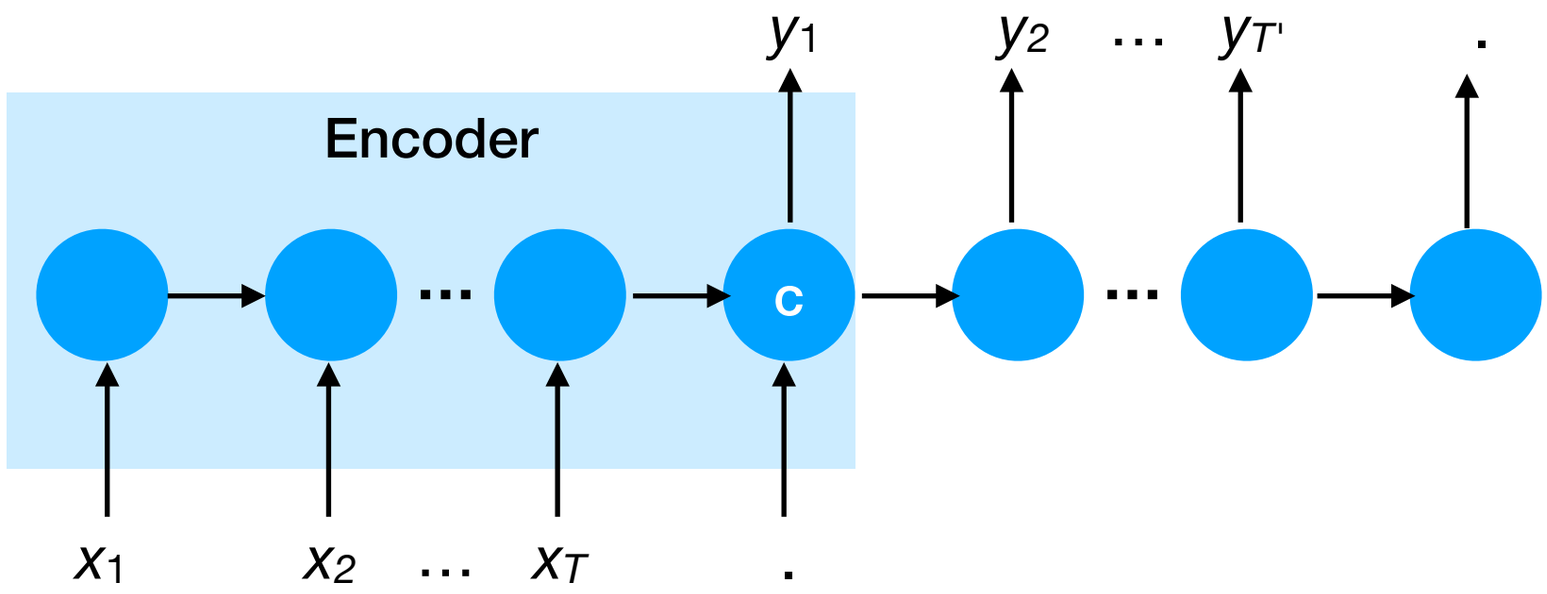
CS/DS 541: Class 21

Jacob Whitehill

seq2seq models

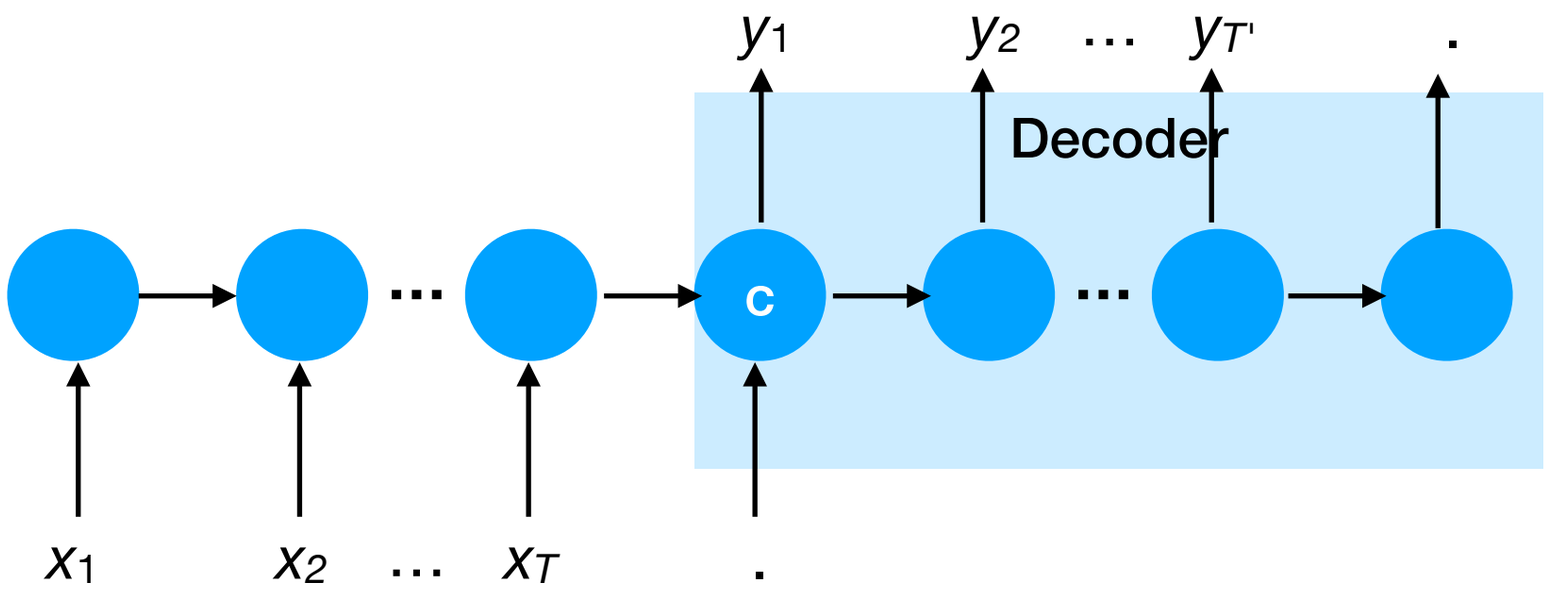
Sequence-to-sequence translation model (seq2seq)

- Recall the seq2seq model from Sutskever et al. (2014) in which the "context" **c** (that encapsulates the meaning of the input sentence) is the last hidden state of the encoder:



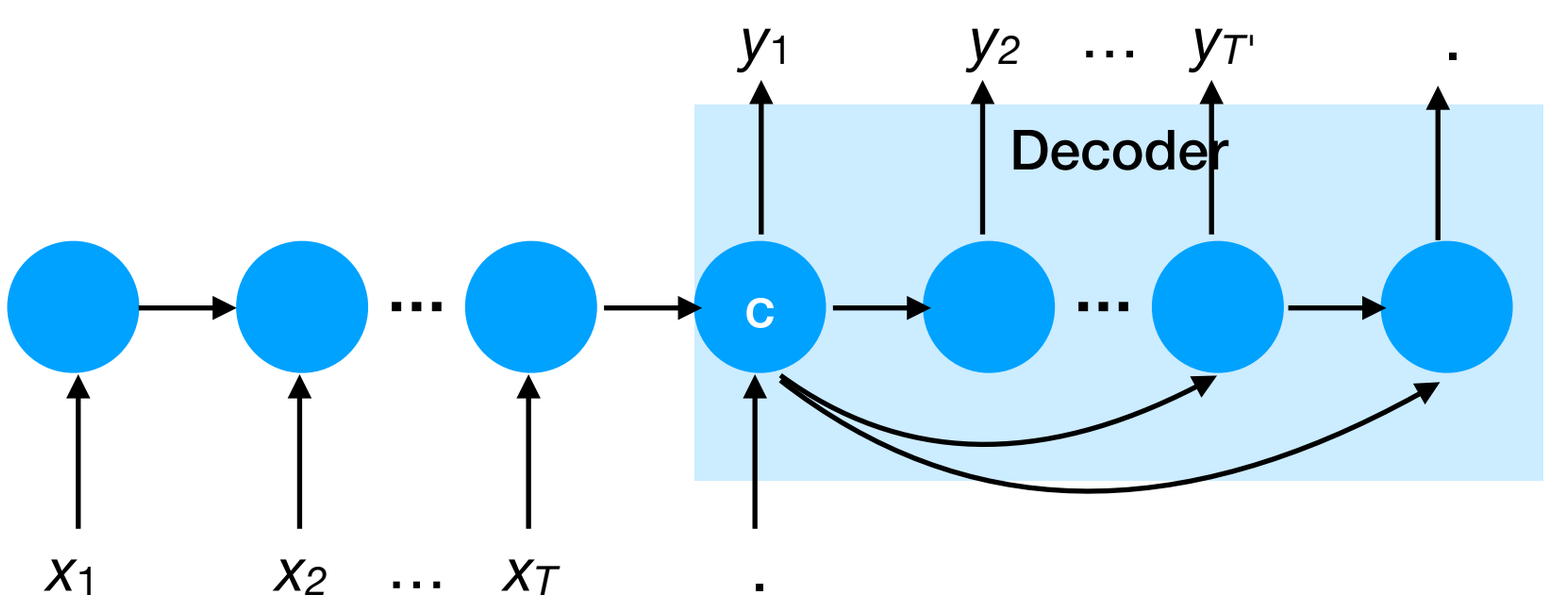
Sequence-to-sequence translation model (seq2seq)

- Recall the seq2seq model from Sutskever et al. (2014) in which the "context" **c** (that encapsulates the meaning of the input sentence) is the last hidden state of the encoder:



seq2seq models with context

- As an alternative to this model (Cho et al. 2014), we can feed **c** explicitly to *all* timesteps of the decoder:



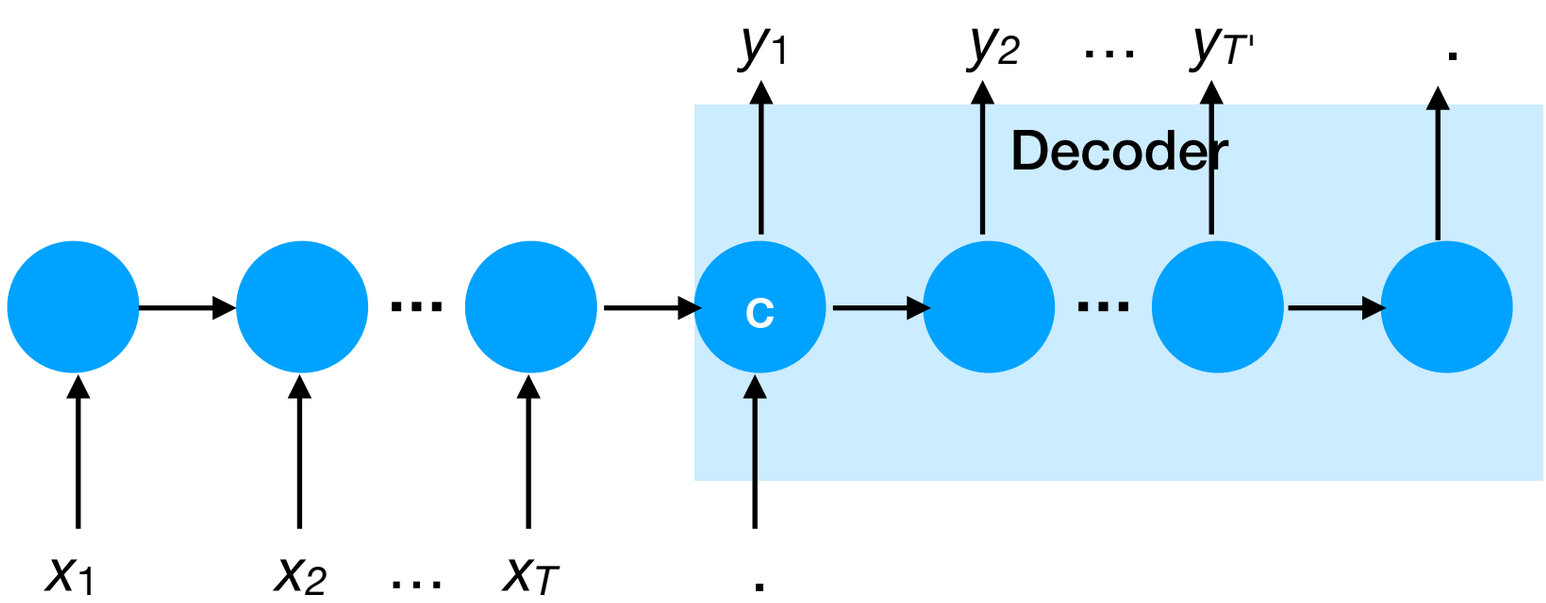
Neural attention models

Attention models

- Seminal paper:
  - Bahdanau et al. 2015

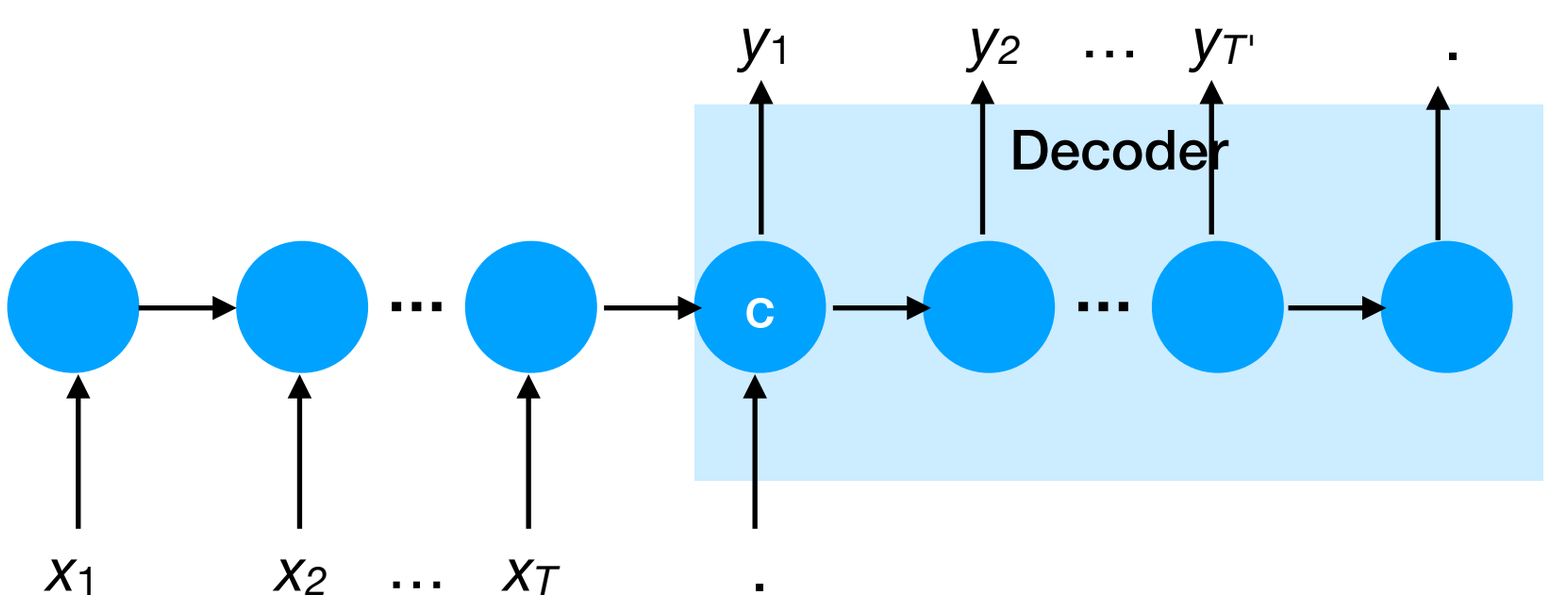
seq2seq models

- While elegantly simple, seq2seq networks may struggle to fit all information about **x** into a single vector **c**.



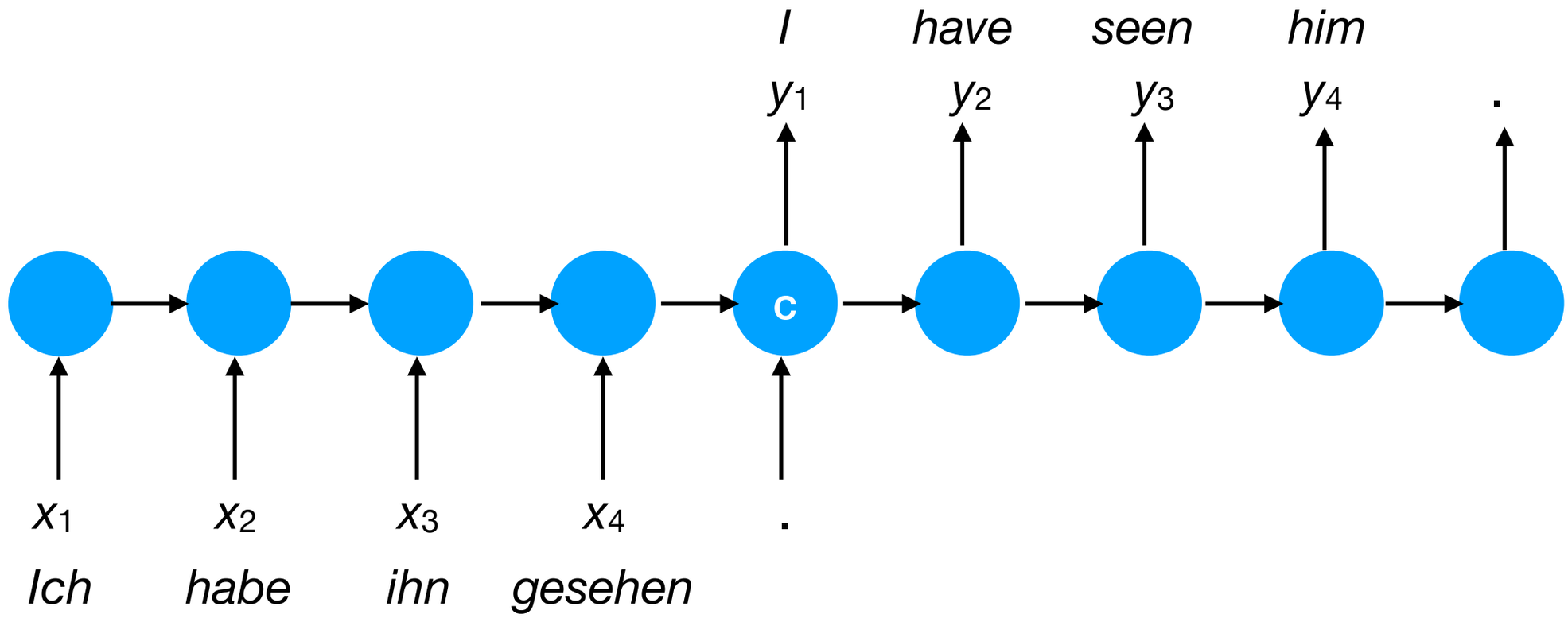
Attention models

- Intuitively, some subsequences of the input may be more relevant than others when producing a particular output symbol y_t.



