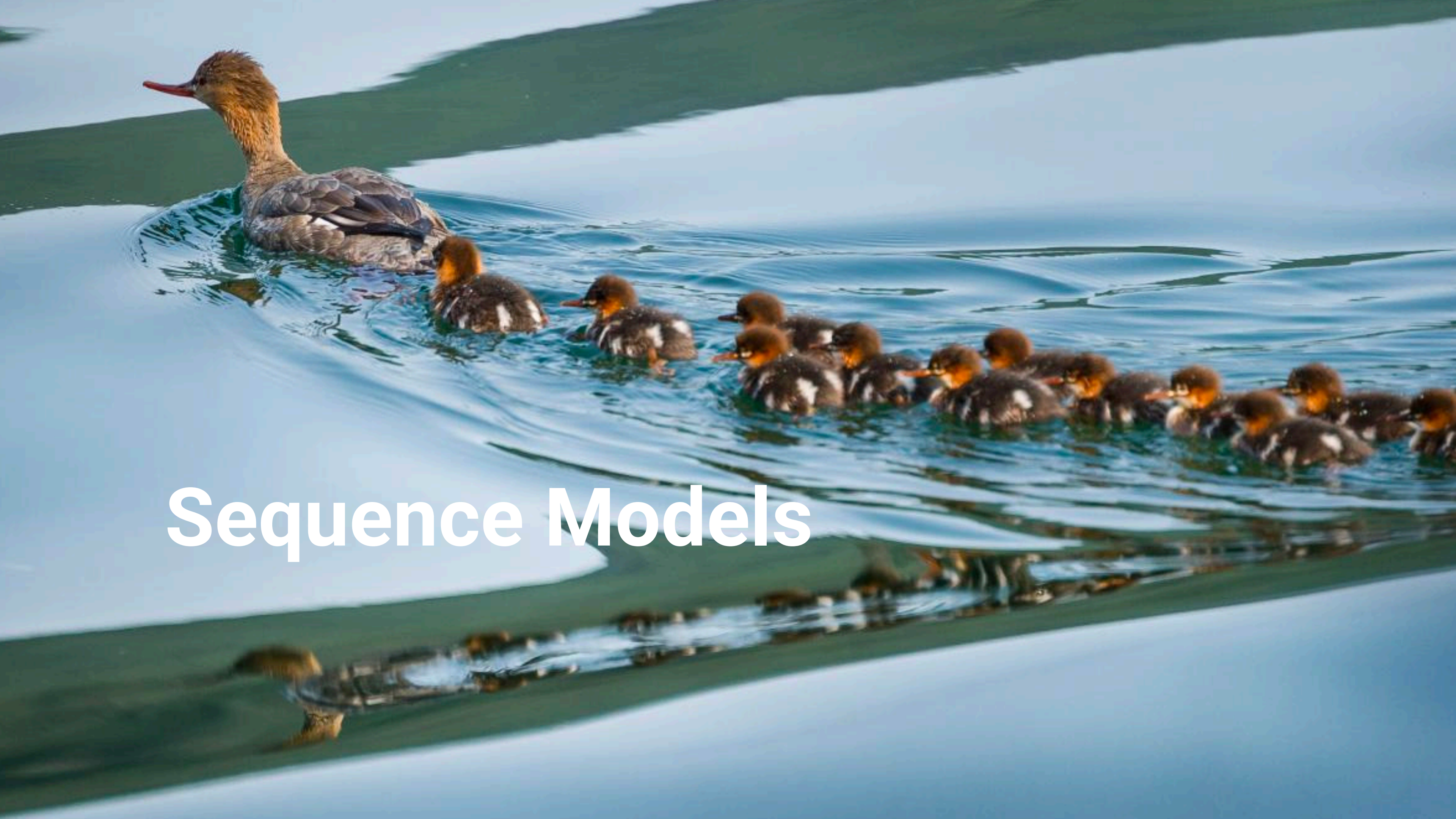
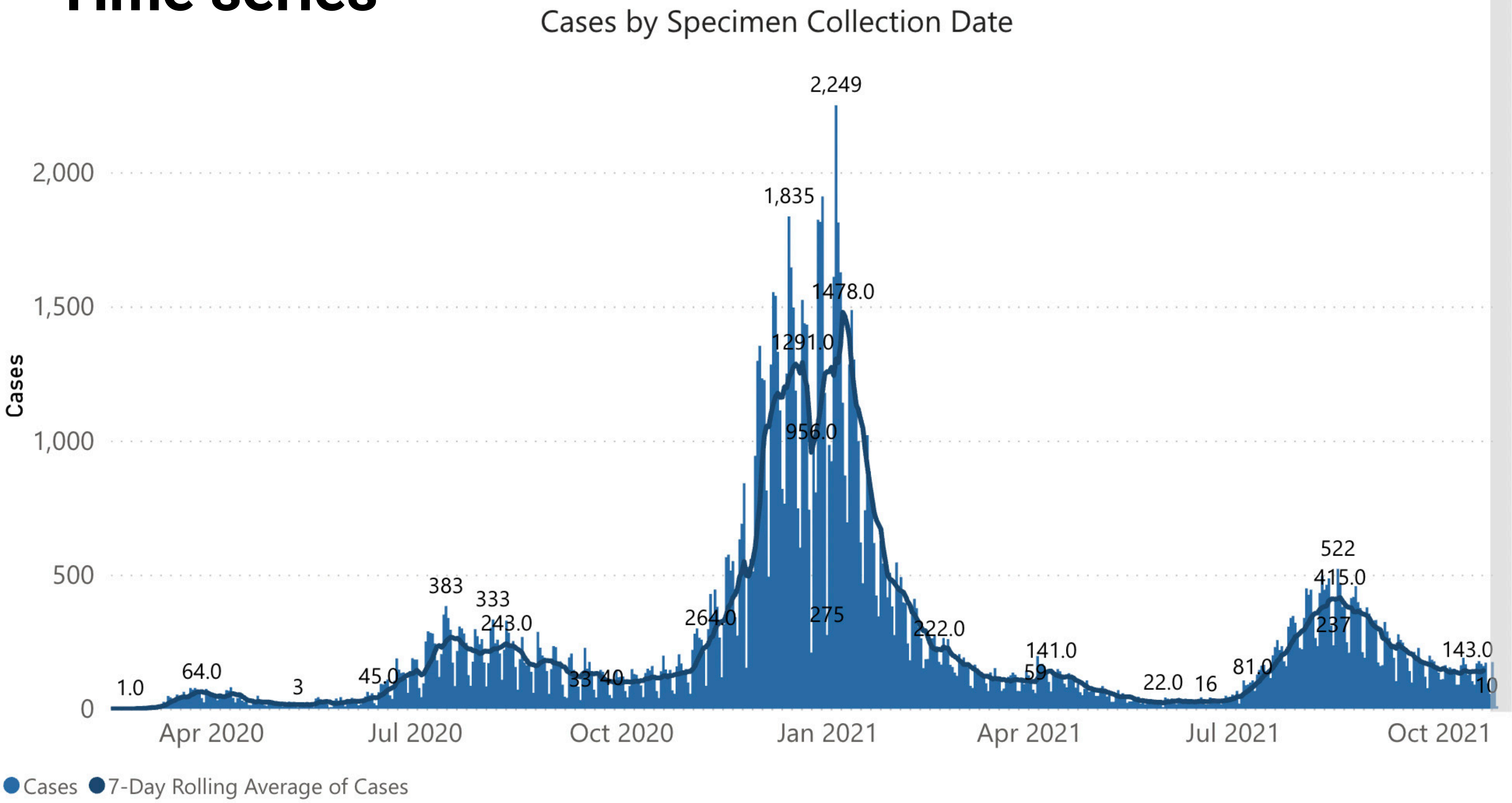


Sequence Models

Time series



Cases by Specimen Collection Date
Cases
2,000
1,500
1,000
500
0
2,249
1,835
1478.0
1291.0
956.0
275
522
415.0
237
383
333
243.0
264.0
222.0
141.0
59
143.0
10
64.0
1.0
3
45.0
33
40
22.0
16
81.0
Apr 2020
Jul 2020
Oct 2020
Jan 2021
Apr 2021
Jul 2021
Oct 2021
Cases
7-Day Rolling Average of Cases

