

Large Language Models: An Introduction for Life Scientists

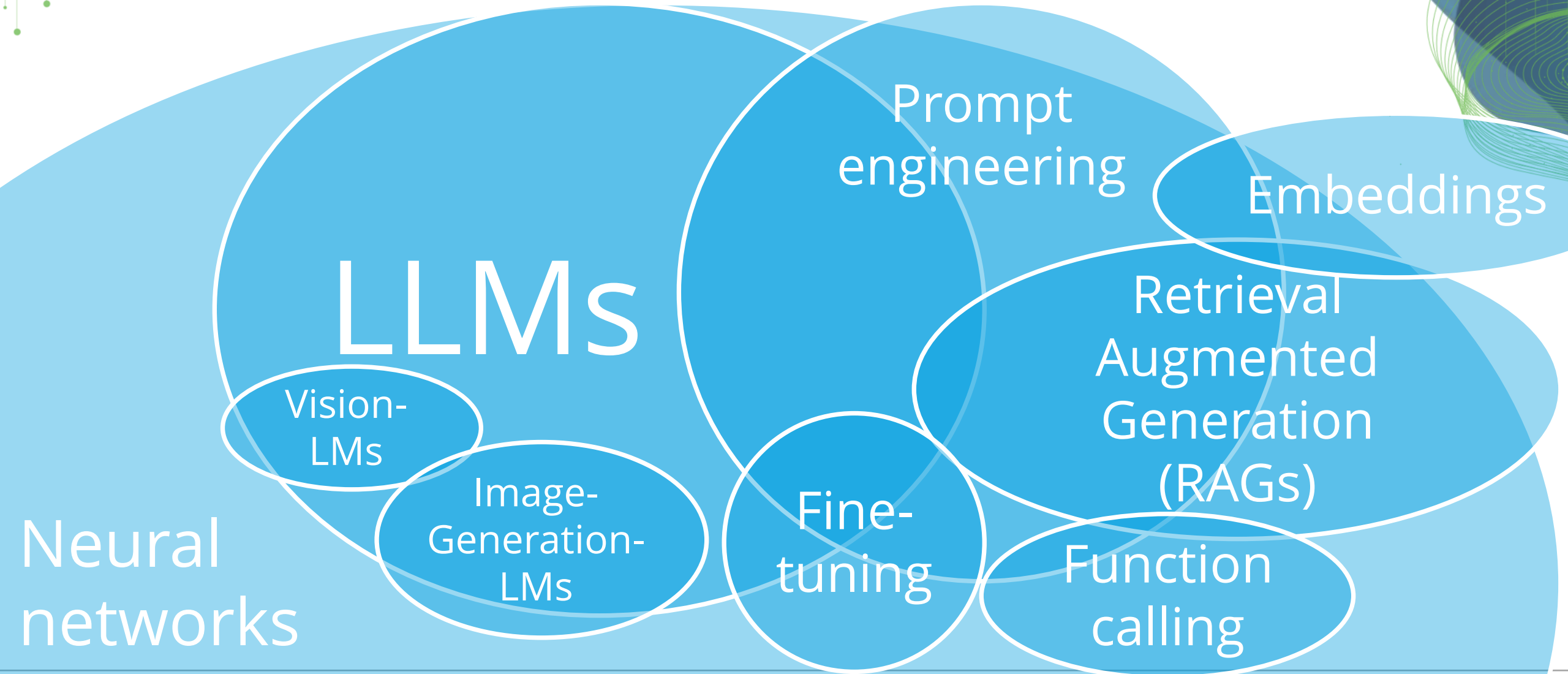
Robert Haase



These slides can be reused under the terms of the [CC-BY4.0](https://creativecommons.org/licenses/by/4.0/) license.

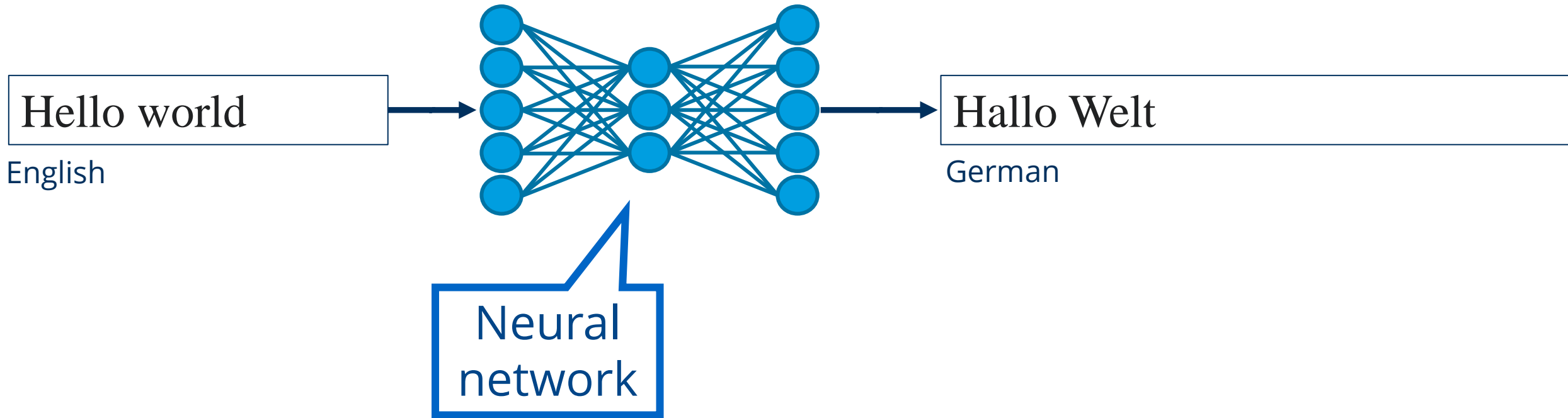
<https://doi.org/10.5281/zenodo.13374649>

Overview



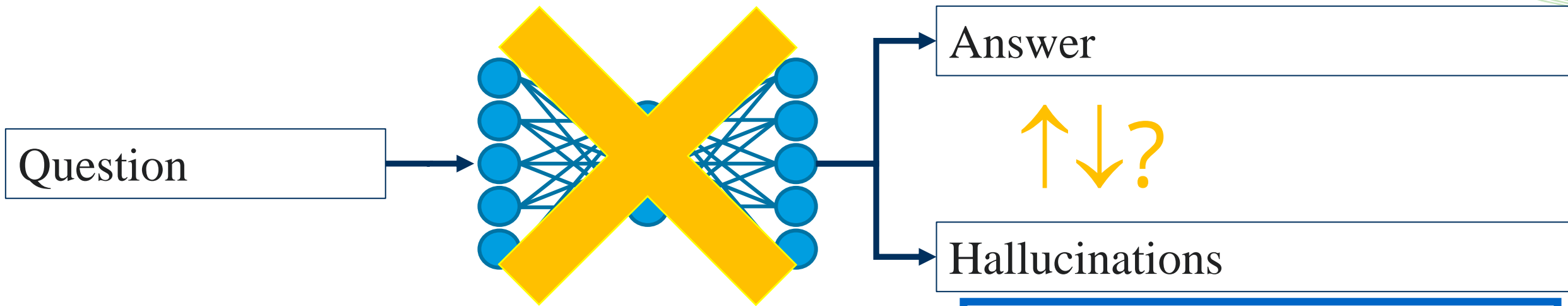
Large Language Models (LLMs)

Text-to-text, translation, ...



Large Language Models (LLMs)

Text-to-text, translation, knowledge retrieval

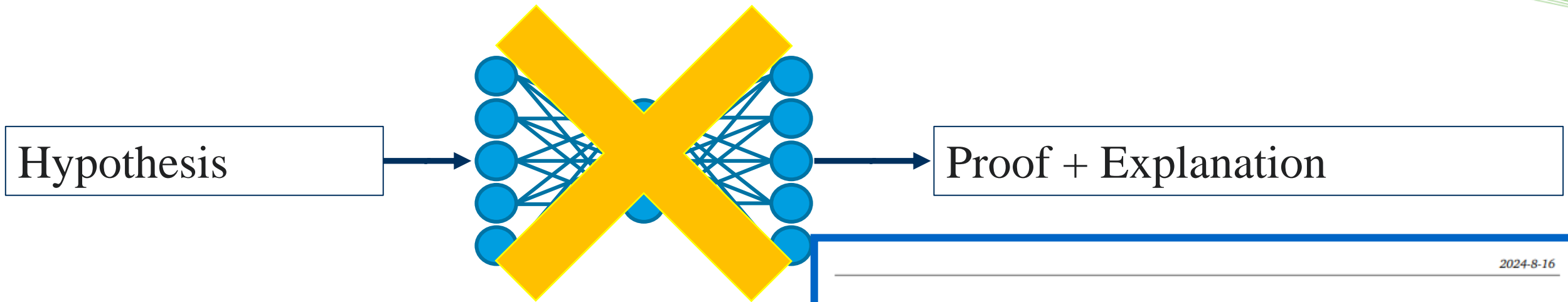


Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks

Patrick Lewis^{†‡}, Ethan Perez^{*},
Aleksandra Piktus[†], Fabio Petroni[†], Vladimir Karpukhin[†], Naman Goyal[†], Heinrich Küttler[†],
Mike Lewis[†], Wen-tau Yih[†], Tim Rocktäschel^{†‡}, Sebastian Riedel^{†‡}, Douwe Kiela[†]

Large Language Models (LLMs)

Text-to-text, translation, reasoning



2024-8-16

The AI Scientist: Towards Fully Automated Open-Ended Scientific Discovery

Chris Lu^{1,2,*}, Cong Lu^{3,4,*}, Robert Tjarko Lange^{1,*}, Jakob Foerster^{2,†}, Jeff Clune^{3,4,5,†} and David Ha^{1,†}

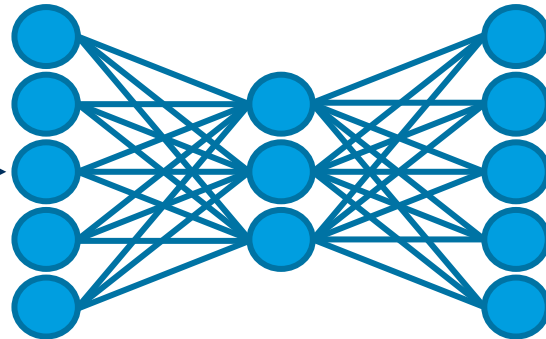
^{*}Equal Contribution, ¹Sakana AI, ²FLAIR, University of Oxford, ³University of British Columbia, ⁴Vector Institute, ⁵Canada CIFAR AI Chair, [†]Equal Advising

Large Language Models (LLMs)

Text-to-text, translation, code generation

Open hela-cells.tif

English



```
from skimage.io import imread  
image = imread("hela-cells.tif")
```

Python

Published as a conference paper at ICLR 2024

SWE-BENCH: CAN LANGUAGE MODELS RESOLVE REAL-WORLD GITHUB ISSUES?

Carlos E. Jimenez^{* 1,2} John Yang^{* 1,2} Alexander Wettig^{1,2}
Shunyu Yao^{1,2} Kexin Pei³ Ofir Press^{1,2} Karthik Narasimhan^{1,2}

¹Princeton University ²Princeton Language and Intelligence ³University of Chicago

Why code generation using LLMs?

Output of LLMs is often criticized for

- Limited reproducibility
- Hard to evaluate (automatically)

Output of LLM-generated code is

- Reproducible
- Can be executed and evaluated (automatically)

Say a

Say a

Write Python code that says "a".

A.

< 2/2 >

A!

Certainly! Here's a simple Python code snippet that prints the letter "a":

```
python  
  
print("a")
```

Copy code

No matter how often you execute this code, the output is always guaranteed to be „a“.

Image Generation LLMs

Text-to-image prompting

Interesting challenges for our community ahead

a histology image of lung cancer cells and some healthy tissue

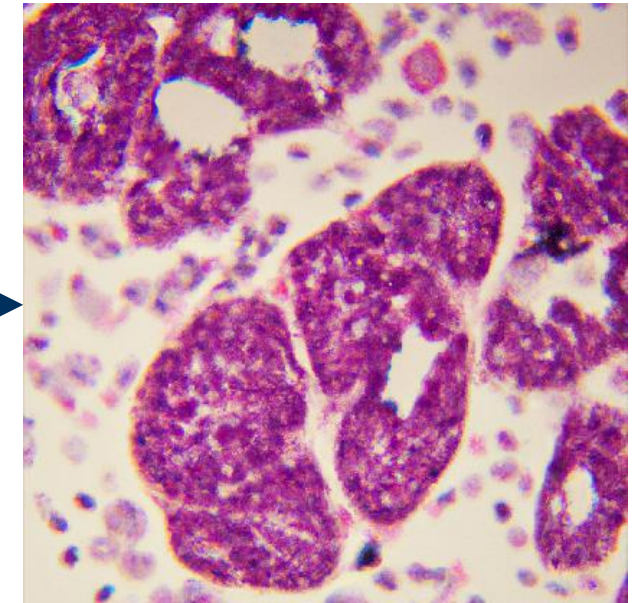
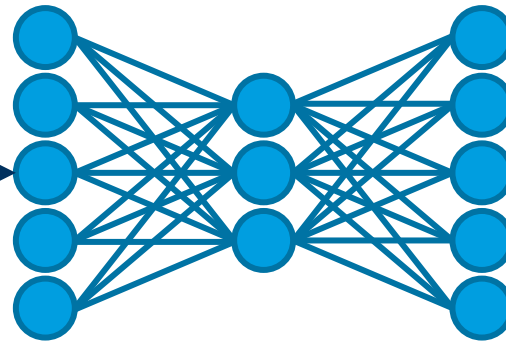
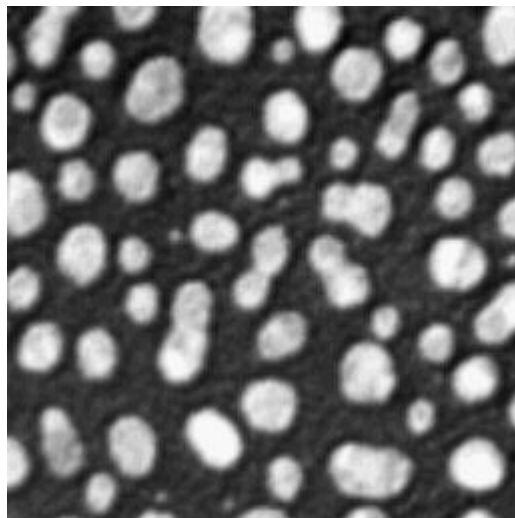
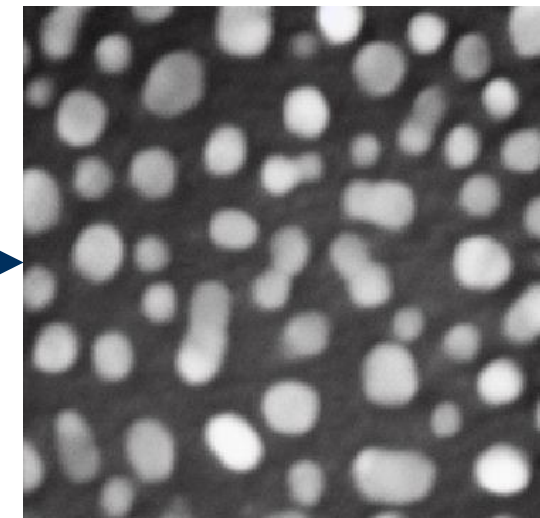
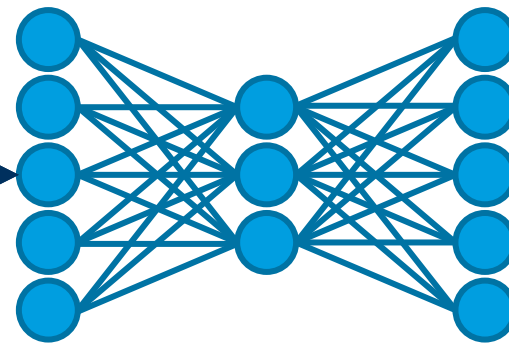


Image Generation LLMs

Image-to-image prompting, image variations



Blur the image

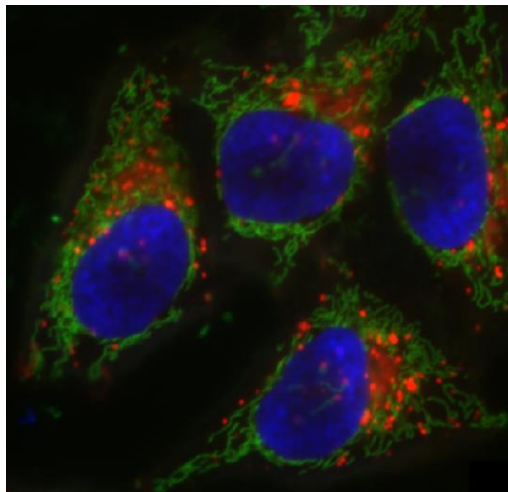


InstructPix2Pix: Learning to Follow Image Editing Instructions

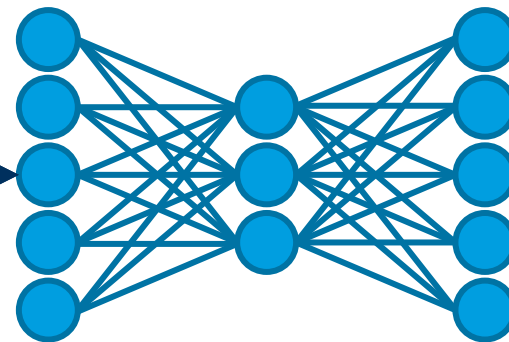
Tim Brooks* Aleksander Holynski* Alexei A. Efros
University of California, Berkeley

Vision Language Models VLMs

Combining image and text: Multi-Modal LMs

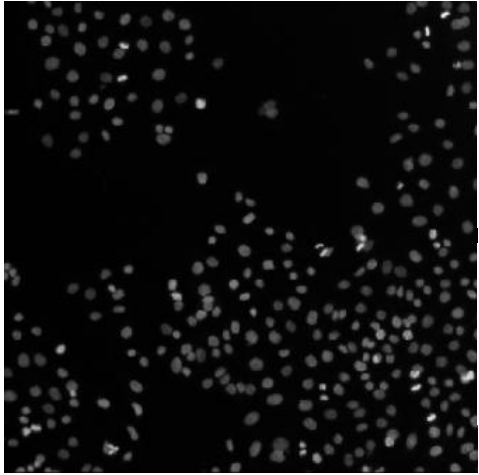


What's in the blue-channel of this image?