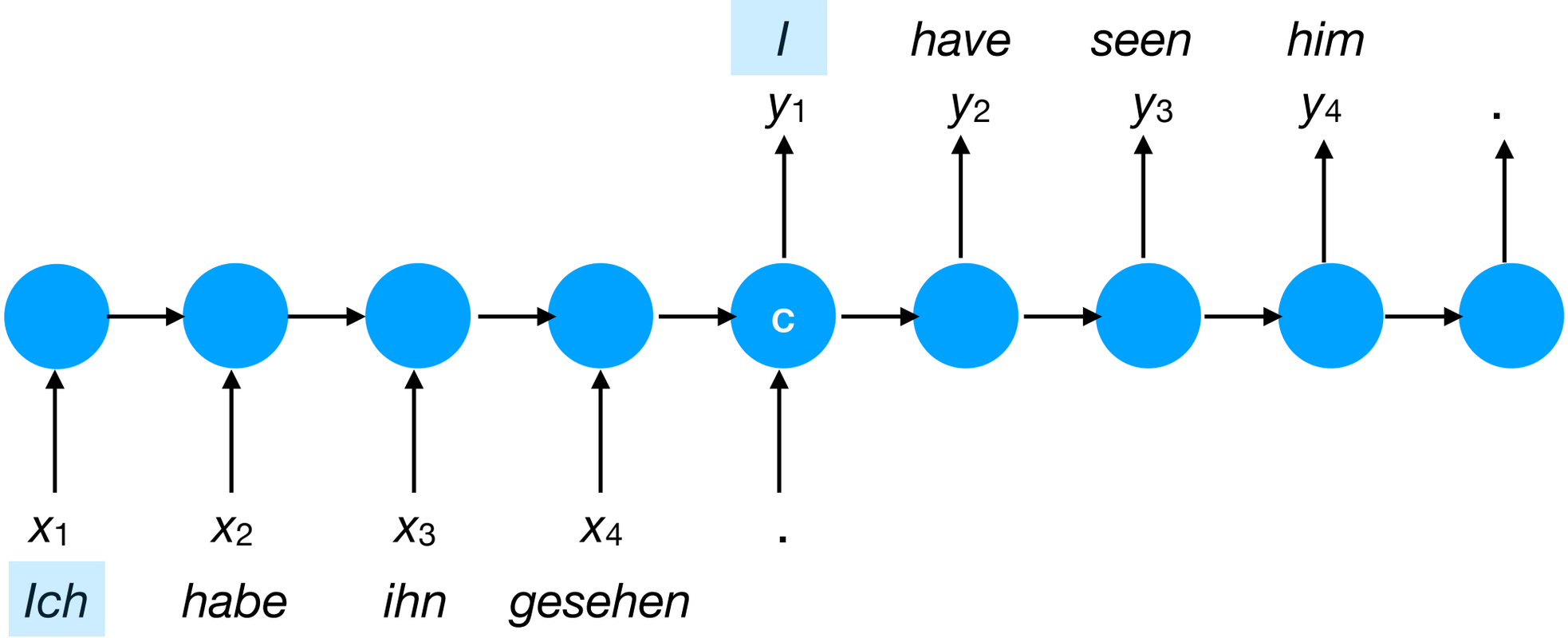
Attention models

- Intuitively, some subsequences of the input may be more relevant than others when producing a particular output symbol y_t.



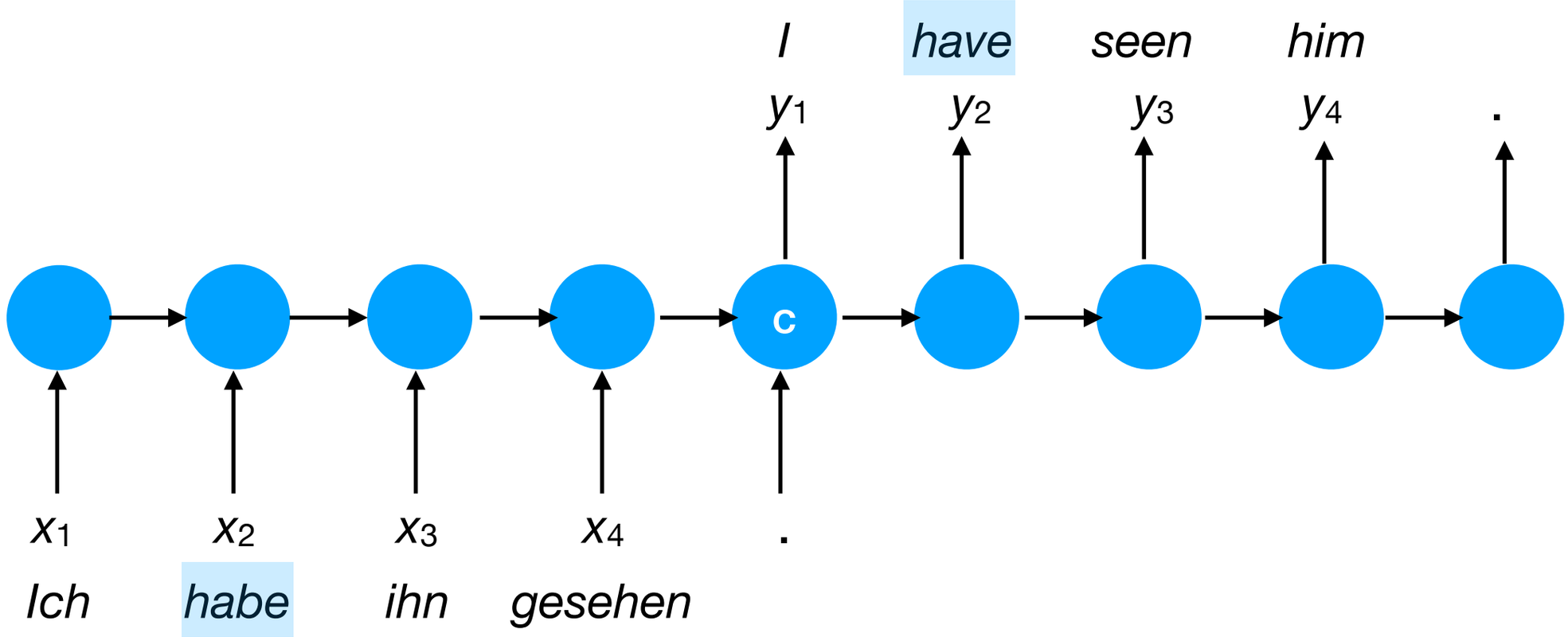
Attention models

- Intuitively, some subsequences of the input may be more relevant than others when producing a particular output symbol y_t.



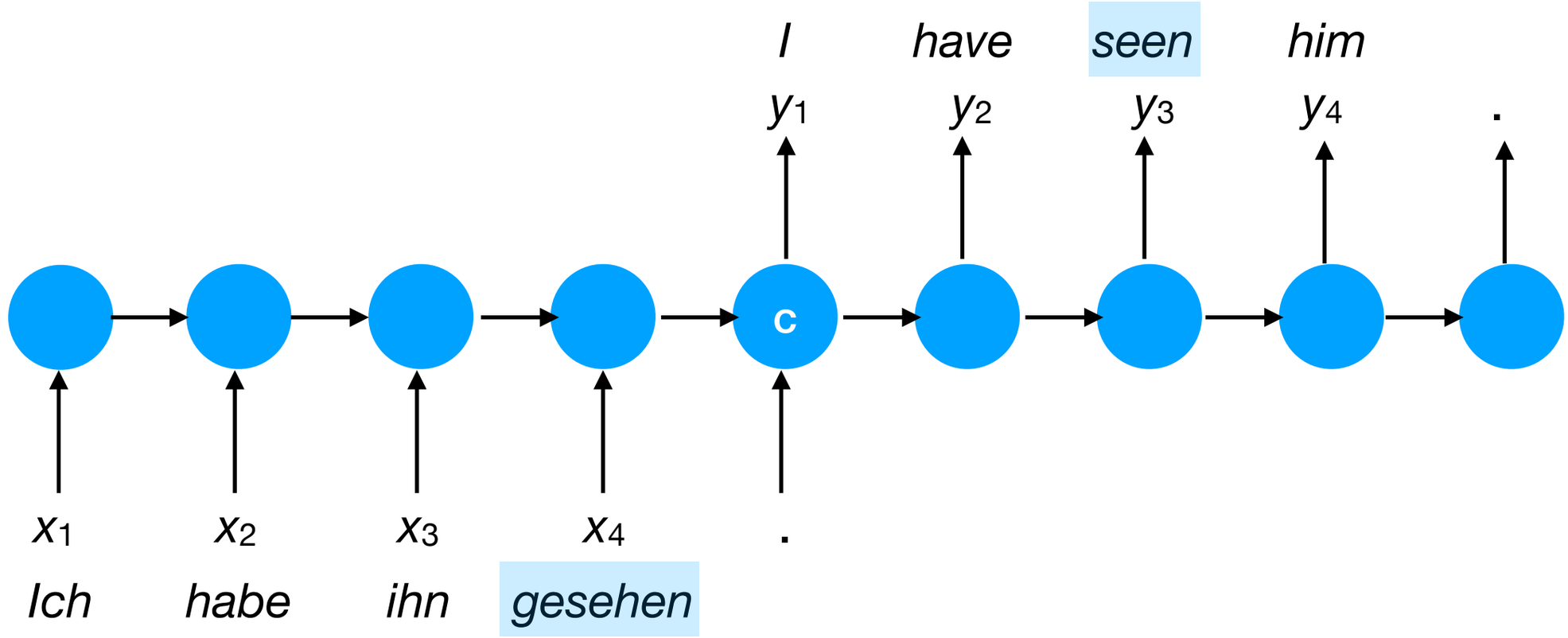
Attention models

- Intuitively, some subsequences of the input may be more relevant than others when producing a particular output symbol y_t.



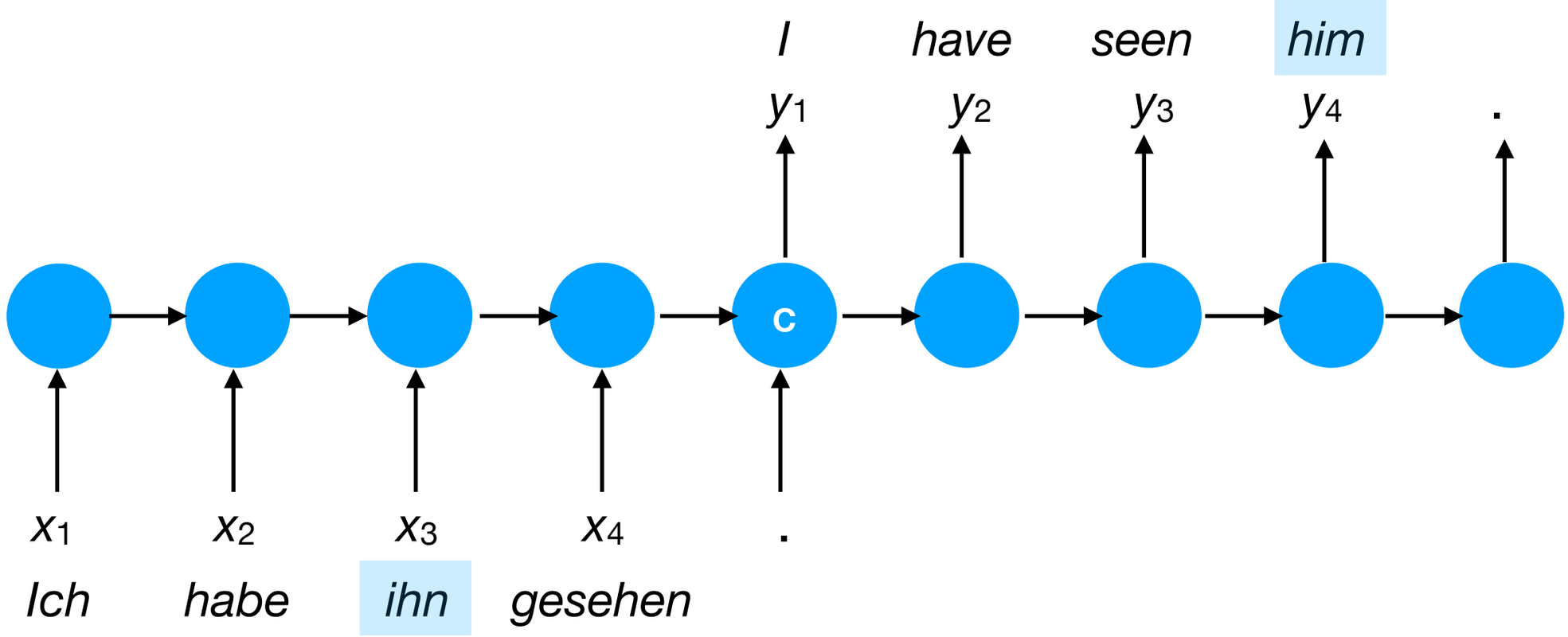
Attention models

- Intuitively, some subsequences of the input may be more relevant than others when producing a particular output symbol y_t.



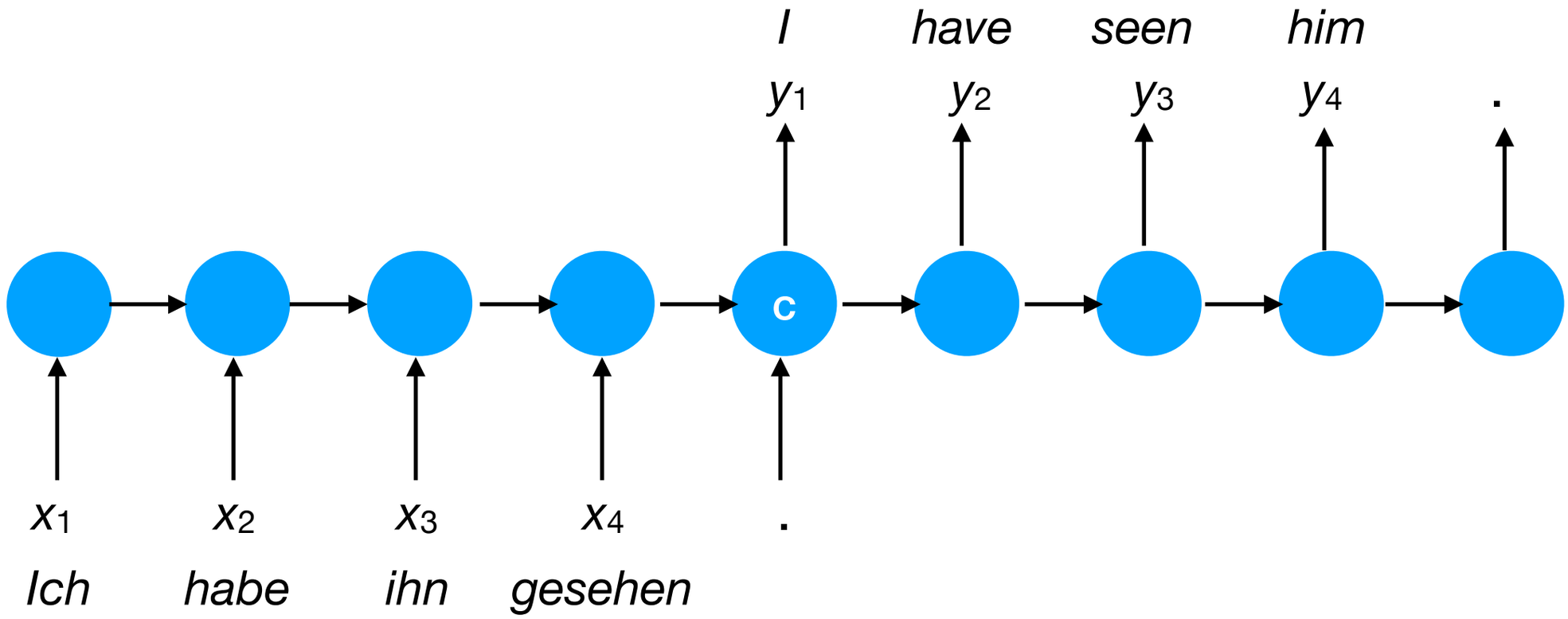
Attention models

- Intuitively, some subsequences of the input may be more relevant than others when producing a particular output symbol y_t.



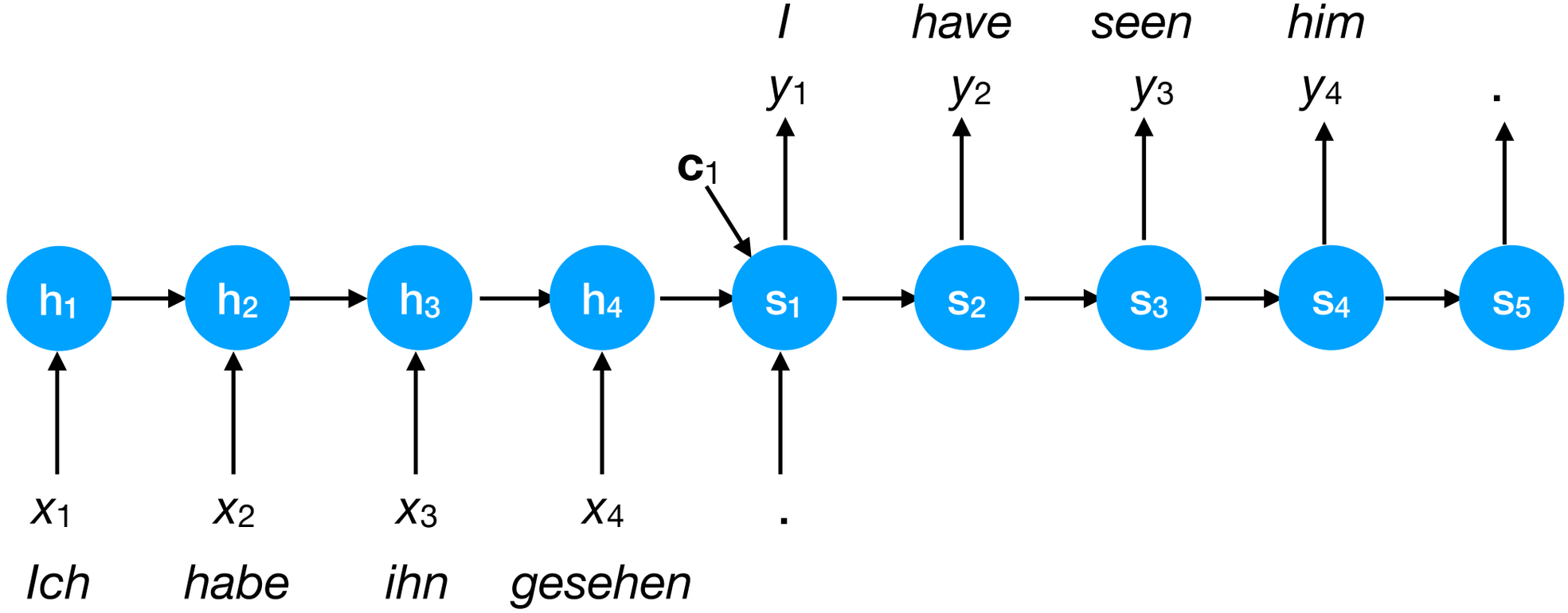
Attention models

- How can we focus the network's **attention** on the **most relevant inputs** when deciding each of the outputs?