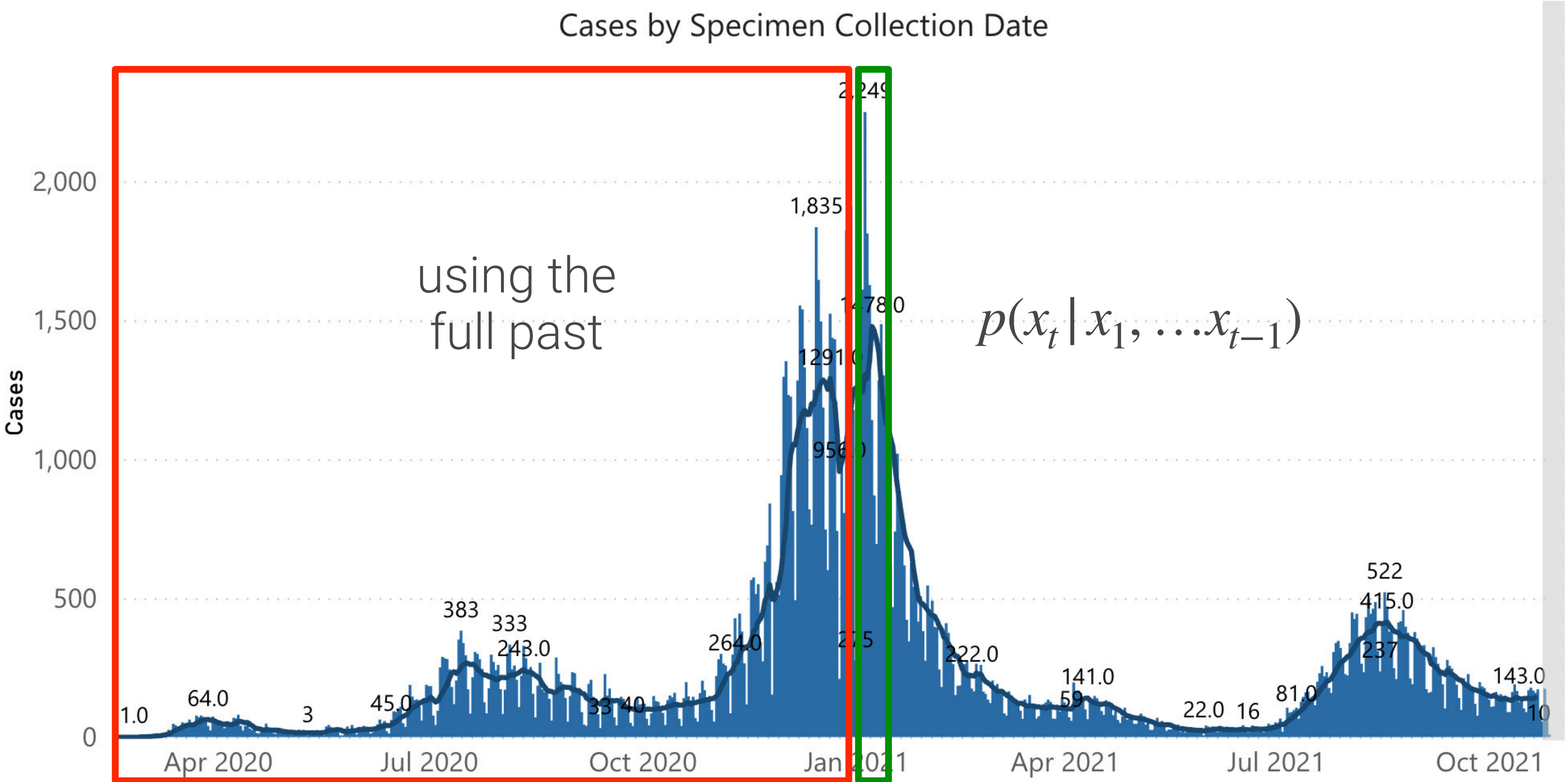
Time Series

- Observations $x_1, x_2, \ldots x_T$
- Joint distribution can always be decomposed via

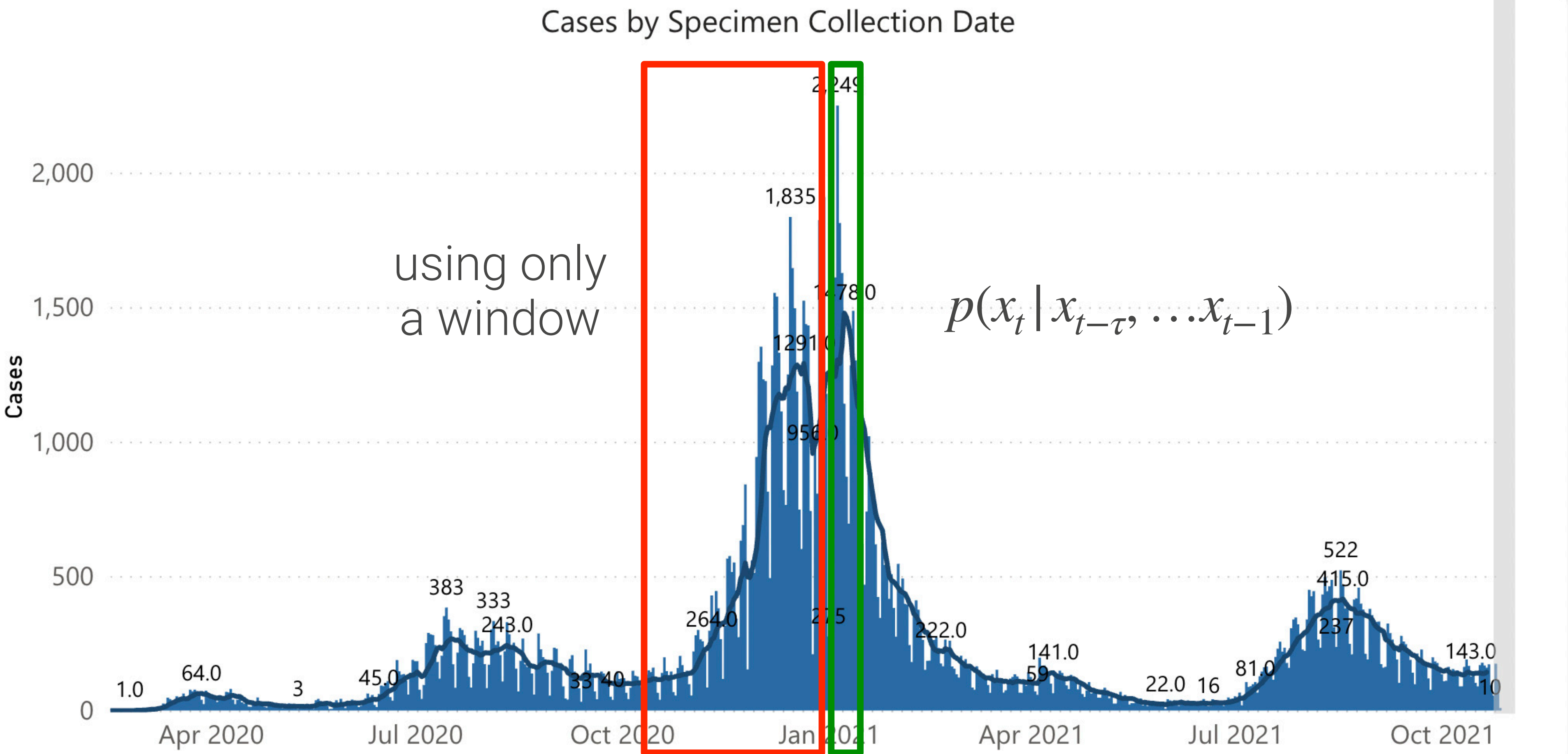
$$p(x_1, x_2, \ldots x_T) = p(x_1)p(x_2 \mid x_1)p(x_3 \mid x_1, x_2)\ldots p(x_T \mid x_1, \ldots x_{T-1})$$

- Causality & time
  Decomposing $p(x)$ forward works better (more accurate) than a backwards decomposition of the same form.
- Can we predict things?