There are nuclei, presumably stained with DAPI and imaged using a fluorescence microscope.

Use case: Code generation



```
[4]: %%bob image
      What's in this microscopy image?
      Answer in one short sentence.
```

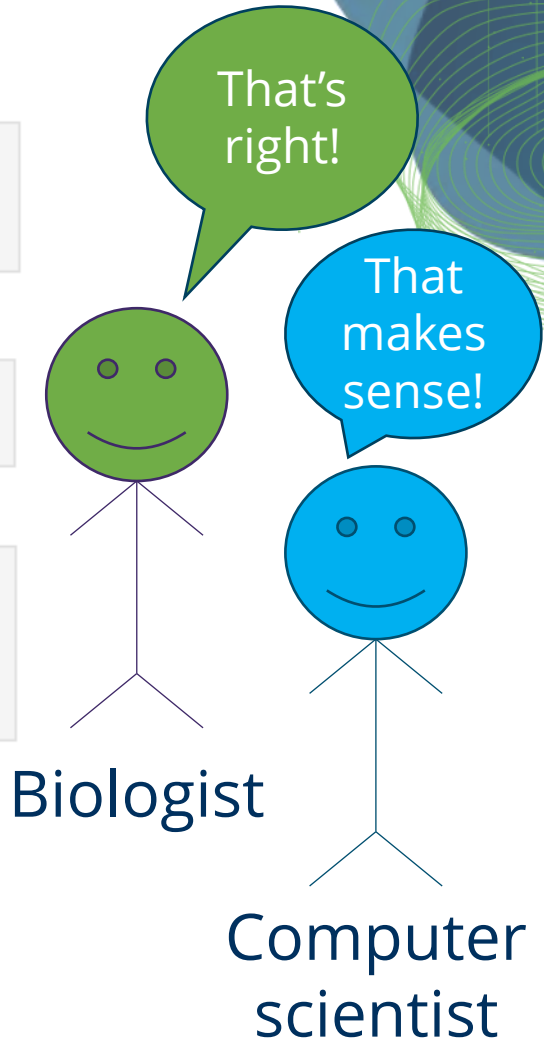
The microscopy image shows numerous cell nuclei scattered across a dark background.

```
[5]: %%bob
      What deep-learning-based Python library could be used to segment the nuclei?
```

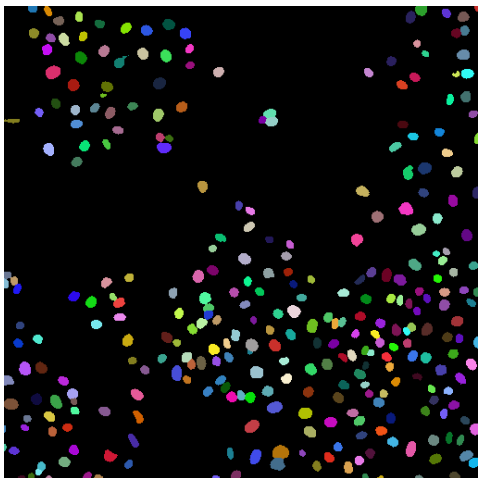
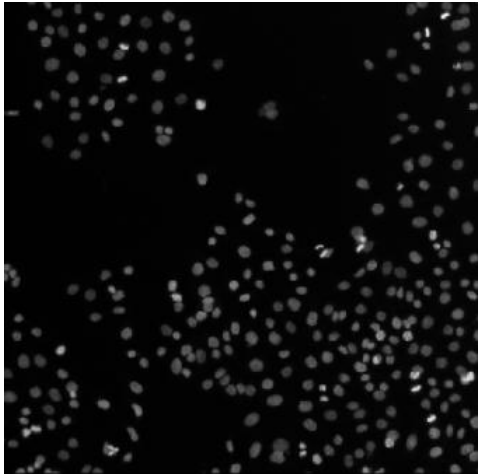
A deep-learning-based Python library that could be used to segment the nuclei is Cellpose.

```
[6]: %%bob
      Please provide Python code for applying this algorithm to my image.
      Make sure the result is a label image of integer type.
      Also visualize the result using an animated curtain.
```

You need to have image analysis knowledge to write prompts like this



Use case: Code generation



```
[7]: from cellpose import models
import numpy as np

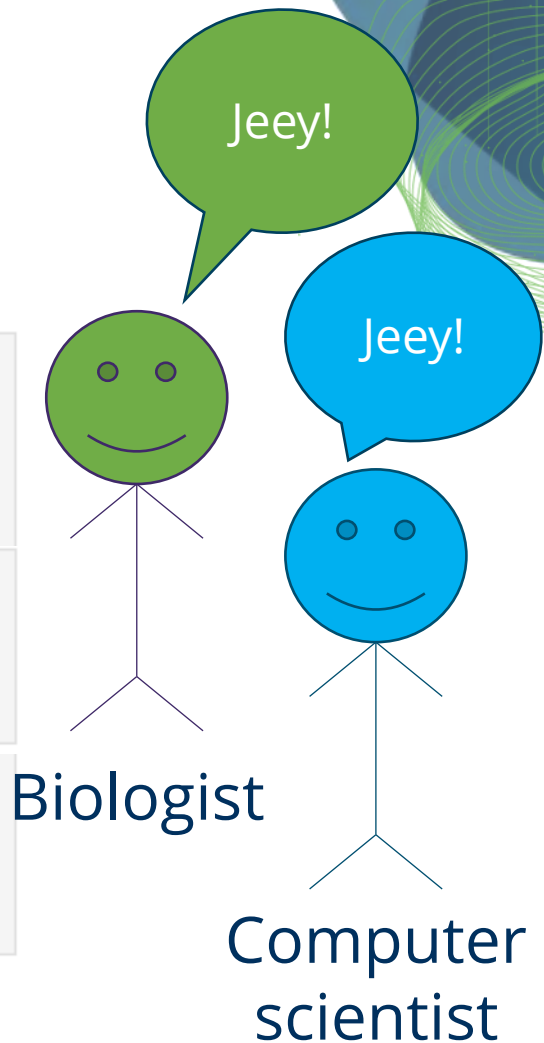
# Load the pre-trained model
model = models.Cellpose(model_type='nuclei')

[6]: %%bob
Please provide Python code for applying this algorithm to my image.
Make sure the result is a label image of integer type.
Also visualize the result using an animated curtain.

# Ensure the result is a label image of integer type
label_image = masks.astype(np.int32)

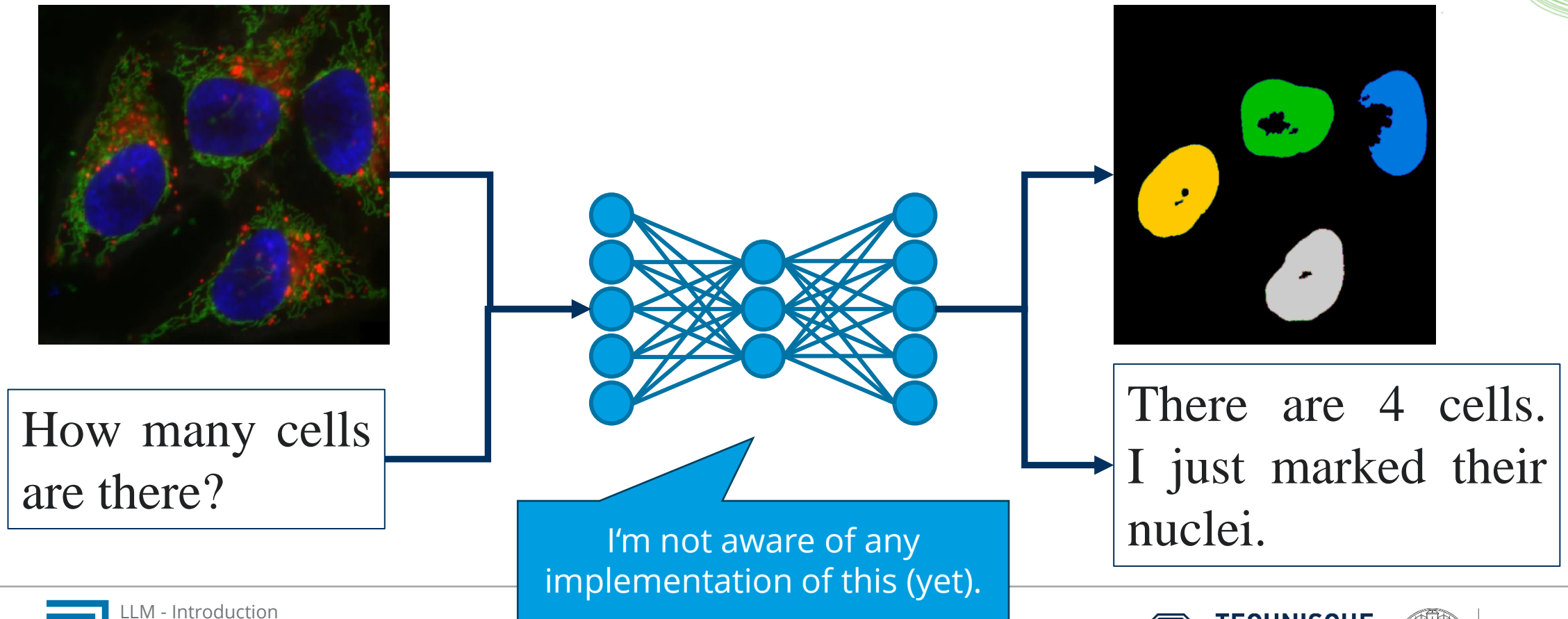
# Visualize using an animated curtain
stackview.animate_curtain(image, label_image)
```

You need programming skills to proof-read this code.



Multi-modal LLMs

Combining image, text and [...] data, to gain new insights.

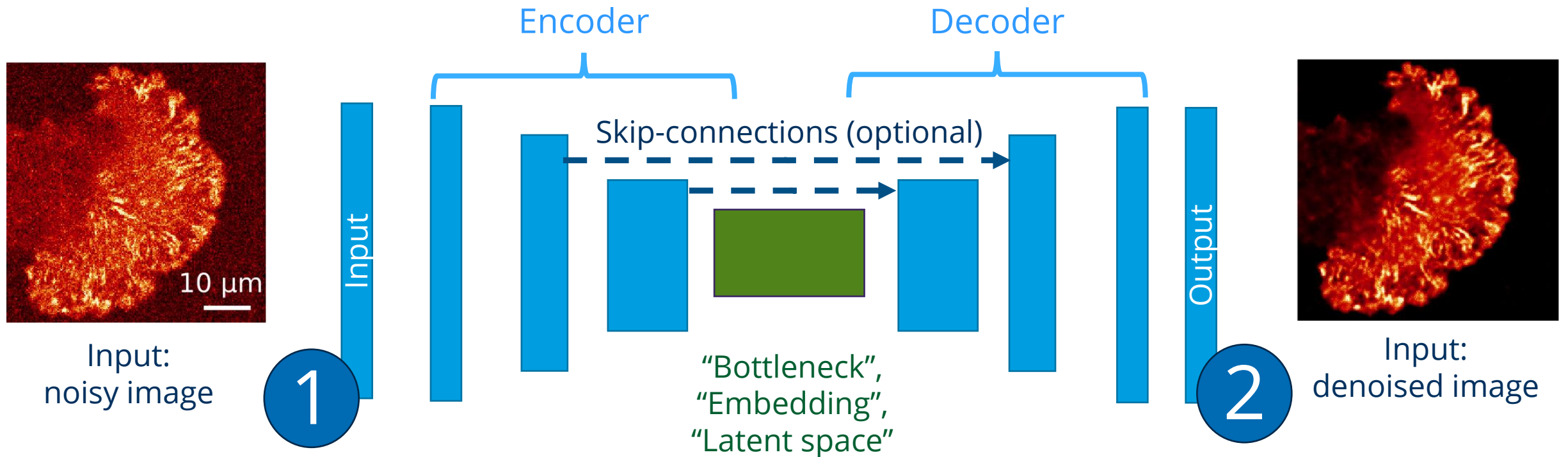


Transformer neural networks



Traditional architecture: Encoder-Decoder Networks

Related: „Auto-encoder“, „Variational Auto-Encoder“, „U-Net“



What's new? Transformers

LLMs use the **transformer** neural network architecture

Related: Generative Pretrained Transformer (GPT)