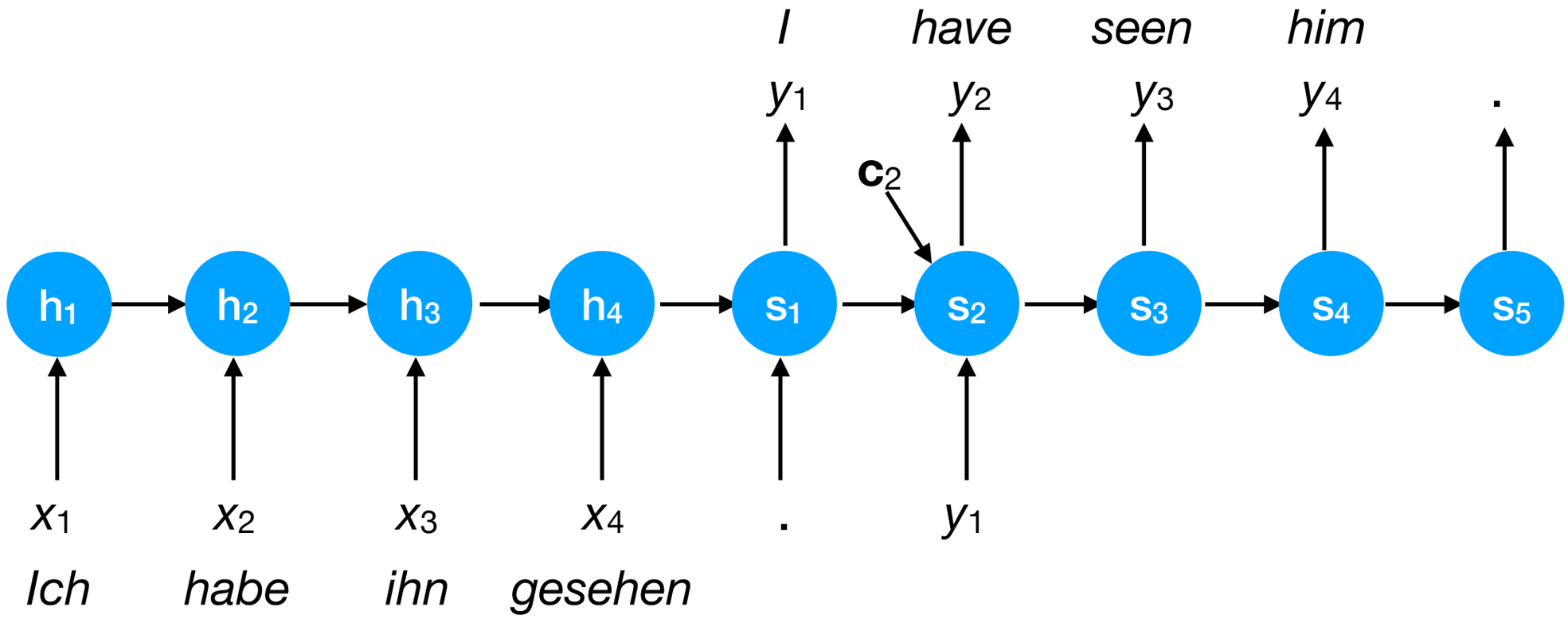
Varying context

- Bahdanau et al. (2015) proposed making the context depend on the particular output y_t: Instead of a fixed $\mathbf{c}$, we compute a different $\mathbf{c}_t$ for each output timestep t.



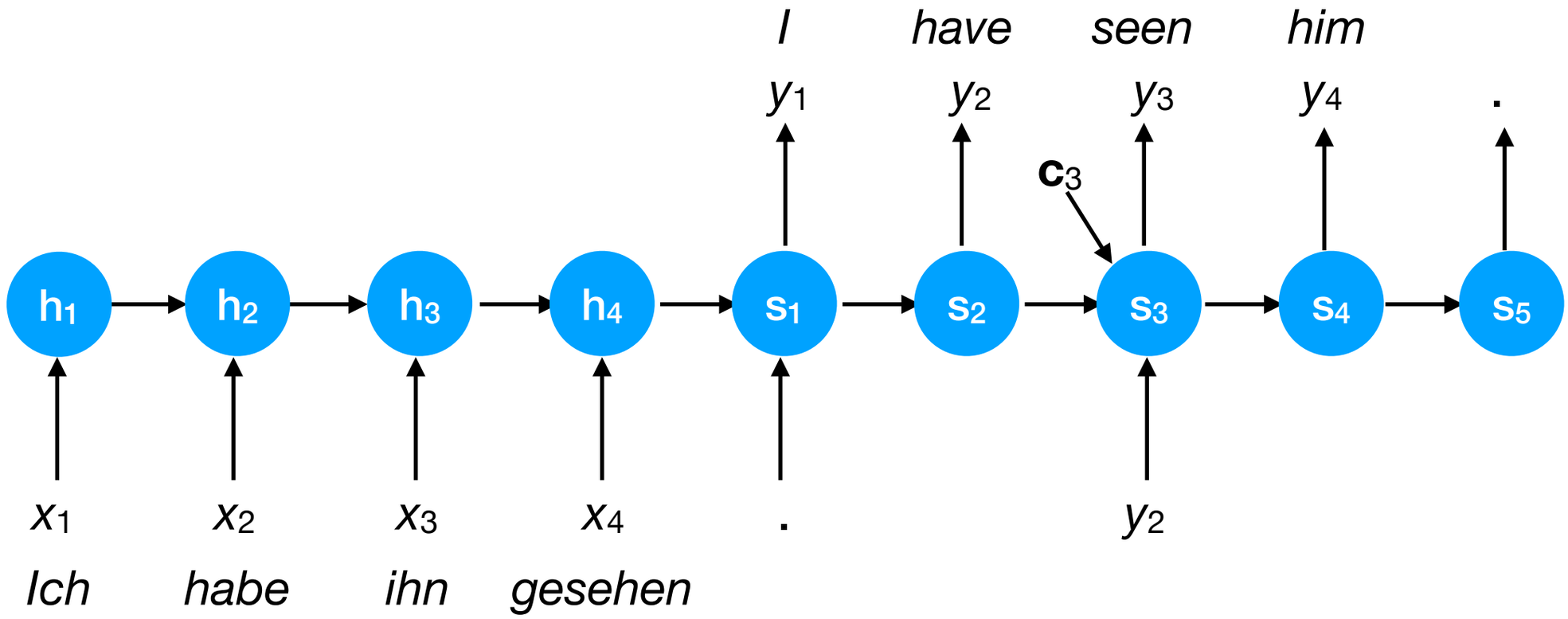
Varying context

- Bahdanau et al. (2015) proposed making the context depend on the particular output y_t: Instead of a fixed $\mathbf{c}$, we compute a different $\mathbf{c}_t$ for each output timestep t.



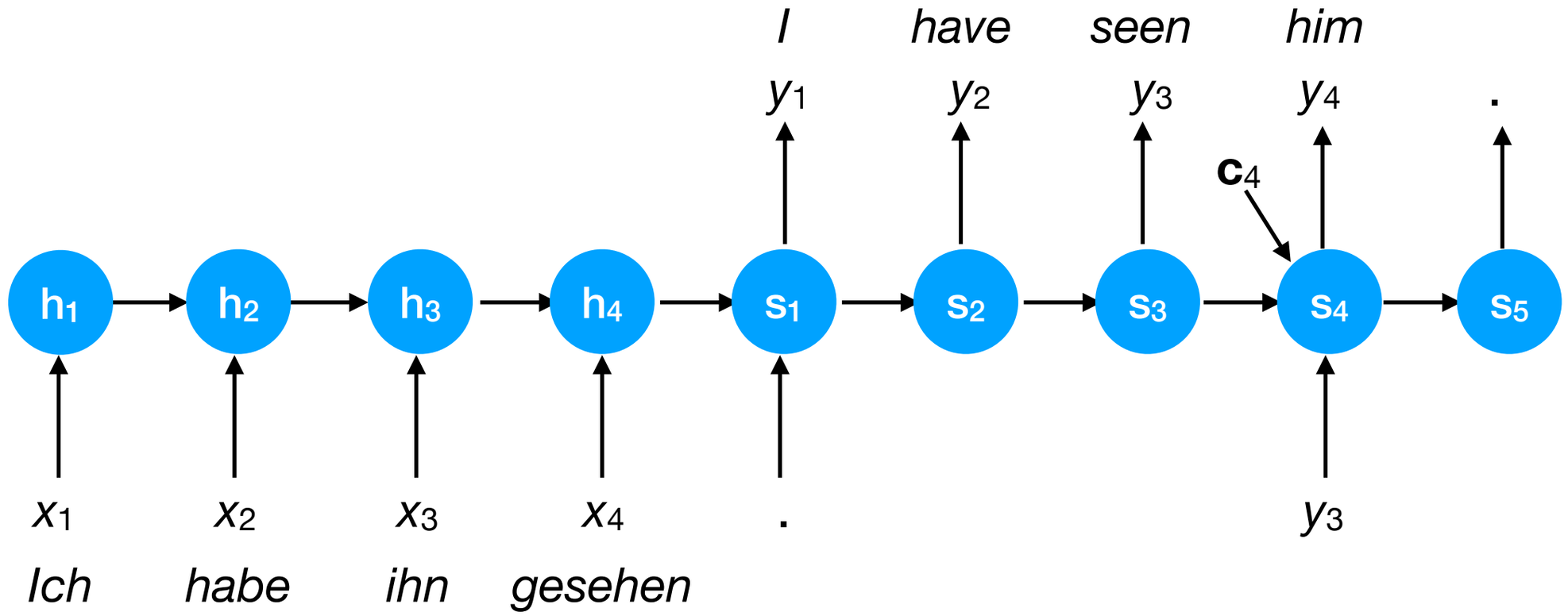
Varying context

- Bahdanau et al. (2015) proposed making the context depend on the particular output y_t: Instead of a fixed $\mathbf{c}$, we compute a different $\mathbf{c}_t$ for each output timestep t.



Varying context

- Bahdanau et al. (2015) proposed making the context depend on the particular output y_t: Instead of a fixed **c**, we compute a different $\mathbf{c}_t$ for each output timestep t.

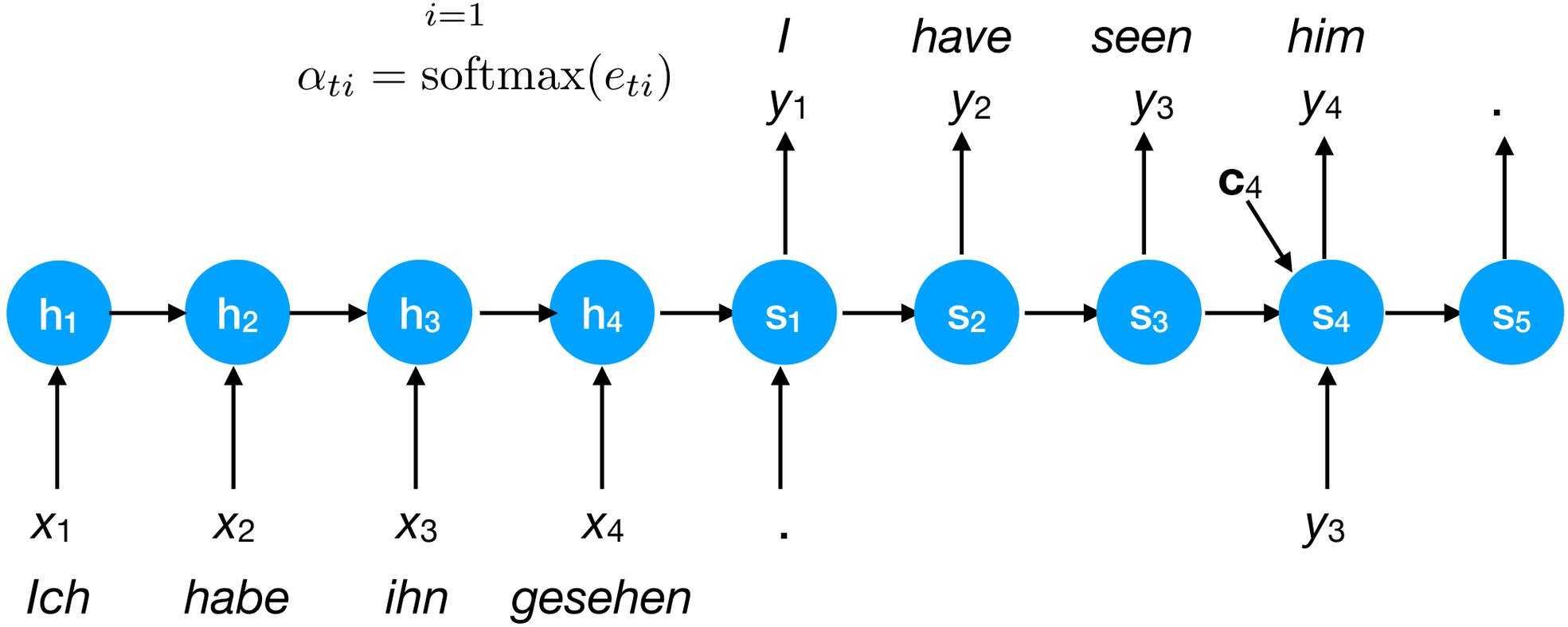


Attention weights

- Each $\mathbf{c}_t$ is a weighted sum of the hidden state sequence $\mathbf{h}_1, \ldots, \mathbf{h}_T$:

$$\mathbf{c}_t = \sum_{i=1}^{T} \alpha_{ti} \mathbf{h}_i$$

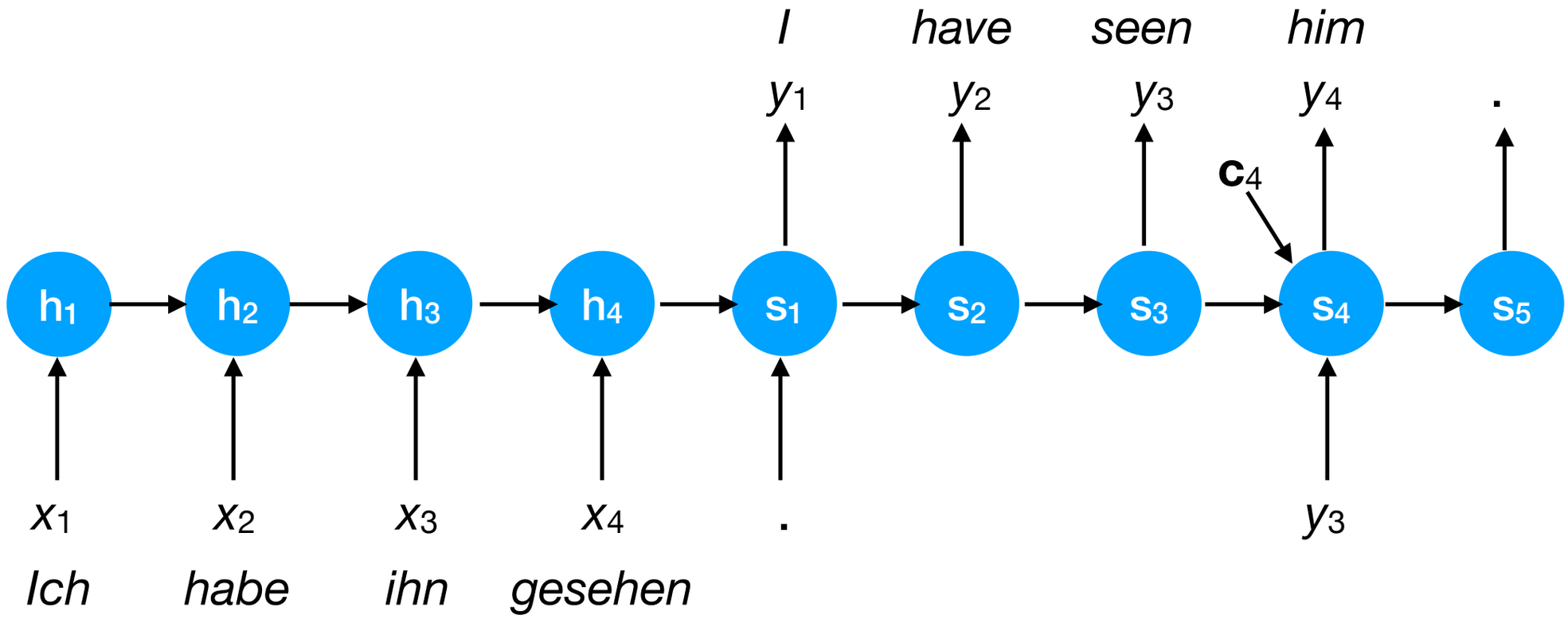
$$\alpha_{ti} = \mathrm{softmax}(e_{ti})$$