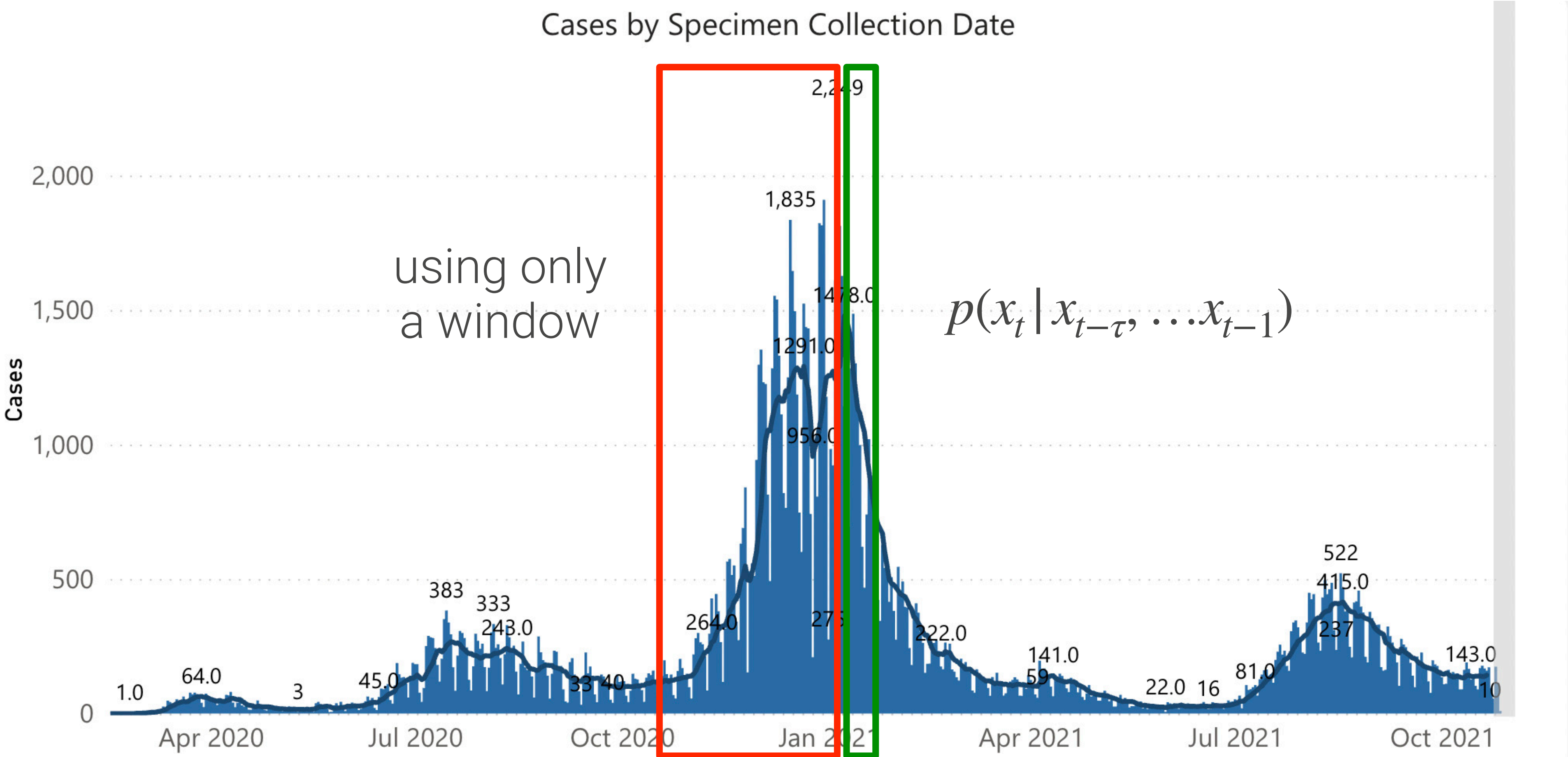
Time Series


Cases by Specimen Collection Date
Cases
2,000
1,500
1,000
500
0
Apr 2020
Jul 2020
Oct 2020
Jan 2021
Apr 2021
Jul 2021
Oct 2021
1,835
522
415.0
383
333
243.0
264.0
222.0
237
141.0
143.0
64.0
45.0
81.0
22.0
16
1.0
3
using the full past
$p(x_t | x_1, \ldots x_{t-1})$

Time Series

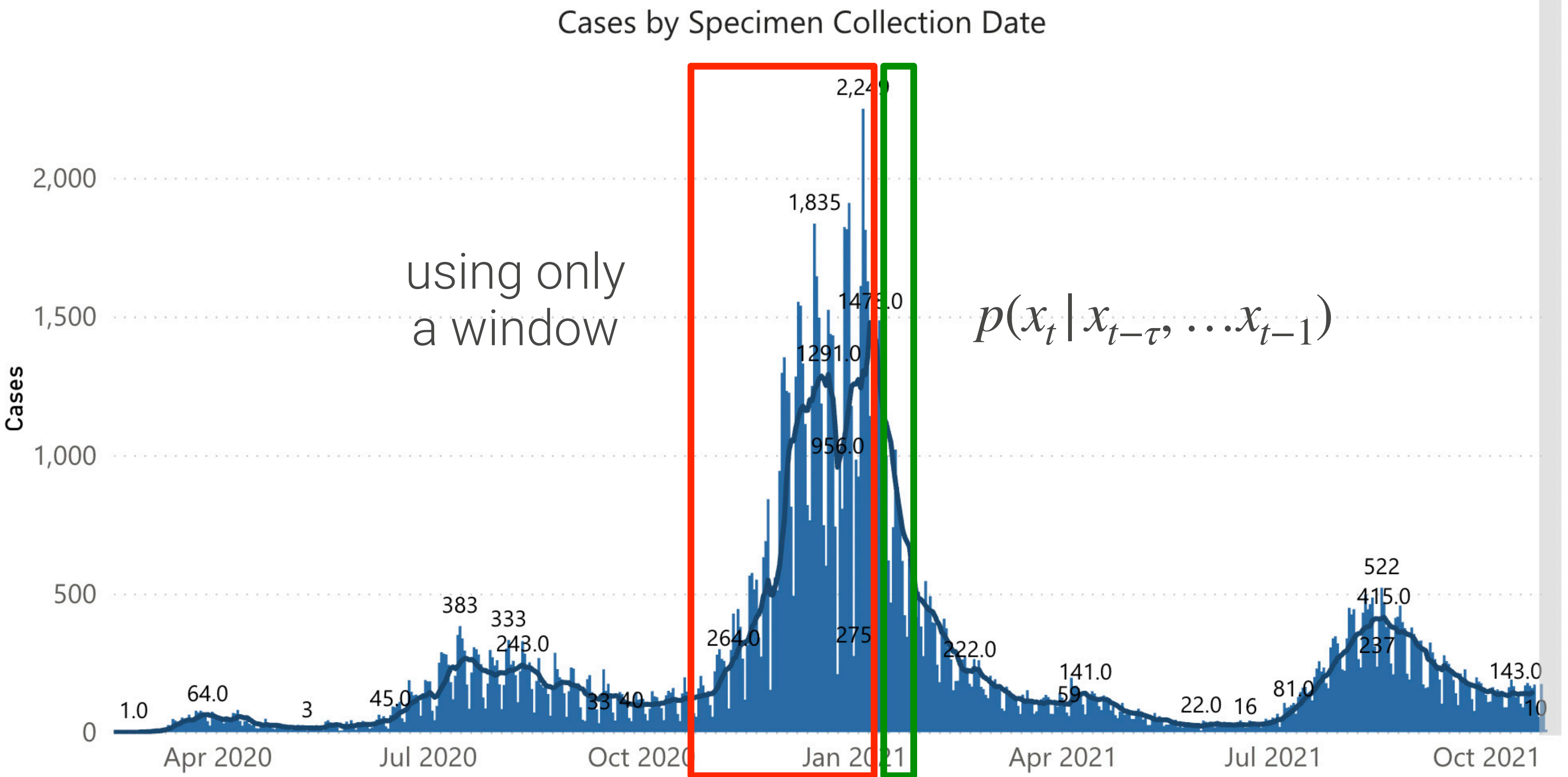

Cases by Specimen Collection Date
Cases
2,000
1,500
1,000
500
0
Apr 2020
Jul 2020
Oct 2020
Jan 2021
Apr 2021
Jul 2021
Oct 2021
1,835
1291.0
956.0
264.0
383
333
243.0
64.0
1.0
45.0
222.0
141.0
22.0
16
81.0
522
415.0
237
143.0
using only
a window
$p(x_t \mid x_{t-\tau}, \ldots x_{t-1})$

Time Series



$$p(x_t \mid x_{t-\tau}, \ldots x_{t-1})$$

Time Series

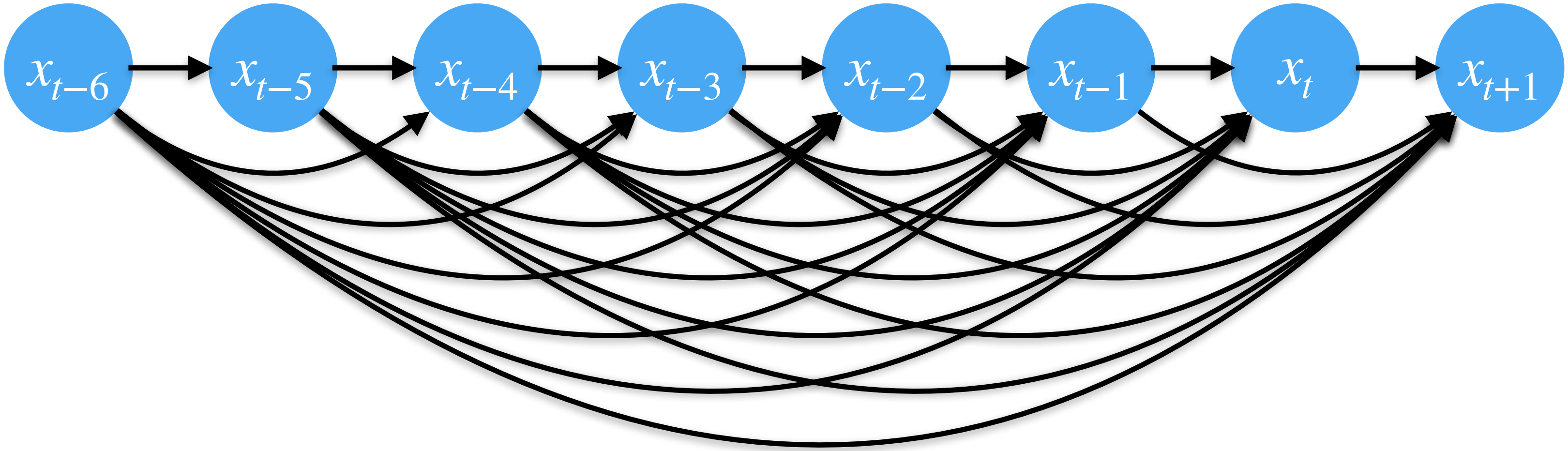


$$p(x_t \mid x_{t-\tau}, \ldots x_{t-1})$$

Time Series (Autoregressive Variant)

