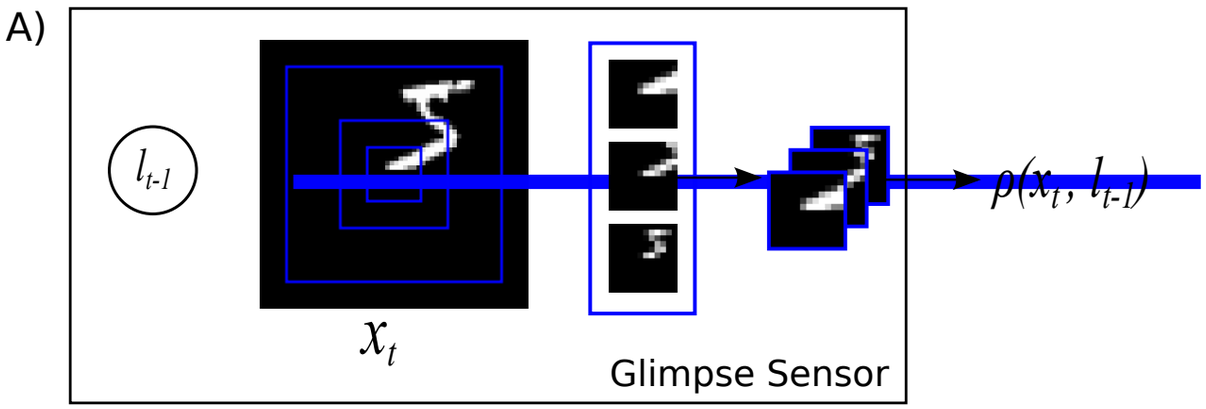
CS/DS 541: Class 22

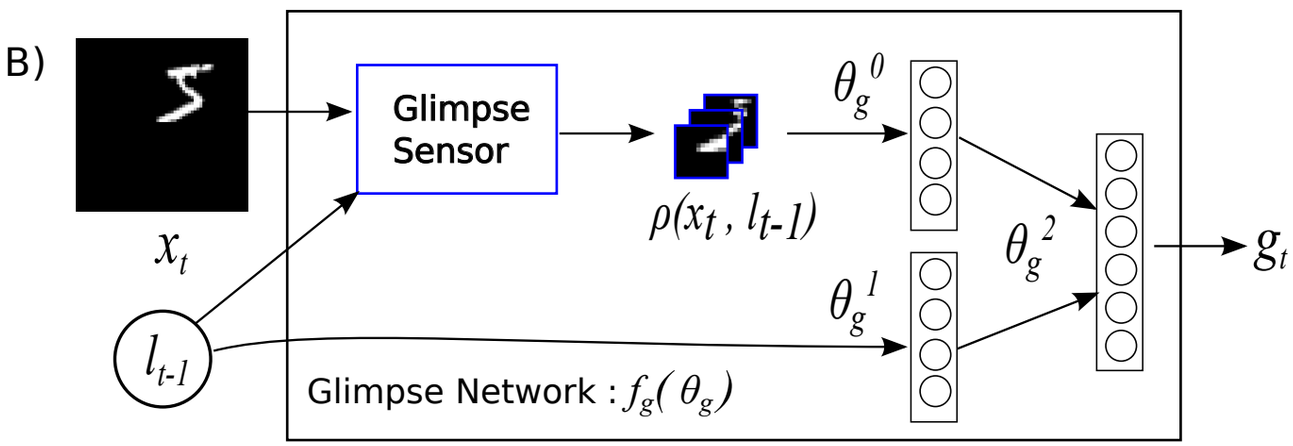
Jacob Whitehill

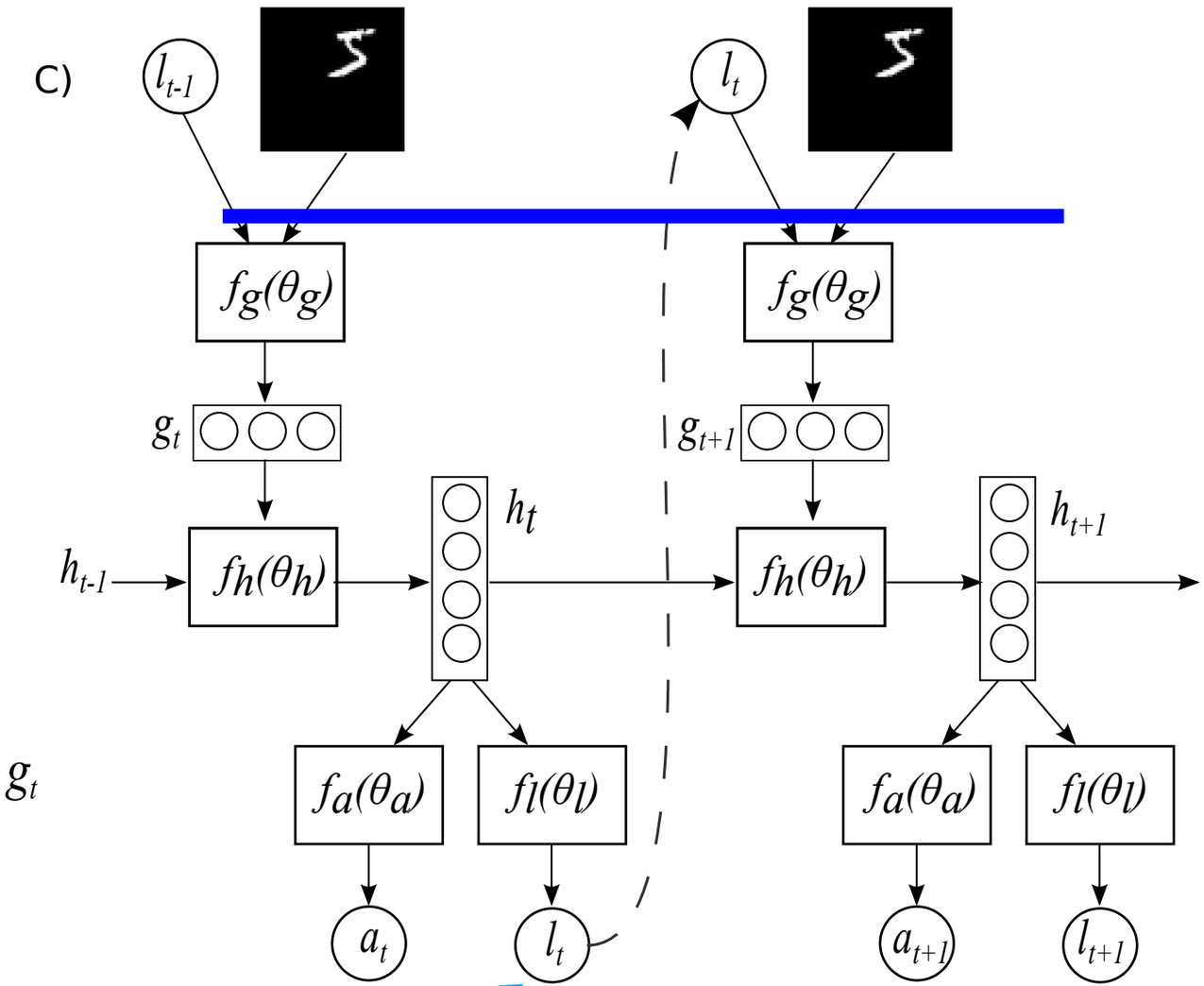
Recurrent models of visual attention

Mnih et al. 2014

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.
  - $P(\tau \mid \theta)$ 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:
  - 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:

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

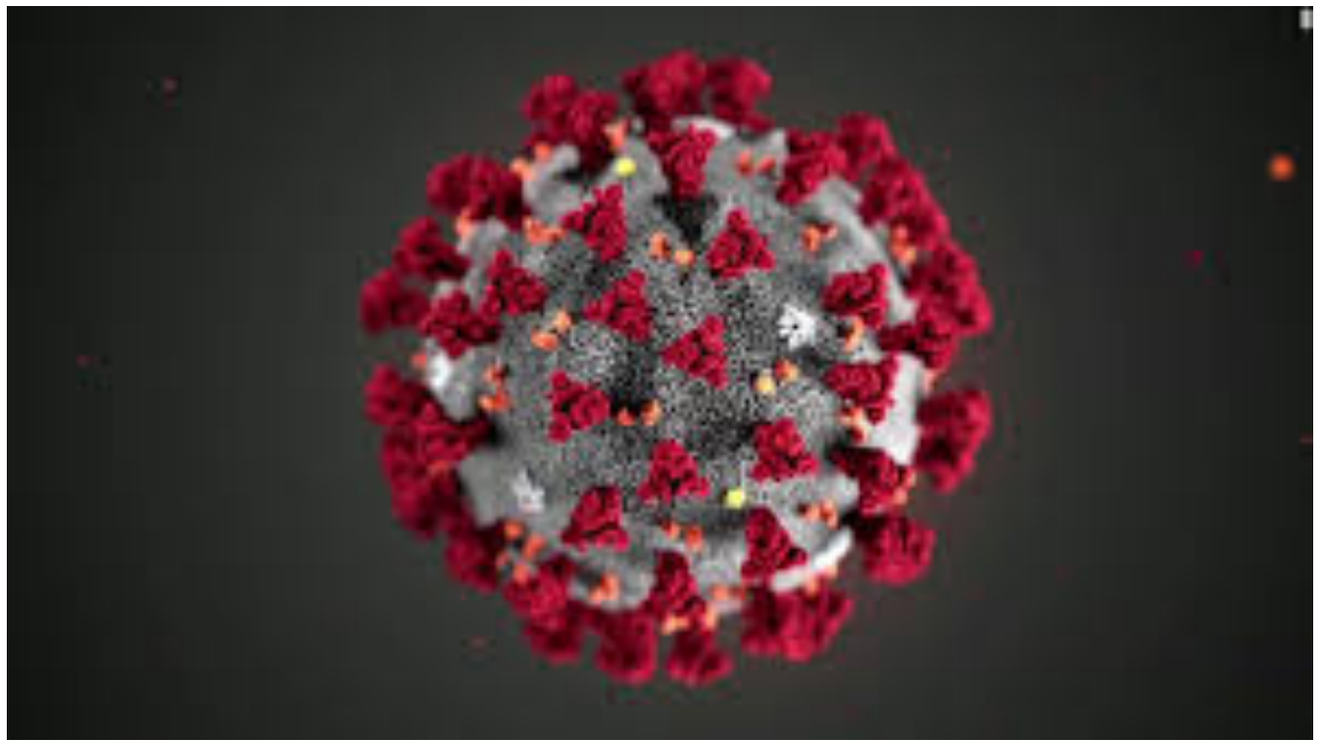
Maximizing expected return

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

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

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

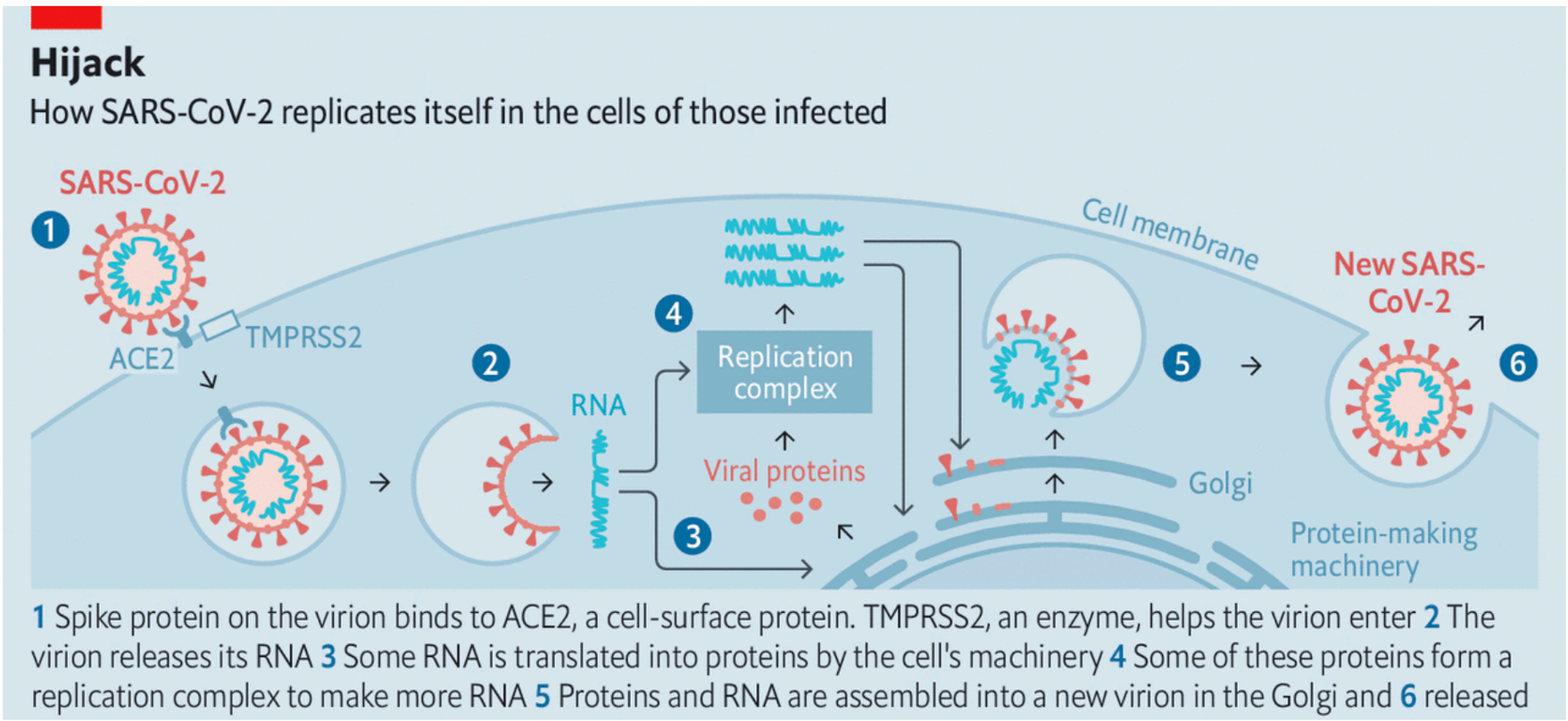
- For SARS-CoV-2 (coronavirus) and many other pathogens, scientists are searching intensely for:

- Vaccines to prevent infection.

- Therapeutic drugs to treat infection.