Attention weights

- Each e_{ti} is determined by the **alignment** between $\mathbf{s}_{t-1}$ and the sequence $\mathbf{h}_1, \ldots, \mathbf{h}_T$:

$$e_{ti} = a(\mathbf{s}_{t-1}, \mathbf{h}_i)$$

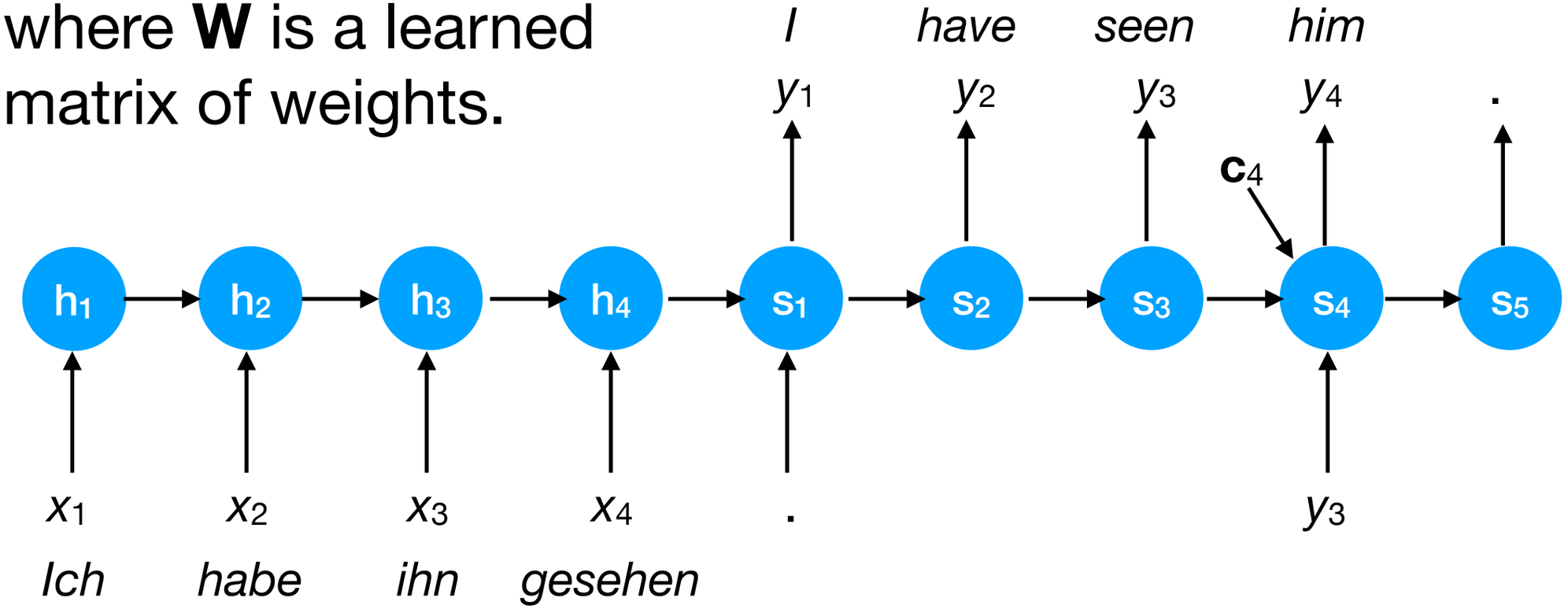


Alignment functions

- Various choices for *a* exist. One of the simplest is just a weighted dot product:

$$a(\mathbf{u}, \mathbf{v}) = \mathbf{u}^\top \mathbf{W} \mathbf{v}$$

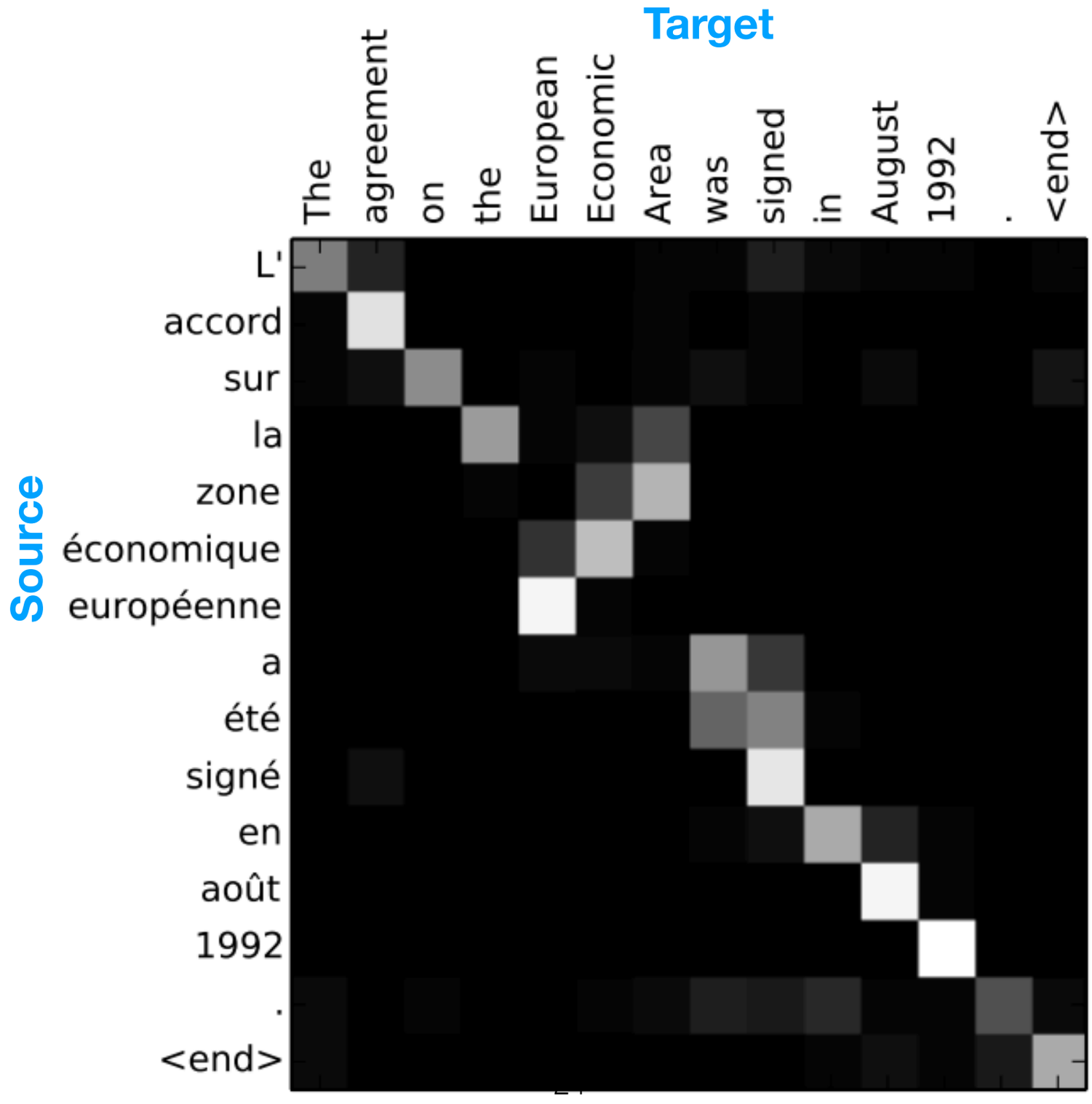
where **W** is a learned matrix of weights.



Results

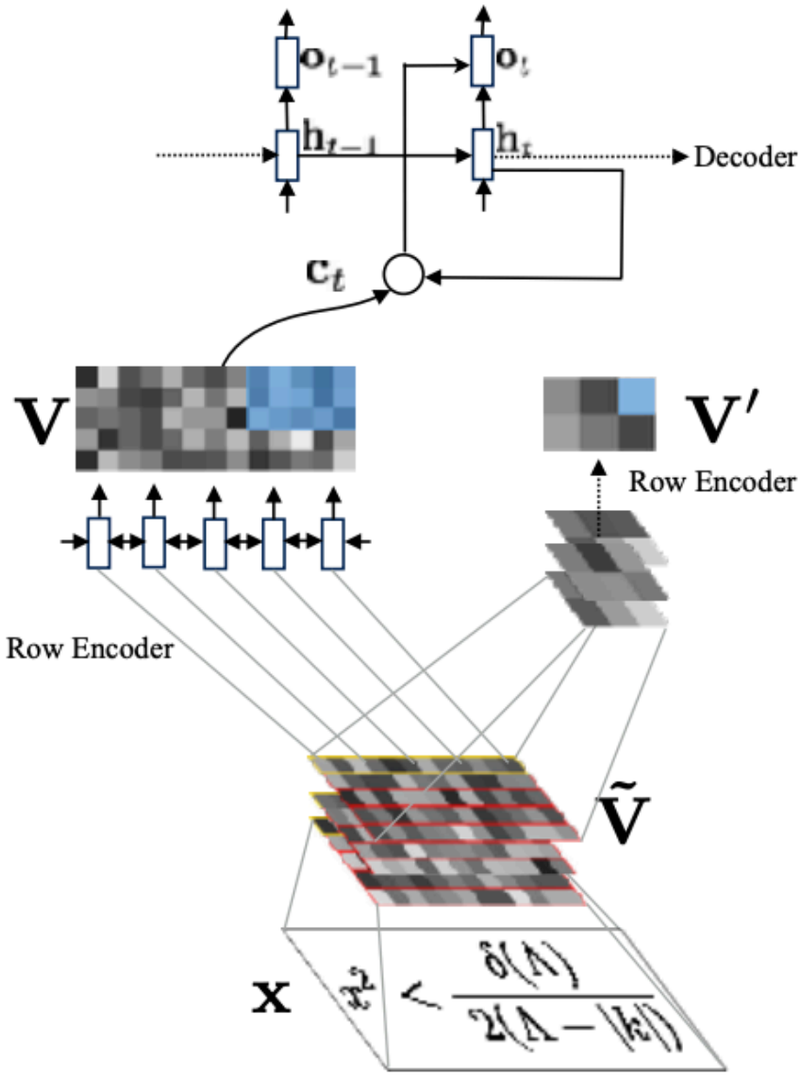
- By adding this **neural attention model** to the standard seq2seq model, the authors were able to achieve higher accuracy on standard machine translation benchmarks.

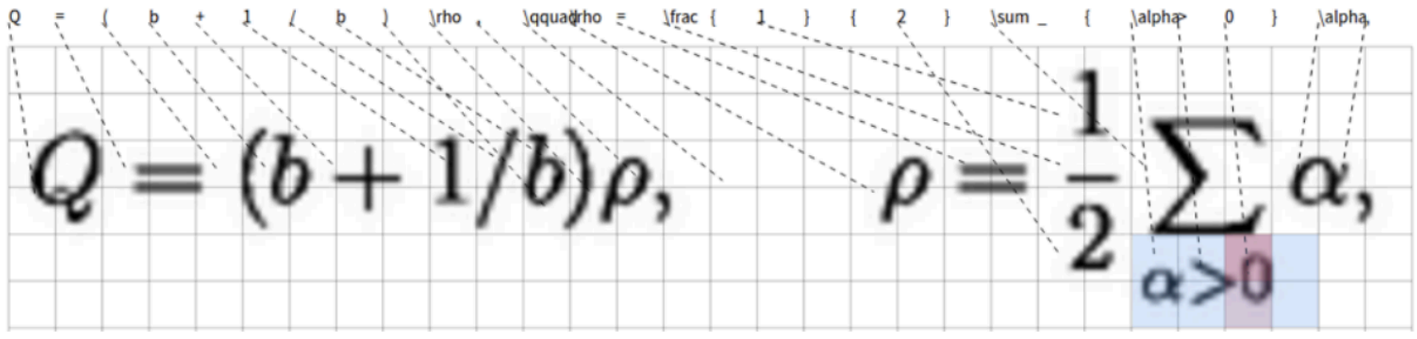
Example alignment



Another application: Image -> LaTeX

- Deng et al. 2017





Recurrent models of visual attention

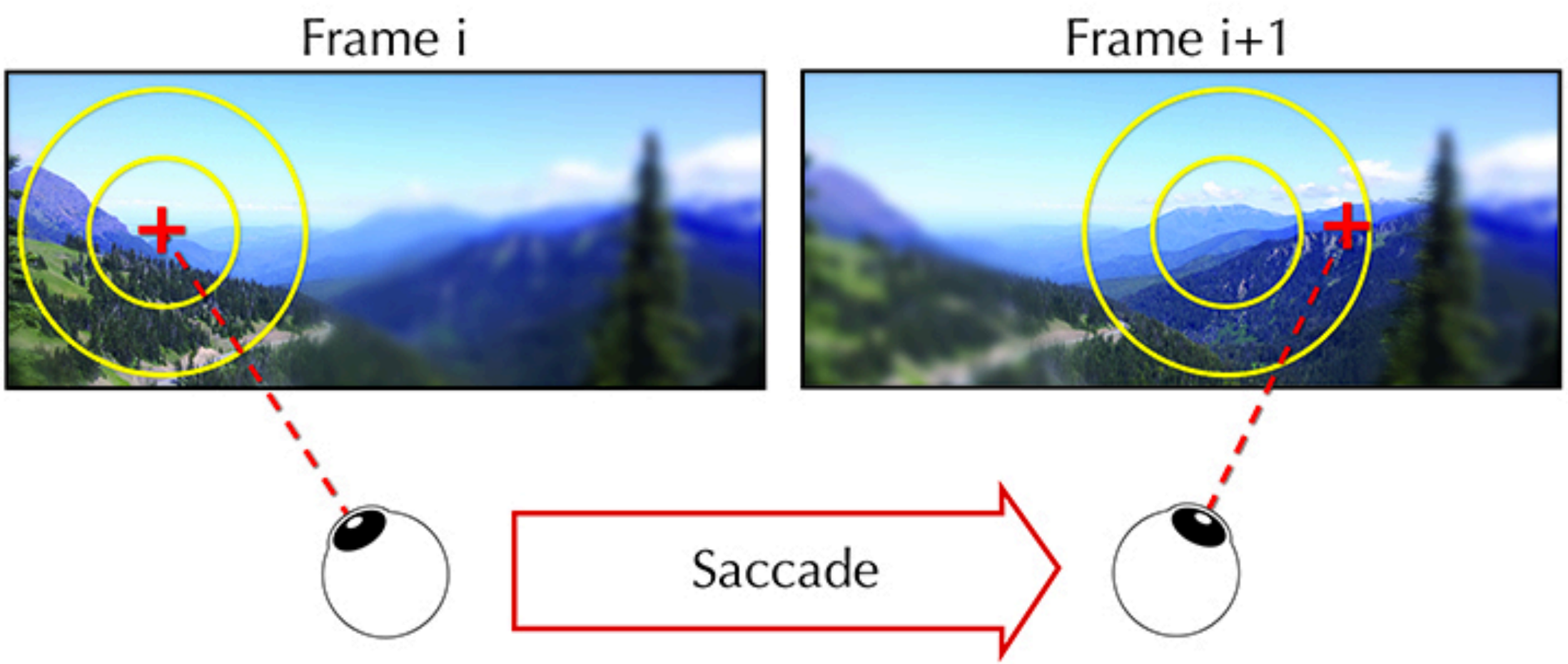
Mnih et al. 2014

Saccades and foveate human vision

- Humans transition the focus of their eye gaze through a sequence of **saccades** that help them concentrate on the most salient information in a visual scene.

Saccades and foveate human vision

- For any focal point on an image, humans' perception is **foveated**, i.e., the spatial resolution decreases as the distance to the focus increases.



https://research.nvidia.com/publication/2017-09_Latency-Requirements-for

Saccades and foveate human vision

- Saccades are a task-dependent method of human attention to visual scenes that varies over time.
- Mnih et al. 2014 explored a DL-based model of recurrent visual attention that implements a similar method.

Recurrent models of visual attention

- Consider a scenario in which a robot is interacting with its environment in order to maximize some long-term reward:

- At each time t:
  - The robot can **look** at a location l_t in the visual scene.
  - Based on the information it learns by looking, the robot **updates** its state representation h_t of the environment.
  - The robot **acts** with action a_t on the environment.
- The robot's goal is to choose actions so as to maximize the **return** $R(\tau)$ where $\tau=(a_1, \ldots, a_T, h_1, \ldots, h_T)$.

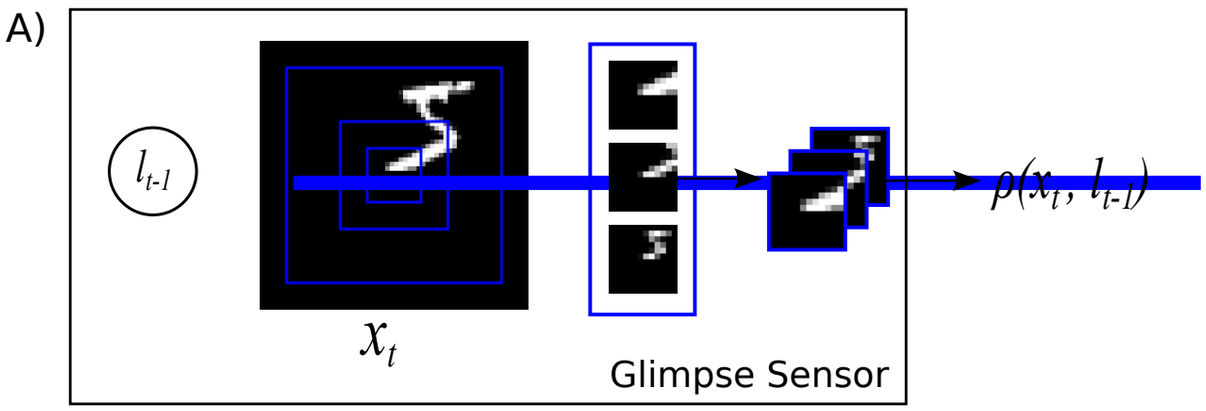
Example

- Consider a robot that picks items from a conveyor belt at a material recovery facility (MRF).

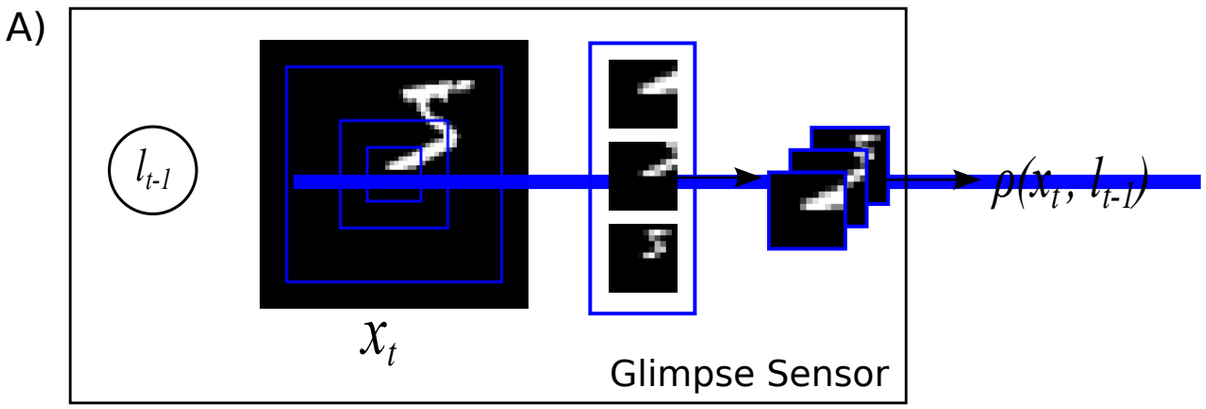
- The robot should look carefully at each item before deciding how to act (what to pick up, how to pick it up).