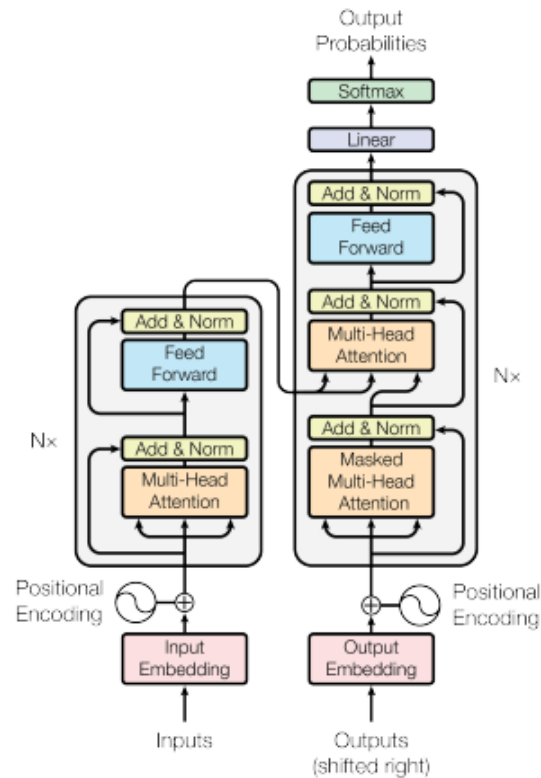
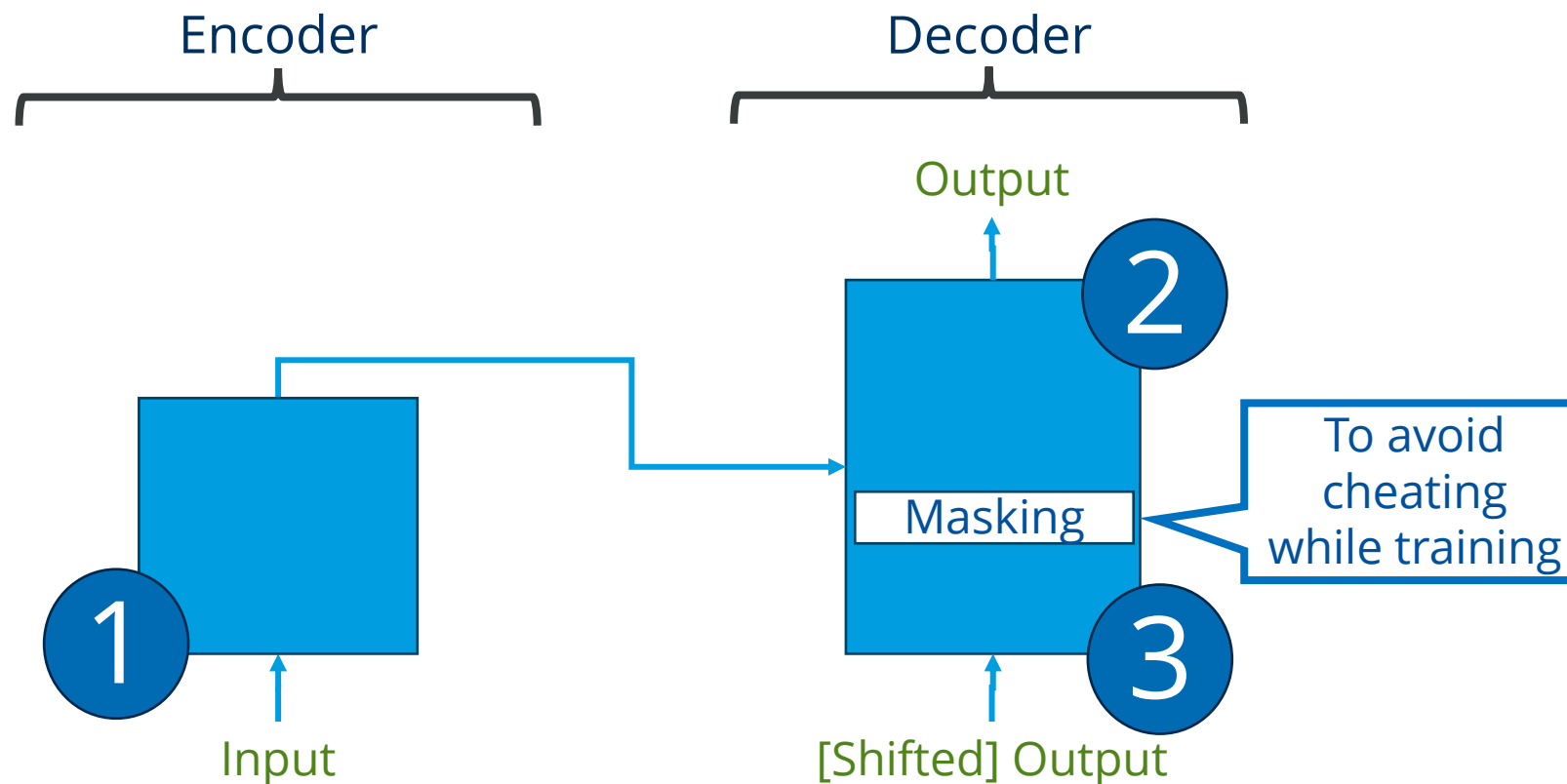


Figure 1: The Transformer - model architecture.

What's new? Transformers

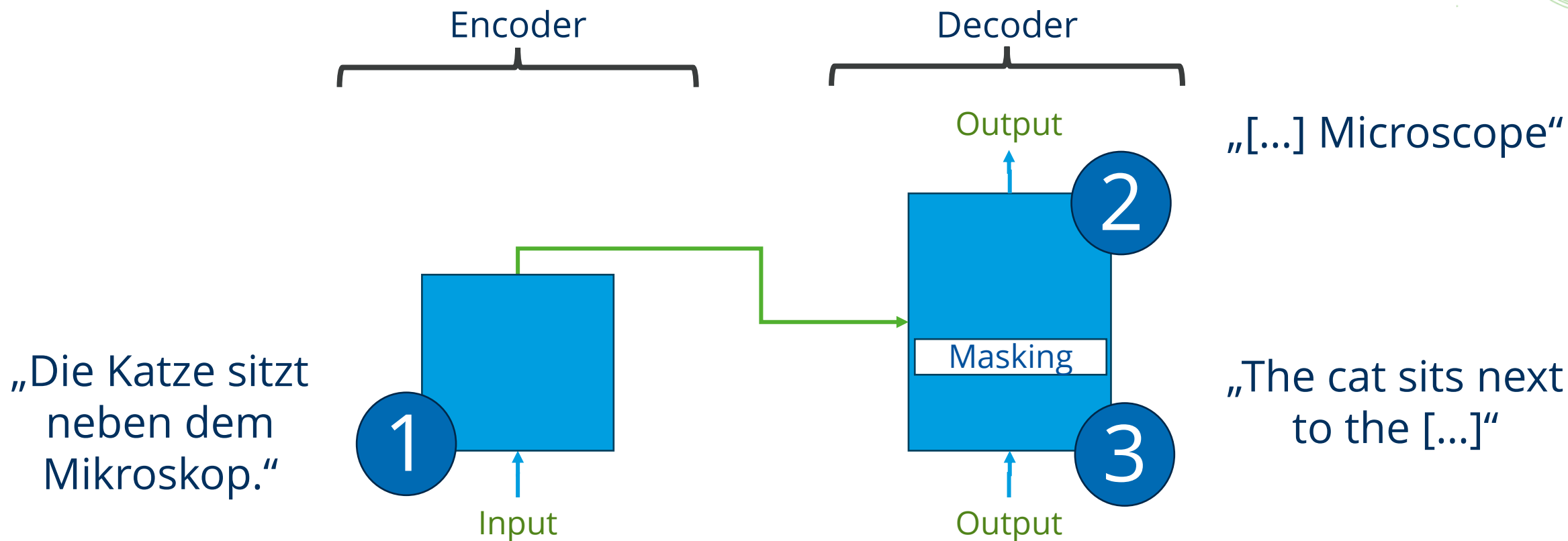
LLMs use the **transformer** neural network architecture

Related: Generative Pretrained Transformer (GPT)



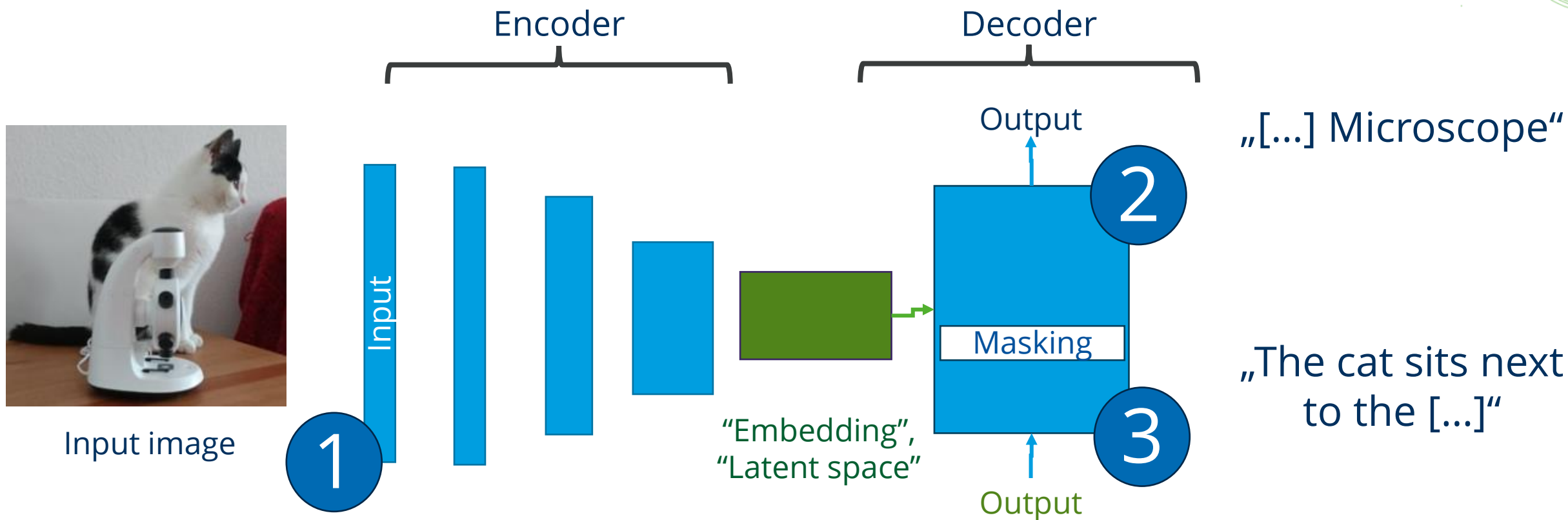
What's new? Transformers

LLMs were originally developed for translation tasks and next-word prediction.







What's new? Transformers

VLMs use combinations of traditional neural network architectures and transformers.



Use case: Descriptive biology

Table 1. Dataset: phenotypes analyzed and example images.

Phenotype	# Imgs	Example
Loss of wing veins (V-)	10	
Ectopic wing veins (V+)	20	
Integrity of wing margin (WM)	23	
Wing surface adhesion (WA)	27	

“... while visual language models are in their infancy, they already show potential for multiple applications in automated phenotyping studies. We encourage the community to carefully test them...”

Prompt

Does this wing look normal or abnormal?



Clearly abnormal because...

Prompt with reference image

This is an example of a normal drosophila wing, then I will give you examples of other wings to classify as normal/abnormal.



Ok!



Clearly abnormal because...



Clearly normal because...

[...]

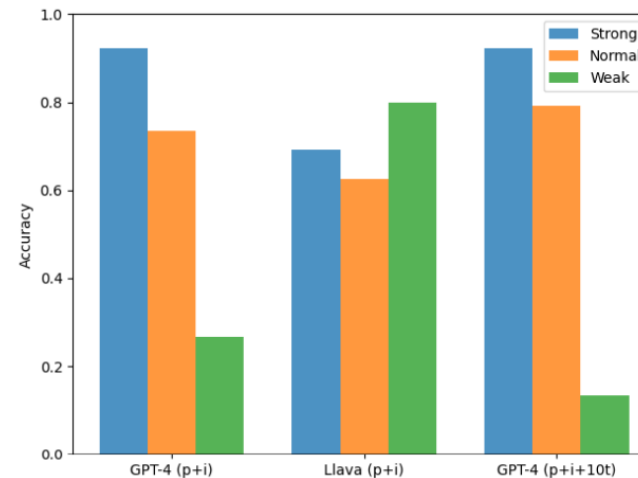
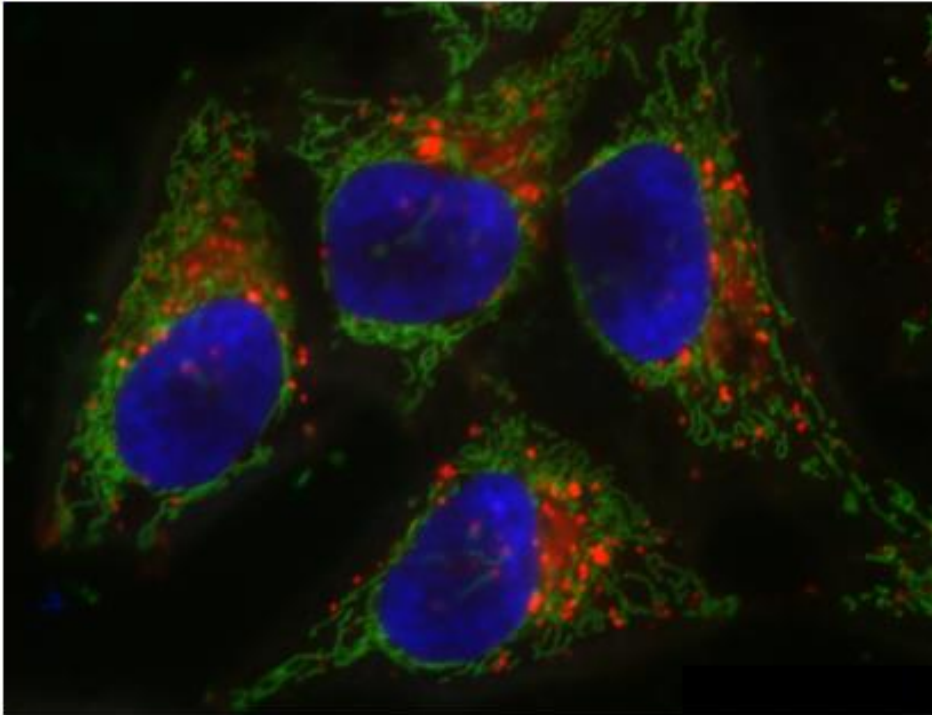


Fig. 3. Accuracy of defect classification by phenotype strength.

Vision language models for counting objects

LLMs were not exactly built for this use-case...

You



How many blue nuclei are in this image?

ChatGPT

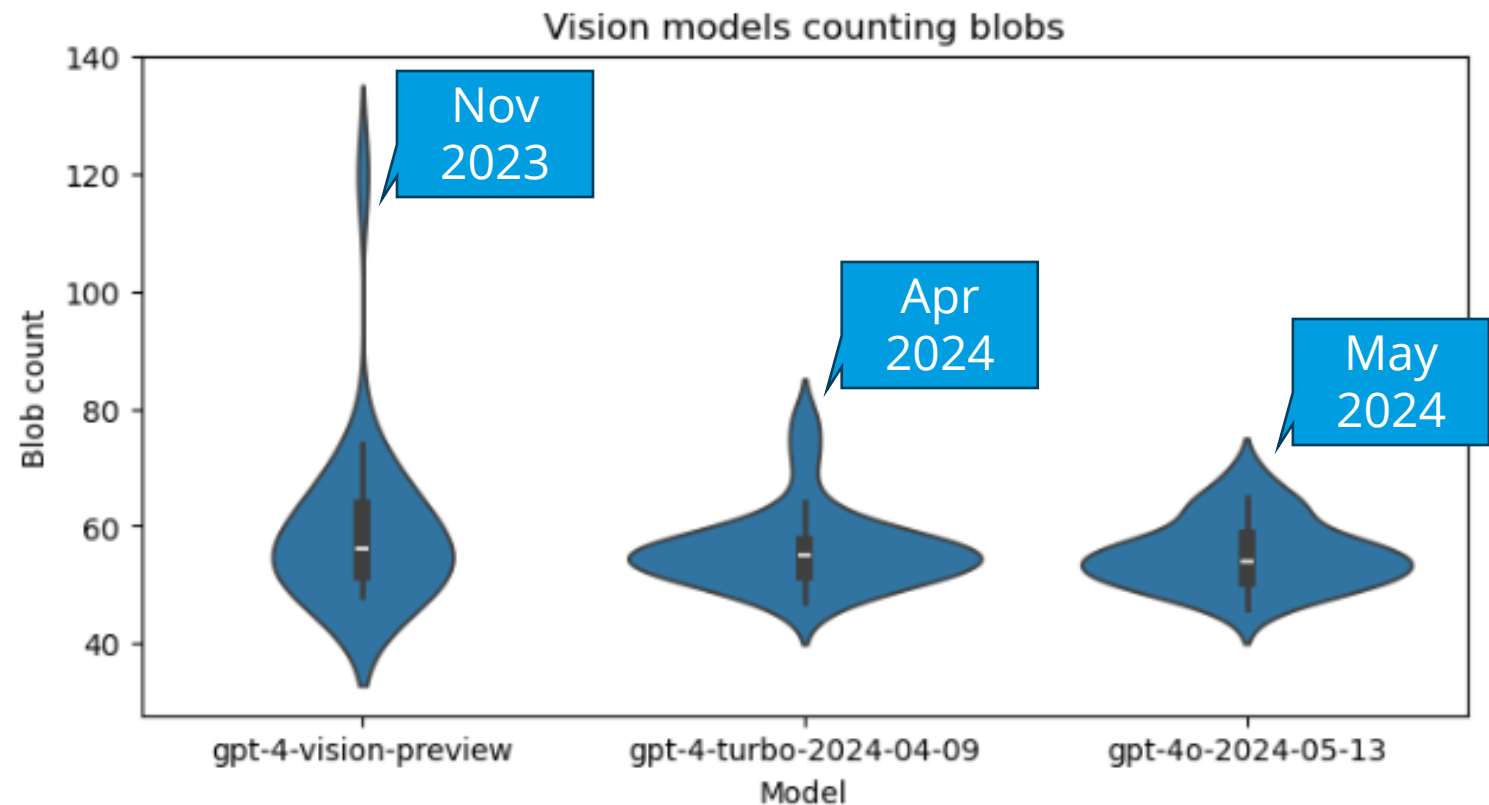
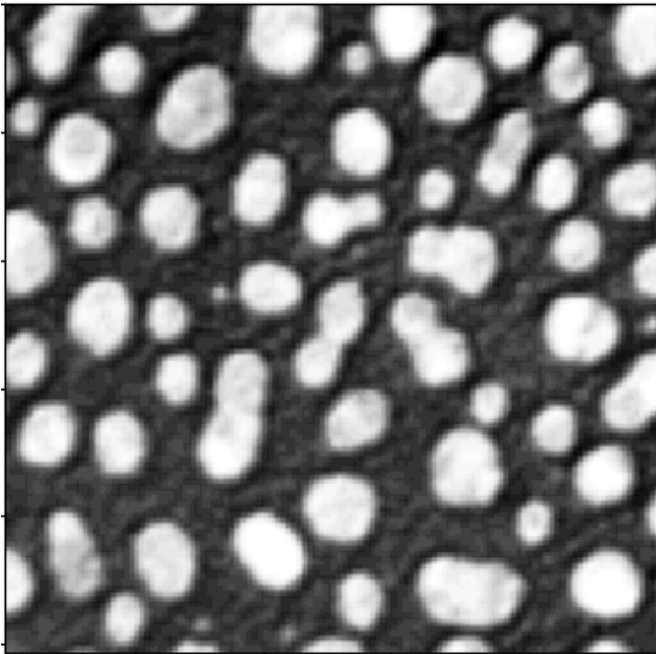
There are three blue nuclei visible in this image.