Drug discovery

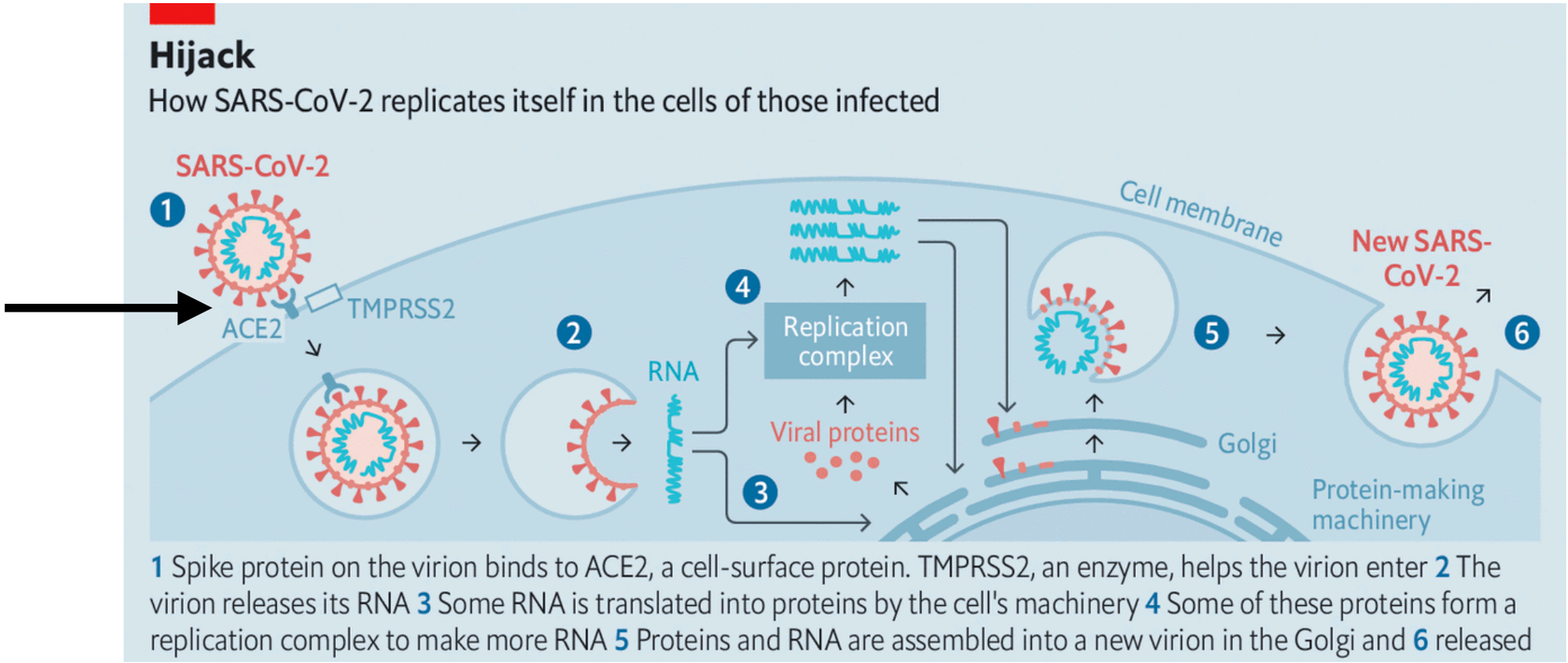
- For drugs against coronavirus, there are several possibilities:



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

Drug discovery

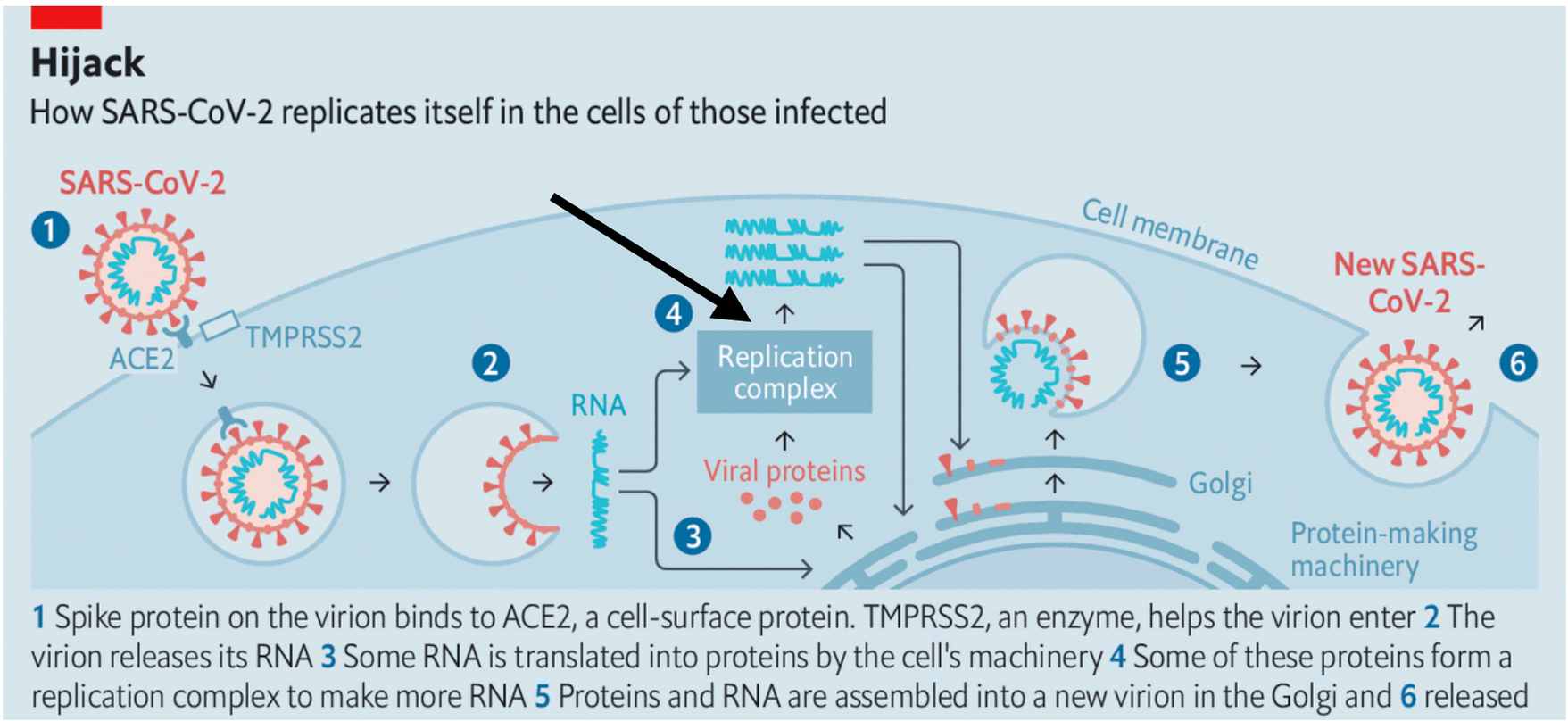
- For drugs against coronavirus, there are several possibilities:
  - Prevent the coronavirus from entering the cell.



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

Drug discovery

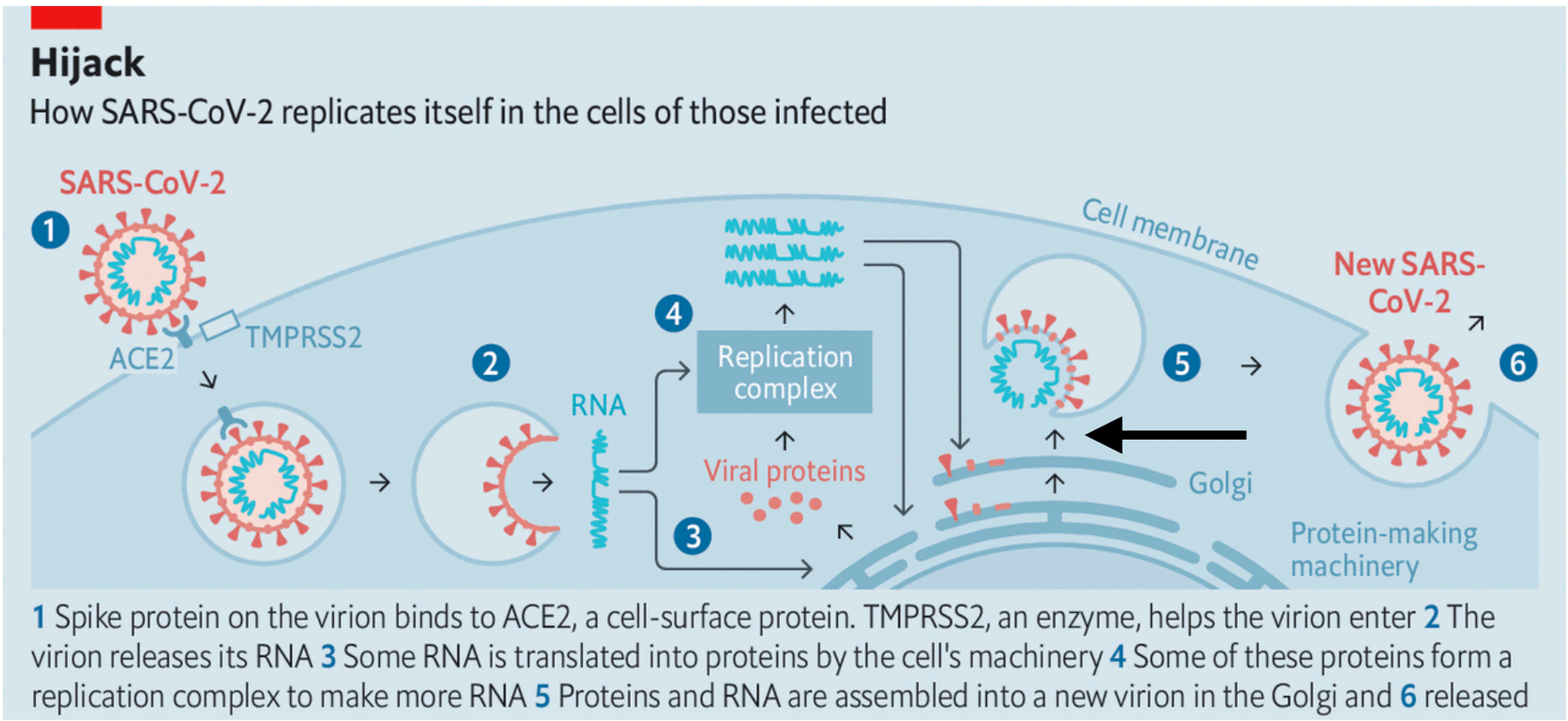
- For drugs against coronavirus, there are several possibilities:
  - Prevent the coronavirus from entering the cell.
  - Disrupt the virus' ability to replicate its RNA.



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

Drug discovery

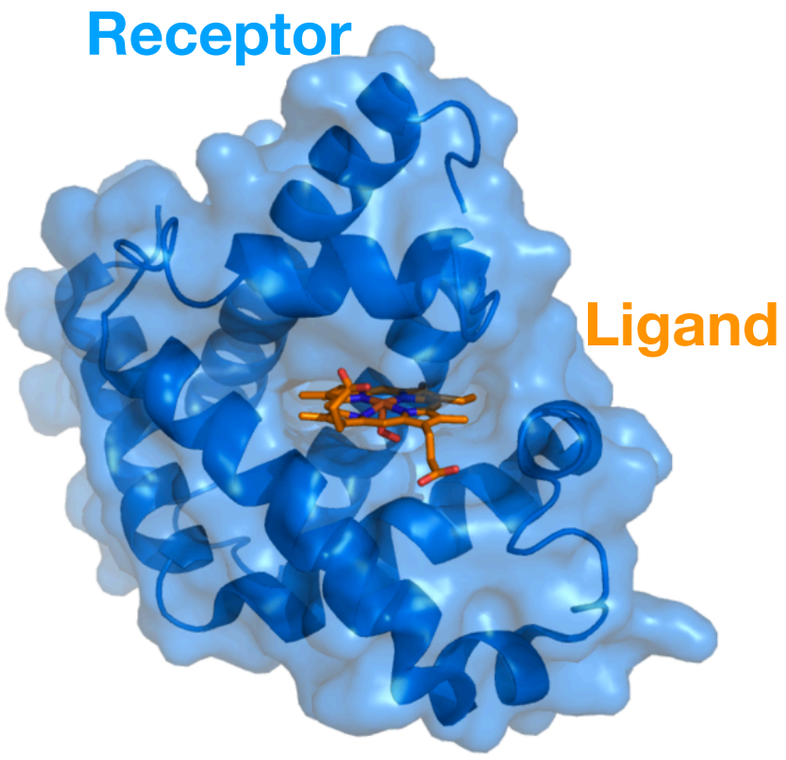
- For drugs against coronavirus, there are several possibilities:
  - Prevent the coronavirus from entering the cell.
  - Disrupt the virus' ability to replicate its RNA.
  - Inhibit the virus' self-assembly process.



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.

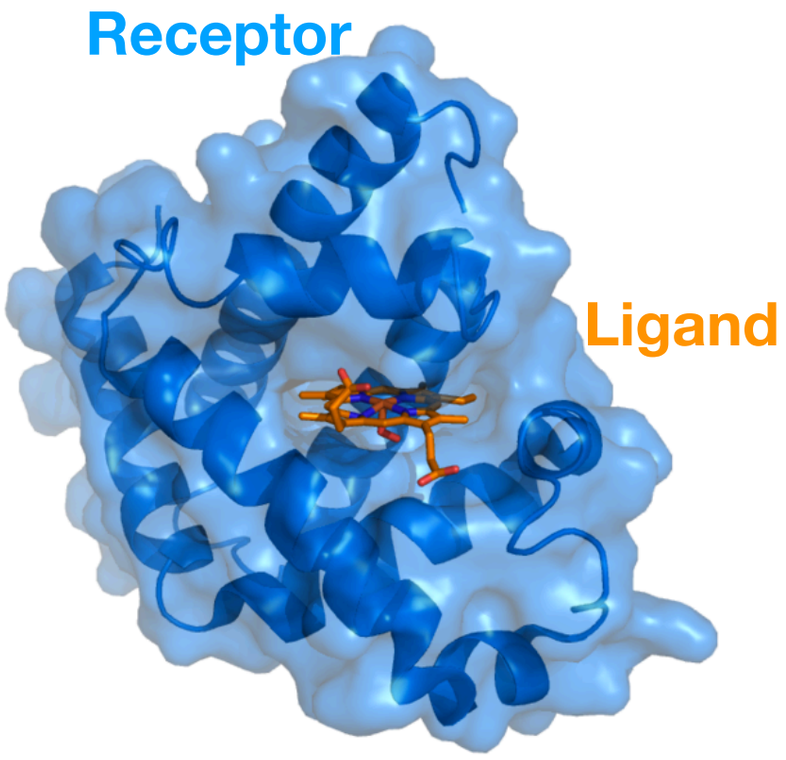


Ligand-receptor binding

- 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.

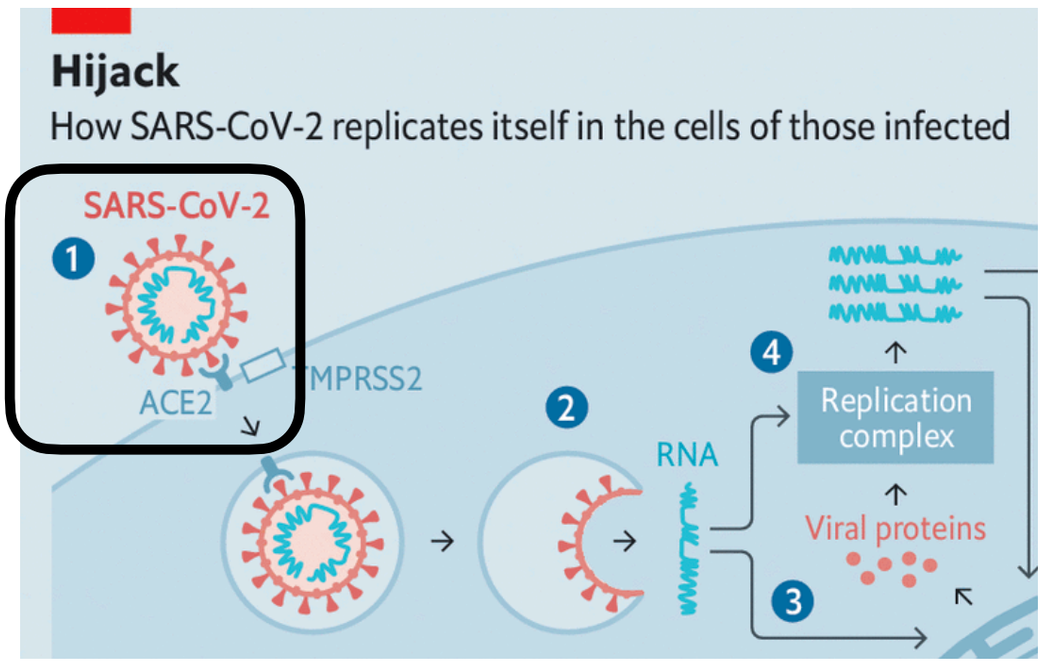
- A useful receptor constitutes a **target** in the drug discovery process.