- Autoregressive estimation

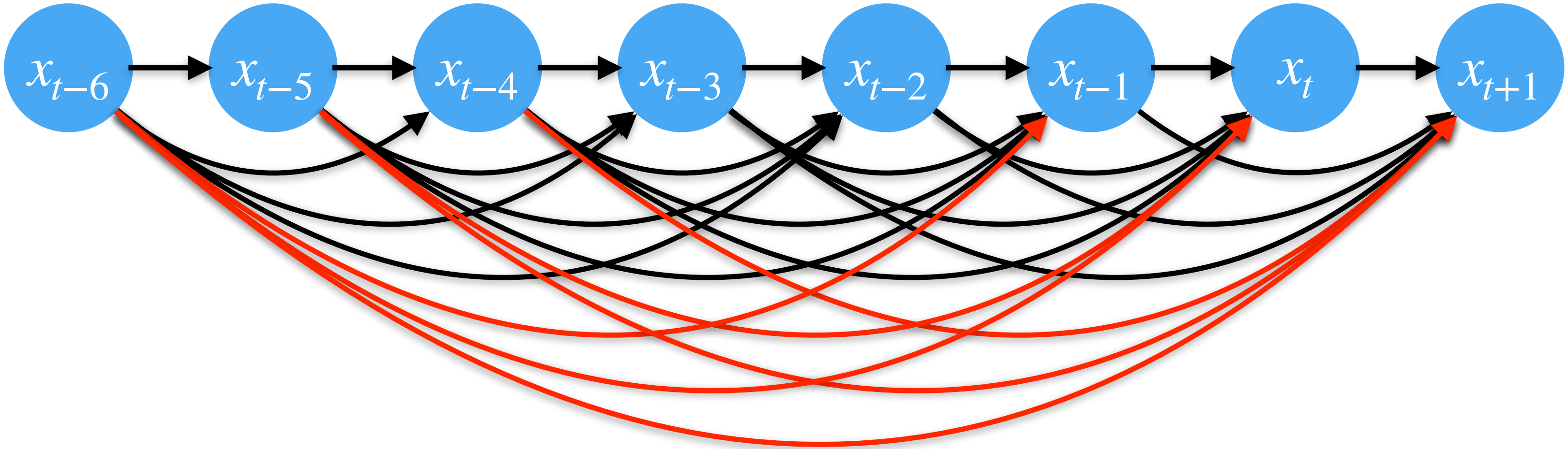
$$x_t \sim p(x_t \mid x_1, \ldots x_{t-1})$$



Time Series (Autoregressive Variant)

- Autoregressive estimation

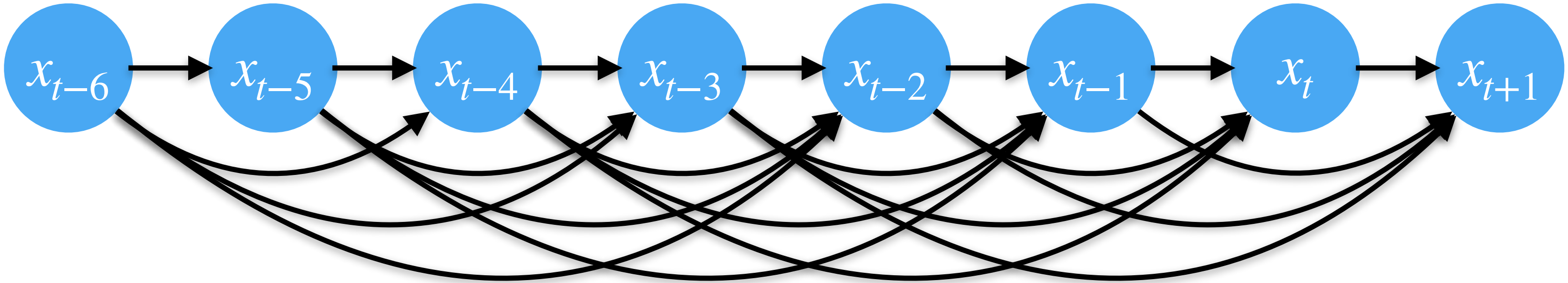
$$x_t \sim p(x_t \mid x_{t-\tau}, \ldots x_{t-1})$$



Time Series (Autoregressive Variant)

- Autoregressive estimation

$$x_t \sim p(x_t \mid x_{t-\tau}, \ldots x_{t-1})$$



- Limit influence to the recent past.
- Taken's theorem: under some regularity conditions using the past τ steps is enough.

Time Series (Autoregressive Variant)

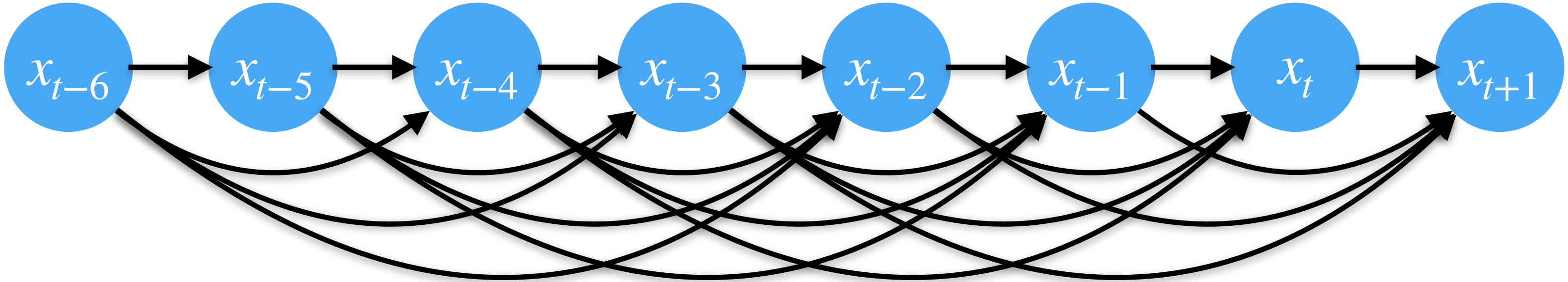
- Autoregressive estimation

$$x_t \sim p(x_t \mid x_{t-\tau}, \ldots x_{t-1})$$

- When does this work?
  - Relevant history about x_t can be found in $(x_{t-\tau}, \ldots x_{t-1})$
  - In practice - we assume that the probability distribution only depends on actual values of $(x_{t-\tau}, \ldots x_{t-1})$ rather than point in time t (stationarity of time series).
- Train regression model for $\bar{y}_t = x_t$ and $\bar{x}_t = (x_{t-\tau}, \ldots, x_{t-1})$

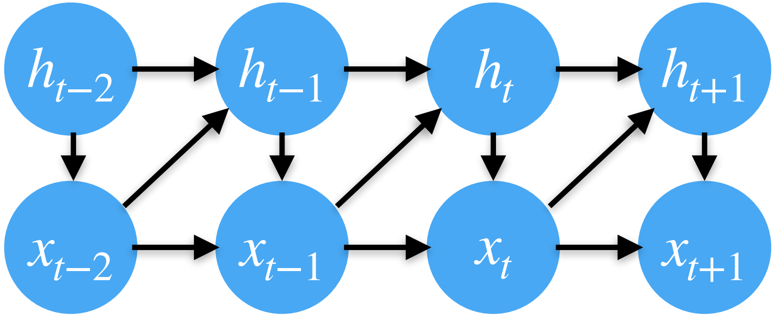
Latent Variables

- Long history might be necessary for good model



- Use latent variable instead to store the history

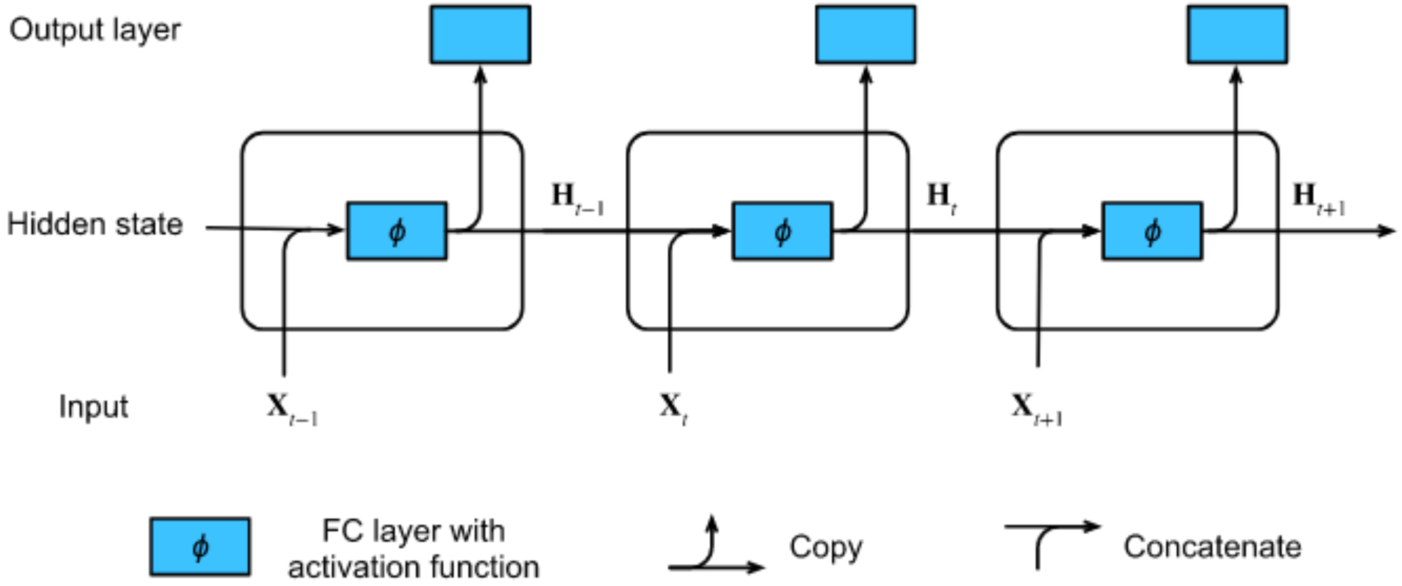
$$h_t = g(x_{t-1}, h_{t-1})$$
$$x_t = f(x_{t-1}, h_t)$$