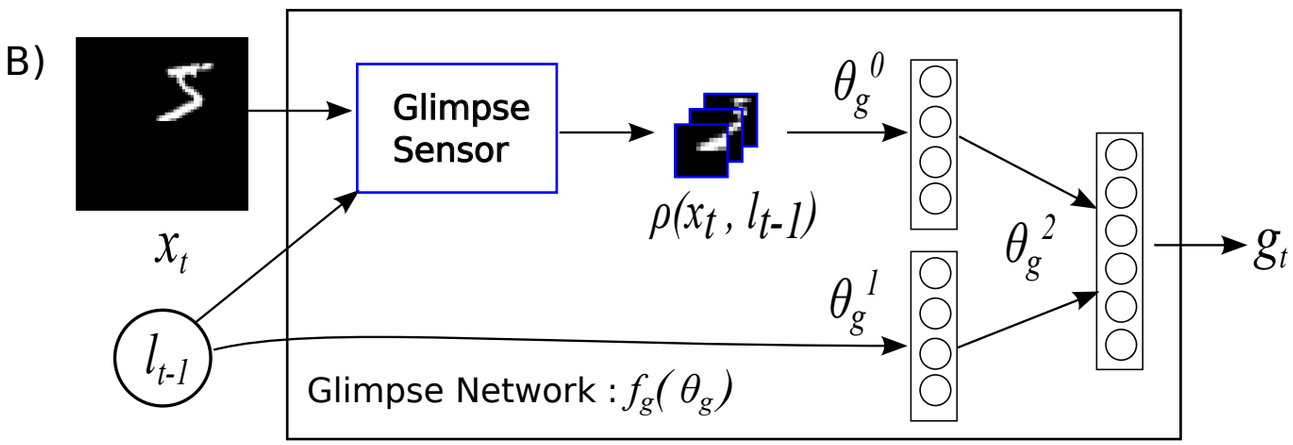
- The reward might be based on amount of sorted garbage, number of machine failures, etc.

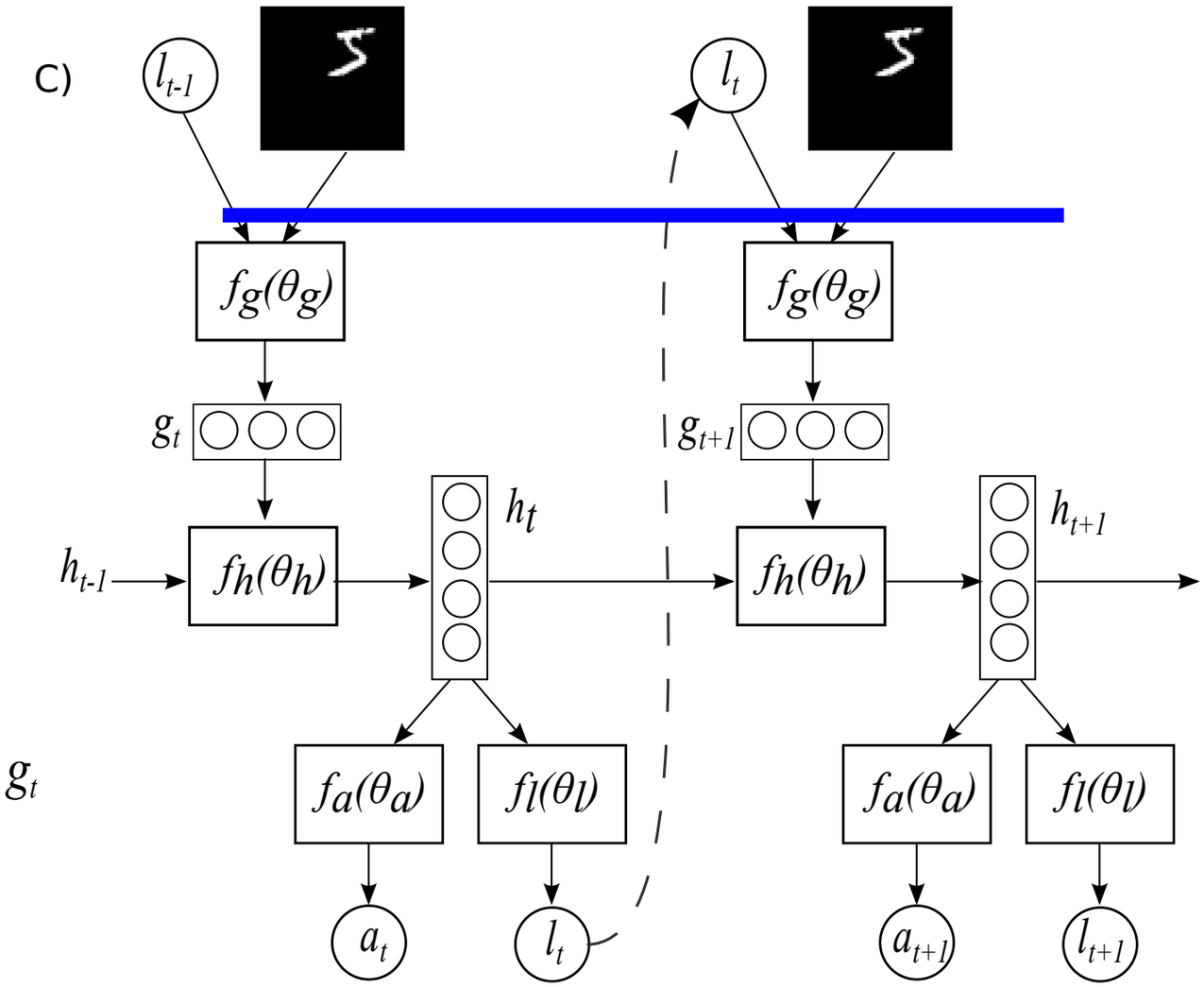
Recurrent models of visual attention



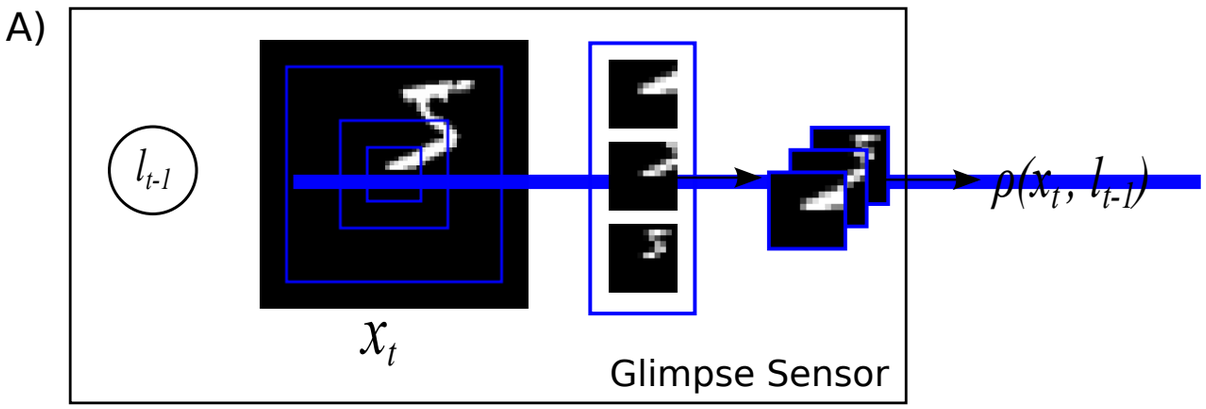
Recurrent models of visual attention

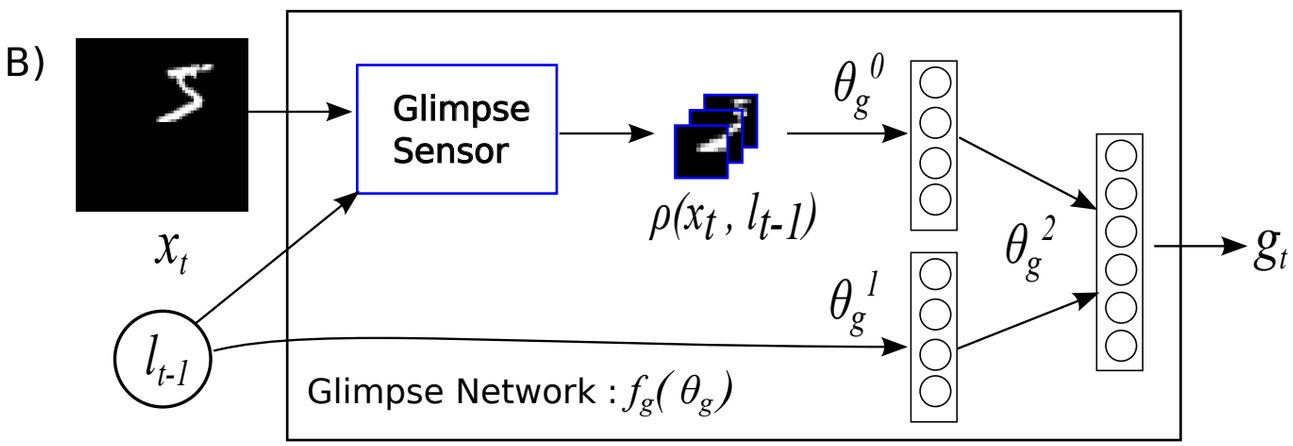


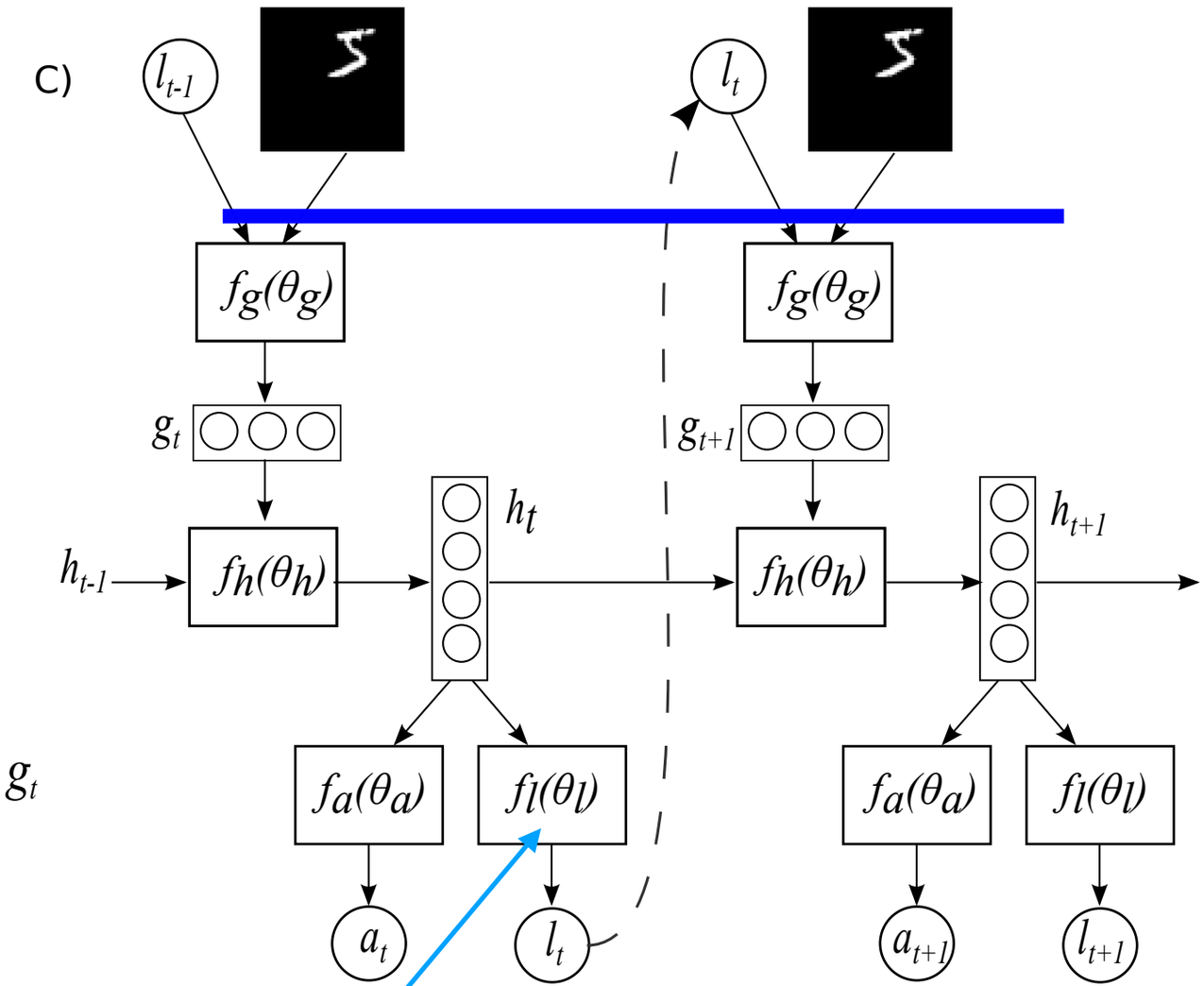




Recurrent models of visual attention

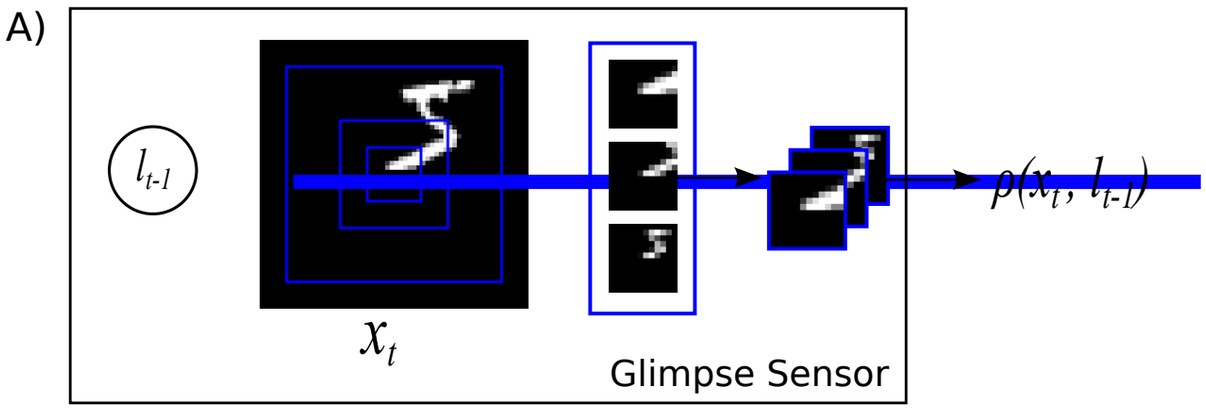


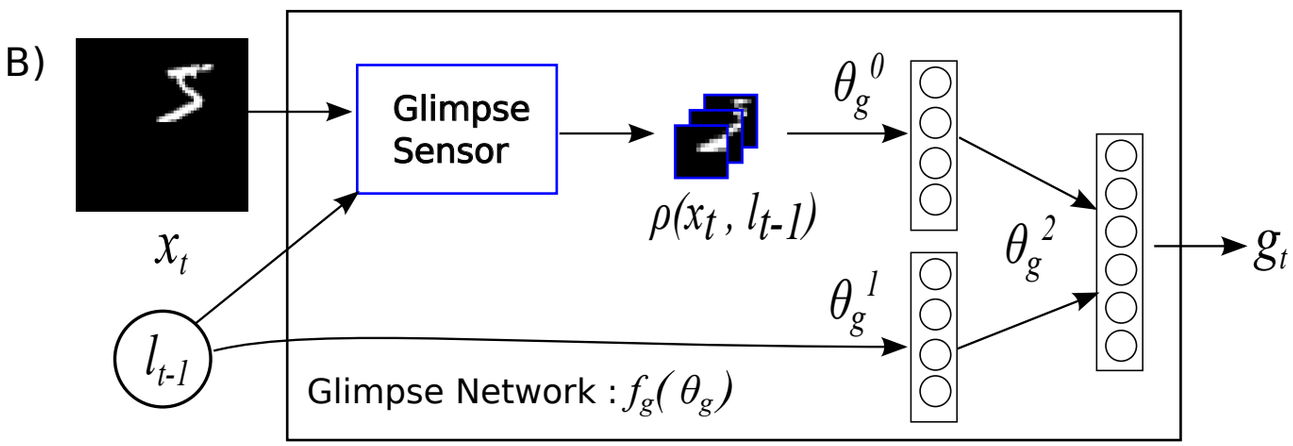


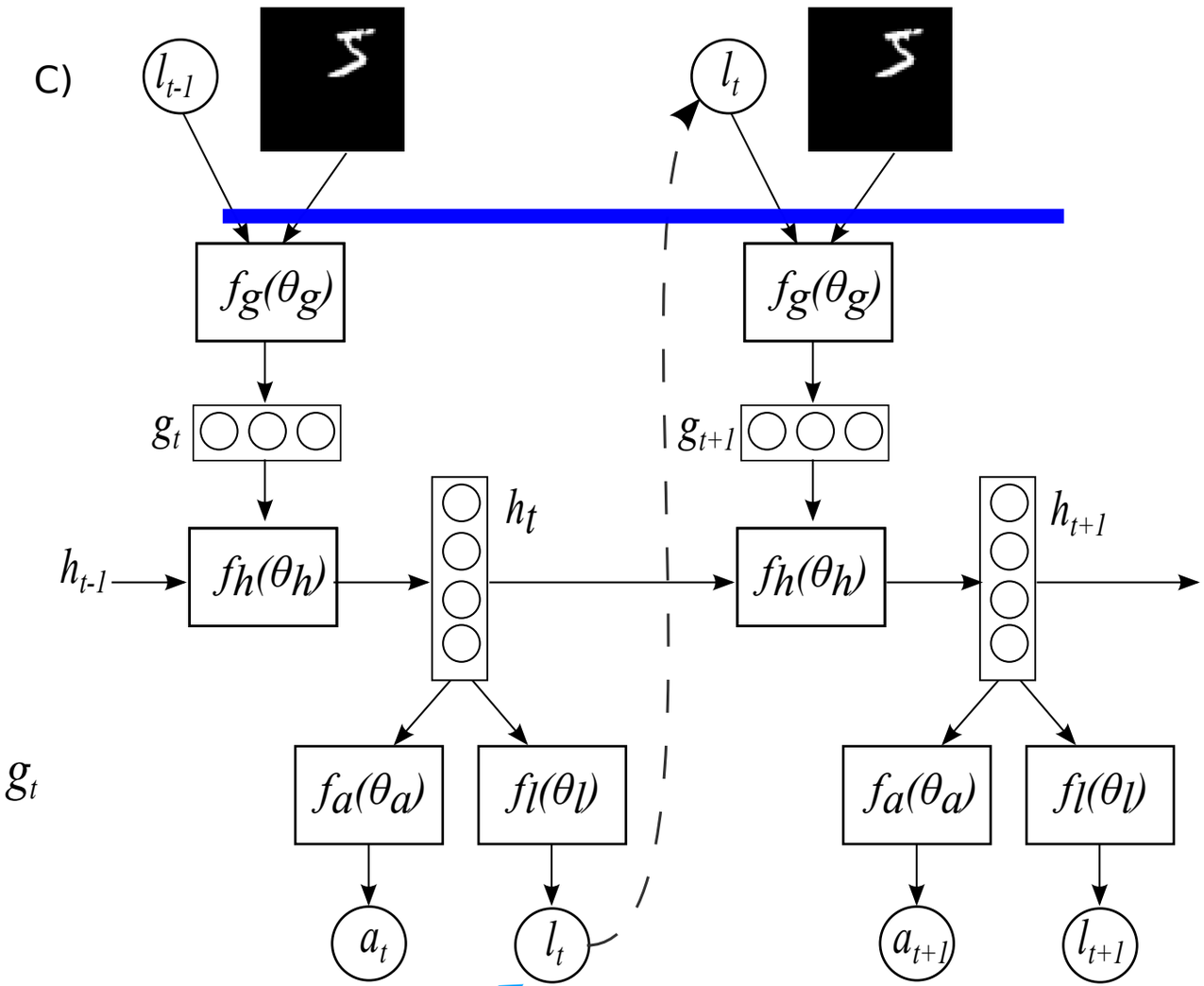


This network produces a probability distribution over all gaze locations l_t.