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

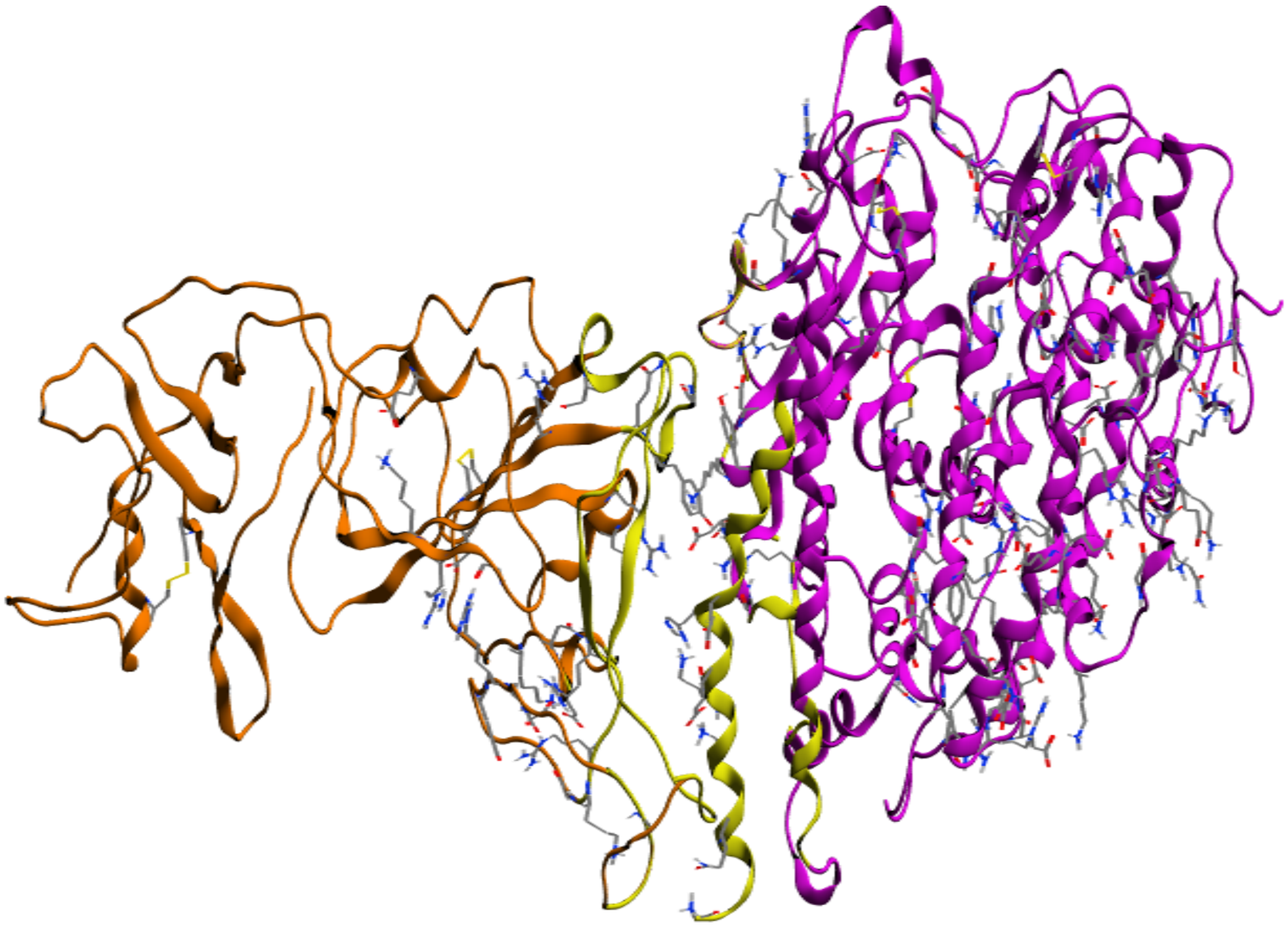


Figure 1) Rendering of nCoV-2019 S-protein and ACE2 receptor complex. Orange ribbons represent the S-protein, purple corresponds to ACE2, and yellow is a highlight of the interface targeted for docking.

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.

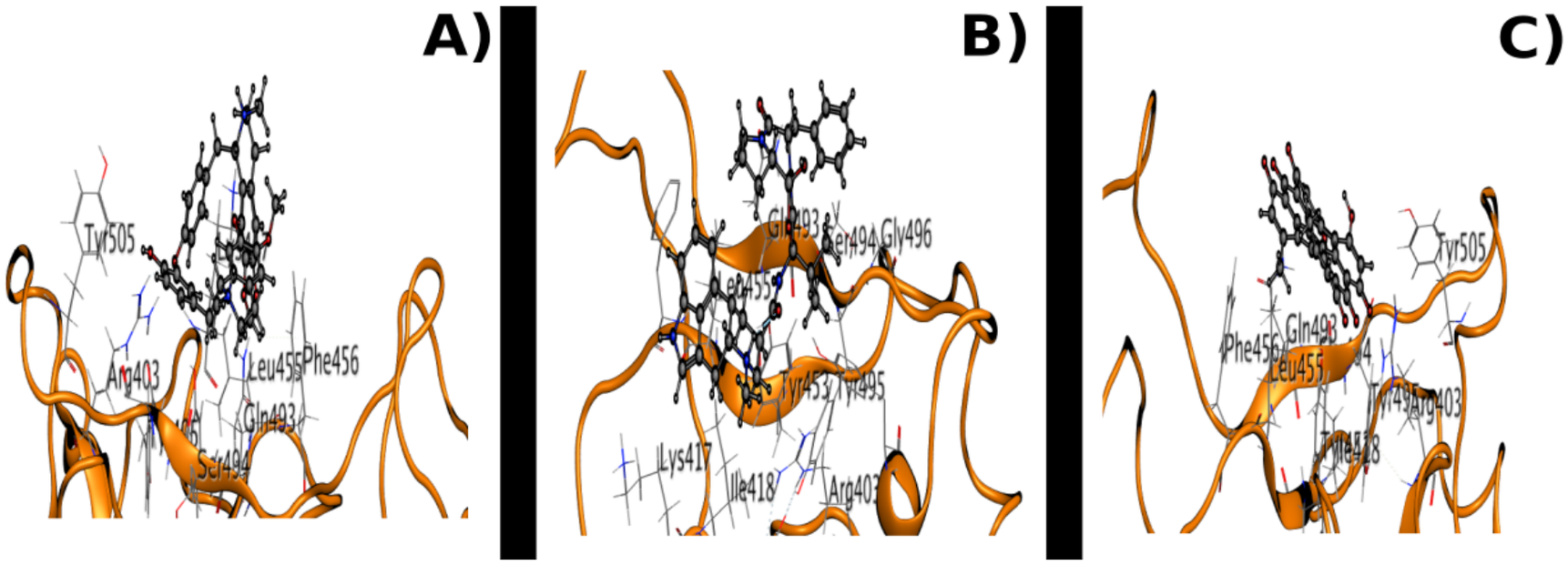


Figure 7) Renderings of three of the top scoring previously regulator approved small-molecules binding with the S-protein receptor recognition region. A) Cepharanthine (ZincID: 30726863). B) Ergoloid (ZincID: 3995616). C) Hypericin (Zinc ID: 3780340). Orange ribbons represent the S-protein.

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.:

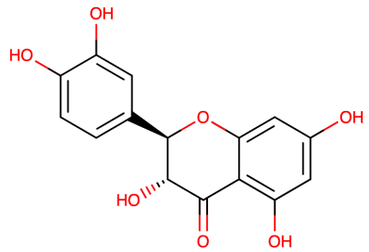
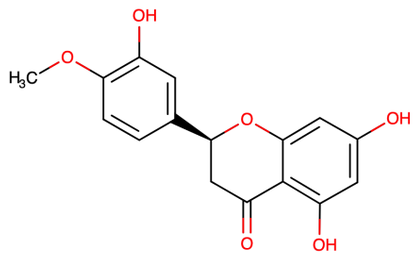
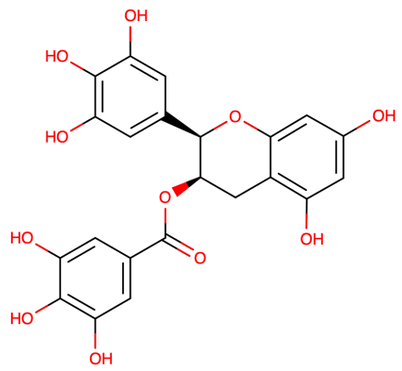
    `x = "OC1=C(C([C@H](O)… mssssw lll slvavtaaqstieeq… ]"`

  - 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

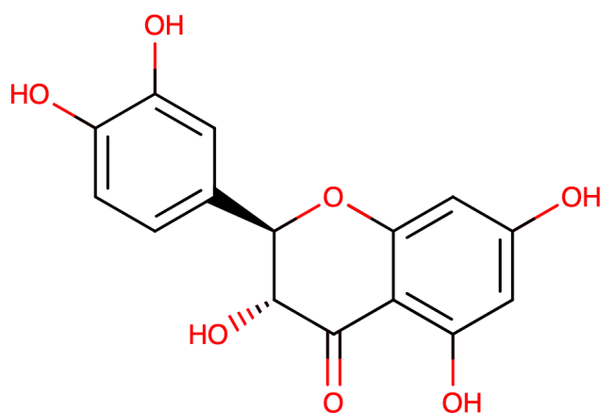
- Using a similar process, we could potentially compute a (different) function *g* that estimates the toxicity *y* of a given compound based on its chemical structure **x**.

Modeling molecules and their interactions

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

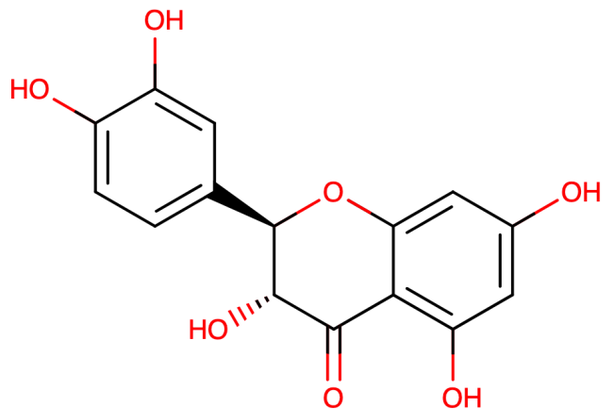
  - Each atom is a node with associated features.

- Each bond is an edge with associated features.



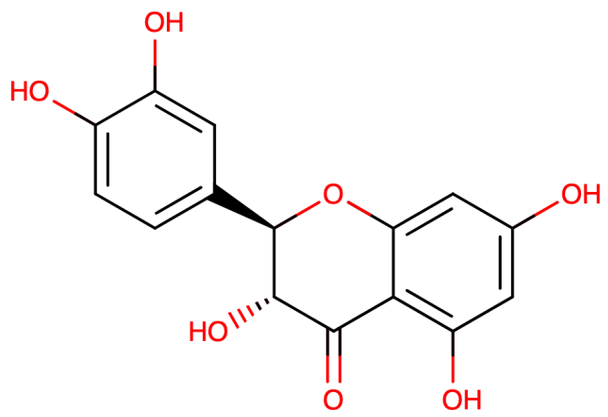
Modeling molecules and their interactions

- 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.



GCN paper and recommended references

- GCN paper (Kipf & Welling 2017).

- Spectral graph theory tutorial.

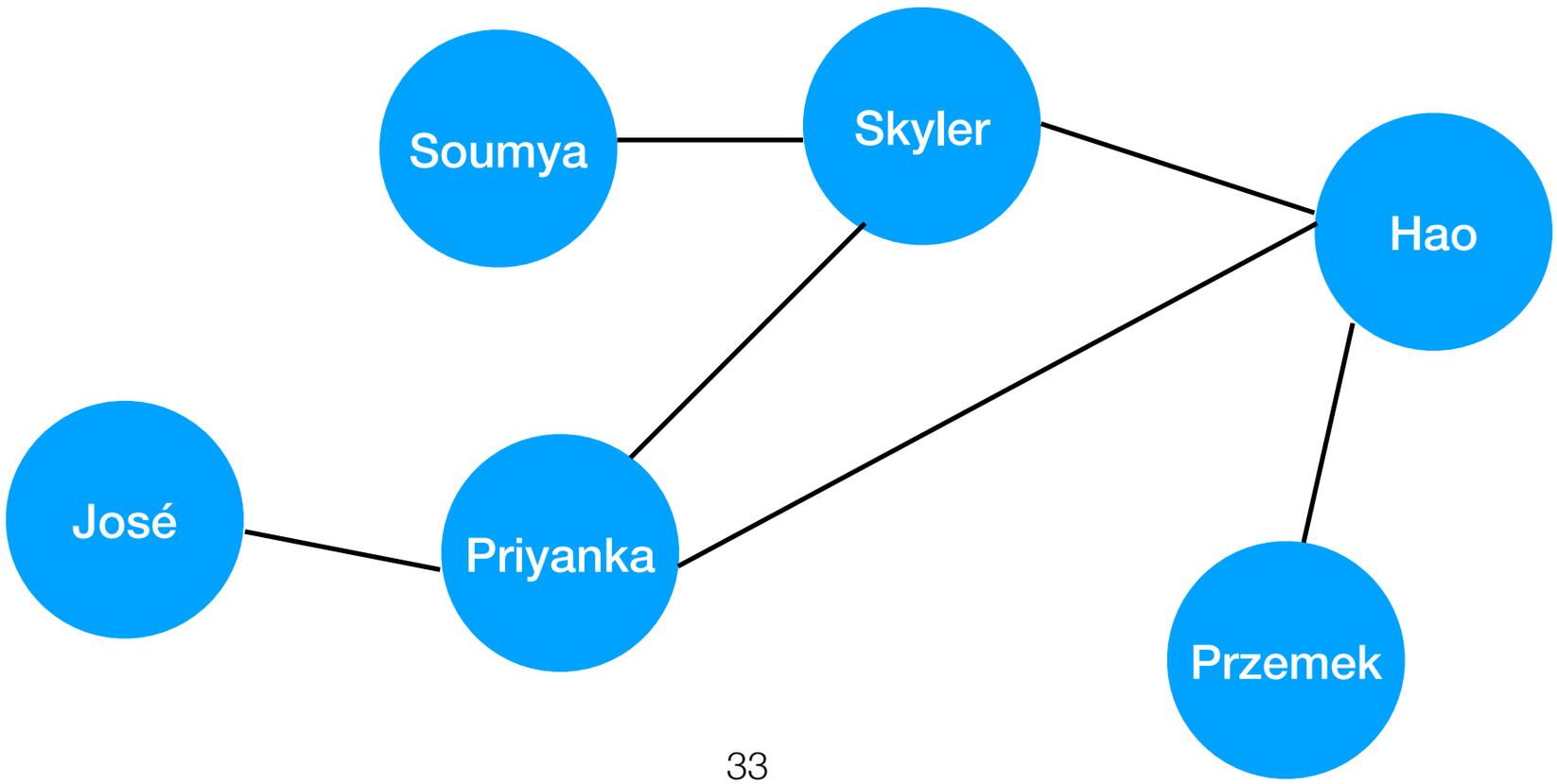
- Fourier analysis on graphs slides.