Plain RNN

g is just a simple deep network

$$h_t = g(x_{t-1}, h_{t-1})$$
$$x_t = f(x_{t-1}, h_t)$$



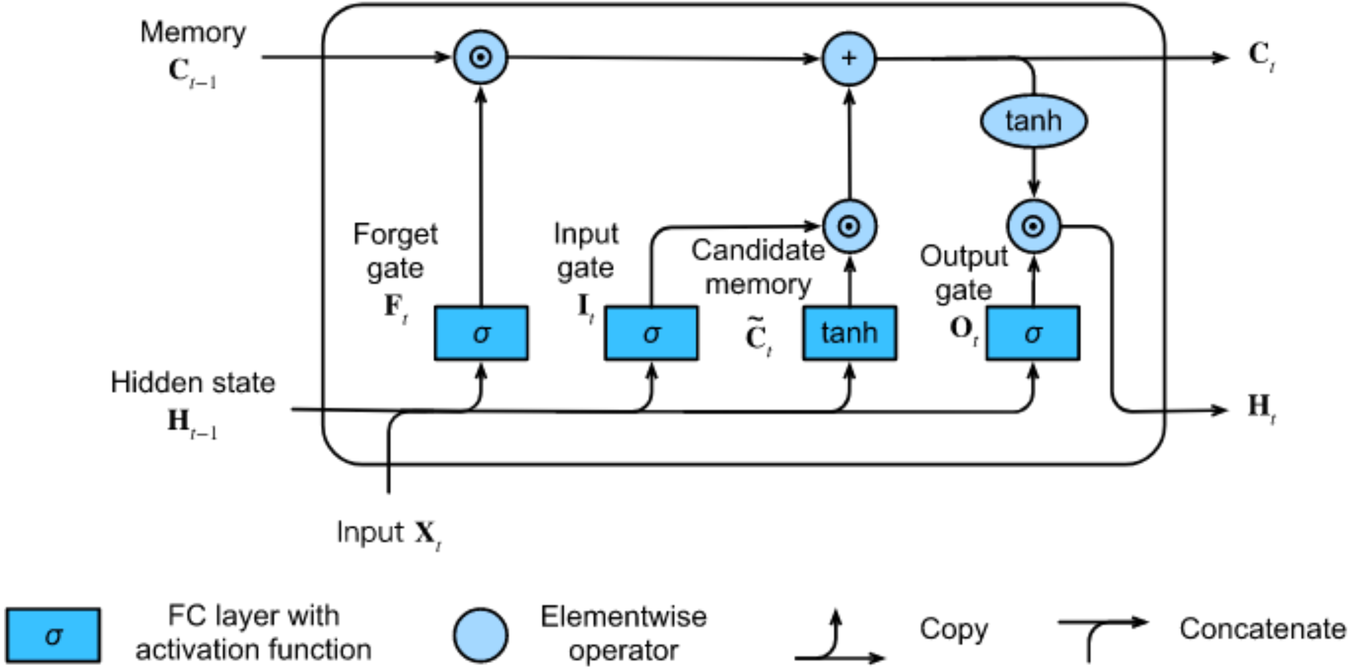
Recursive Neural Network Variants

- **Plain RNN**
  g is just a simple deep network
- **Long Short Term Memory** (Hochreiter&Schmidhuber '98)
  g is a complex memory device to remember past state
- **Gated Recurrent Unit** (Cho et al '14)
  g is a slightly less complex memory device to remember past state (and works typically slightly worse)

$$h_t = g(x_{t-1}, h_{t-1})$$
$$x_t = f(x_{t-1}, h_t)$$

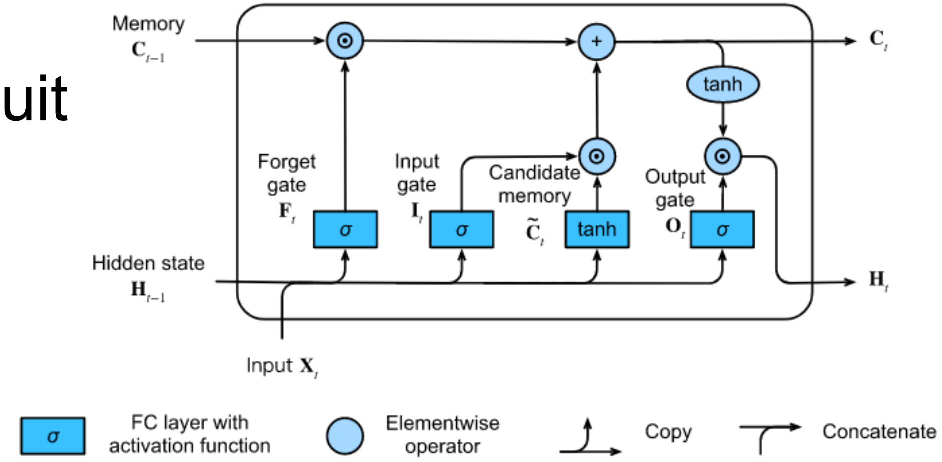
Long Short Term Memory (Hochreiter & Schmidhuber '98)

- Mimic memory cell in a circuit



Long Short Term Memory (Hochreiter & Schmidhuber '98)

- Mimic memory cell in a circuit

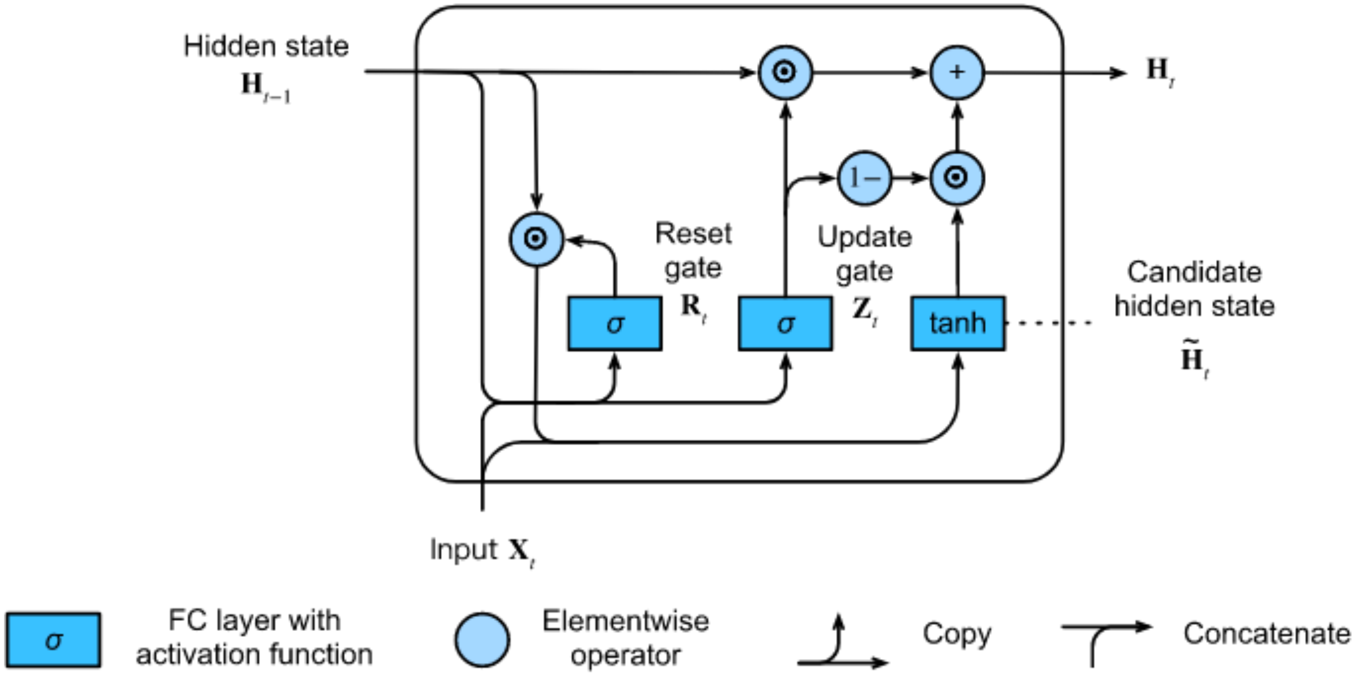


$$i_t = \sigma(W_i(x_t, h_{t-1}) + b_i)$$
$$f_t = \sigma(W_f(x_t, h_{t-1}) + b_f)$$
$$o_t = \sigma(W_o(x_t, h_{t-1}) + b_o)$$
$$c_t = f_t \cdot c_{t-1} i_t \cdot \tanh(W_c(x_t, h_{t-1}) + b_c)$$
$$h_t = o_t \cdot \tanh c_t$$