Recurrent models of visual attention

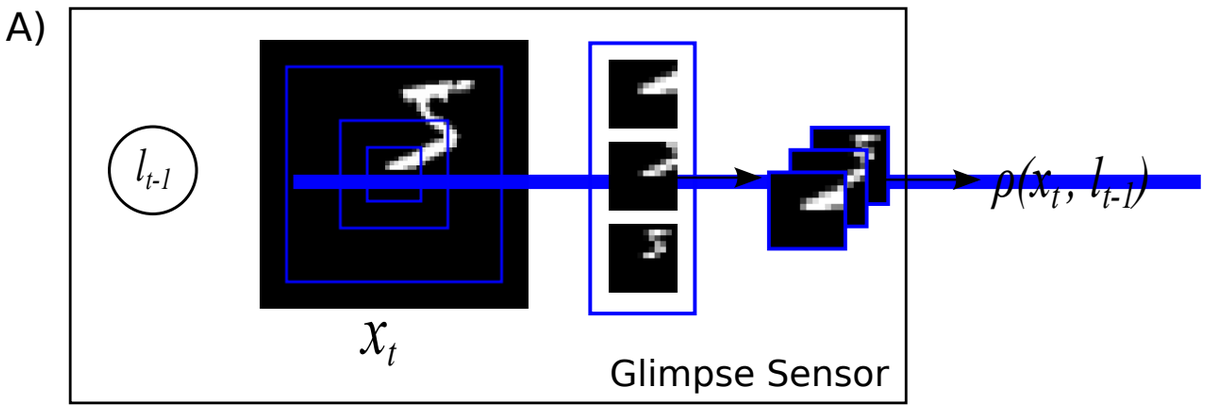


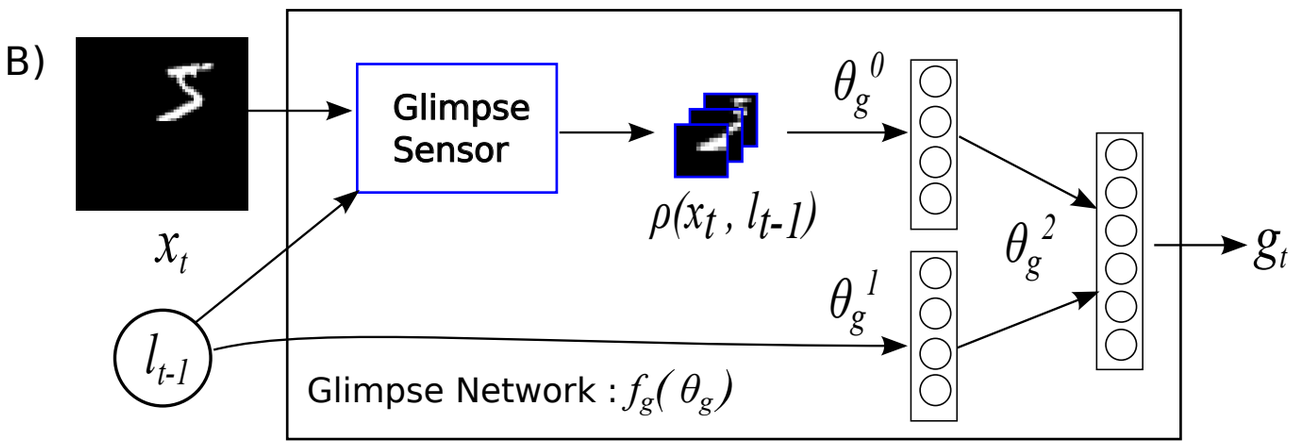


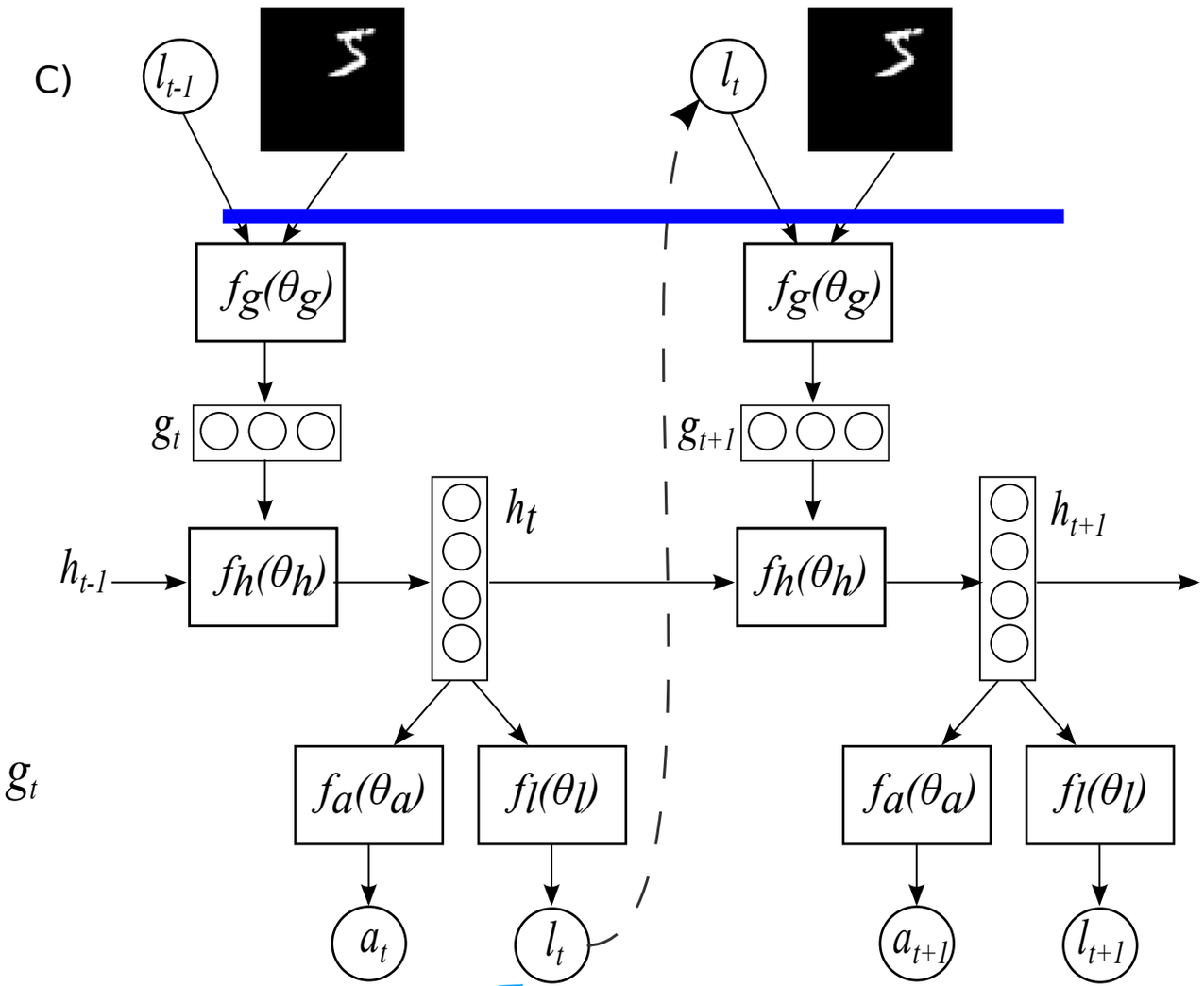


A sample from this probability distribution is then drawn to decide the actual (discrete) target l_t.

Recurrent models of visual attention







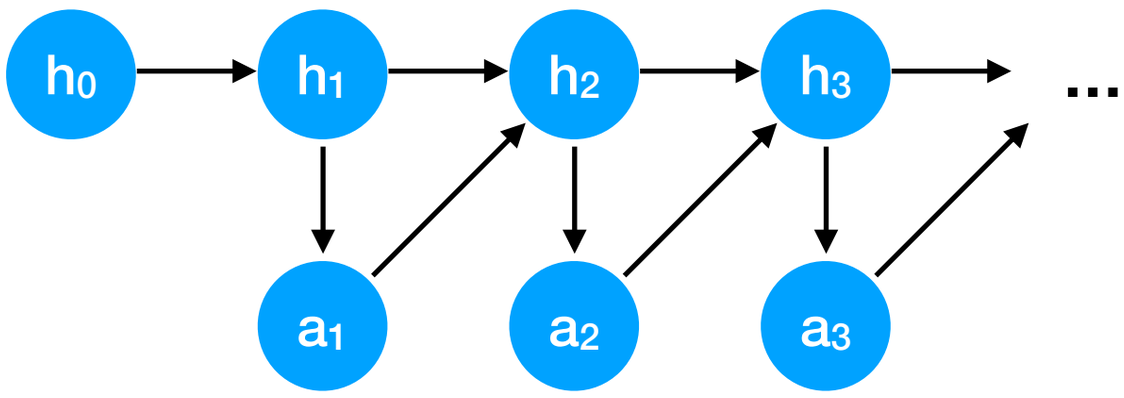
Sampling is non-differentiable, and thus we have broken back-propagation :-(.

REINFORCE

- Fortunately, Williams 1992 devised an algorithm called REINFORCE that offers a way to conduct gradient descent to optimize the RNN’s parameters despite the non-differentiability.

REINFORCE

- Premise:
  - We have a policy network $\boldsymbol{\pi}_\theta(h_t) = P(a_t \mid h_t, \theta)$ that computes the probability distribution over actions a_t, given h_t and θ, at each timestep t.
- We can represent the relationship between the actions and hidden states using a graphical model:



REINFORCE

- Premise:
  - We have a policy network $\boldsymbol{\pi}_\theta(h_t) = P(a_t \mid h_t, \theta)$ that computes the probability distribution over actions a_t, given h_t and θ, at each timestep t.
  - $P(\tau \mid \theta)$ is the probability distribution over trajectories, given the policy network's parameters θ. It factorizes:

$$P(\tau \mid \theta) = P(a_1, \ldots, a_T, h_0, h_1, \ldots, h_T \mid \theta) = P(h_0) \prod_{t=1}^{T} P(a_t \mid h_t, \theta) P(h_t \mid h_{t-1}, a_{t-1})$$

REINFORCE

- Premise:
  - We have a policy network $\boldsymbol{\pi}_\theta(h_t) = P(a_t \mid h_t, \theta)$ that computes the probability distribution over actions a_t, given h_t and θ, at each timestep t.
  - $P(\tau \mid \theta)$ is the probability distribution over trajectories, given the policy network's parameters θ. It factorizes:

$$P(\tau \mid \theta) = P(a_1, \ldots, a_T, h_0, h_1, \ldots, h_T \mid \theta) = P(h_0) \prod_{t=1}^{T} P(a_t \mid h_t, \theta) P(h_t \mid h_{t-1}, a_{t-1})$$

Action probability: we usually know this.

REINFORCE

- Premise:
  - We have a policy network $\boldsymbol{\pi}_\theta(h_t) = P(a_t \mid h_t, \theta)$ that computes the probability distribution over actions a_t, given h_t and θ, at each timestep t.
  - $P(\tau \mid \theta)$ is the probability distribution over trajectories, given the policy network's parameters θ. It factorizes:

$$P(\tau \mid \theta) = P(a_1, \ldots, a_T, h_0, h_1, \ldots, h_T \mid \theta) = P(h_0) \prod_{t=1}^{T} P(a_t \mid h_t, \theta) P(h_t \mid h_{t-1}, a_{t-1})$$

System dynamics: we often do not know this.

REINFORCE

- Premise:
  - The prob. dist. of the initial state is called $P(h_0)$.
  - We have a return function R over trajectories τ.
  - We want to maximize the **expected return** over all possible trajectories τ:

$$J(\theta) = \mathbb{E}_{\tau \sim P(\tau \mid \theta)}[R(\tau)] = \int_{\tau} P(\tau \mid \theta) R(\tau) d\tau$$

Maximizing expected return

- We might want to conduct SGD on J w.r.t. θ.

- Unfortunately, the integral is generally intractable.

$$J(\theta) = \mathbb{E}_{\tau \sim P(\tau \mid \theta)}[R(\tau)] = \int_{\tau} P(\tau \mid \theta) R(\tau) d\tau$$

Maximizing expected return

- We might want to conduct SGD on J w.r.t. θ.

- Unfortunately, the integral is generally intractable.

$$J(\theta) = \mathbb{E}_{\tau \sim P(\tau \mid \theta)}[R(\tau)] = \int_{\tau} P(\tau \mid \theta) R(\tau) d\tau$$

- Instead, we use the **log-likelihood ratio trick**…

$$\nabla_\theta P(\tau \mid \theta) = P(\tau \mid \theta) \nabla_\theta \log P(\tau \mid \theta)$$

Maximizing expected return

$$\nabla_\theta J(\theta) = \int \nabla_\theta P(\tau \mid \theta) R(\tau) d\tau$$

Maximizing expected return

$$\nabla_\theta J(\theta) = \int \nabla_\theta P(\tau \mid \theta) R(\tau) d\tau$$

$$= \int P(\tau \mid \theta) \nabla_\theta \log P(\tau \mid \theta) R(\tau) d\tau$$

Maximizing expected return

$$\nabla_\theta J(\theta) = \int \nabla_\theta P(\tau \mid \theta) R(\tau) d\tau$$

$$= \int P(\tau \mid \theta) \nabla_\theta \log P(\tau \mid \theta) R(\tau) d\tau$$

$$= \mathbb{E}_{\tau \sim P(\tau \mid \theta)} \left[\nabla_\theta \log P(\tau \mid \theta) R(\tau) \right]$$

Maximizing expected return

$$\nabla_\theta J(\theta) = \int \nabla_\theta P(\tau \mid \theta) R(\tau) d\tau$$

$$= \int P(\tau \mid \theta) \nabla_\theta \log P(\tau \mid \theta) R(\tau) d\tau$$

$$= \mathbb{E}_{\tau \sim P(\tau \mid \theta)} \left[\nabla_\theta \log P(\tau \mid \theta) R(\tau) \right]$$

$$\approx \frac{1}{n} \sum_{i=1}^{n} \nabla_\theta \log P(\tau \mid \theta) R(\tau)$$

Maximizing expected return

$$\begin{aligned}
\nabla_\theta J(\theta) &= \int \nabla_\theta P(\tau \mid \theta) R(\tau) d\tau \\
&= \int P(\tau \mid \theta) \nabla_\theta \log P(\tau \mid \theta) R(\tau) d\tau \\
&= \mathbb{E}_{\tau \sim P(\tau \mid \theta)} \left[\nabla_\theta \log P(\tau \mid \theta) R(\tau) \right] \\
&\approx \frac{1}{n} \sum_{i=1}^{n} \nabla_\theta \log P(\tau \mid \theta) R(\tau)
\end{aligned}$$

- In other words, we can approximate the gradient of *J* by sampling many trajectories and averaging.

Maximizing expected return

$$P(\tau \mid \theta) = P(h_0) \prod_{t=1}^{T} P(h_t \mid h_{t-1}, a_{t-1}) P(a_t \mid h_t, \theta)$$

Maximizing expected return

$$P(\tau \mid \theta) = P(h_0) \prod_{t=1}^{T} P(h_t \mid h_{t-1}, a_{t-1}) P(a_t \mid h_t, \theta)$$

$$\Longrightarrow$$

$$\log P(\tau \mid \theta) = \log P(h_0) + \sum_{t=1}^{T} \log P(h_t \mid h_{t-1}, a_{t-1}) + \sum_{t=1}^{T} \log P(a_t \mid h_t, \theta)$$

Maximizing expected return

$$P(\tau \mid \theta) = P(h_0) \prod_{t=1}^{T} P(h_t \mid h_{t-1}, a_{t-1}) P(a_t \mid h_t, \theta)$$

$$\Longrightarrow$$

$$\log P(\tau \mid \theta) = \log P(h_0) + \sum_{t=1}^{T} \log P(h_t \mid h_{t-1}, a_{t-1}) + \sum_{t=1}^{T} \log P(a_t \mid h_t, \theta)$$

$$\Longrightarrow$$ **The first two terms do not depend on θ.**

$$\nabla_\theta \log P(\tau \mid \theta) = \sum_{t=1}^{T} \nabla_\theta \log P(a_t \mid h_t, \theta)$$

Maximizing expected return

$$P(\tau \mid \theta) = P(h_0) \prod_{t=1}^{T} P(h_t \mid h_{t-1}, a_{t-1}) P(a_t \mid h_t, \theta)$$

$$\Longrightarrow$$

$$\log P(\tau \mid \theta) = \log P(h_0) + \sum_{t=1}^{T} \log P(h_t \mid h_{t-1}, a_{t-1}) + \sum_{t=1}^{T} \log P(a_t \mid h_t, \theta)$$

$$\Longrightarrow$$

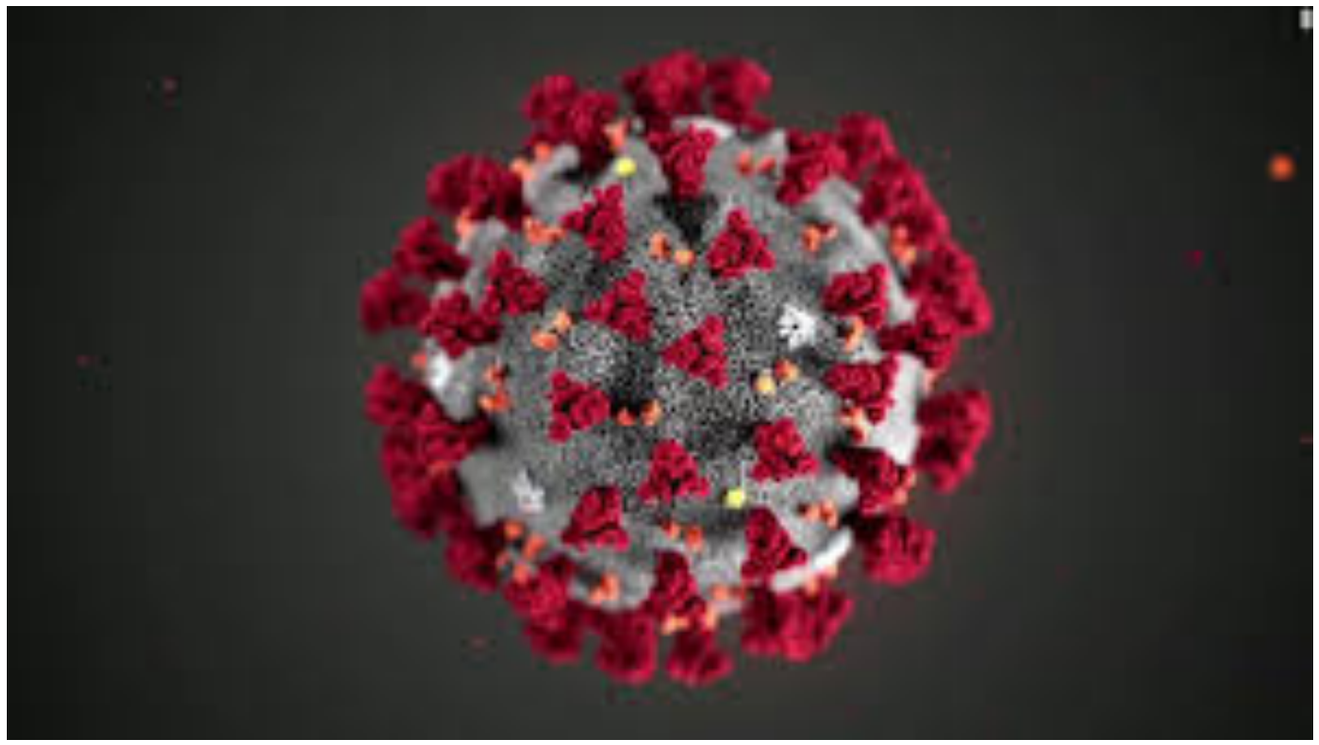
$$\nabla_\theta \log P(\tau \mid \theta) = \sum_{t=1}^{T} \nabla_\theta \log P(a_t \mid h_t, \theta)$$

$$= \sum_{t=1}^{T} \nabla_\theta \pi_\theta(h_t)$$

This is just the softmax output of our action prediction network.