$n = 1$

Vision language models for counting objects

Prompt: „Analyse the following image by counting the bright blobs. Respond with the number only.“ (n=25)



Benchmarking



Benchmarking LLMs for Bio-image Analysis

Code Generation

Example test-case inspired by HumaEval (Chen et al 2021)

```
[1]: def workflow_segmentation_measurement_summary(image):  
    """  
    This function implements a workflow consisting of these steps:  
    * threshold intensity input image using Otsu's method  
    * label connected components  
    * measure area of the labeled objects  
    * determine mean area of all objects  
    """  
    import skimage  
    import numpy as np  
    binary_image = image > skimage.filters.threshold_otsu(image)  
    label_image = skimage.measure.label(binary_image)  
    stats = skimage.measure.regionprops(label_image)  
    areas = [s.area for s in stats]  
    return np.mean(areas)
```

```
[2]: def check(candidate):  
    import numpy as np  
  
    assert candidate(np.asarray([  
        [0,0,0,0,0],  
        [1,1,1,0,0],  
        [1,1,1,0,0],  
        [1,1,0,0,0],  
        [0,0,0,0,0],  
    ])) == 8
```

Prompt

Reference solution

Unit test (excerpt)

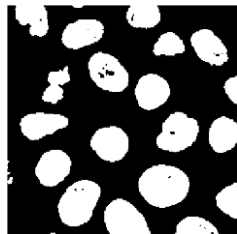
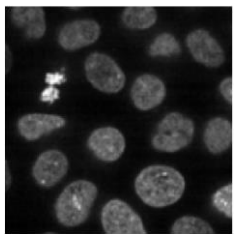
Open for your (!) contributions

14 more currently on their way...

We formulated 57 of such test-cases (yet)

Benchmarking LLMs for Bio-image Analysis

Use case: segment the image and measure the average area of objects.



	area
0	955.0
1	31.0
2	815.0
3	1166.0
4	1135.0
⋮	



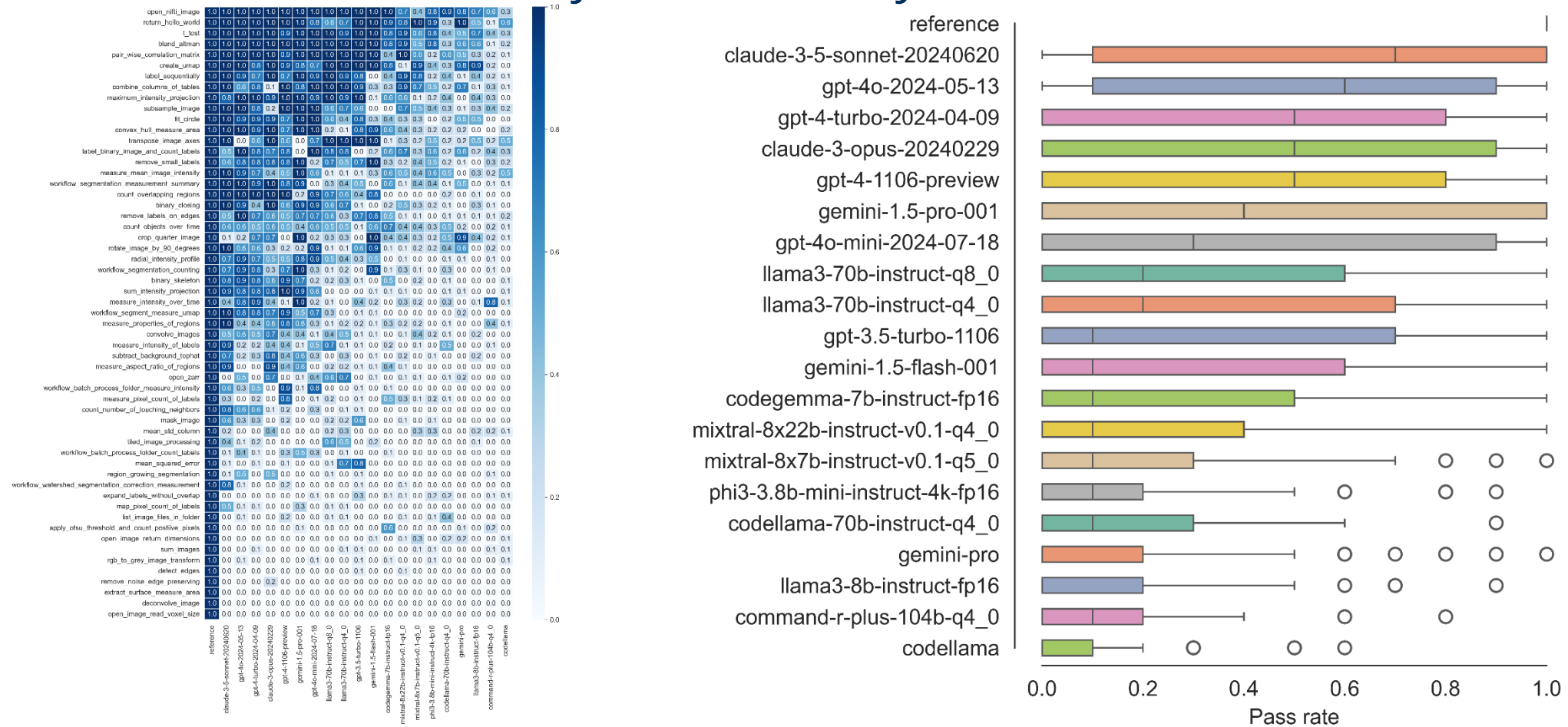
858.04

Unit-test pass-rate (n=10):

workflow_segmentation_measurement_summary	1.0	1.0	1.0	0.9	1.0	0.8	0.9	0.0	0.3	0.4	0.5	0.0	0.6	0.1	0.4	0.4	0.1	0.5	0.0	0.1	0.1	
reference																						
claude-3-5-sonnet-20240620																						
gpt-4o-2024-05-13																						
gpt-4-turbo-2024-04-09																						
claude-3-opus-20240229																						
gpt-4-1106-preview																						
gemini-1.5-pro-001																						
gpt-4o-mini-2024-07-18																						
llama3-70b-instruct-q8_0																						
llama3-70b-instruct-q4_0																						
gpt-3.5-turbo-1106																						
gemini-1.5-flash-001																						
codegemma-7b-instruct-fp16																						
mixtral-8x22b-instruct-v0.1-q4_0																						
mixtral-8x7b-instruct-v0.1-q5_0																						
phi3-3.8b-mini-instruct-4k-fp16																						
codellama-70b-instruct-q4_0																						
gemini-pro																						
llama3-8b-instruct-fp16																						
command-r-plus-104b-q4_0																						
codellama																						

Benchmarking LLMs for Bio-image Analysis

Summary: 57 use-cases (yet), 20 LLMs (yet), n=10



Prompt Engineering



Prompt Engineering

Modify a prompt until it works.

Write python code to open the image file 'data/blobs.tif' and threshold it using Otsu's method. Apply connected component labeling to the result to retrieve a label image. Visualize the label image.



Here is the python code to achieve the task:

```
import cv2
import numpy as np
import matplotlib.pyplot as plt

# Load the image
img = cv2.imread('data/blobs.tif', 0)

# Apply Otsu's thresholding
_, thresholded = cv2.threshold(img, 0, 255, cv2.THRESH_BINARY + cv2.THRESH_OTSU)

# Apply connected component labeling
num_labels, labels = cv2.connectedComponents(thresholded)

# Visualize the label image
plt.figure(figsize=(10, 10))
plt.imshow(labels, cmap='nipy_spectral')
plt.colorbar()
plt.show()
```

Make sure you have the necessary libraries (OpenCV, NumPy, and Matplotlib) installed in your Python environment before running the code.

Prompt Engineering

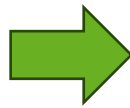
Modify a prompt until it works.

Write python code to open the image file 'data/blobs.tif' and threshold it using Otsu's method.

Apply connected component labeling to the result to retrieve a label image. Visualize the label image.

Do not explain the solution.

Only write Python code and short comments are ok.



```
import cv2
import numpy as np

# Read the image
img = cv2.imread('data/blobs.tif', cv2.IMREAD_GRAYSCALE)

# Apply Otsu's thresholding
_, thresh = cv2.threshold(img, 0, 255, cv2.THRESH_BINARY + cv2.THRESH_OTSU)

# Apply connected component labeling
num_labels, label_image = cv2.connectedComponents(thresh)

# Visualize the Label image
cv2.imshow('Label Image', label_image.astype(np.uint8) * 50)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

Prompt Engineering

Modify a prompt until it works.

Write python code to open the image file 'data/blobs.tif' and threshold it using Otsu's method.

Apply connected component labeling to the result to retrieve a label image. Visualize the label image.

Do not explain the solution.

Only write Python code and short comments are ok.

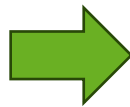
Assume the code is executed in a Jupyter notebook.

Do not use any of these Python libraries:

* OpenCV

* PIL

* Pillow



```
import matplotlib.pyplot as plt
from skimage import io, filters, measure

# Load the image
image = io.imread('data/blobs.tif')

# Threshold using Otsu's method
thresh = filters.threshold_otsu(image)
binary_image = image > thresh

# Connected component labeling
label_image = measure.label(binary_image)

# Visualize the label image
plt.figure(figsize=(8,8))
plt.imshow(label_image, cmap='nipy_spectral')
plt.axis('off')
plt.show()
```

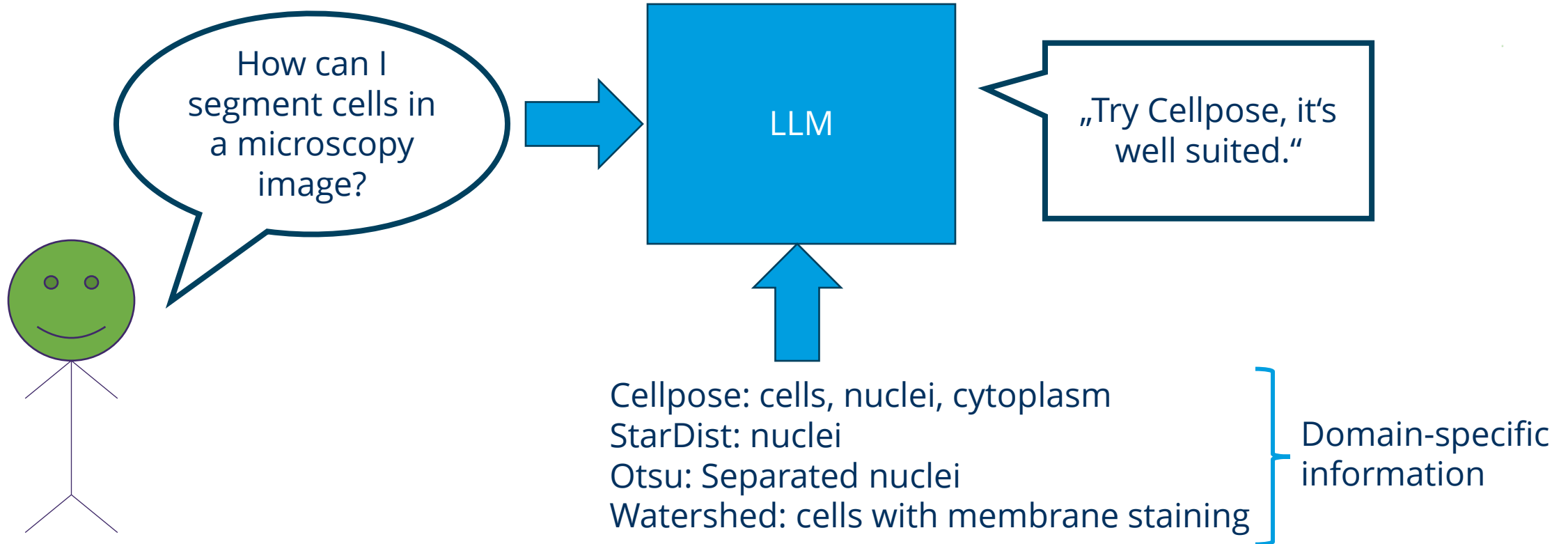
„Long“ Context

Retrieval Augmented Generation



Retrieval Augmented Generation (RAG)

An attempt to make LLMs answer more accurate / truthful



Retrieval Augmented Generation

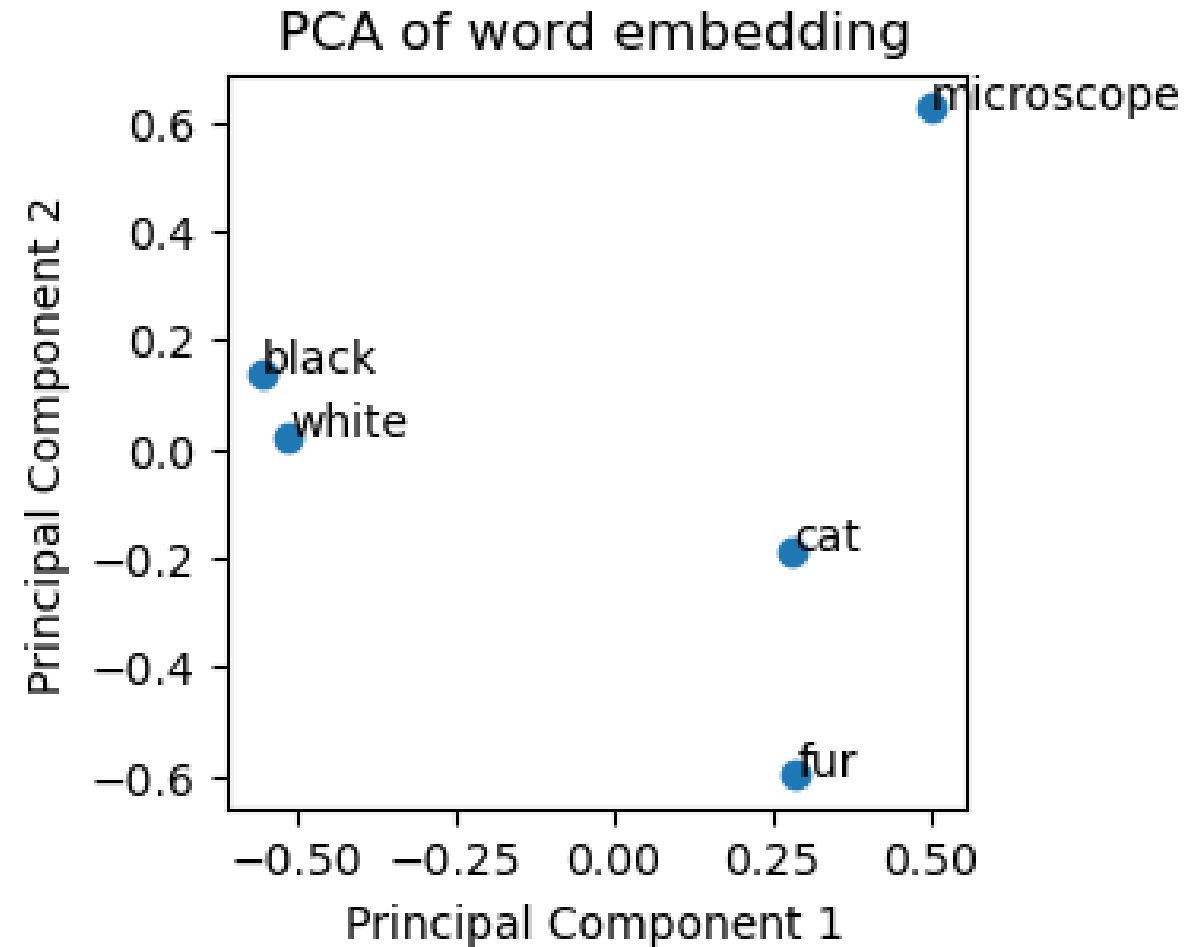
Uses embeddings to
Identify related documents