Machine learning on graphs

Machine learning on graphs

- Consider an undirected graph where each node is associated with a feature vector.

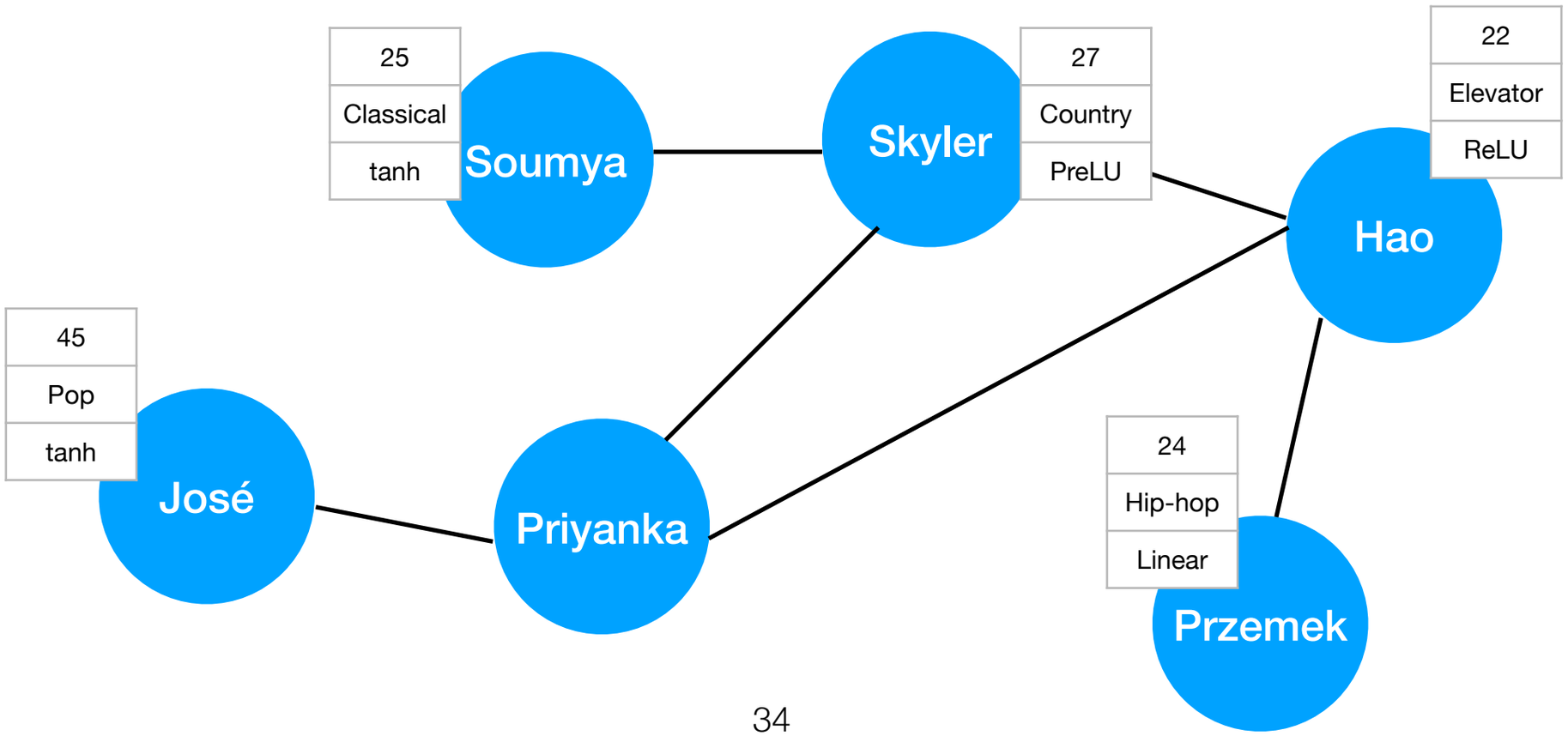
- Example: social network of CS/DS 541.



Machine learning on graphs

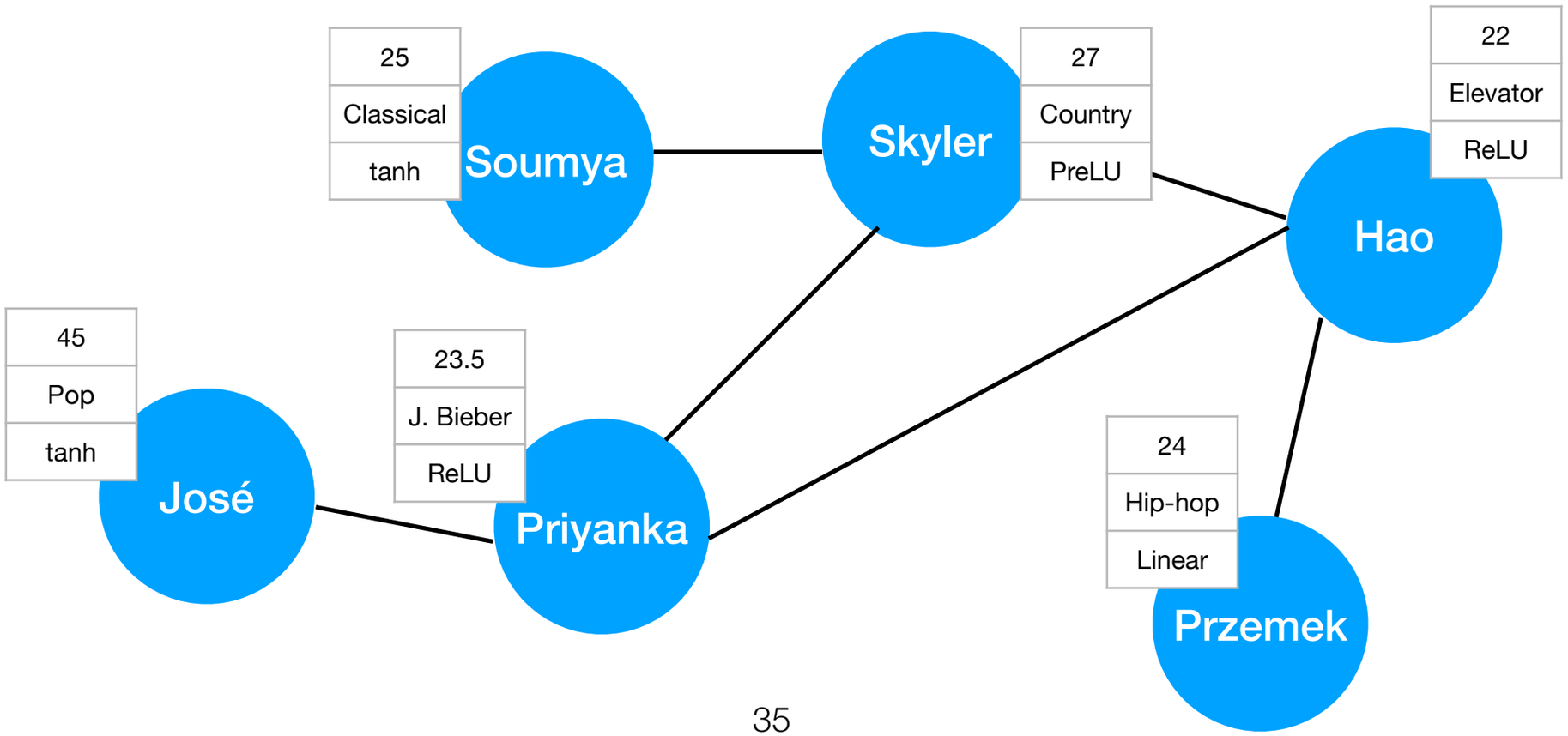
- Consider an undirected graph where each node is associated with a feature vector.

- Example: social network of CS/DS 541.



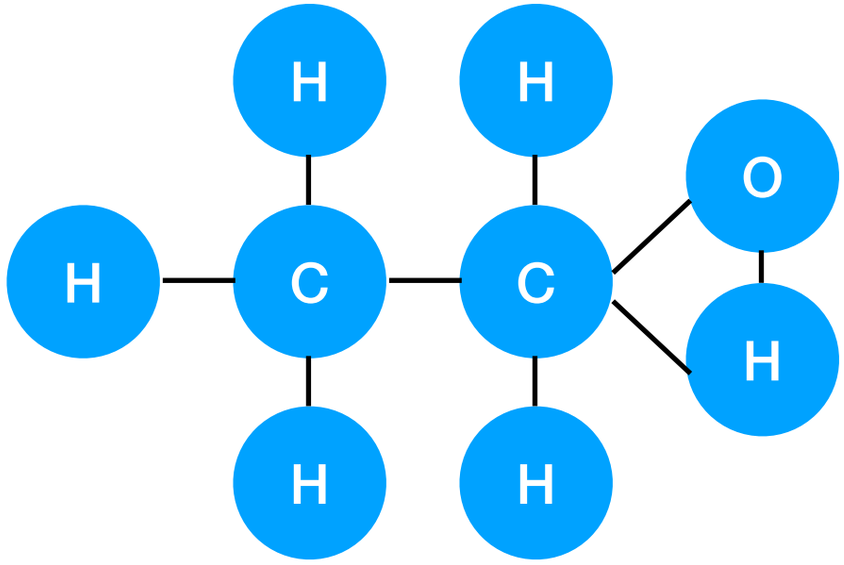
Machine learning on graphs

- Based on the graph topology and set of observed feature vectors, can we infer the unobserved feature vector?
  - This is a semi-supervised ML problem.



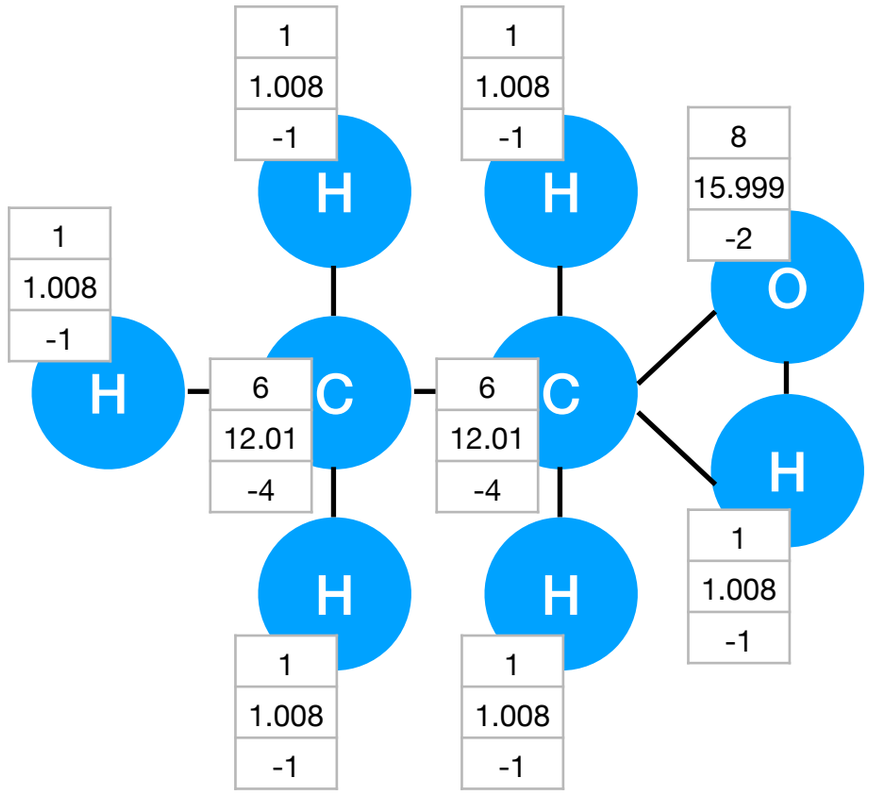
Machine learning on graphs

- Example: computational chemistry.



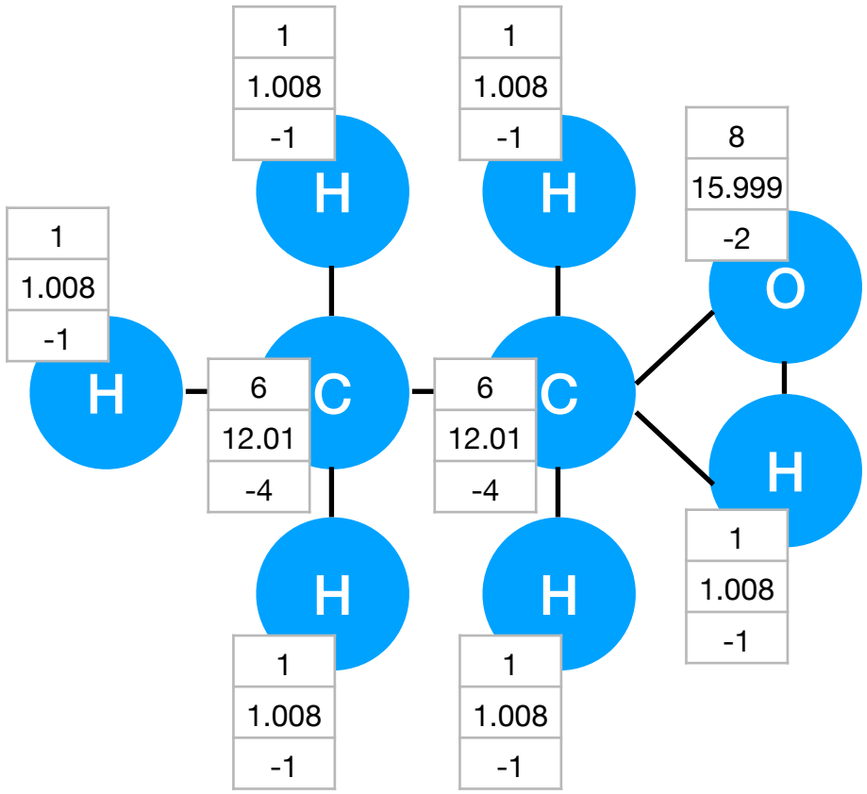
Machine learning on graphs

- Example: computational chemistry.



Machine learning on graphs

- Can we predict the binding affinity of a molecule to a particular receptor given its molecular structure?