REINFORCE: summary

- REINFORCE provides a tractable approximate method (based on sampling) to compute the gradient of a cost function that depends in a non-differentiable way on a variable that was sampled from a probability distribution produced by the neural network.



Accelerating drug discovery via computational chemistry & machine learning

https://cdn.cnn.com/cnnnext/dam/assets/200130165125-corona-virus-cdc-image-super-tease.jpg

Drug discovery

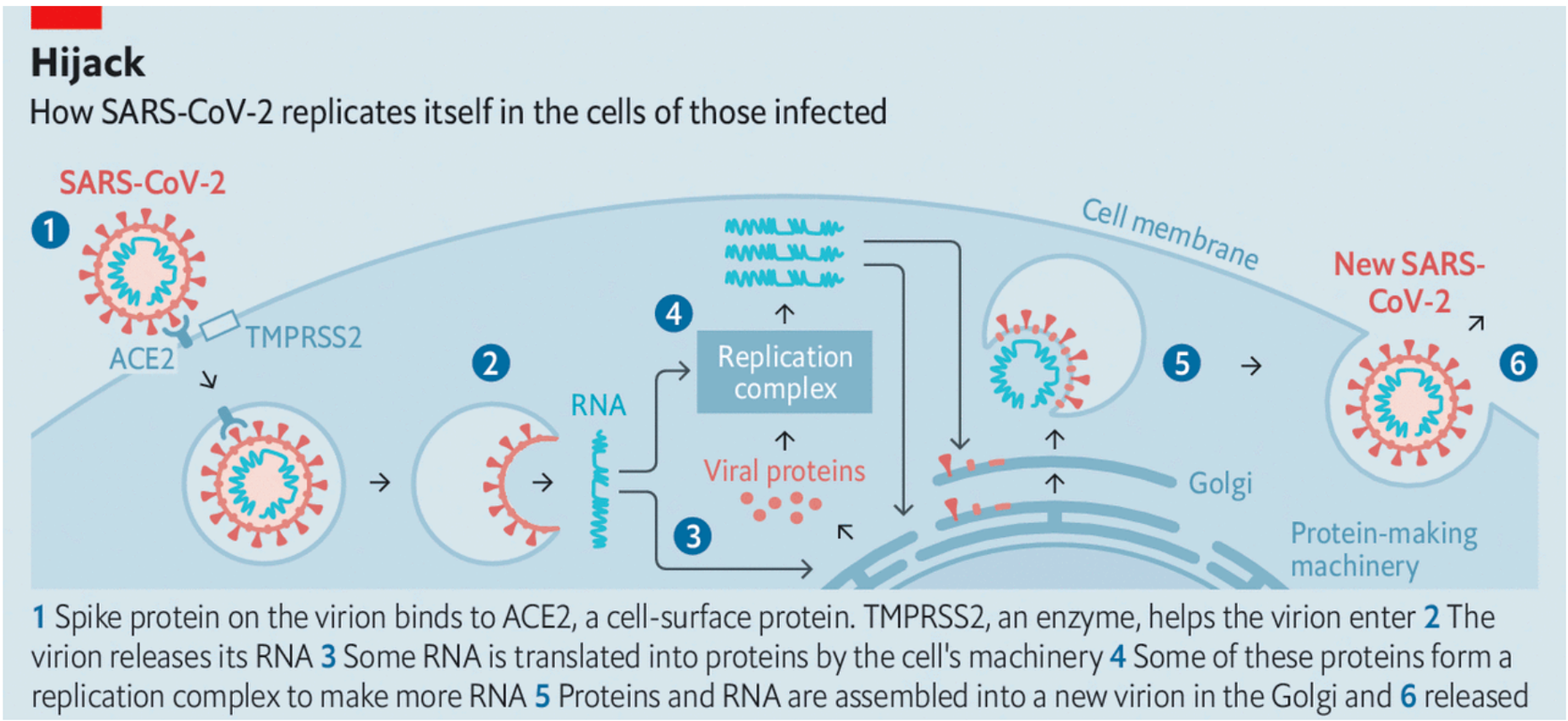
- For SARS-CoV-2 (coronavirus) and many other pathogens, scientists are searching intensely for:

- Vaccines to prevent infection.

- Therapeutic drugs to treat infection.

Drug discovery

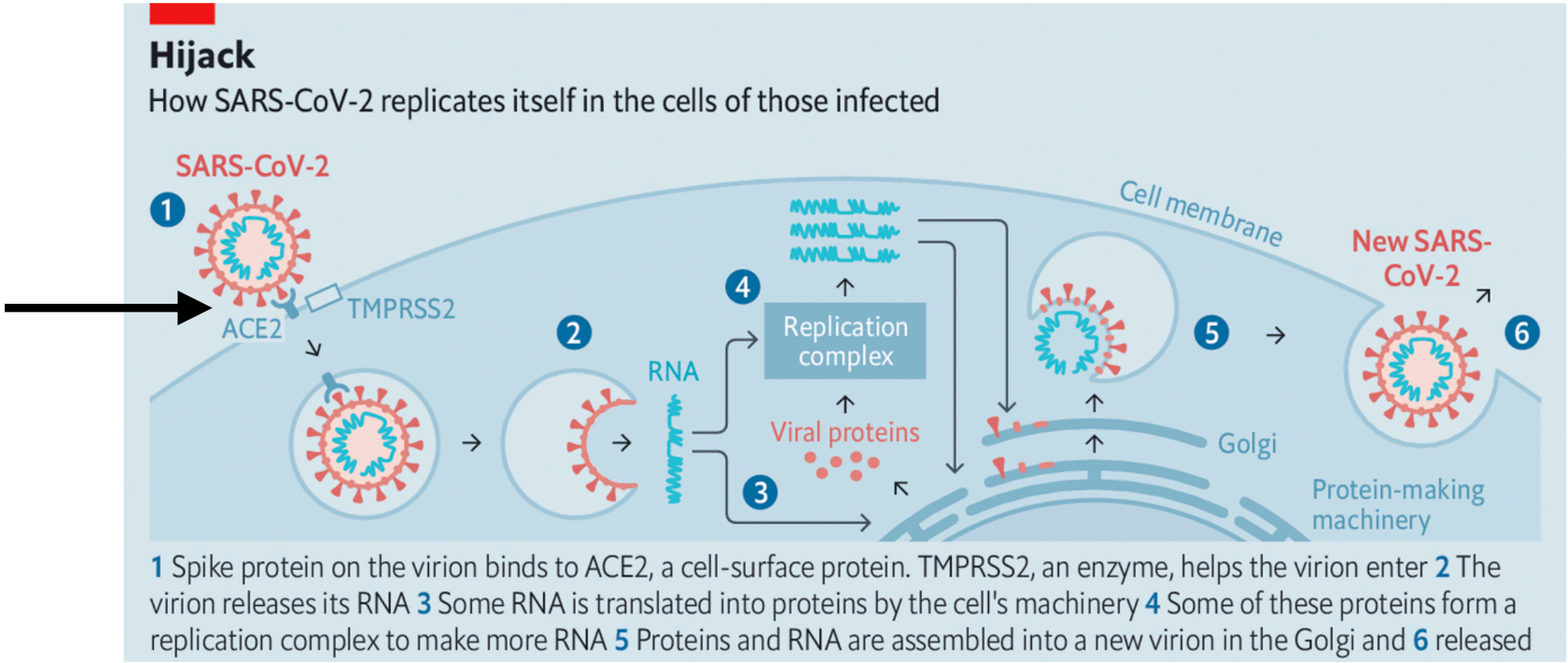
- For drugs against coronavirus, there are several possibilities:



https://www.economist.com/sites/default/files/images/print-edition/20200314_FBC902.png

Drug discovery

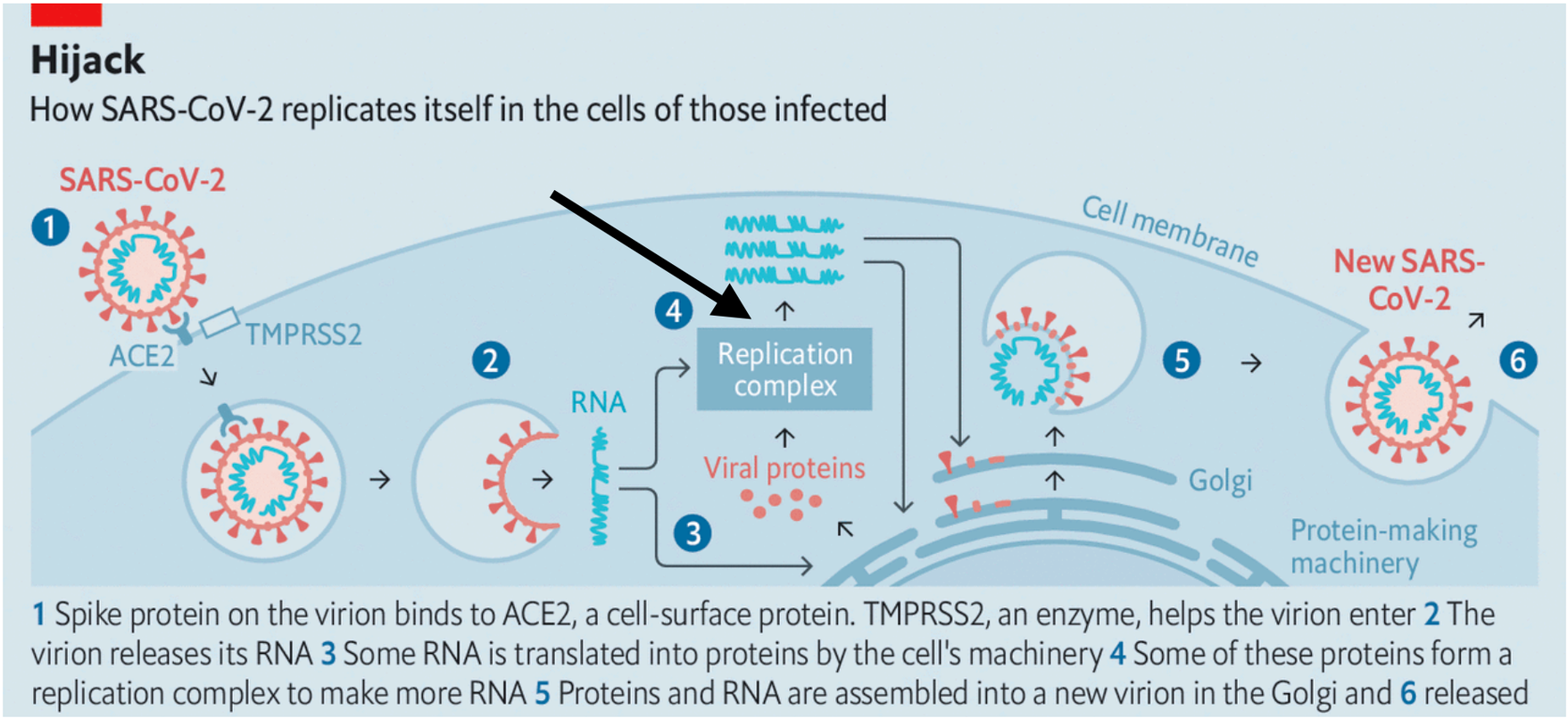
- For drugs against coronavirus, there are several possibilities:
  - Prevent the coronavirus from entering the cell.



https://www.economist.com/sites/default/files/images/print-edition/20200314_FBC902.png

Drug discovery

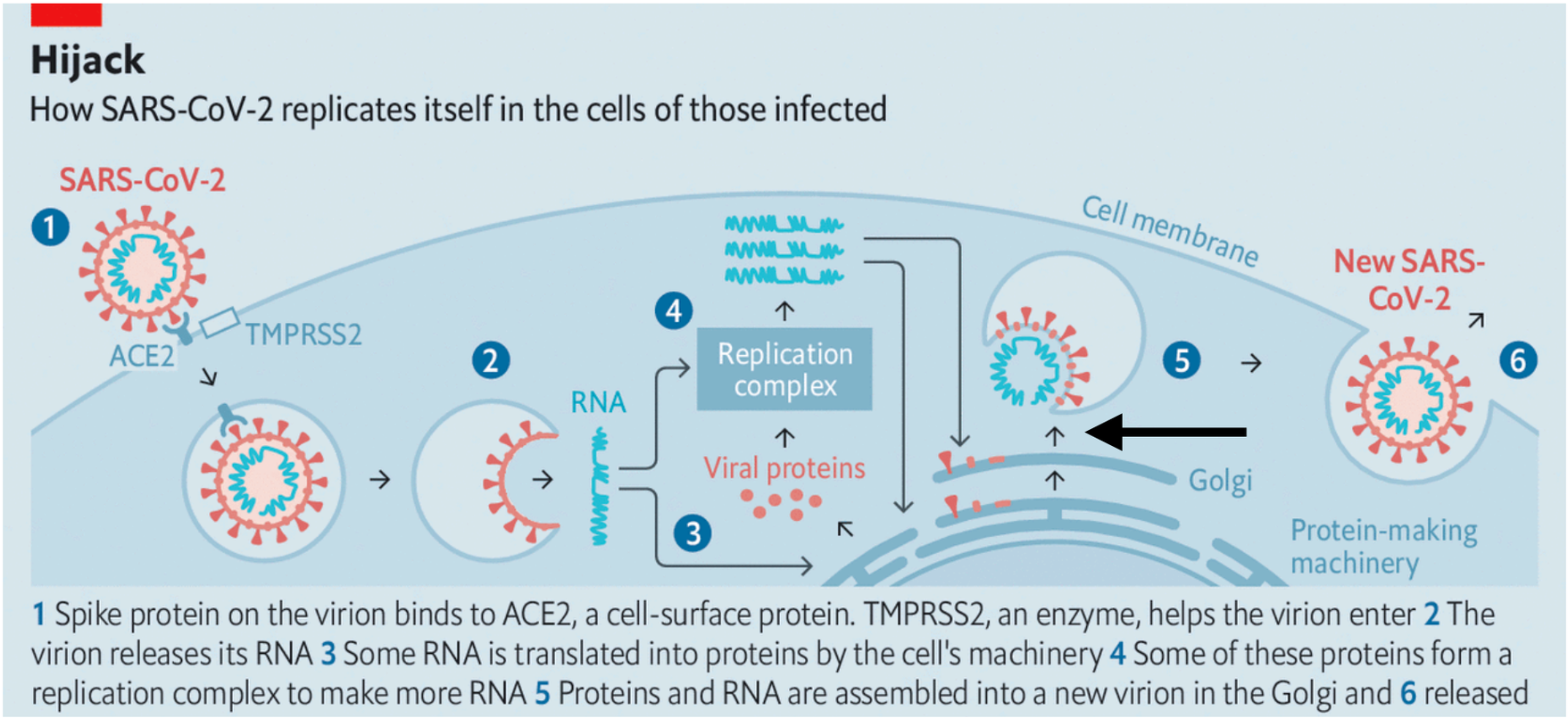
- For drugs against coronavirus, there are several possibilities:
  - Prevent the coronavirus from entering the cell.
  - Disrupt the virus' ability to replicate its RNA.



https://www.economist.com/sites/default/files/images/print-edition/20200314_FBC902.png

Drug discovery

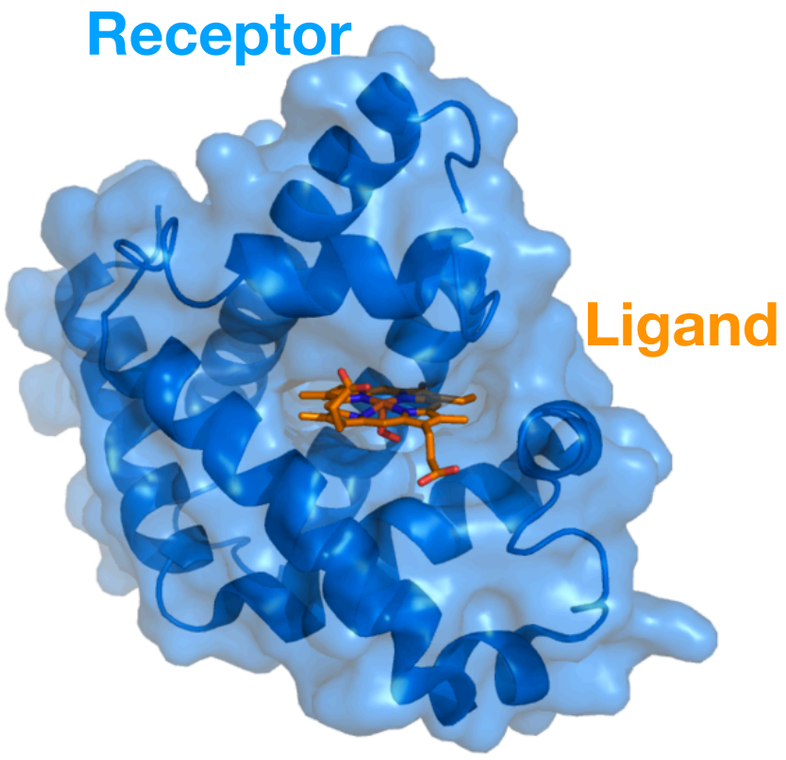
- For drugs against coronavirus, there are several possibilities:
  - Prevent the coronavirus from entering the cell.
  - Disrupt the virus' ability to replicate its RNA.
  - Inhibit the virus' self-assembly process.



https://www.economist.com/sites/default/files/images/print-edition/20200314_FBC902.png

Ligand-receptor binding

- Drugs are molecules (**ligands**) that bind to a protein receptor, which can belong to either the human host or the virus itself.