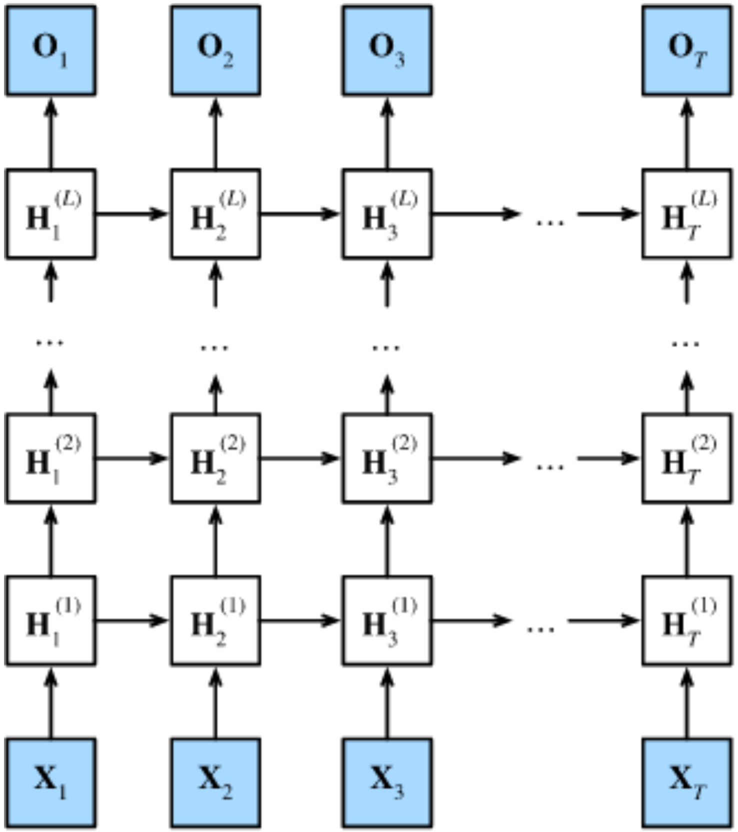
- Different variants exist (e.g. for output gate)

Gated Recurrent Unit (Cho et al. '14)

Simplified state relative to LSTM (faster, smaller)



Using RNNs with Hidden State

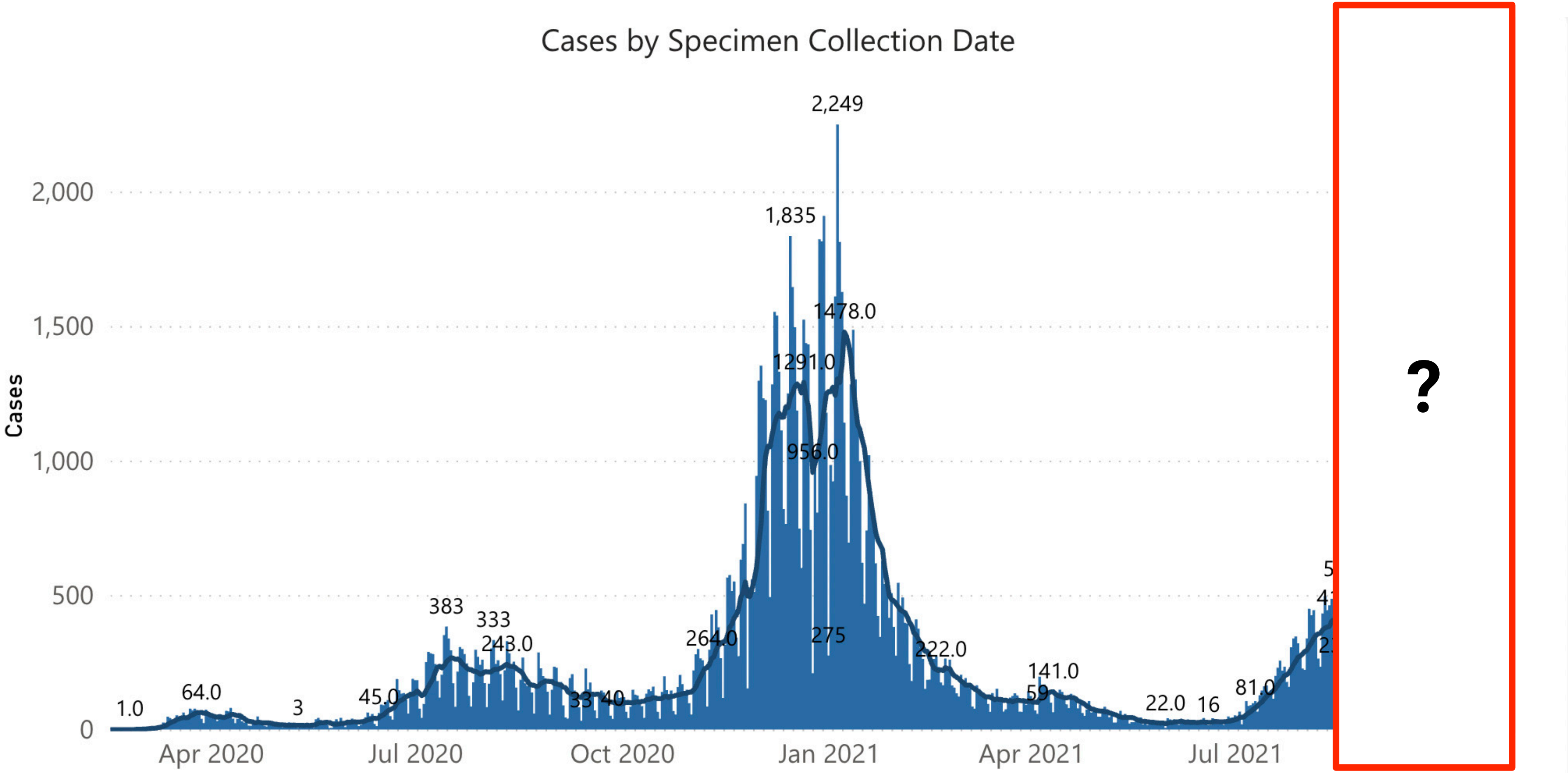


- Stack multiple layers of hidden state
  (deep and simple is better than shallow and complex)
- Training can be expensive (back-propagation through long chain)
- In practice, truncate gradient to avoid expensive chain