```
def embed(text):  
    from openai import OpenAI  
    client = OpenAI()  
  
    response = client.embeddings.create(  
        input=text,  
        model="text-embedding-3-small"  
    )  
    return response.data[0].embedding
```

```
vector = embed("Hello world")
```

```
len(vector)
```

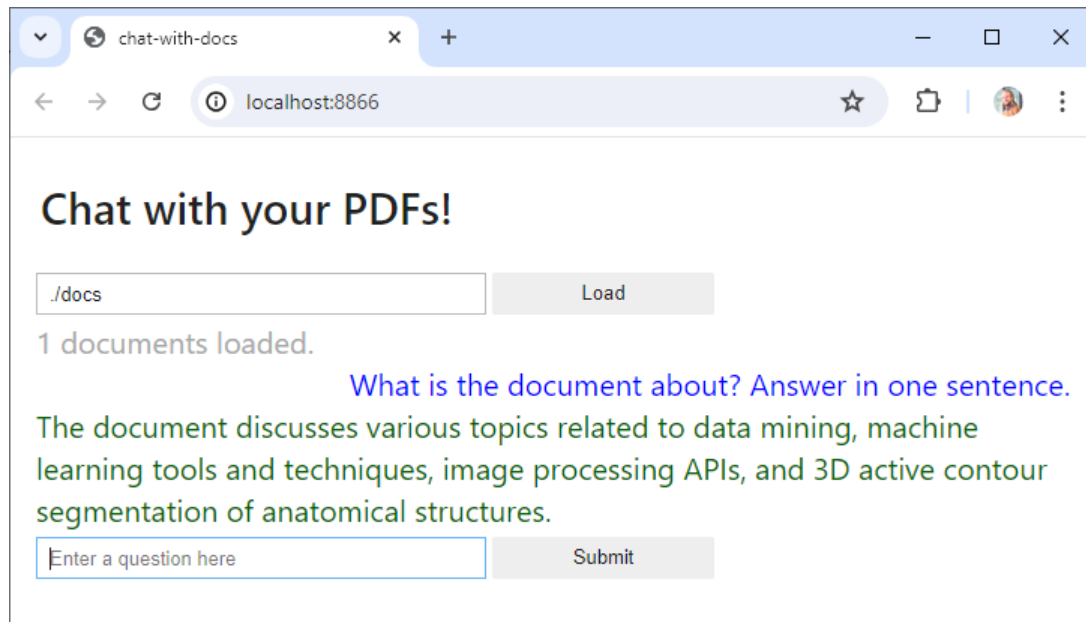
1536



Exercise: Chat With Docs!

Run this chatbot locally and load your documents

Explore the code. It is surprisingly simple!



```
class Chat():
    def __init__(self):
        self.former_folder = None

    def load(self, folder):
        if self.former_folder == folder:
            return self.count_documents()
        self.former_folder = folder

        documents = SimpleDirectoryReader(folder, required_exts=[".pdf"]).load_data()
        index = VectorStoreIndex.from_documents(documents)
        self._query_engine = index.as_query_engine()
        self._documents = documents
        return self.count_documents()

    def count_documents(self):
        return len(np.unique([d.metadata["file_name"] for d in self._documents]))

    def query(self, question):
        return self._query_engine.query(question)
```

Function calling



Function Calling

1. choosing a function

Given a list of tools...

- `get_current_time`
- `order_food`
- `book_room`

... and a task:

Please book meeting room 3 for Robert at 3pm.

Which is the right tool to use?

Some kind of next-word prediction task



`book_room`

Given a list of tools...

- * `get_current_time`
- * `order_food`
- * `book_room`

... and a task:
"Please book meeting room 3 for Robert at 3pm."
Which is the right tool to use?

The right tool to use for the task "Please book meeting room 3 for Robert at 3pm" is:

- `book_room`

Function calling

2. Parameterizing the function

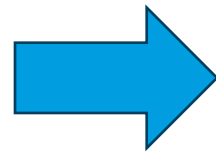
Given a function signature...

`book_room(room, time, person)`

... and a task:

Please book meeting room 3 for Robert at 3pm.

How could I use the tool?



`book_room("Meeting Room 3", "3pm", "Robert")`

Some kind of translation task

Function calling

2. Parameterizing the function

Given a function signature...

`book_room(room, time, person)`

... and a task:

Please book meeting room 3 for Robert at 3pm.

How could I use the tool?

Given a function signature...
`book_room(room, time, person)`
... and a task:
"Please book meeting room 3 for Robert at 3pm."
How could I use the tool?



To use the `book_room` function for the task "Please book meeting room 3 for Robert at 3pm," you would call the function with the appropriate arguments. Based on the function signature `book_room(room, time, person)`, the arguments should be:

- `room`: "meeting room 3"
- `time`: "3pm"
- `person`: "Robert"

Here's how you could use the tool:

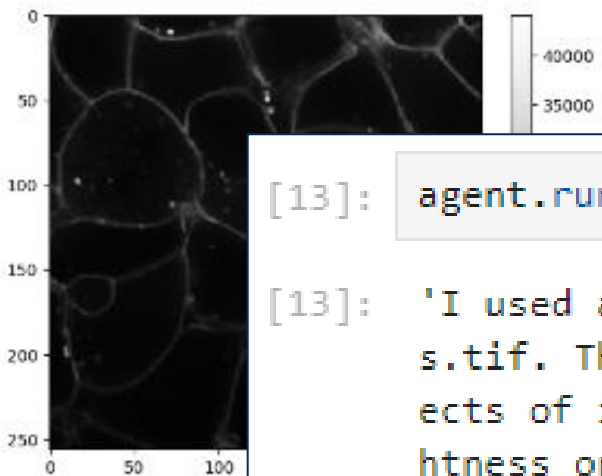
```
python  
  
book_room("meeting room 3", "3pm", "Robert")
```



LLMs deciding which function to call

Function calling doing bio-image analysis

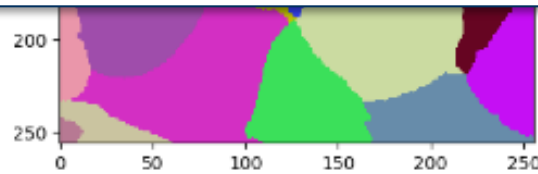
```
[10]: agent.run("Show the image.")
showing data/membranes.tif
```



```
shape (256, 256)
dtype uint16
size 128.0 kB
```

```
[11]: agent.run("Please segment the image data/membranes.tif")
segmenting (Local-minima-seeded watershed) data/membranes.tif
[11]: 'The segmented image has been stored as segmented_data/membranes.tif'
[12]: agent.run("Please show the segmented data/membranes.tif image.")
showing segmented_data/membranes.tif
```

```
[13]: agent.run("Which algorithm did you use? Why did you use this algorithm?")
[13]: 'I used an image segmentation algorithm to segment the image data/membrane
s.tif. This algorithm was chosen because it is effective at separating obj
ects of interest in the image based on their characteristics, such as brig
htness or color, which can help in further analysis and processing of the
image data.'
```



Fine-tuning LLMs

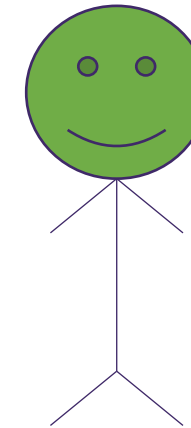
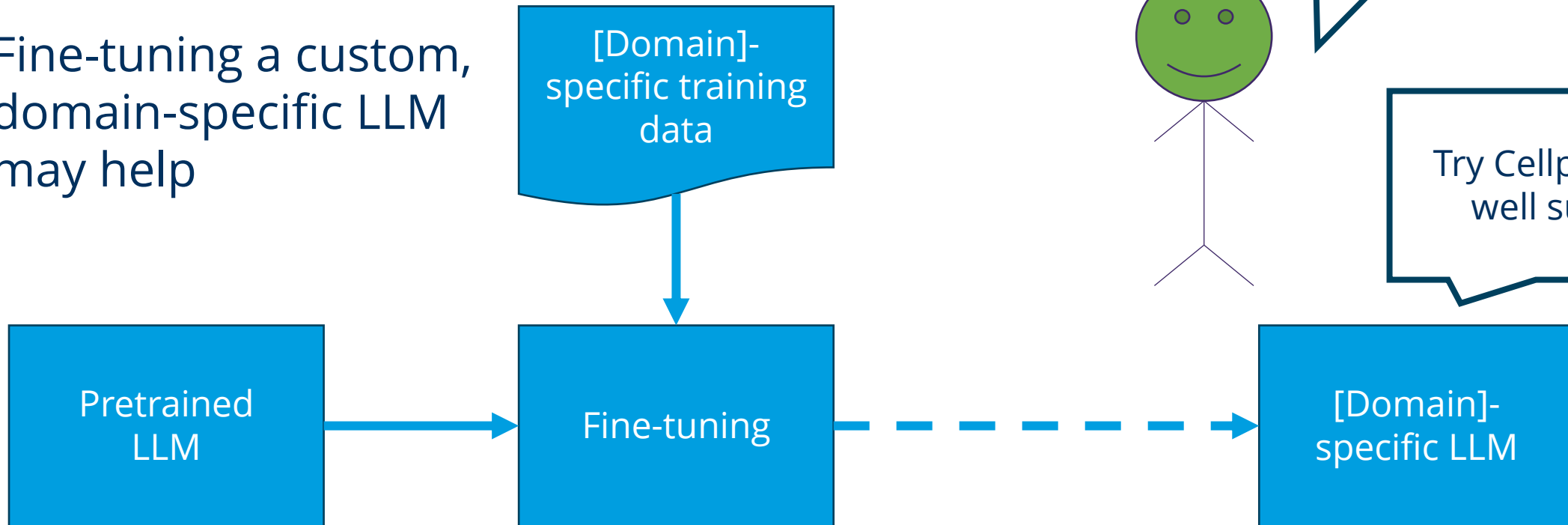


Fine-tuning

Classical prompt-engineering is limited

- Response time ↑
- Costs ↑

Fine-tuning a custom, domain-specific LLM may help

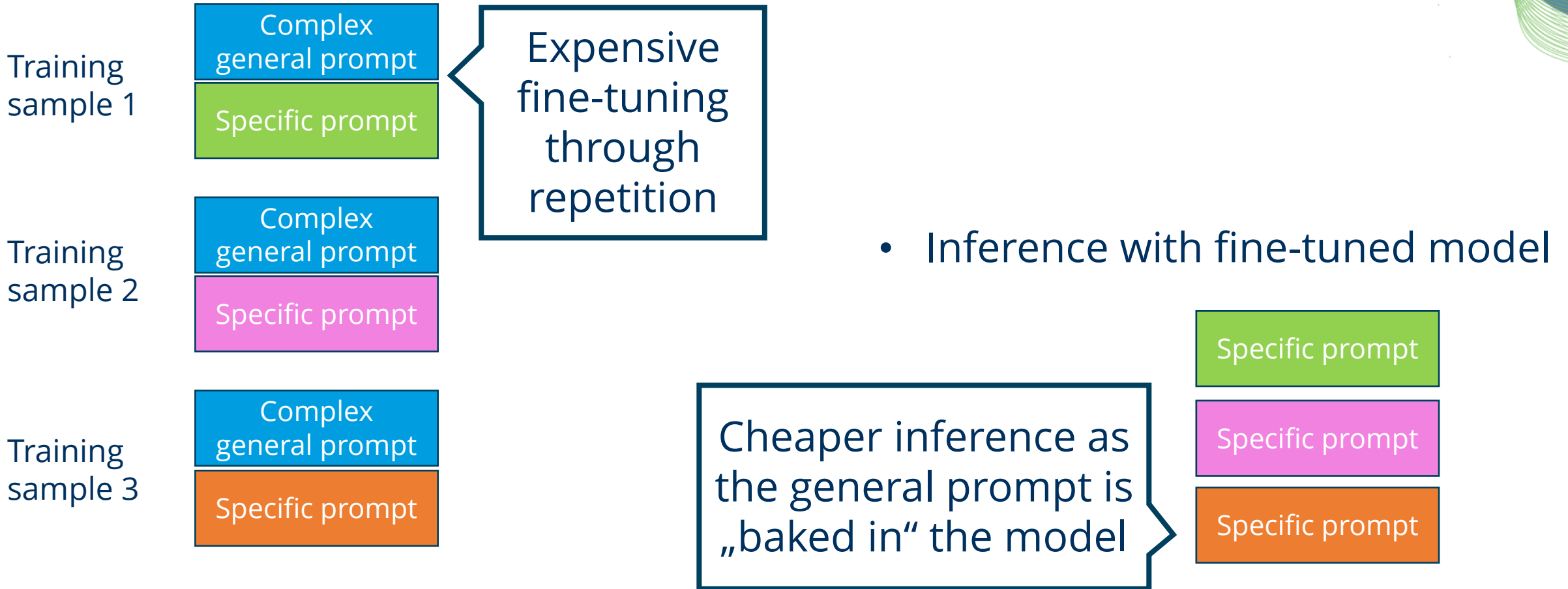


How can I segment cells in a microscopy image?

Try Cellpose, it's well suited.

Fine-tuning OpenAI's gpt-3.5-turbo

Training data should include successful general/system prompts



- Inference with fine-tuned model

LLM fine-tuning in practice: Python

The screenshot shows a web browser window with the URL `scads.github.io/generative-ai-notebooks/70_fine_tuning/40_fine_tuning.html`. The page title is "Fine-tuning an OpenAI GPT from questions and answers". The main content area contains a text box with a "Question:" and an "Answer:". The question is "How can I open CZI or LIF files using Python?". The answer explains that the `AICSImageIO` package is used and provides a code snippet:

```
python
from aicsimageio import AICSImage
aics_image = AICSImage("../data/EM_C_6_c0.ome.tif")

np_image = aics_image.get_image_data("ZYX")
```

Below the code, there is a "See also:" section with a link to a "lazyprogrammer tutorial".

The screenshot shows a web browser window with the URL `scads.github.io/generative-ai-notebooks/71_fine_tuning_hf/fine-tune-gemma.html`. The page title is "Fine-tuning Gemma on local hardware". The main content area contains a paragraph explaining that the notebook is for fine-tuning the `google/gemma-2b-it` model using Huggingface infrastructure locally. It mentions that Gemma is provided under and subject to the Gemma Terms of Use found at `ai.google.dev/gemma/terms`. Below this, there is a "Read more" section with links to "QLoRA", "Gemma Cookbook", and "example sft_qlora". There is also a "Troubleshooting" section with two bullet points:

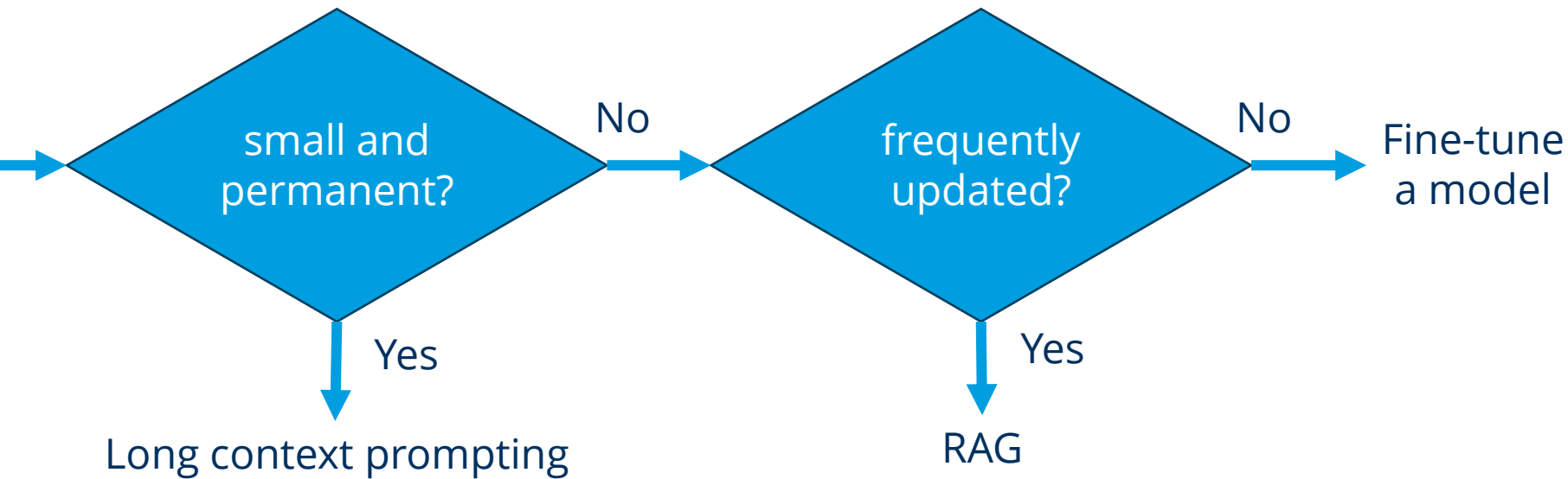
- If you run this notebook on Windows and receive error messages mentioning that CUDA initialization failed, make sure you have `bitsandbytes` version 0.43.2 or larger installed.
- If you run out of GPU memory, make sure you use the right hardware. This notebook was developed using an RTX 3080 mobile GPU with 16 GB of memory.

Challenge: Get good training data.

Prompt engineering decision tree (opinionated)

For deciding between classical prompt engineering, RAG and fine-tuning, these questions may provide guidance:

Is your knowledge base ...



Additional criteria:

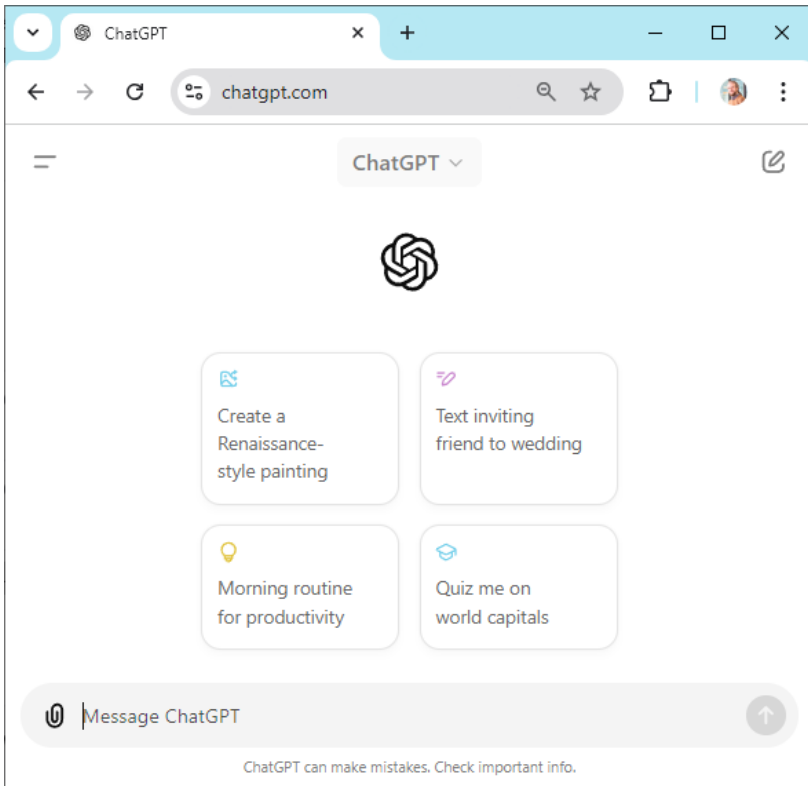
- Compute hardware
- Compute time
- Costs

LLMs are everywhere

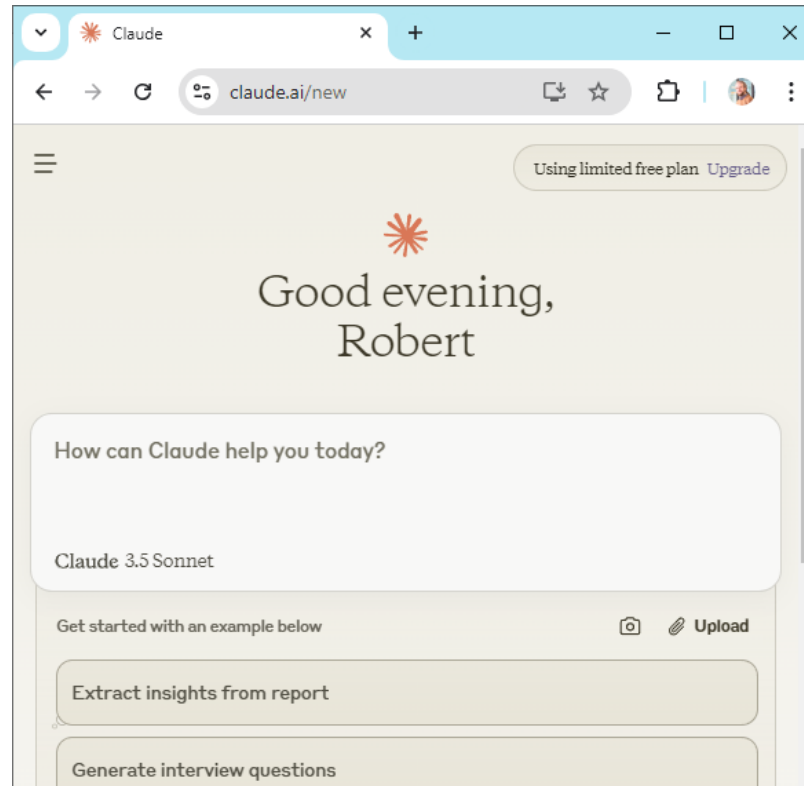


LLMs are everywhere

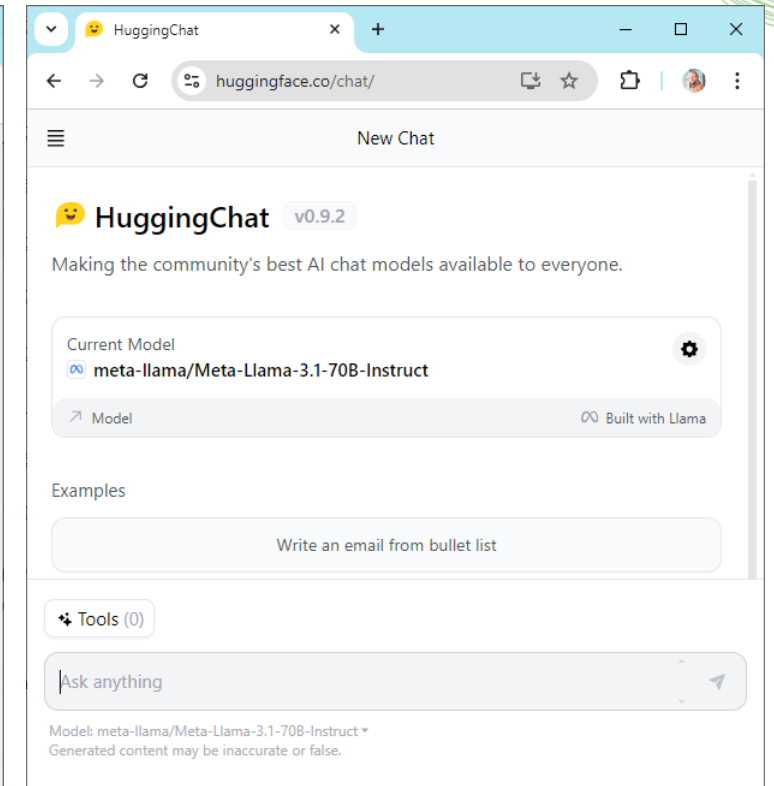
ChatGPT



Claude

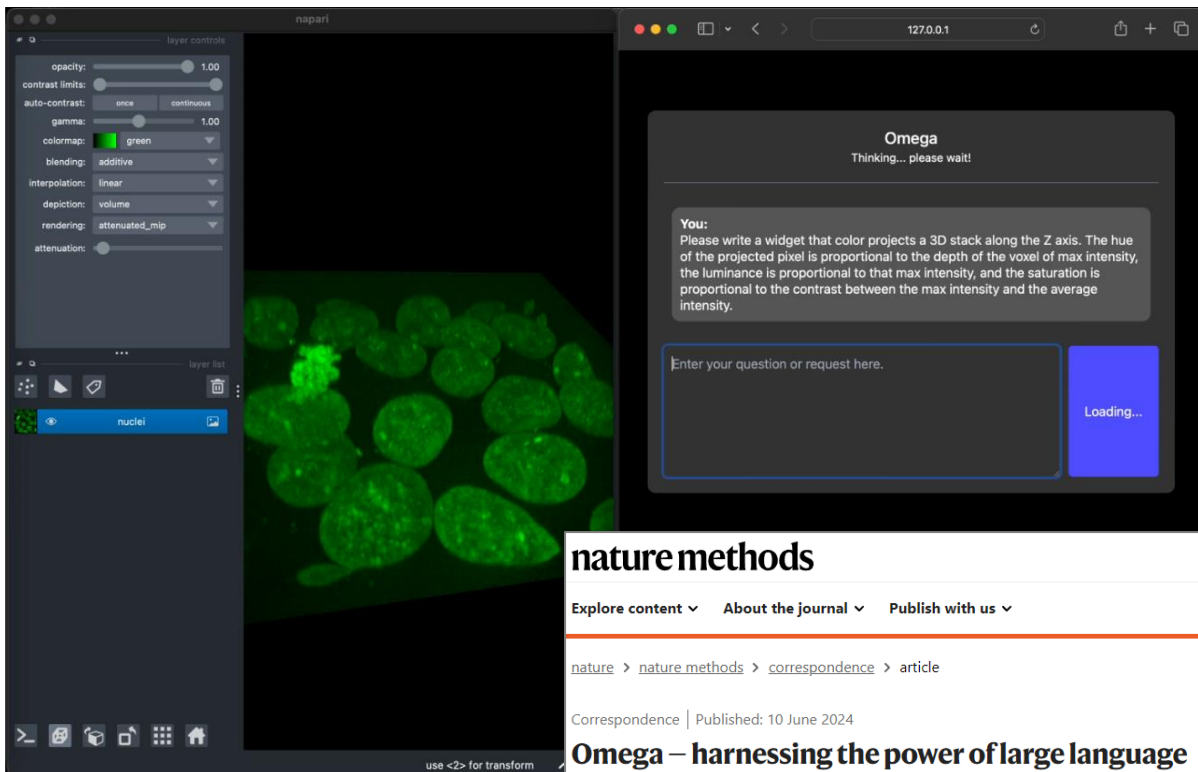


Llama 3.1 405B



LLMs are everywhere

Napari-chatGPT / Omega



<https://github.com/royerlab/napari-chatgpt>
<https://www.nature.com/articles/s41592-024-02310-w>

nature methods

Explore content ▾ About the journal ▾ Publish with us ▾

[nature](#) > [nature methods](#) > [correspondence](#) > article

Correspondence | Published: 10 June 2024

Omega – harnessing the power of large language models for bioimage analysis

Loïc A. Royer

Nature Methods (2024) | [Cite this article](#)

58 Altmetric | [Metrics](#)

Bioimage-io ChatBot

GloBIAS Seminar Series
Sept. 27th 2024
Oct. 18th 2024

Hi there! I'm Meiman. I am help you navigate the bioimaging tools and provide information about bioimaging. How can I help you today?

How can I segment dense nuclei in a fluorescence microscopy image?

To segment dense nuclei in fluorescence microscopy images, you can use several tools and methods available in the bioimaging community. Here are some recommended approaches and tools based on your needs:

1. Cellpose: This tool is designed for 2D nuclei segmentation and can handle dense and overlapping nuclei. It outputs a binary mask showing where nuclei were detected. More information can be found [here](#).
2. CellProfiler: This software offers a workflow for nuclei segmentation in 2D images. It can detect and segment nuclei, outputting a binary mask. More details are available [here](#).
3. ImageJ/Fiji: The Watershed method in ImageJ/Fiji is effective for separating touching and overlapping nuclei. A detailed tutorial on using this method for DAPI-stained nuclei can be found [here](#).
4. Ilastik: This tool allows for interactive learning and segmentation. You can create label classes for different objects (e.g., background, nuclei, mitotic nuclei, cytoplasm) and use live prediction mode to refine the segmentation. More information can be found in the [Ilastik documentation](#).
5. Scikit-image: This Python library provides various functions for image processing, including segmentation. An example of segmenting human cell nuclei using scikit-image can be found in the [scikit-image documentation](#).

These tools and methods should help you effectively segment dense nuclei in your fluorescence microscopy images.

[More Details](#)

Ready to chat! Type your message and press enter!

[Send](#) [Edit Profile](#) [Reset](#) [Feedback](#) [Options](#)

Warning: BioImage.IO Chatbot can make mistakes. Consider verifying important information.
Tips: Customize your profile to get personalized answers.
If you find our chatbot helpful for you, please click the AI4Life icon.

<https://bioimage.io/chat/>
<https://www.nature.com/articles/s41592-024-02370-y>

nature methods

Explore content ▾ About the journal ▾ Publish with us ▾

[nature](#) > [nature methods](#) > [correspondence](#) > article

Correspondence | Published: 09 August 2024

BioImage.IO Chatbot: a community-driven AI assistant for integrative computational bioimaging

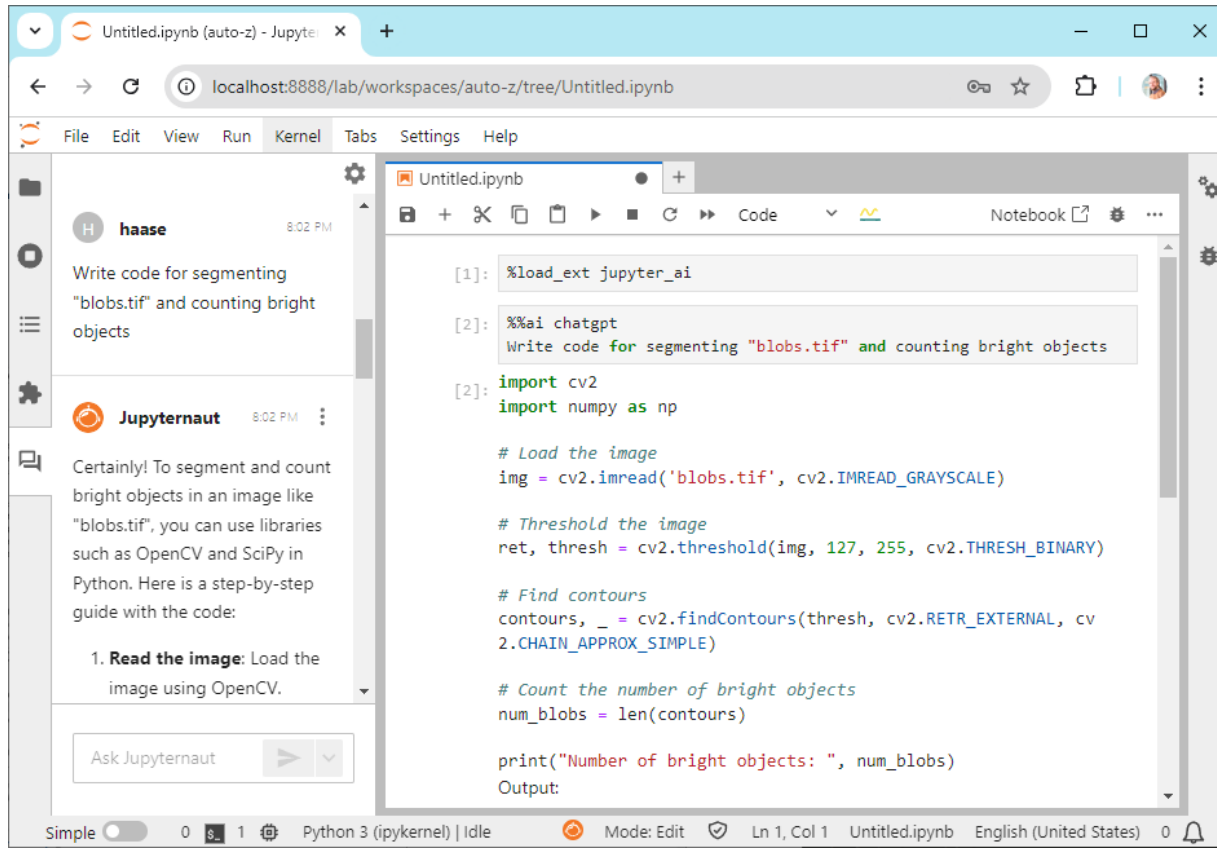
Wanlu Lei, Caterina Fuster-Barceló, Gabriel Reder, Arrate Muñoz-Barrutia & Wei Ouyang

Nature Methods **21**, 1368–1370 (2024) | [Cite this article](#)

865 Accesses | 1 Altmetric | [Metrics](#)

LLMs are everywhere

Jupyter-AI



The screenshot shows the Jupyter-AI interface in a browser. The address bar is `localhost:8888/lab/workspaces/auto-z/tree/Untitled.ipynb`. The notebook contains the following code:

```
[1]: %load_ext jupyter_ai
[2]: %%ai chatgpt
Write code for segmenting "blobs.tif" and counting bright objects
[2]: import cv2
import numpy as np

# Load the image
img = cv2.imread('blobs.tif', cv2.IMREAD_GRAYSCALE)

# Threshold the image
ret, thresh = cv2.threshold(img, 127, 255, cv2.THRESH_BINARY)

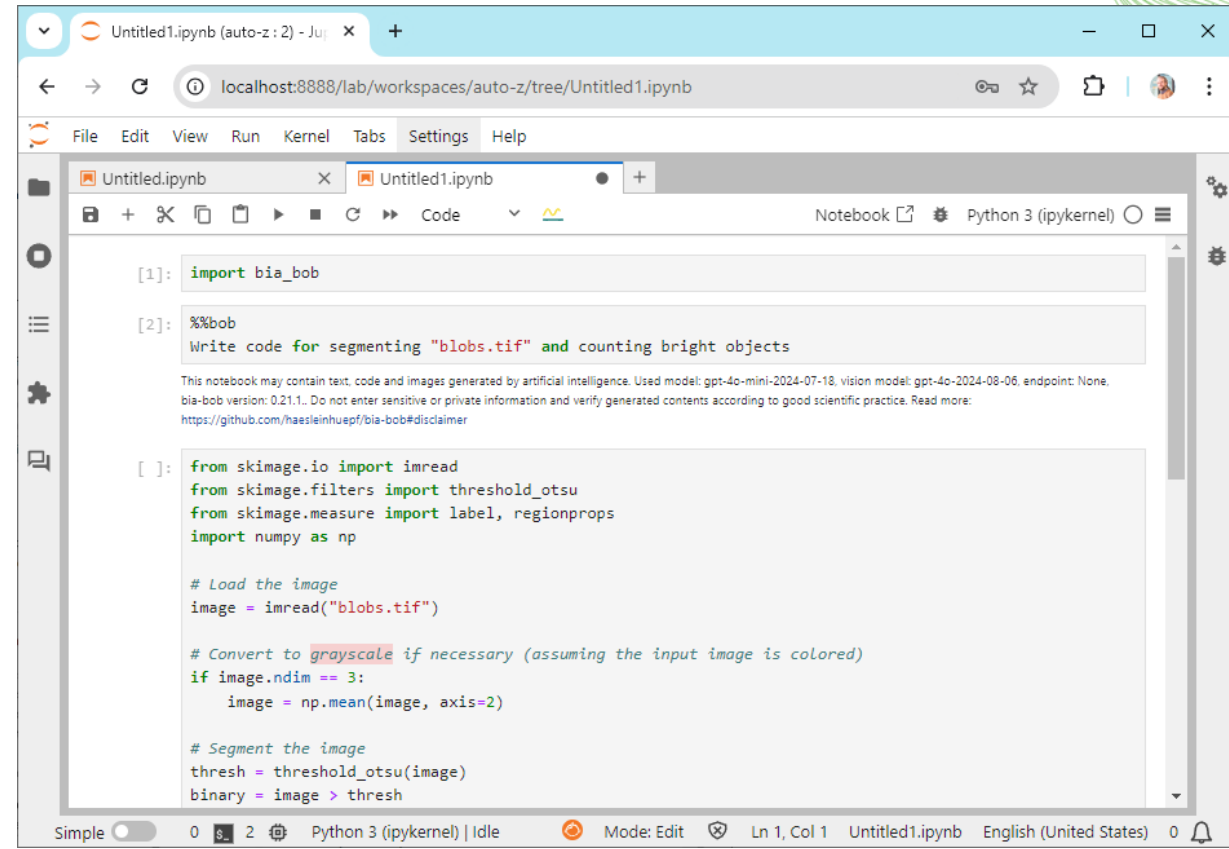
# Find contours
contours, _ = cv2.findContours(thresh, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)

# Count the number of bright objects
num_blobs = len(contours)

print("Number of bright objects: ", num_blobs)
Output:
```

The left sidebar shows a chat window with the user's prompt: "Write code for segmenting 'blobs.tif' and counting bright objects". The chatbot response provides a step-by-step guide and the code above.

bia-bob



The screenshot shows the bia-bob interface in a browser. The address bar is `localhost:8888/lab/workspaces/auto-z/tree/Untitled1.ipynb`. The notebook contains the following code:

```
[1]: import bia_bob
[2]: %%bob
Write code for segmenting "blobs.tif" and counting bright objects
[ ]: from skimage.io import imread
from skimage.filters import threshold_otsu
from skimage.measure import label, regionprops
import numpy as np

# Load the image
image = imread("blobs.tif")

# Convert to grayscale if necessary (assuming the input image is colored)
if image.ndim == 3:
    image = np.mean(image, axis=2)

# Segment the image
thresh = threshold_otsu(image)
binary = image > thresh
```

The left sidebar shows a chat window with the user's prompt: "Write code for segmenting 'blobs.tif' and counting bright objects". The chatbot response provides a disclaimer and the code above.

LLMs are everywhere

aider

```
macbook$ aider demo.py
Added demo.py to the chat
Using git repo: .git

demo.py> add a name param to the `greeting` function. add all the types.

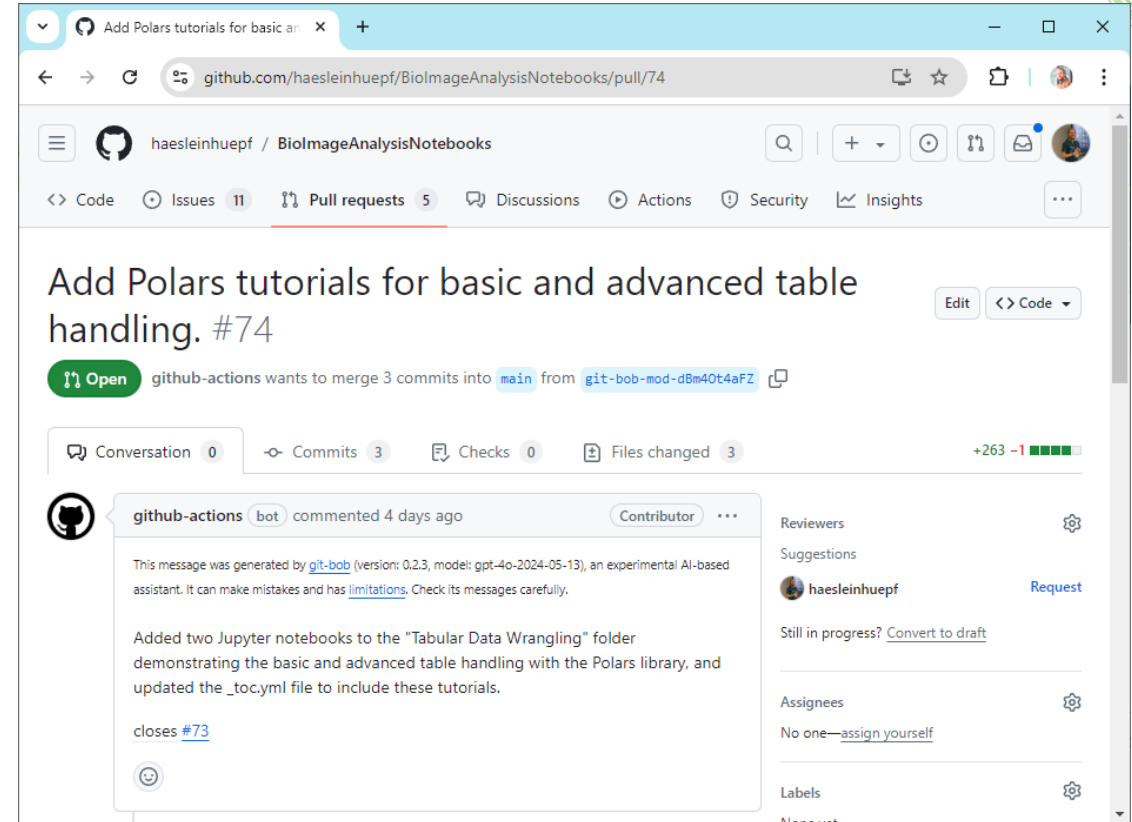
I will update the greeting function to accept a name parameter and include type hints for the parameter and the return type.

demo.py
<<<<<<< ORIGINAL
def greeting():
    print("Hello, world!")
=====
def greeting(name: str) -> None:
    print(f"Hello, {name}!")
>>>>>>> UPDATED

Applied edit to demo.py
Commit 9237455 aider: Updated greeting function to accept a name parameter and include type hints.

demo.py> |
```

git-bob



LLMs are everywhere

Github Copilot [Workspace]

Issue Polars #73

Specification

Do the two new .ipynb files demonstrate basic and advanced table handling with the polars library and are they listed in the .toc.yml file?

Current

- No, the two new .ipynb files demonstrating basic and advanced table handling with the polars library are not present.
- No files related to the polars library are listed in the docs/_toc.yml file.

Proposed

- Yes, the two new .ipynb files demonstrating basic and advanced table handling with the polars library are now present.
- docs/40_tabular_data_wrangling/basic_polars_table_handling.ipynb demonstrates basic table handling with the polars library.
- docs/40_tabular_data_wrangling/advanced_polars_table_handling.ipynb demonstrates advanced table handling with the polars library.
- Yes, the files related to the polars library are listed in the docs/_toc.yml file.
- The docs/_toc.yml file now includes entries for docs/40_tabular_data_wrangling/basic_polars_table_handling.ipynb and docs/40_tabular_data_wrangling/advanced_polars_table_handling.ipynb under the "Tabular data, plots and statistics" section.

Generate plan

Cursor

File Edit Selection View ...

BioImageAnalysisNotebooks

brightness_and_contrast.ipynb | my_library.py

docs > 03_advanced_python > my_library.py > square

```
1 def wuzzle(number):
7     return math.sqrt(number * 1.2)
8
```

Add a numpy-style docstring


Ctrl+@ Accept Ctrl+@ Reject Follow-up instructions... Ctrl+Shift+K