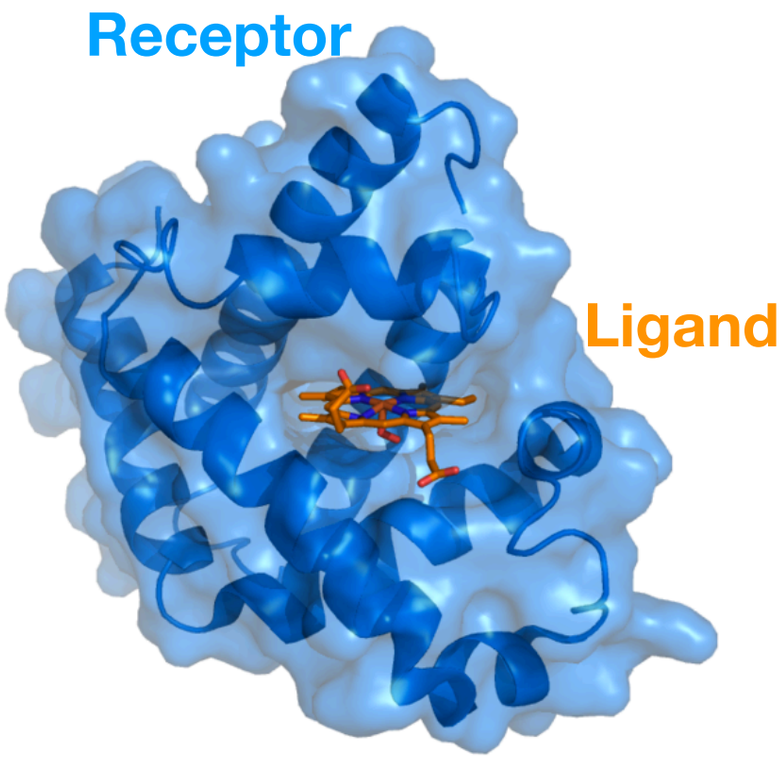


Ligand-receptor binding

- Drugs are molecules (**ligands**) that bind to a protein receptor, which can belong to either the human host or the virus itself.

- Once bound to the receptor, the protein's behavior may change; this can be harnessed to create a drug.

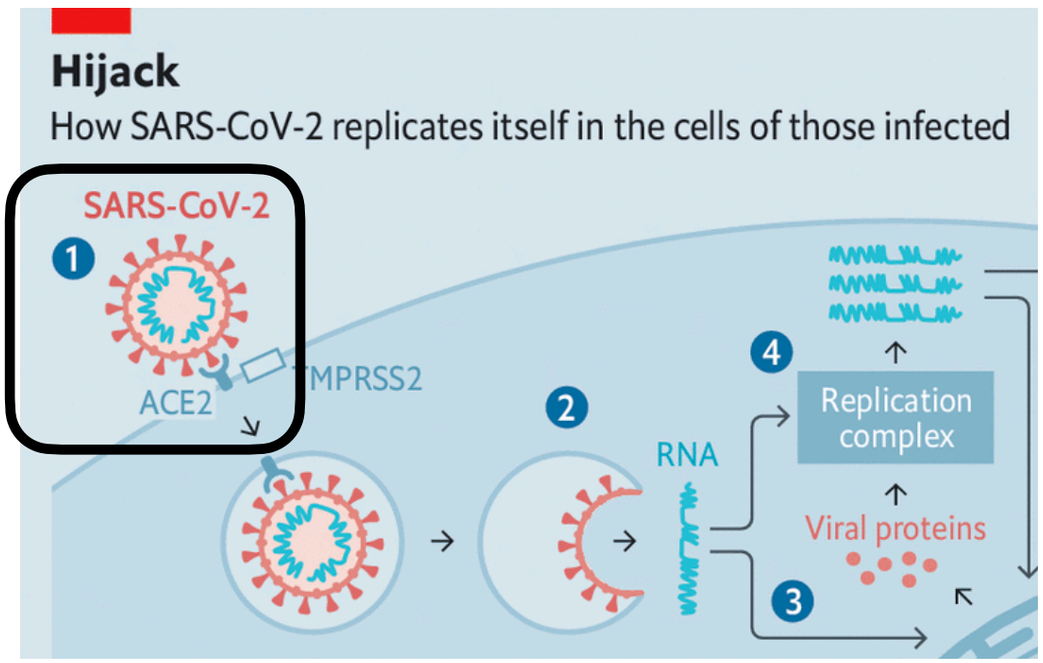
- A useful receptor constitutes a **target** in the drug discovery process.



Ligand-receptor binding

- For SARS-CoV-2, one particular target is its spike protein that binds to the ACE2 on the human cells' membranes.

- If we can find a molecule that binds with enough affinity (attraction between ligand and receptor), then we might inhibit the virus from entering the cell.



SARS-CoV-2 and ACE2

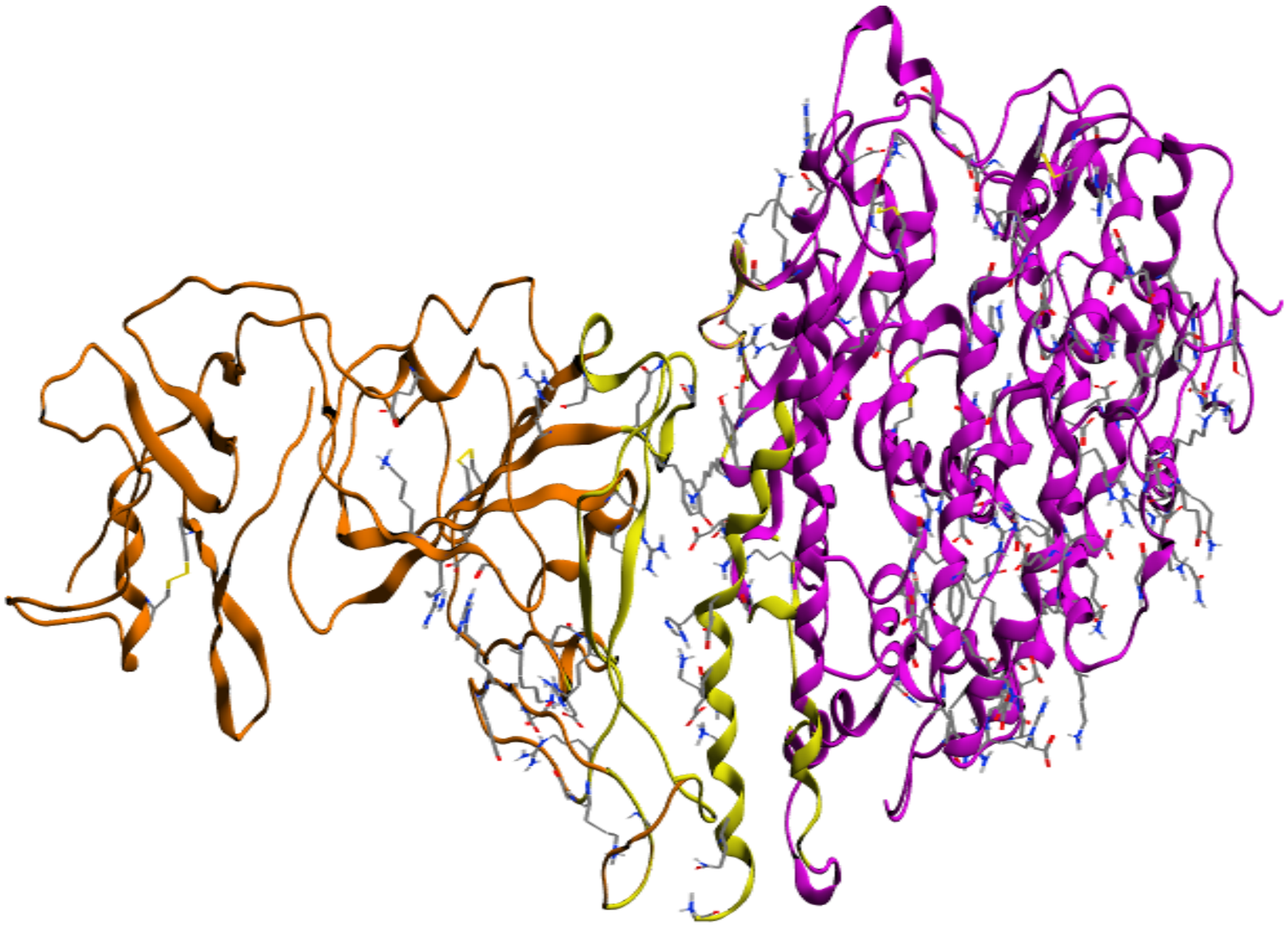


Figure 1) Rendering of nCoV-2019 S-protein and ACE2 receptor complex. Orange ribbons represent the S-protein, purple corresponds to ACE2, and yellow is a highlight of the interface targeted for docking.

Smith and Smith 2020

Drug discovery

- Drug discovery is extremely expensive and slow (often 5-10 years):
  1. Identify a target in either the host cells or the virus.
  2. Search for a compound that can bind to the target.
     a. Rational design based on the structure of the target.
     b. High-throughput screening of existing compound libraries.
  3. Perform in-vivo studies on animals to measure efficacy and safety.
  4. Perform clinical trials on humans to measure efficacy and safety.

Drug discovery

- Drug discovery is extremely expensive and slow (often 5-10 years):
  1. Identify a target in either the host cells or the virus.
  2. Search for a compound that can bind to the target.
     a. Rational design based on the structure of the target.
     b. High-throughput screening of existing compound libraries.
     c. Computational techniques to identify a promising “starting point” for a drug.
  3. Perform in-vivo studies on animals to measure efficacy and safety.
  4. Perform clinical trials on humans to measure efficacy and safety.

Drug discovery

- One technique for automatically identify potential drug compounds is to simulate the **molecular dynamics** (MD) of how the compound interacts with the receptor.

https://www.youtube.com/watch?v=meEX8jDF9-k

Mollica et al. 2015

Drug discovery

- Researchers at the Oak Ridge National Laboratory recently explored how MD simulations conducted on a supercomputer could identify potentially useful compounds from a set of already approved drugs.

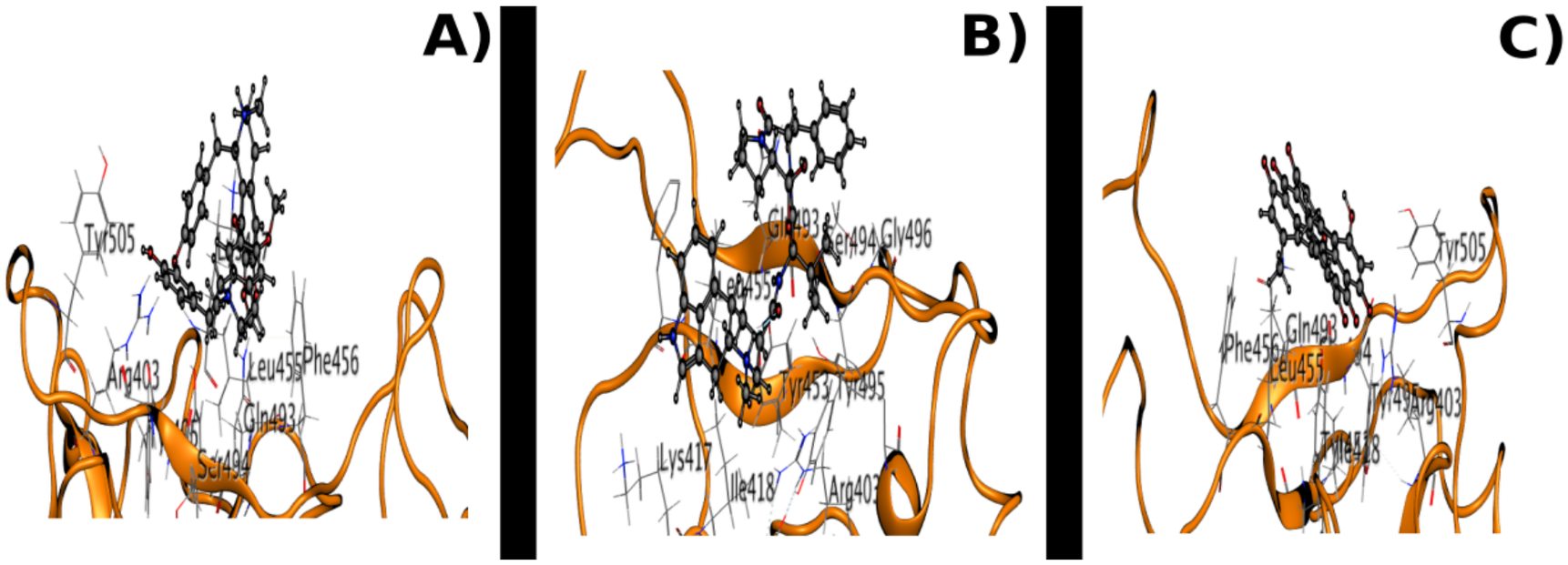


Figure 7) Renderings of three of the top scoring previously regulator approved small-molecules binding with the S-protein receptor recognition region. A) Cepharanthine (ZincID: 30726863). B) Ergoloid (ZincID: 3995616). C) Hypericin (Zinc ID: 3780340). Orange ribbons represent the S-protein.

Smith and Smith 2020

Drug discovery

- However, MD simulations are computationally extremely intensive.

- Smith and Smith's 2020 study was conducted on one of the world's largest supercomputers.

- Is there a way to identify candidate compounds more quickly, i.e., without MD simulations?

- Machine learning may be a useful tool.

Machine learning for drug discovery

- We can formulate the problem as follows:

  - Let **x** represent the components and structure of a candidate ligand (e.g., chemical formula, amino acid sequence, 3-D molecular structure) as well as the target, e.g.:

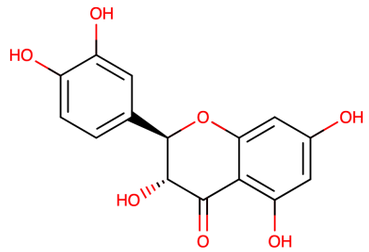
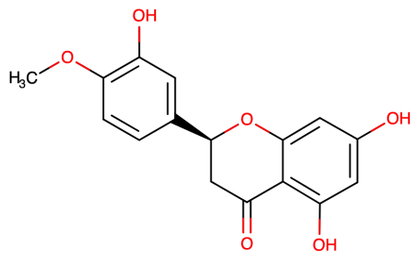
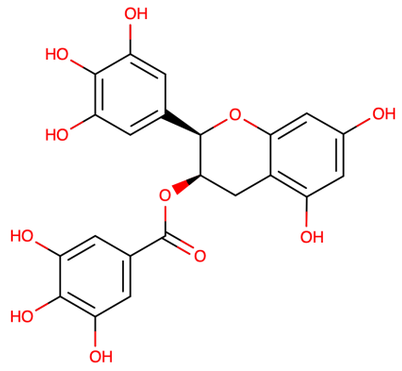
    `x = "OC1=C(C([C@H](O)… mssssw lllslvavtaaqstieeq… ]"`

  - Let *y* represent the affinity between the compound and the target receptor, e.g.:

    $y = 2.4\ \mu M$

  - We want to compute a function *f* that maps **x** to its associated *y*.

Machine learning for drug discovery

x (chemical structure)	*y* (affinity)*
 **Taxifolin**	**2.4 *μM***
 **Hesperitin**	**2.2 *μM***
 **Epigallocatechin gallate**	**0.8 *μM***

*These numbers are fictitious.

Machine learning for drug discovery

- A key requirement is that *f* is more accurate and/or faster to compute than other techniques to determine the binding affinity (e.g., MD simulations).

- Once we have *f*, we can:

  - Estimate the binding affinity for a novel compound+target combination.

  - Search for the best compound from a large set.

Machine learning for drug discovery

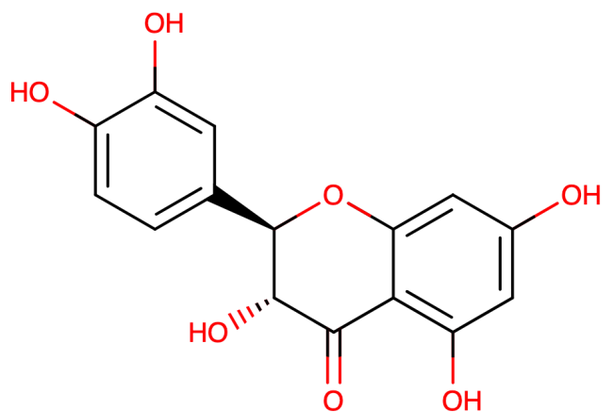
- Using a similar process, we could potentially compute a (different) function *g* that estimates the toxicity *y* of a given compound based on its chemical structure **x**.

Modeling molecules and their interactions

- We can represent the structure and attributes of a chemical molecule as a **graph**:

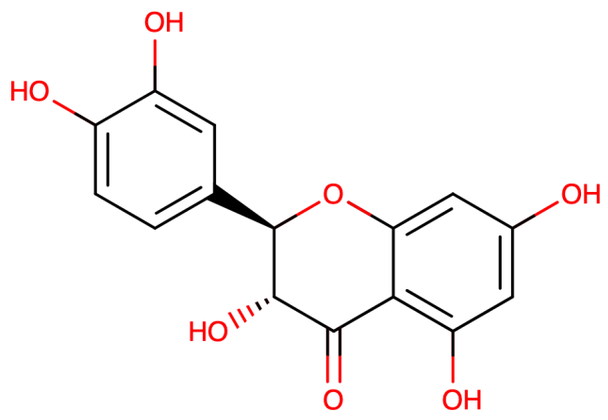
  - Each atom is a node with associated features.

- Each bond is an edge with associated features.



Modeling molecules and their interactions

- We want to extract high-level features from sub-graphs, irrespective of their locations in the whole graph.
- This is akin to a convolution kernel that operates on 2D sub-images in the same way, irrespective of location.



Modeling molecules and their interactions

- Starting next lecture, we will examine Graph Convolutional Networks (GCNs).
- Example application to drug discovery.