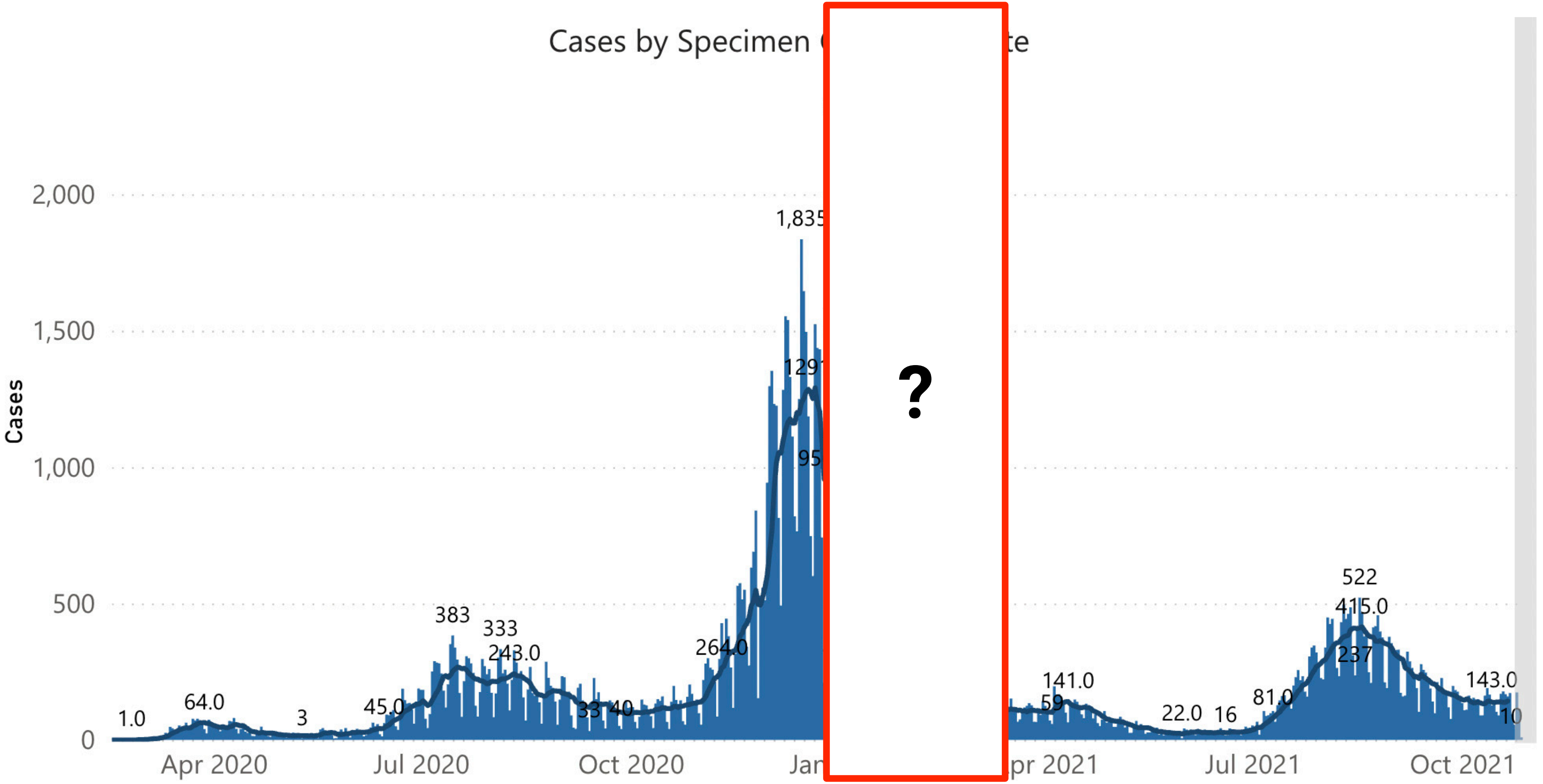
Use framework defaults

Pitfalls

Interpolation vs. **Prediction**



Interpolation vs. Prediction



Training a sequence model

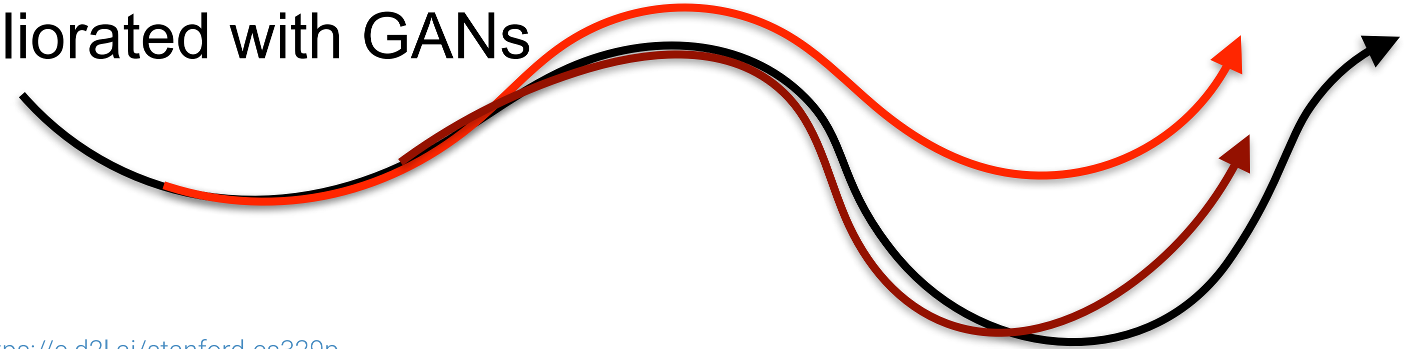
- **IID Data**
  - Random partition into training / validation (and test)
  - Equally reliable estimates (and bagging, too)
- **Dependent data**
  - **Don’t train on the future the predict the past**
  - Train on $x_1, \ldots, x_t$ and estimate $x_{t+1}, \ldots x_T$
  - Stationarity of time series important if you want to use all past history. Otherwise need to fix covariate drift.

Example - Product Recommendations

- **Time series**
  - Stationarity (Christmas comes every year)
  - Nonstationarity (MacBook Pro 14 in 2021)
- **Caution**
  - Concept shift/drift (e.g. COVID-19 related change to purchase more durable goods vs. eating out)
  - External causes for nonstationarity. Conditioned on that, we might have independent data (e.g. umbrella sales governed by weather).

Teacher vs. Student Forcing

- Autoregressive model $x_t = f(x_{t-\tau}, \ldots x_{t-1})$
- Iterate to get $x_t, x_{t+1}, x_{t+2}, \ldots$
  - Prediction using iterates can lead to rapid divergence.
  - Training on real data ensures that we're only one step away from truth.
- Can be ameliorated with GANs or VAE.

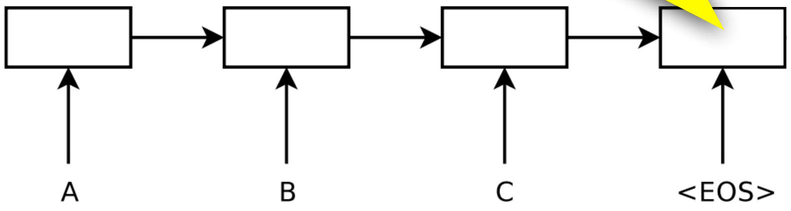



ITERATIVE
OUTPUT

Seq2Seq for Machine Translation, Sutskever, Vinyals, Le ‘14

- Encode source sequence s via LSTM to representation $\phi(s)$
- Decode to target sequence one character at a time

$\phi(s)$

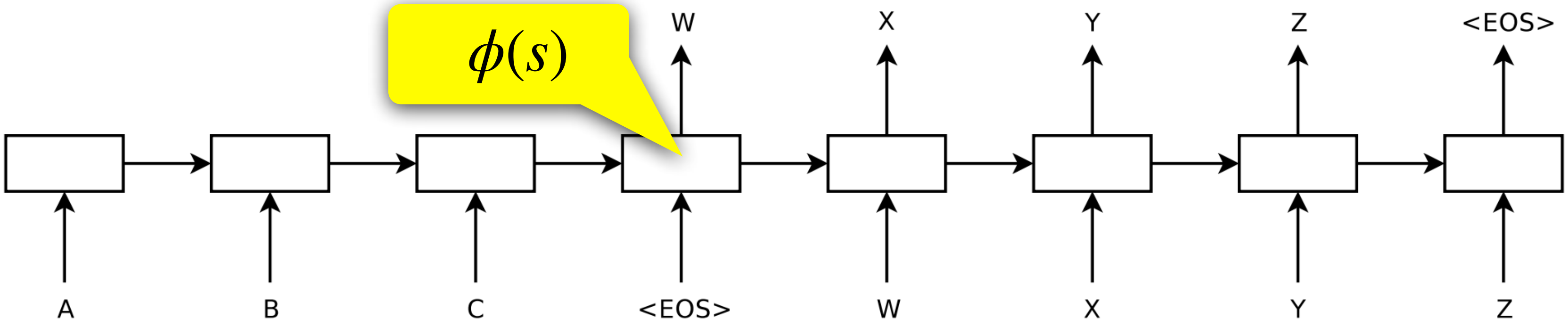


- ‘The table is round.’ - ‘Der Tisch ist rund.’
- ‘The table is very beautiful with r ,
  blah blah blah blah’ - ‘Error … ’

Representation not rich enough

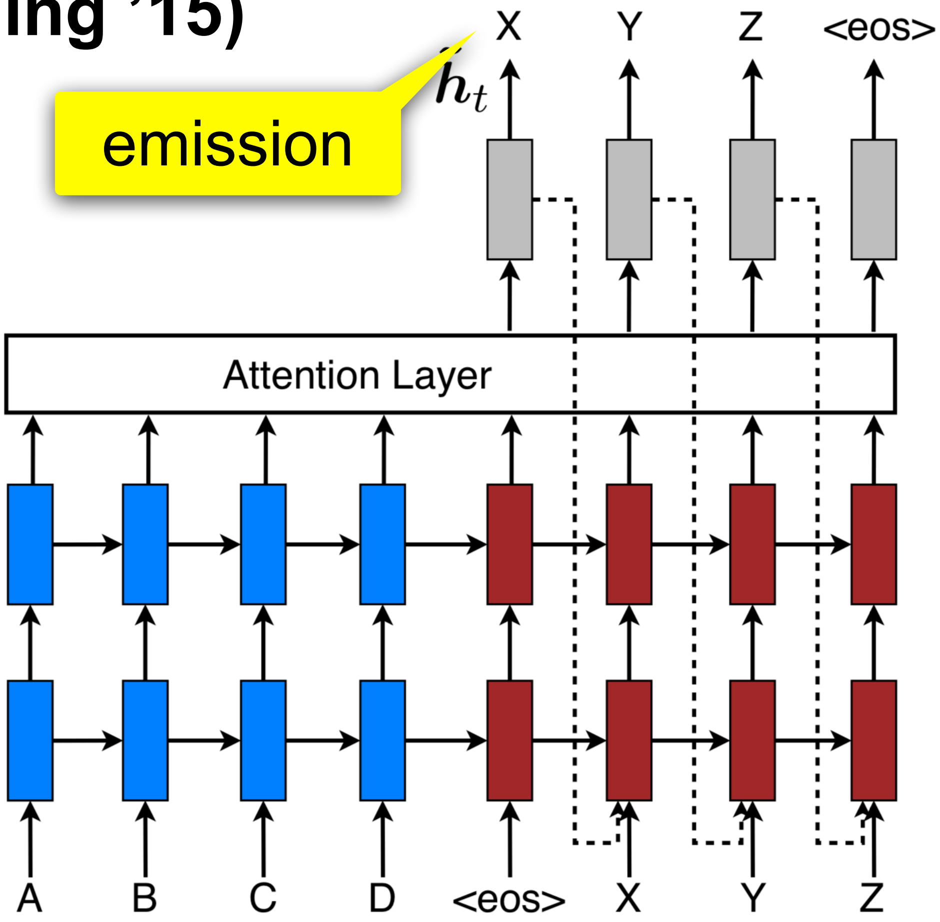
Seq2Seq for Machine Translation, Sutskever, Vinyals, Le ‘14

- Encode source sequence s via LSTM to latent representation $\phi(s)$
- Decode to target sequence one character at a time



- Need memory for long sequences
- Attention to iterate over source
  (we can look up source at any time after all)

Seq2Seq with attention (Bahdanau, Cho, Bengio ’14) (Pham, Luong, Manning ’15)