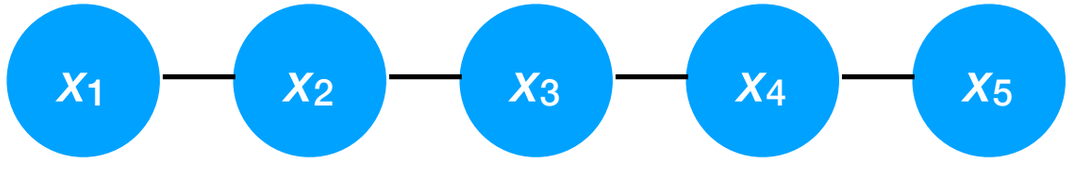


2.2 *μM*

Machine learning on graphs

- In contrast to vectors and images, the topology of graphs is **variable**.

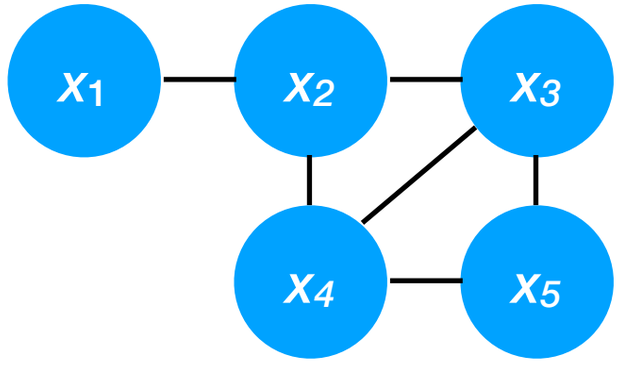
- E.g., while each element of an *n*-vector is connected to 1-2 neighbors



Machine learning on graphs

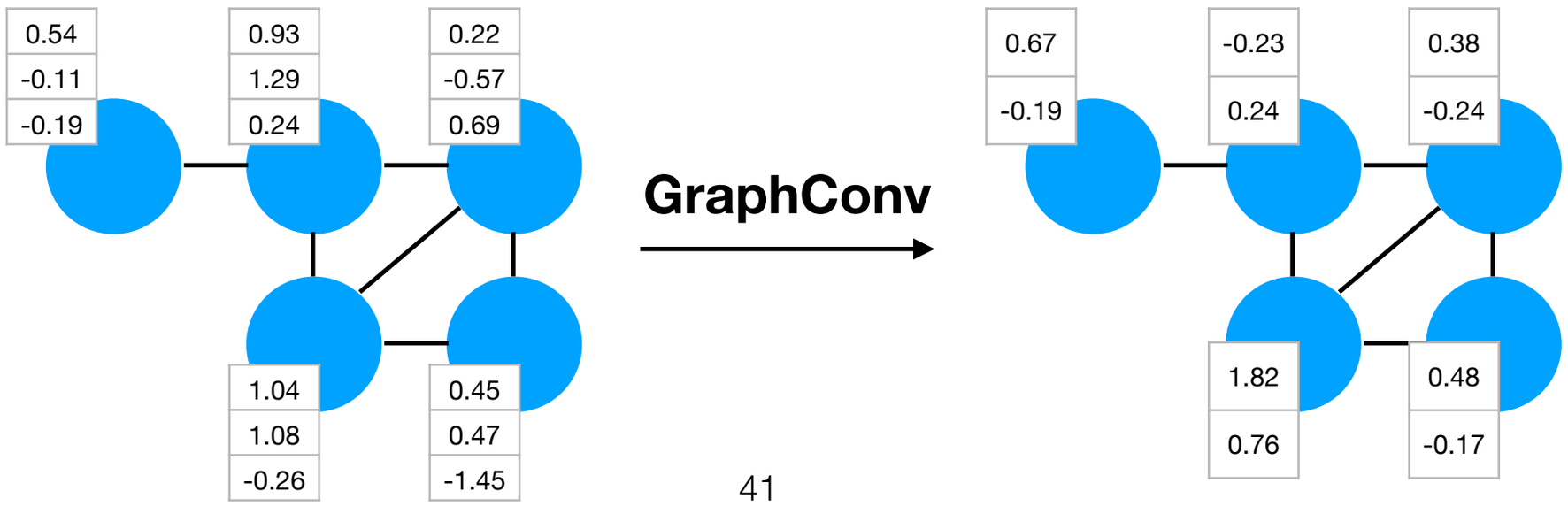
- In contrast to vectors and images, the topology of graphs is **variable**.

- E.g., while each element of an *n*-vector is connected to 1-2 neighbors, each graph node has up to *n*-1 neighbors.



Machine learning on graphs

- With GCNs, the feature vector at each node is transformed (by each GraphConv layer) based on the feature vectors of nearby nodes.

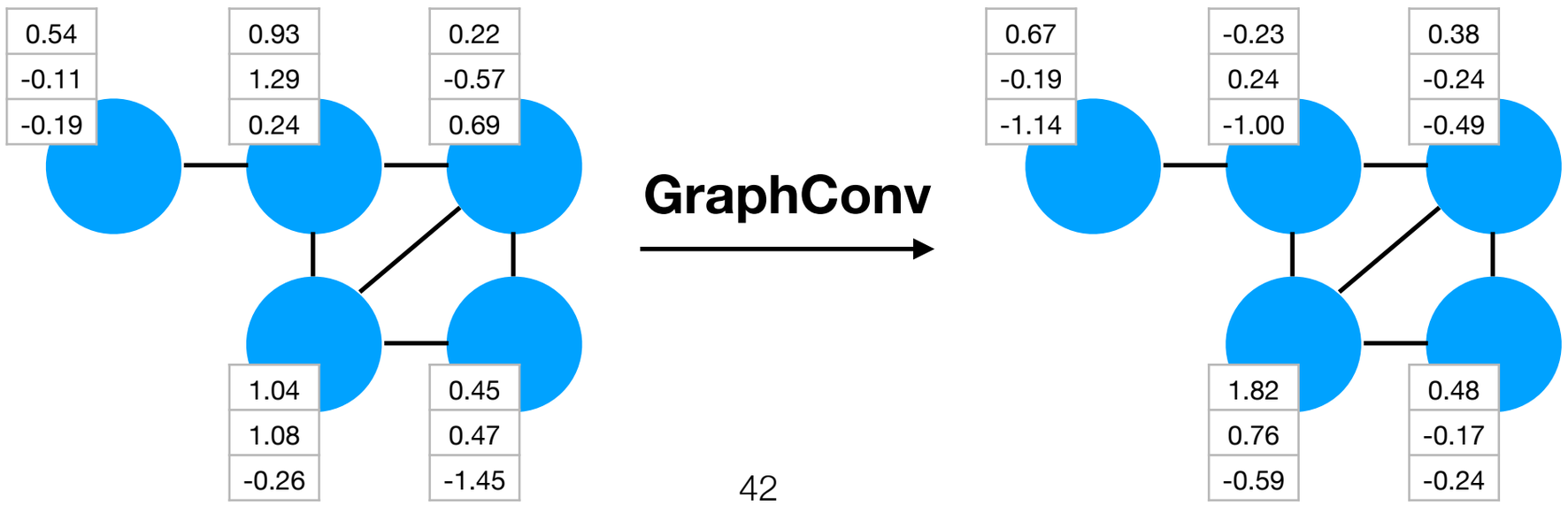


Machine learning on graphs

- With GCNs, the feature vector at each node is transformed (by each GraphConv layer) based on the feature vectors of nearby nodes.

- With *true* graph convolution, “nearby” actually means *all* nodes.

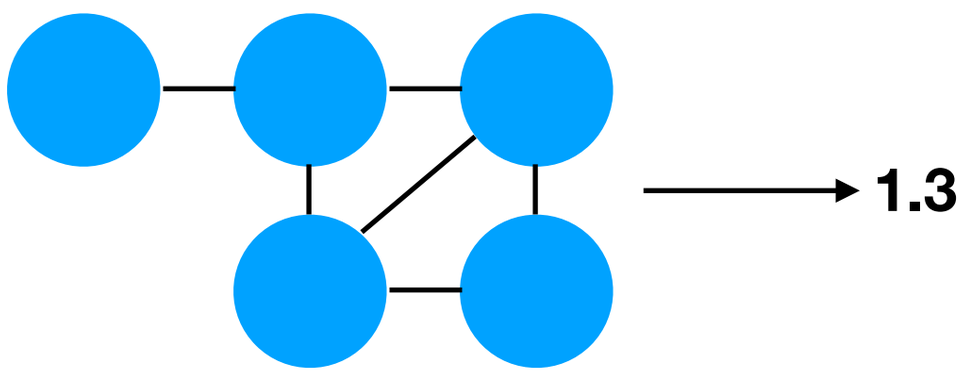
- However, most GCNs actually perform *approximate* convolution where the set of influencing nodes is local.

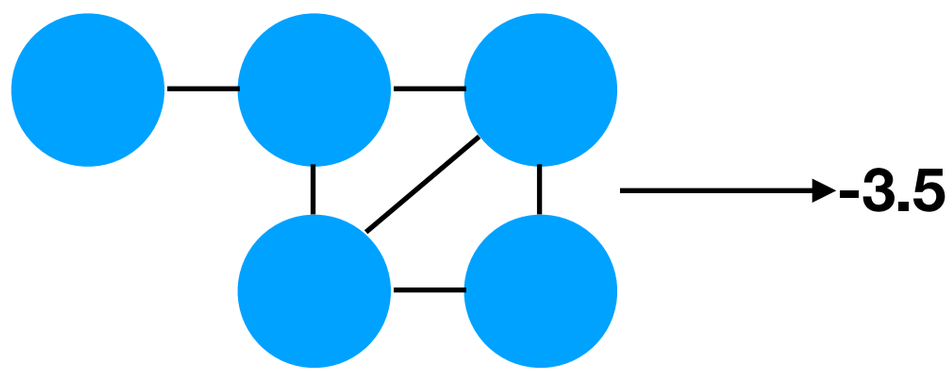


Machine learning on graphs

- When doing ML on graphs, ultimately we must either:

    1. Restrict the topology of all input graphs to be the same (but with different feature vectors):
        - Supervised: apply a tailored aggregation function that depends on a specific graph topology.
        - Semi-supervised: classify the unlabeled nodes.

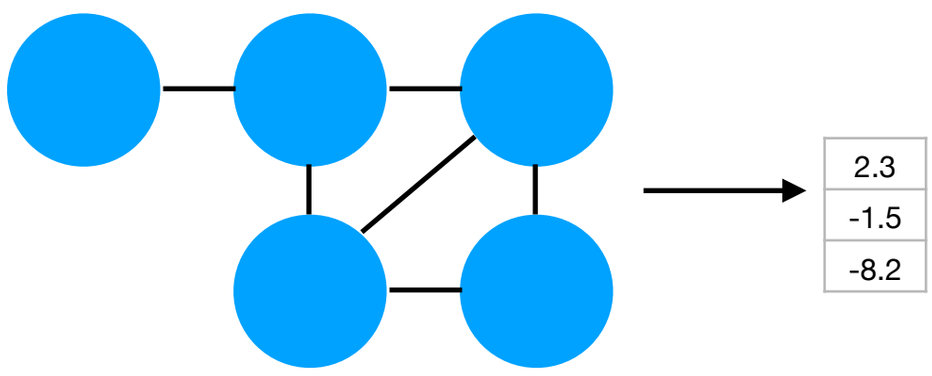




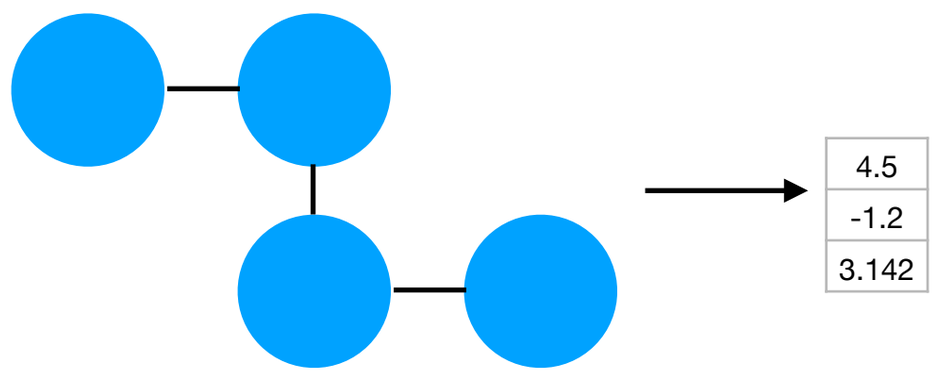
Machine learning on graphs

- …or:

    2. Apply pooling to convert the graph into a fixed-length feature vector for downstream processing.



$\mathbf{x}^{(1)}$ $\mathbf{h}^{(1)}$



$\mathbf{x}^{(2)}$ $\mathbf{h}^{(2)}$

Rethinking convolution

Convolution is linear

- Recall that convolution is linear and can thus be implemented by multiplication with a matrix **W**, e.g.:

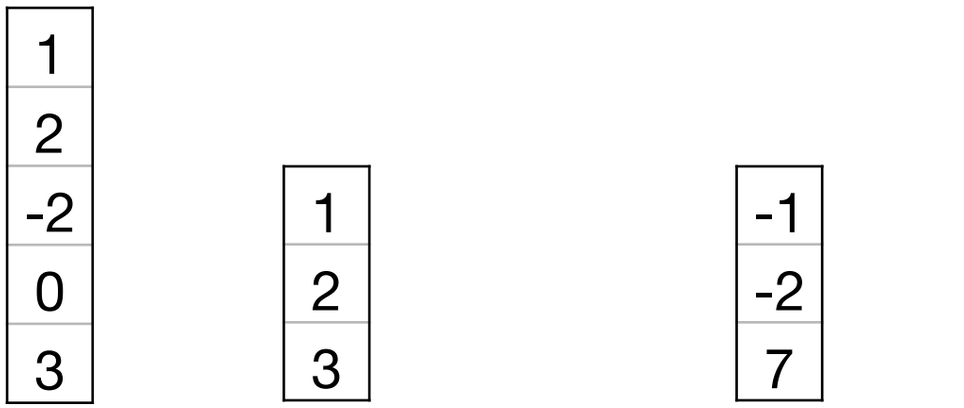


Image **Filter** **Feature map**

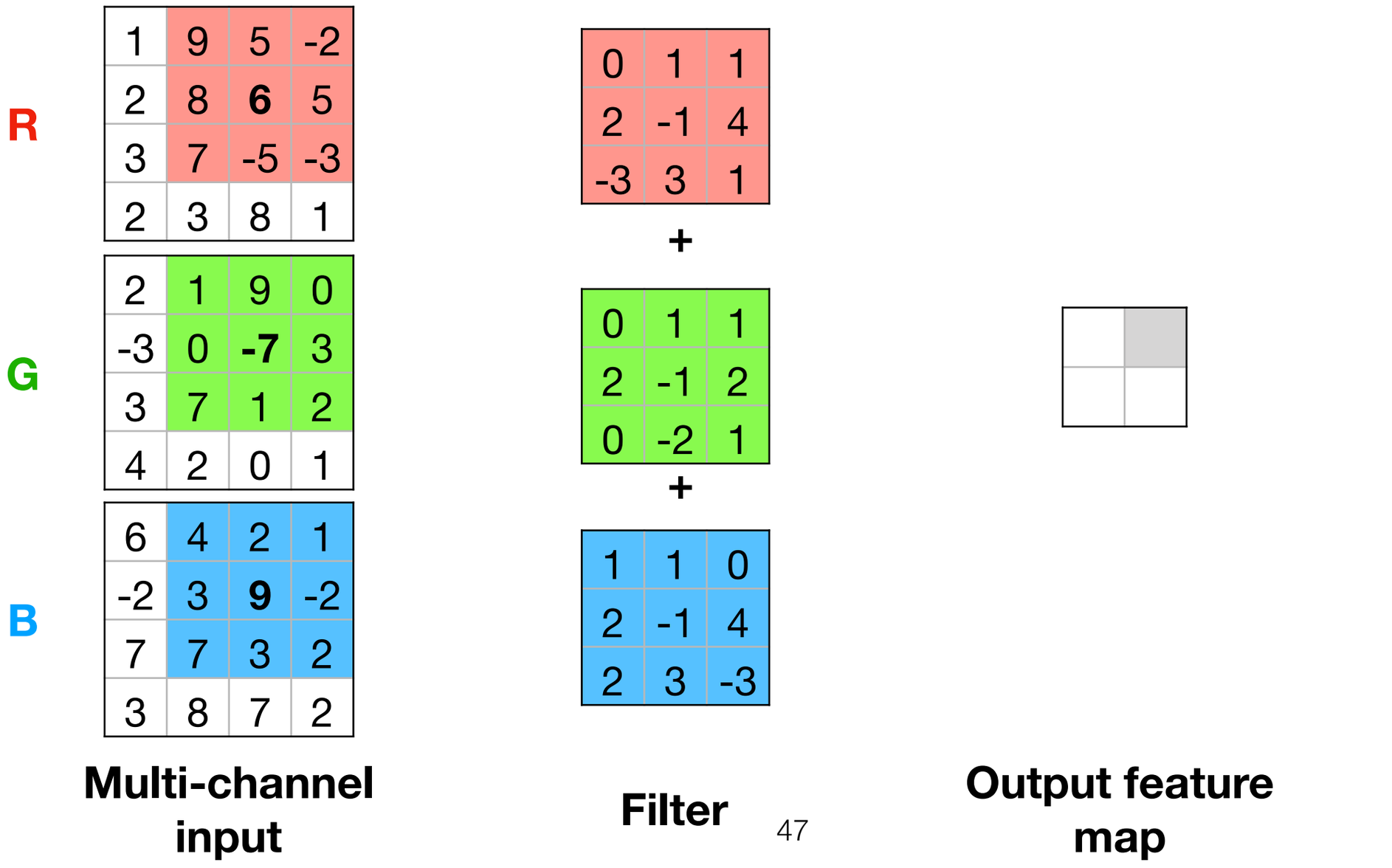
$$\begin{bmatrix} 1 & 2 & 3 & 0 & 0 \\ 0 & 1 & 2 & 3 & 0 \\ 0 & 0 & 1 & 2 & 3 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \\ -2 \\ 0 \\ 3 \end{bmatrix} = \begin{bmatrix} -1 \\ -2 \\ 7 \end{bmatrix}$$

Wx = h

W **x** **h**

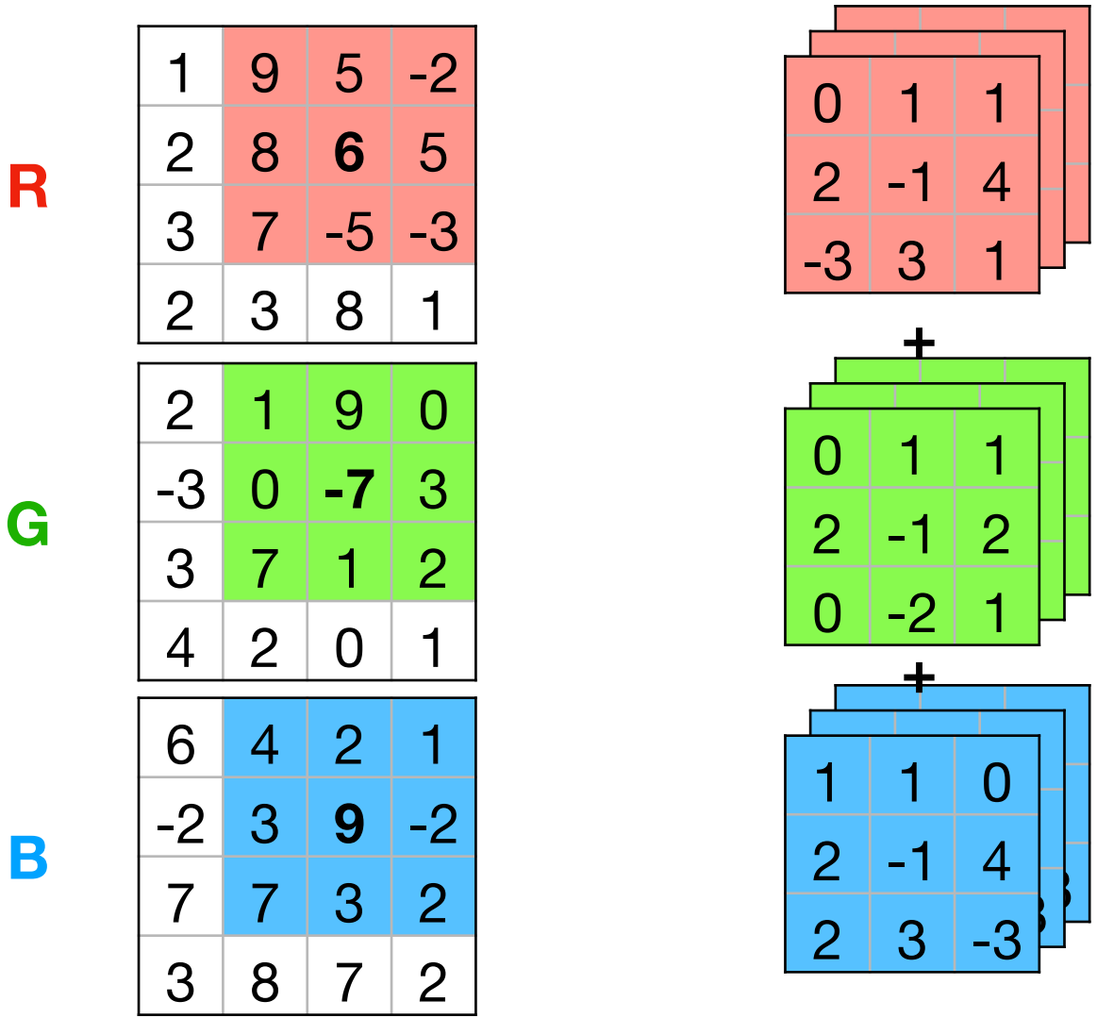
Review of 3D convolution

- Recall how 3D convolution (3x3 filter) on a stack of input images works.



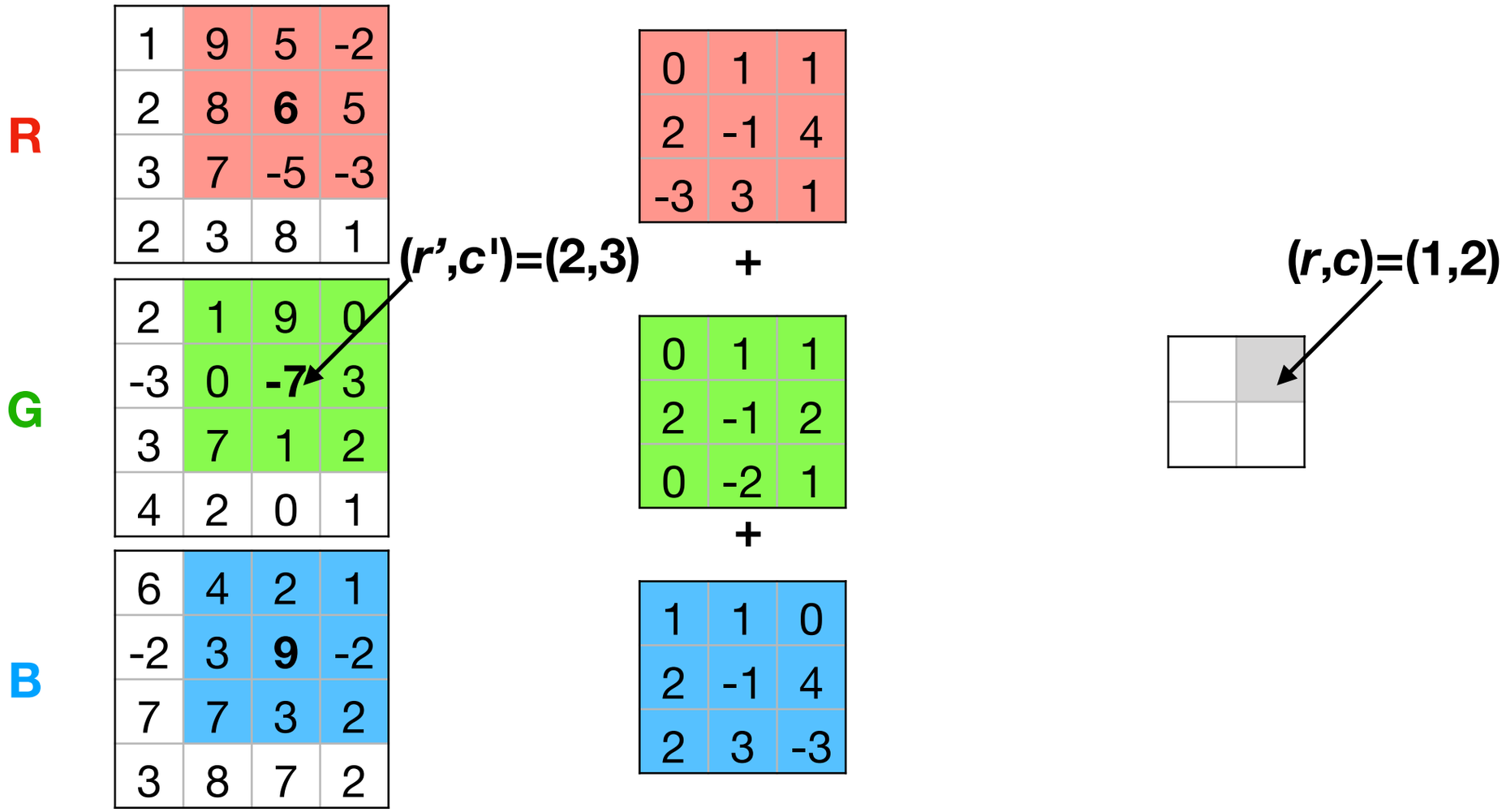
Review of 3D convolution

- We can also apply multiple such filters to obtain a stack of output feature maps.



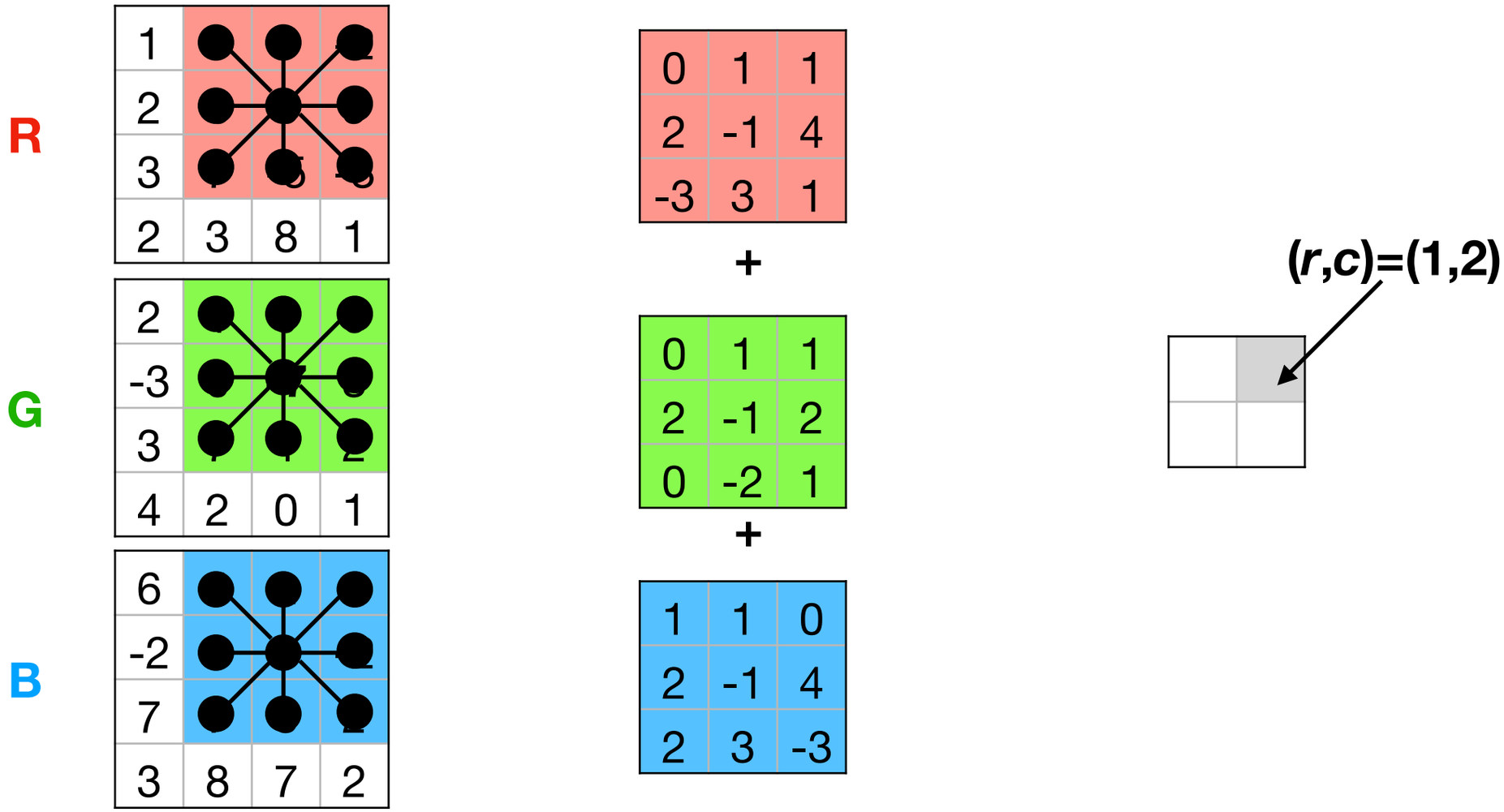
Review of 3D convolution

- The value of each element (*r*,*c*) of the output feature map depends on the set of (*k***k*) **neighbors** of the corresponding element (*r*',*c*') in the input image.