```
9 def square(number):
10     """
11     Calculate the square of a number.
12
13     Parameters
14     -----
15     number : int or float
16         The number to be squared.
17
18     Returns
19     -----
20     int or float
21         The square of the input number.
22     """
23     return number * number
```

main* 11:01 0/10 0

Ln 9, Col 1 Spaces: 4 UTF-8 CRLF Python 3.12.4 (base: conda) Cursor Tab

LLMs are everywhere



Microscope Image Analysis GPT

By Antonios Lioutas

Expert in microscopy image analysis with Python, skilled in tools like scikit-image, SimpleITK, Cellpose, Napari, Starfish, Dask, Numpy and Pandas.

★ 4.2
Ratings (10+)

Other
Category

300+
Conversations

Conversation Starters

- How do I use Cellpose for cell segmentation?
- What's the best way to analyze fluorescence images in Python?
- Can you help me debug this SimpleITK code?
- Tips for improving image analysis with Dask?

Capabilities

- ✓ Code Interpreter & Data Analysis
- ✓ Browsing

<https://chatgpt.com/g/g-FGdNx7Mll-microscope-image-analysis-gpt>

GPTs


Discover and create custom versions of ChatGPT that combine instructions, extra knowledge, and any combination of skills.

Q image analysis

All

- Image to Video**
Image analysis and video creation
By community builder 10K+
- Smarter Image Analysis**
Deep and powerful image analysis, returning more accurate and relevant details th...
By Finn J 1K+
- Bio-image Analysis GPT**
Bio-image Analysis with Python, a GPT created with content from the BioImageAna...
By Robert Haase 1K+
- PPT and PDF Analyst with Image Analysis**
Professional tone, analyzing PPTs, PDFs, and imagery.
By fuying ding 1K+
- Microscope Image Analysis GPT**
Expert in microscopy image analysis with Python, skilled in tools like scikit-image, S...
By Antonios Lioutas 300+
- Image Analyse and Description for Recreation**
Provides detailed forensic and artistic image analysis and will try to recreate the im...
By Andreas Dötsch 500+
- Stable Diffuser**
Expert image analysis, image-to-image, and image & prompt generation for DALL...
By Kenneth Whelan 900+
- SR image analysis**
I am analysing the spatial distribution of siglec and IgM molecules. I have images ...
By Vivek Ravikumar 20+
- Image Analysis Assistant**
图像分析助理, 专注于监督和深度学习
By fisherdaddy.com 7

<http://chat.openai.com/gpts>



Bio-image Analysis GPT

By Robert Haase

Bio-image Analysis with Python, a GPT created with content from the BioImageAnalysisNotebooks by R. Haase, G. Witz, M. Fernandes, M.L. Zoccoler, S. Taylor, M. Lampert, T. Korten, licensed CC-BY 4.0 and BSD3 unless mentioned otherwise.
<https://haesleinhuepf.github.io/BioImageAnalysisNotebooks>

★ 4.2
Ratings (20+)

Other
Category

1K+
Conversations

Conversation Starters

- How can I segment bright blobs in a fluorescence microscopy...
- How can I segment cells in a fluorescence microscopy image...
- How can I measure the number of neighbors in a label image?
- How can I erode labels in a label image?

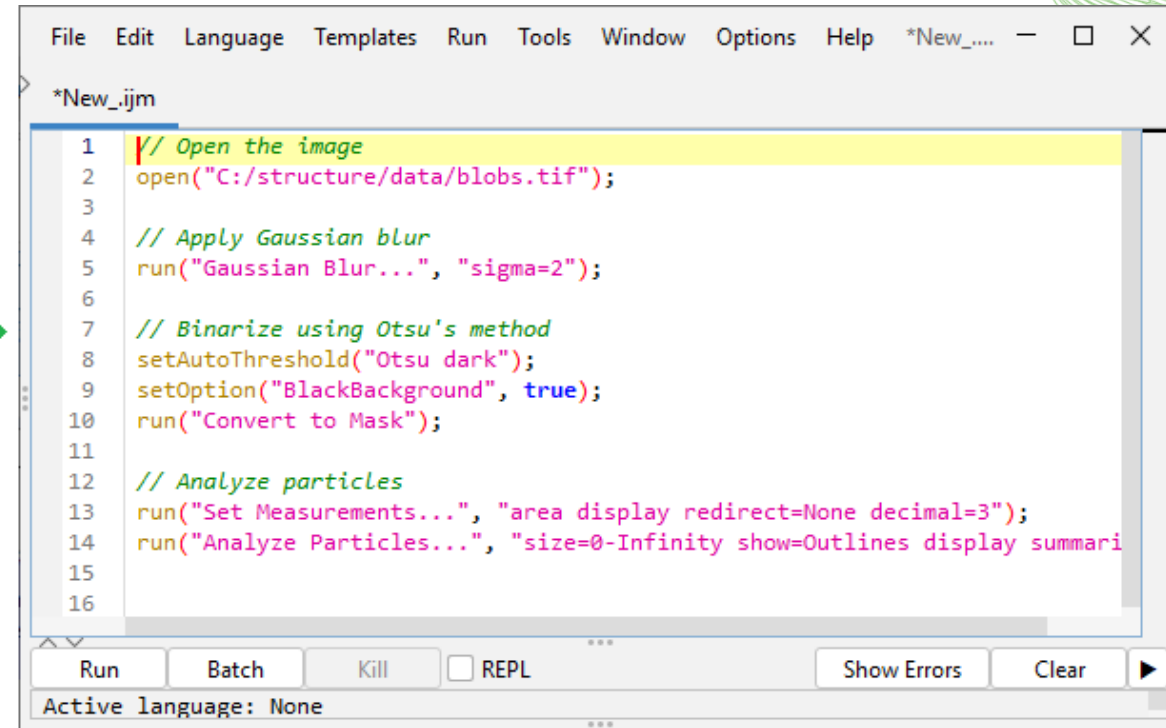
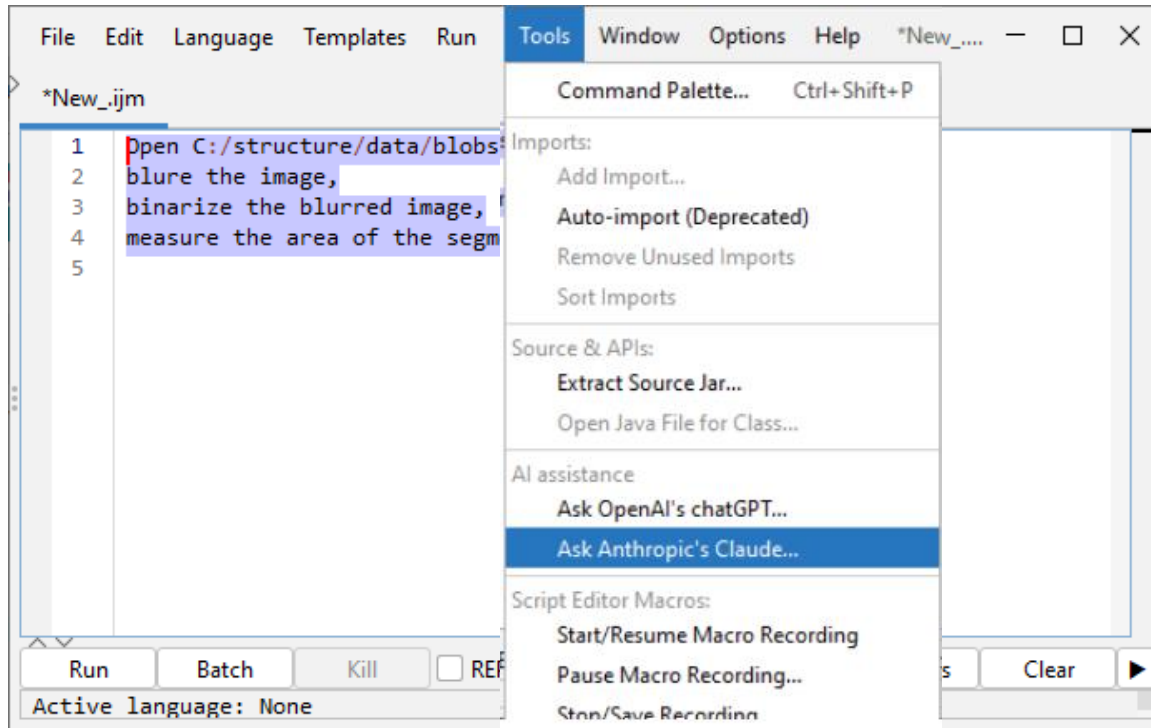
Capabilities

- ✓ Browsing

<https://chatgpt.com/g/g-psAohb1OY-bio-image-analysis-gpt>

LLMs are everywhere

Fiji's Script Editor

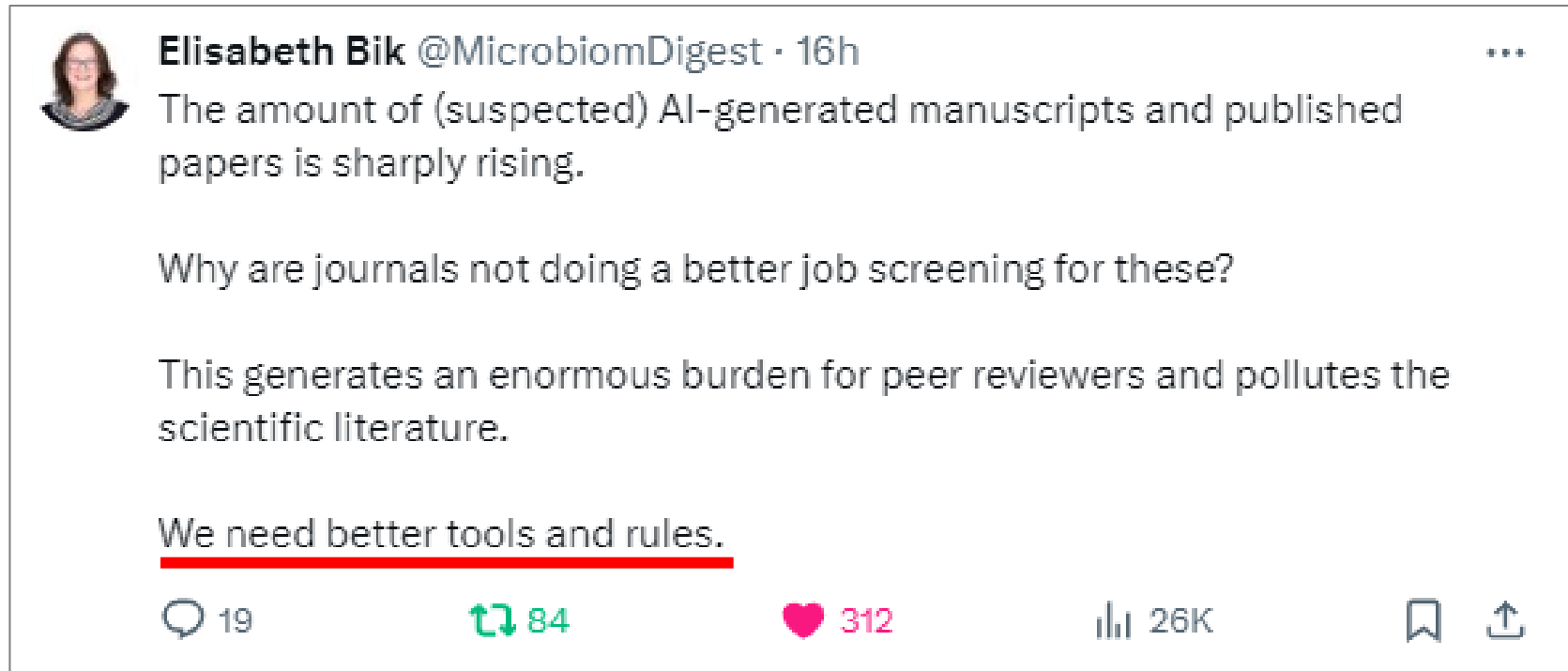


Good Scientific Practice



Challenges

Generative artificial intelligence imposes a risk to science



Elisabeth Bik @MicrobiomDigest · 16h

The amount of (suspected) AI-generated manuscripts and published papers is sharply rising.

Why are journals not doing a better job screening for these?

This generates an enormous burden for peer reviewers and pollutes the scientific literature.

We need better tools and rules.

19 84 312 26K

The image shows a tweet from Elisabeth Bik (@MicrobiomDigest) posted 16 hours ago. The tweet discusses the increasing amount of suspected AI-generated manuscripts and published papers, questioning why journals are not doing a better job of screening for them. It notes that this creates a heavy burden for peer reviewers and pollutes scientific literature. The tweet concludes with the underlined statement: "We need better tools and rules." The engagement metrics at the bottom of the tweet are: 19 replies, 84 retweets, 312 likes, and 26K views. The tweet also includes icons for replies, retweets, likes, bookmarks, and a share icon.

Rules...

„When making their results publicly available, researchers should, in the spirit of research integrity, disclose whether or not they have used generative models, and if so, which ones, for what purpose and to what extent.“

Check your institutions' and funders' guidelines.

Statement by the Executive Committee of the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) on the Influence of Generative Models of Text and Image Creation on Science and the Humanities and on the DFG's Funding Activities

September 2023

Good scientific practice

If you use code from ...

a human expert

an expert LLM

You must ...

- Understand the code (roughly)
- Question used methods
- Check results carefully
- Test code on samples the expert didn't see



Good scientific practice

If you use code from ...

a human expert

an expert LLM

You must ...

- Pay the expert
- Mention the expert
- Share responsibility
- Ask the expert endless questions



\$100/h



co-author



\$0.1/h

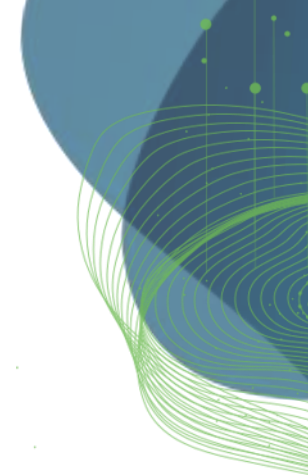


in methods



Exercises

Robert Haase



Exercises: Generative AI Notebooks (Python)

The screenshot shows the homepage of the ScaDS.AI Generative AI Notebooks. The page title is "Generative Artificial Intelligence Notebooks". The main text states: "This page is a collection of Jupyter Notebooks about Generative Artificial Intelligence (GenAI) - focusing on how to use certain techniques using Python. It aims at Python programmers who want to dive into GenAI for generating text, code and images using commercial and open source/weights models." Below this, it says: "Contributions and feedback are very welcome! In case you see room for improvement, please create a github issue and/or consider contributing." A "Topics" section lists the following: "The notebook collection aims covering these topics:"

- Large Language Models (LLMs)
- Vision Language Models (VLMs)
- Multi-modal Language Models
- Text/Code/Image generation
- Prompt Engineering
- Retrieval-augmented-generation
- Model fine-tuning

A sidebar on the left contains a search bar and a navigation menu with categories: "Generative Artificial Intelligence Notebooks", "Setup" (with a sub-item "Setting up your computer"), "LLM basics" (with sub-items: "Prompting basics", "Accessing LLMs", "Chatbots", "Function / Tool calling"), and "Multi-Modal LLMs" (with sub-items: "Image generation", "Image manipulation").

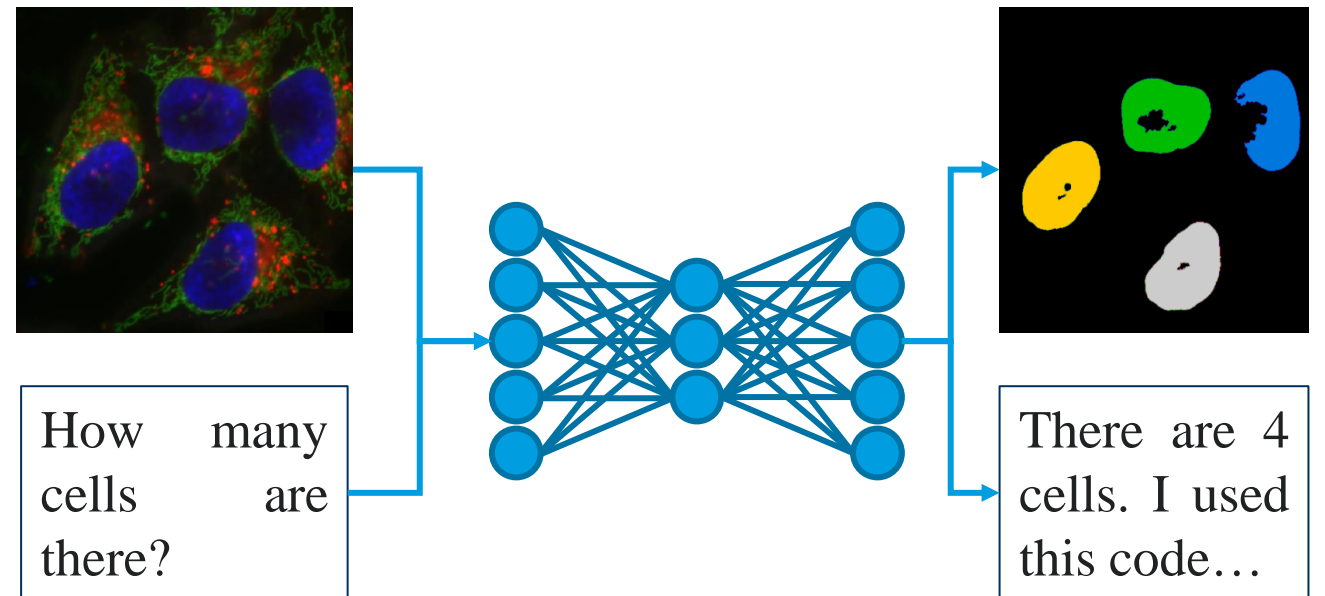
Requires Python programming skills

The screenshot shows the "Setting up your computer" page. The title is "Setting up your computer". The main text says: "This chapter provides instructions for setting up your computer." Below this, there is a section titled "Computer hardware" with the text: "To execute the notebooks in this collection, it is highly recommended to use a computer with a CUDA compatible NVidia Graphics card with at least 4GB of memory. These notebooks were tested on a Windows 10 Laptop with an NVidia RTX 3050 Mobile GPU." Another section titled "Setting up Python and Conda environments" contains the text: "When working with Python, we will make use of many plugins and software libraries which need to be organized. One way of doing this, is by managing Conda environments. A conda environment can be seen as a virtual desktop, or virtual computer, accessible via the terminal. If you install some software into one Conda environment, it may not be accessible from another environment. If a Conda environment breaks, e.g. incompatible software was installed, you can just make a new one and start over." At the bottom, there is a "See also" section with a link: "Managing Scientific Python environments using Conda, Mamba and friends". The sidebar on the left is similar to the first screenshot, but the "Setting up your computer" item is expanded.

Installation instructions;
GPU-Workstation recommended

Summary & outlook

- LLMs can generate code to analyze biological microscopy images
- Open-source benchmarks can help targeting further improvement
- Challenges:
 - Identify best strategy (fine-tuning, prompt-engineering, RAGs)
 - Multi-modal / multi-agent approaches
 - Introduce good scientific practice -> trustworthy AI



Acknowledgements



Christian Tischer
(EMBL Heidelberg)



Nico Scherf
(MPI CBS Leipzig)



Jean-Karim Hériché
(EMBL Heidelberg)



Loïc A. Royer
(CZ Biohub)

Mara Lampert (ScaDS.AI / TU Dresden), Kevin Yamauchi (ETH Zurich / Basel), Seth Hinz, Teun A.P.M. Huijben, Jordão Bragantini (CZ Biohub San Francisco), Ilan Theodoro (CZ Biohub and the State University of Campinas, Brazil), Ian C. (University of British Columbia)

Communities & platforms



<https://doi.org/10.5281/zenodo.13374649>

Funding



Diese Maßnahme wird gefördert durch die Bundesregierung aufgrund eines Beschlusses des Deutschen Bundestages. Diese Maßnahme wird mitfinanziert durch Steuermittel auf der Grundlage des von den Abgeordneten des Sächsischen Landtags beschlossenen Haushaltes.