Seq2Seq with attention (Bahdanau, Cho, Bengio ’14) (Pham, Luong, Manning ’15)

$$\alpha_{ij} \propto \exp(a(\tilde{h}_{i-1}, h_j))$$

query

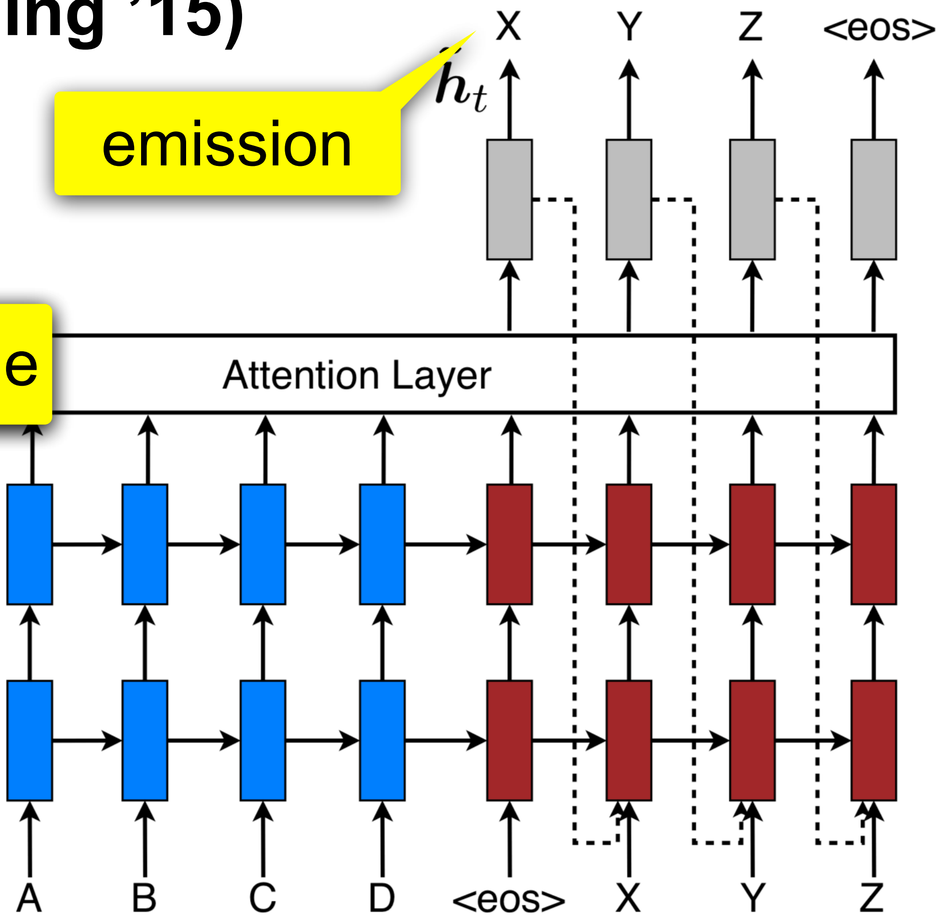
key & value

$$c_i = \sum_{j=1}^{n} \alpha_{ij} h_j$$

$$\tilde{h}_i = f(s_{i-1}, y_{i-1}, c_i)$$

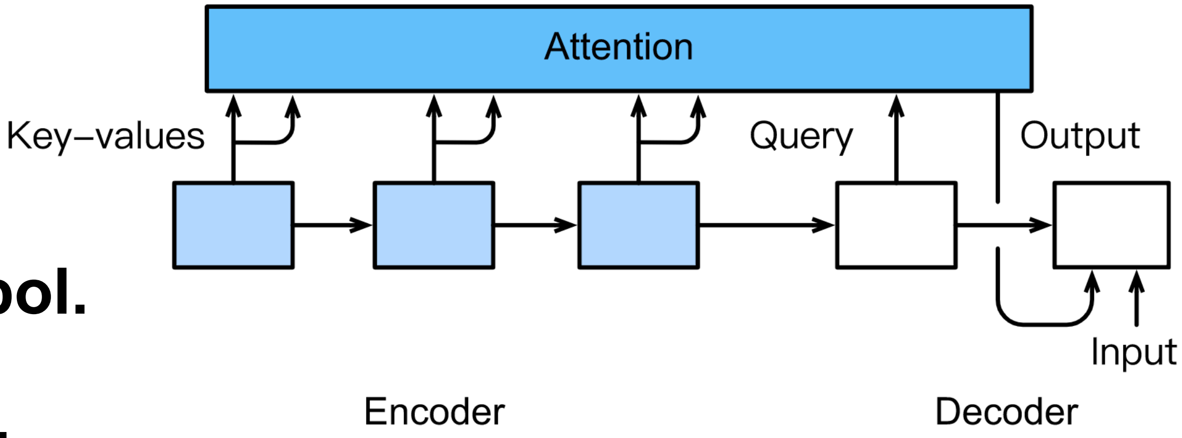
$$y_i$$

emission

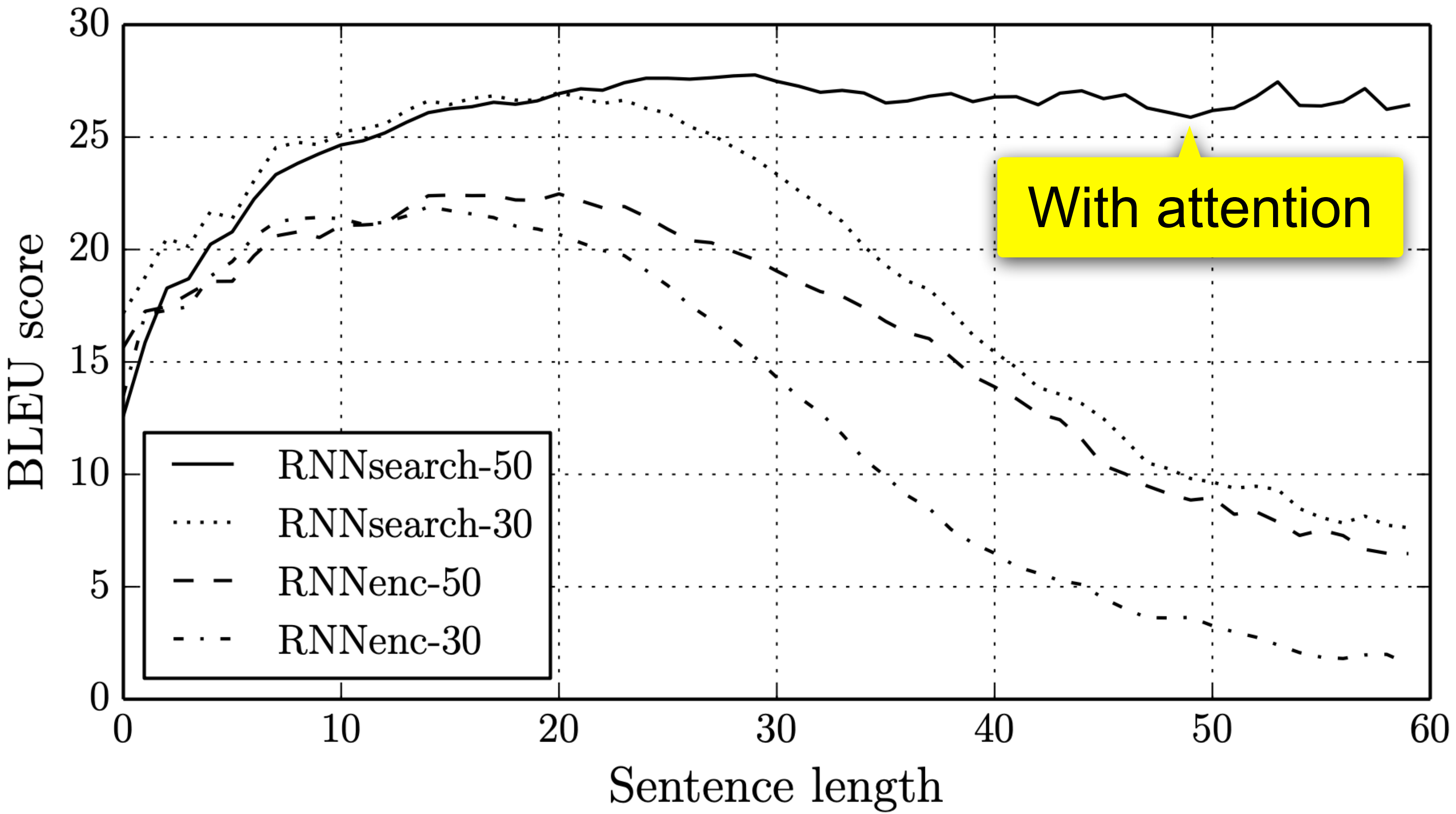


Seq2Seq with attention (Bahdanau, Cho, Bengio ’14) (Pham, Luong, Manning ’15)

- Iterative attention model
  - Compute (next) attention weights
  - Aggregate next state
  - Emit next symbol
- Repeat
- **Memory networks emit only one symbol.**
- **NMT with attention emits many symbols.**



Seq2Seq with attention (Bahdanau, Cho, Bengio ’14)



Lots more to come (Lecture 10++)

- Sequence models require long history
- Expensive to store and train
- Expensive to compute

- Use representation of sequence directly
- Use attention to compute state
- Can use bidirectional strategy naturally (simply attend to past and future) for sequence embeddings.