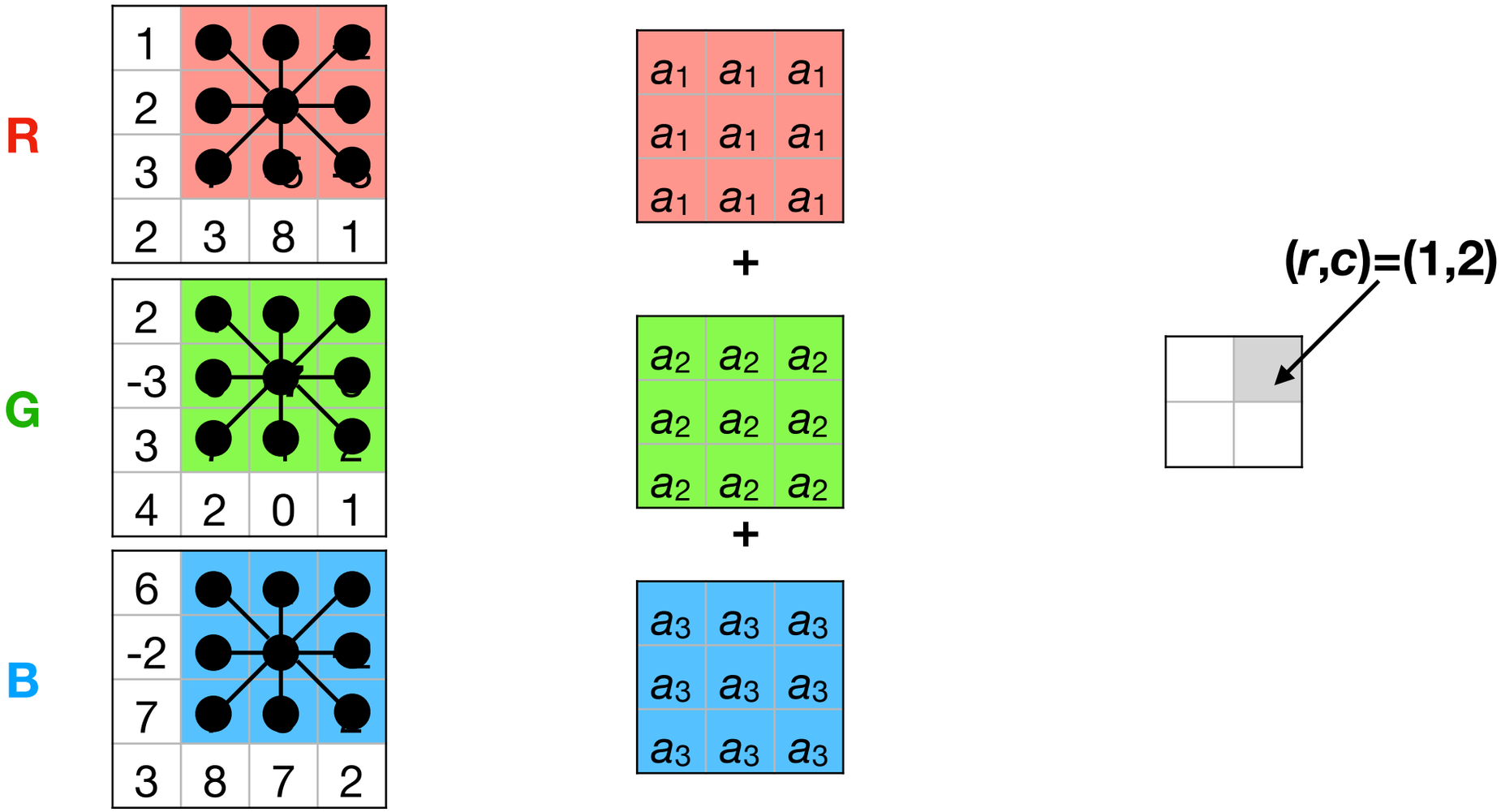
Review of 3D convolution

- The value of each element (*r*,*c*) of the output feature map depends on the set of (*k***k*) **neighbors** of the corresponding element (*r*',*c*') in the input image.



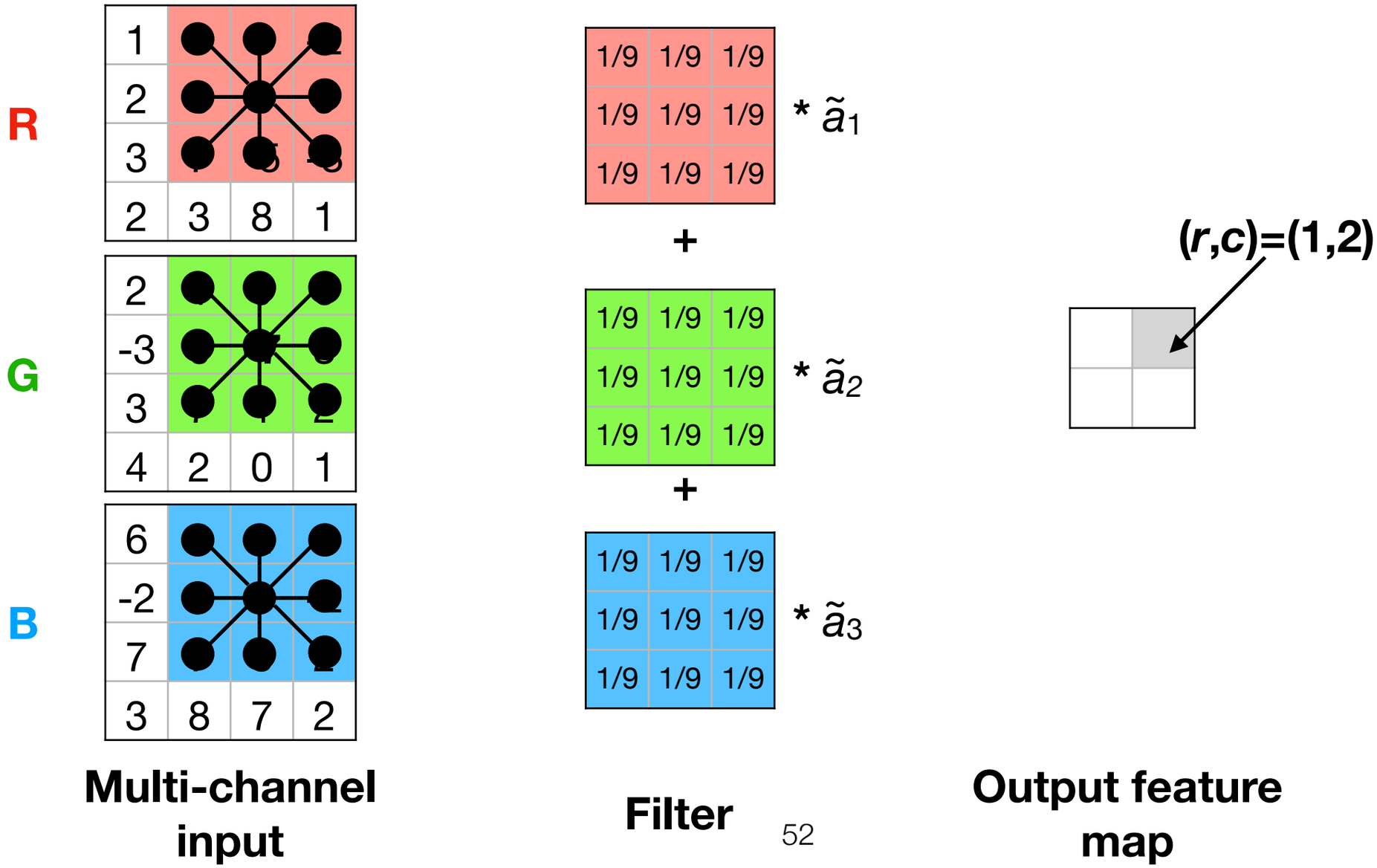
Special case

- Let's consider a special case where all the elements of each channel of the convolution filter are equal, i.e., there are just 3 learned parameters a_1, a_2, a_3.



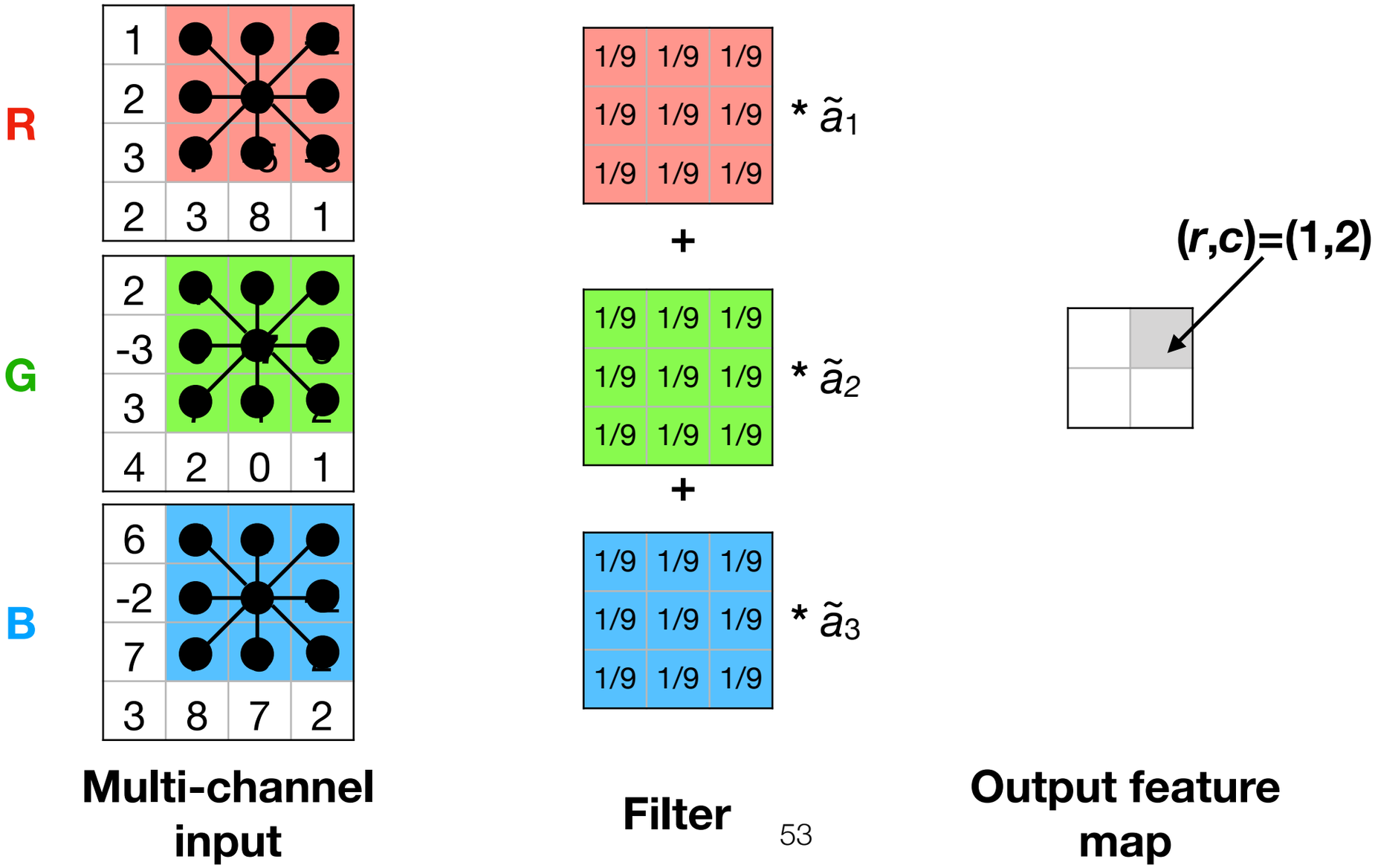
Special case

- Let's consider a special case where all the elements of each channel of the convolution filter are equal, i.e., there are just 3 learned parameters a_1, a_2, a_3.



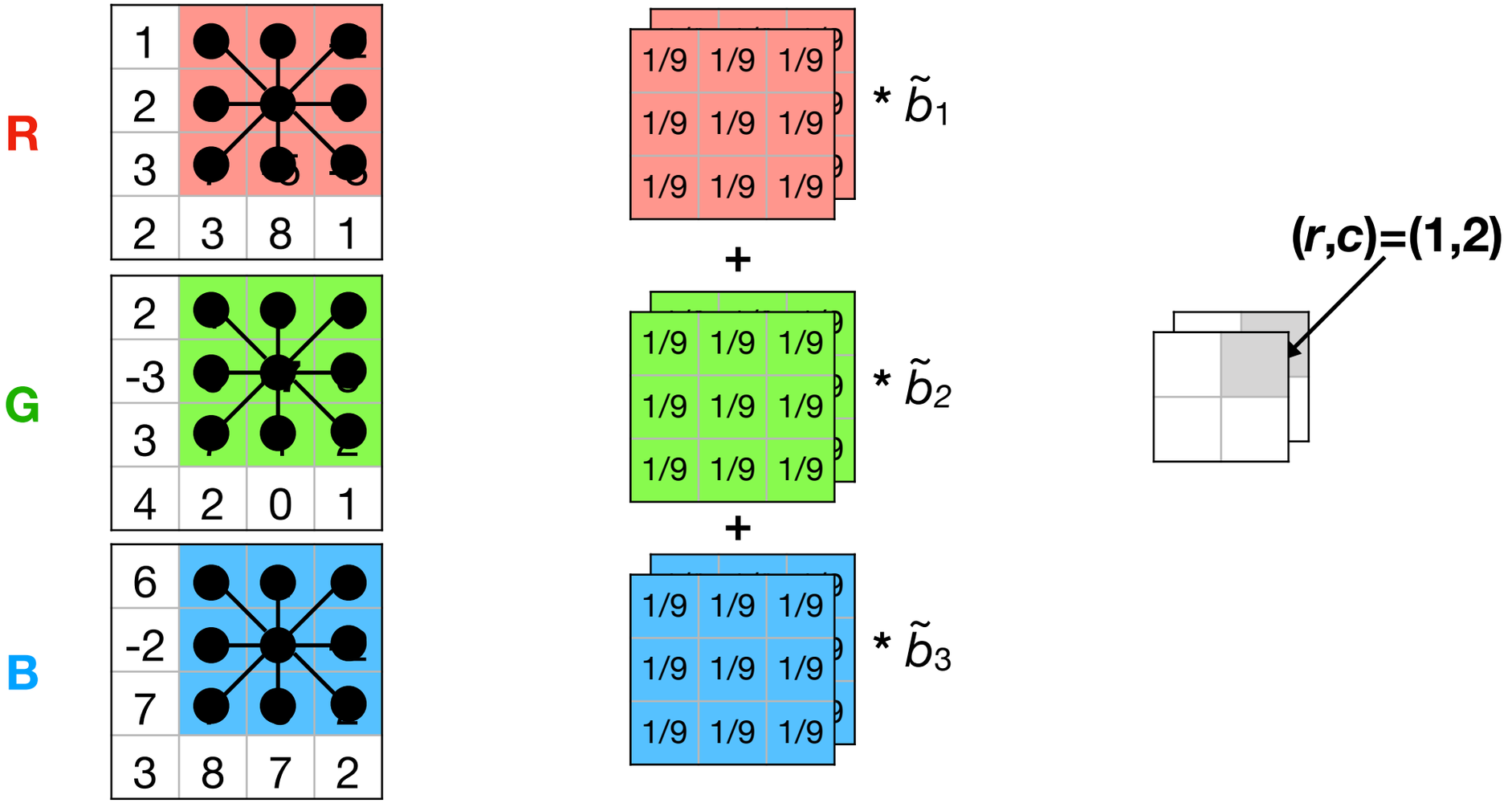
Special case

- In this case, the output feature map at (*r*,*c*) equals:
  $\tilde{a}_1$ * (Avg value of neighbors of (*r*',*c*')^R) +
  $\tilde{a}_2$ * (Avg value of neighbors of (*r*',*c*')^G) +
  $\tilde{a}_3$ * (Avg value of neighbors of (*r*',*c*')^B)



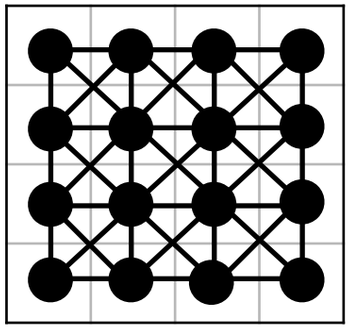
Special case

- For a second filter, the output feature map at (*r*,*c*) would equal:
  $\tilde{b}_1$ * (Avg value of neighbors of (*r*',*c*')^R) +
  $\tilde{b}_2$ * (Avg value of neighbors of (*r*',*c*')^G) +
  $\tilde{b}_3$ * (Avg value of neighbors of (*r*',*c*')^B)



Images as graphs

- Let us now represent an image as an undirected graph with:
  - One node per pixel.
  - An edge between each pair of adjacent pixels.



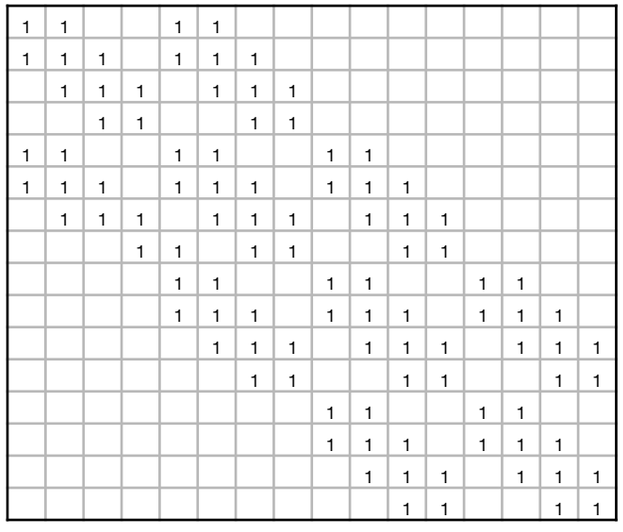
Images as graphs

- We can represent this graph using an **adjacency matrix.**

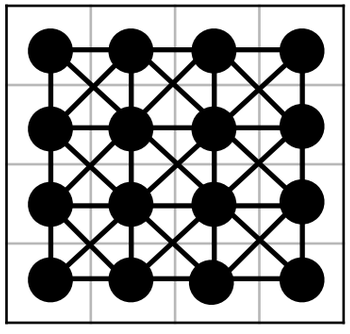
- In particular, for an image with *n* pixels (*n*=16 in this example), let $\mathbf{A} \in \{0,1\}^{n \times n}$ where A_{ij}=1 iff pixel *i* neighbors pixel *j*.

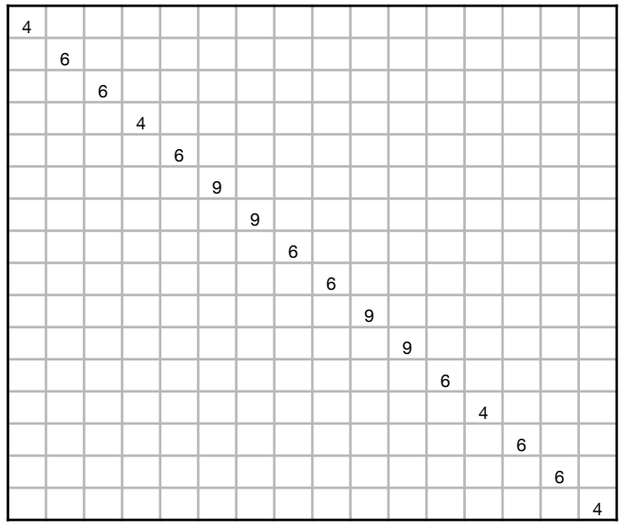
- Also define diagonal matrix **D** to count the total number of neighbors of each node *i*: $D_{ii} = \sum_j A_{ij}$

If we include self-connections (A_{ii}=1) then D_{ii} is 9 for interior pixels, 6 for edges, and 4 for corners in this example.)



A

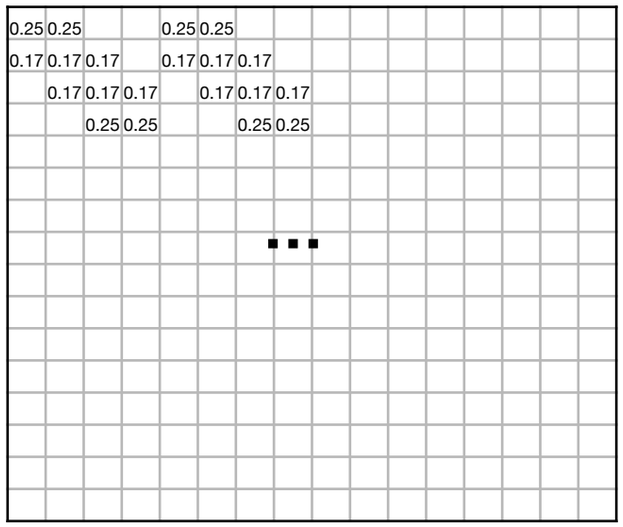




D

Images as graphs

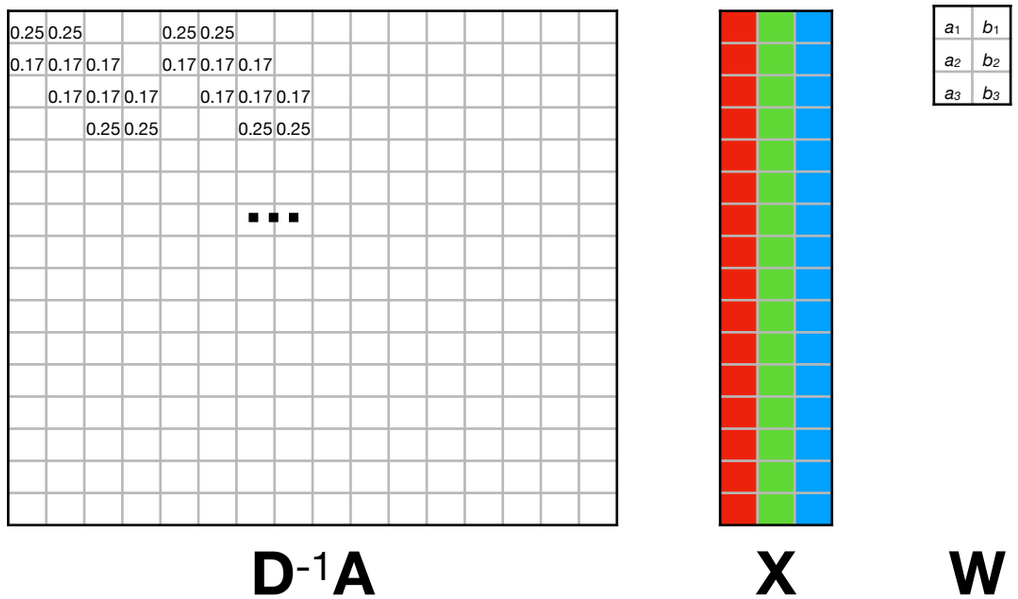
- By computing $\mathbf{D}^{-1}\mathbf{A}$, we can compute a normalized adjacency matrix:



$\mathbf{D}^{-1}\mathbf{A}$

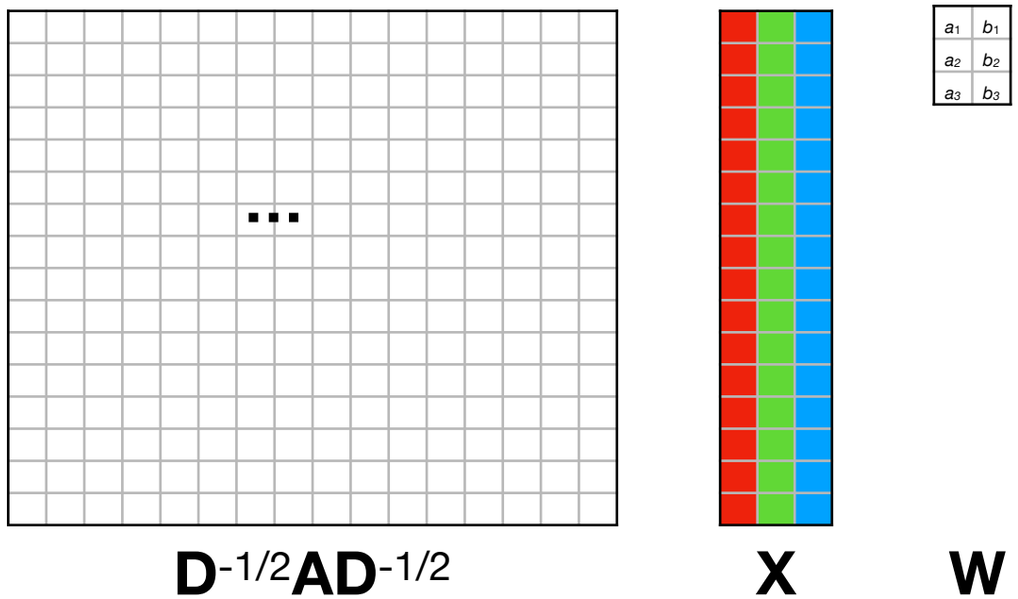
Images as graphs

- We can now express this convolution as the product of the normalized adjacency matrix $\mathbf{D}^{-1}\mathbf{A}$ with the input features $\mathbf{X}$ and the filter weights $\mathbf{W}$:



Images as graphs

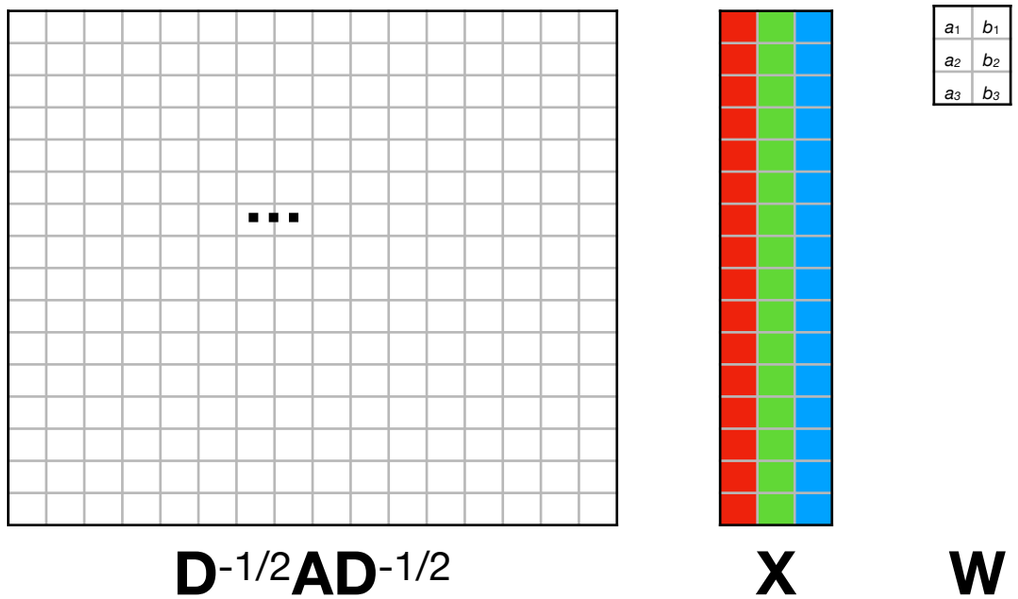
- We can now express this convolution as the product of the normalized adjacency matrix $\mathbf{D}^{-1}\mathbf{A}$ with the input features $\mathbf{X}$ and the filter weights $\mathbf{W}$:



- A related matrix is given by $\mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$.

Images as graphs

- We can now express this convolution as the product of the normalized adjacency matrix $\mathbf{D}^{-1}\mathbf{A}$ with the input features $\mathbf{X}$ and the filter weights $\mathbf{W}$:



$$\mathbf{h} = \sigma\left(\mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}}\mathbf{X}\mathbf{W}\right)$$

- A related matrix is given by $\mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$.

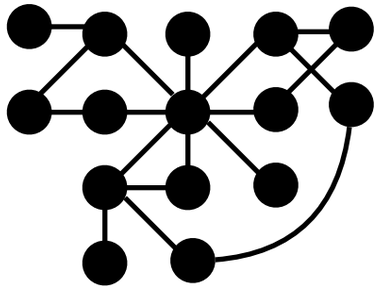
- We can then apply a non-linear activation function σ.

Images as graphs

- We can apply the same procedure to general graphs.

- Moreover, the number of neighbors of each node does not have to be the same.

$$\mathbf{h} = \sigma\left(\mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}}\mathbf{X}\mathbf{W}\right)$$



- Intuitively, we convolve the features of each graph node with a fixed filter and renormalize according to the node's neighbors.

- This is the basic operation of a **graph convolutional neural network** (GCN) (as proposed by Kipf & Welling 2017).

Exercises

https://cs230.stanford.edu/files/cs230exam_win19.pdf

Exercise 1

(1 point) After training a neural network, you observe a large gap between the training accuracy (100%) and the test accuracy (42%). Which of the following methods is commonly used to reduce this gap?

(i) Generative Adversarial Networks

(ii) Dropout

(iii) Sigmoid activation

(iv) RMSprop optimizer

Exercise 2

(2 points) Which of the following propositions are true about a CONV layer? **(Check all that apply.)**

(i) The number of weights depends on the depth of the input volume.

(ii) The number of biases is equal to the number of filters.

(iii) The total number of parameters depends on the stride.

(iv) The total number of parameters depends on the padding.

Exercise 3

(2 points) How does splitting a dataset into train, dev and test sets help identify overfitting?

Exercise 4

(3 points) You want to solve a classification task. You first train your network on 20 samples. Training converges, but the training loss is very high. You then decide to train this network on 10,000 examples. Is your approach to fixing the problem correct? If yes, explain the most likely results of training with 10,000 examples. If not, give a solution to this problem.

Exercise 5

(2 points) You are given the following piece of code for forward propagation through a single hidden layer in a neural network. This layer uses the sigmoid activation. Identify and correct the error.

```
import numpy as np

def forward_prop(W, a_prev, b):
    z = W*a_prev + b
    a = 1/(1+np.exp(-z)) #sigmoid
    return a
```

Exercise 6

- Why does it make no sense to use BatchNorm when the mini-batch size is 1?

Exercise 7

(2 points) You have an input volume of $32 \times 32 \times 3$. What are the dimensions of the resulting volume after convolving a 5×5 kernel with zero padding, stride of 1, and 2 filters? (1 formula)

Exercise 8

(2 point) You have 500 examples in total, of which only 175 were examples of preterm births (positive examples, label = 1). To compensate for this class imbalance, you decide to duplicate all of the positive examples, and then split the data into train, validation and test sets. Explain what is a problem with this approach. (1-2 sentences)

Exercise 9

(1 point) You fix the issue. Subject matter experts tell you that the model should absolutely not miss preterm births, but false positives are okay. Your best model achieves 100% recall. Does it mean the model works well? Explain. (1 sentence)