black hat
USA 2022

GPT-3 and Me

How Supercomputer-scale Neural Network Models Apply to Defensive Cybersecurity Problems

Younghoo Lee, Principal Research Scientist, Sophos
Joshua Saxe, Chief Scientist, Sophos

Your presenters

- **Joshua Saxe / @joshua_saxe**
- Chief Scientist at Sophos
- Author of “Malware Data Science” from No Starch Press, with Hillary Sanders
- 15 years of experience doing academic and applied research at the intersection of cybersecurity and machine learning

- **Younghoo Lee**
- Principal Research Engineer at Sophos
- Author of multiple peer-reviewed papers and patents on applying machine learning to cybersecurity
- Principal researcher behind multiple ML systems deployed to millions of networks

Information Classification: General

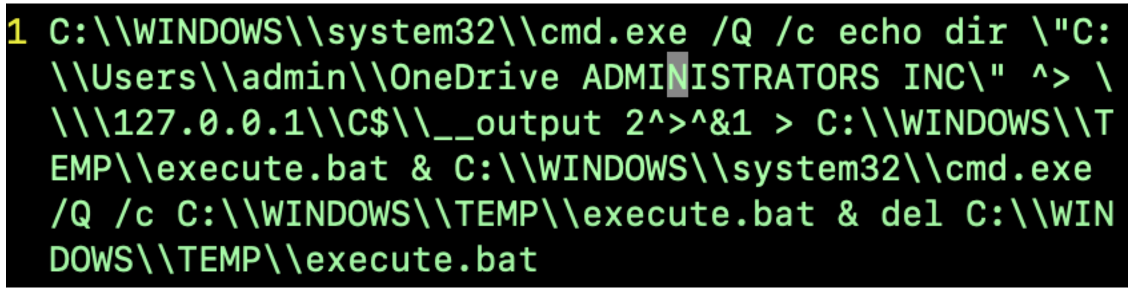
Theses in this talk

- Model scale has fundamentally changed machine learning's capabilities
- Self-supervised learning has fundamentally changed machine learning's capabilities
- It's time that the security data science research community comes together to understand what this means for our work
- This talk just reflects early work in this space, we hope much more community work will ensure

Information Classification: General

Results in this talk:

A GPT-3 based approach can describe complex commandlines, reducing SOC analyst cognitive load



```
C:\\WINDOWS\\system32\\cmd.exe /Q /c echo dir \"C:
\\Users\\admin\\OneDrive ADMINISTRATORS INC\" ^> \
\\127.0.0.1\\C$\\__output 2^>^&1 > C:\\WINDOWS\\T
EMP\\execute.bat & C:\\WINDOWS\\system32\\cmd.exe
/Q /c C:\\WINDOWS\\TEMP\\execute.bat & del C:\\WIN
DOWS\\TEMP\\execute.bat
```

GPT-3 based approach

The command will create a file called "execute.bat" in the C:\\WINDOWS\\TEMP folder. It will then run the command "dir C:\\Users\\admin\\OneDrive ADMINISTRATORS INC" and output the results to the __output file on the local machine. The batch file will then execute itself and delete itself afterwards.

Results in this talk:

A GPT-3 based approach can learn to detect malicious documents using tiny training volumes

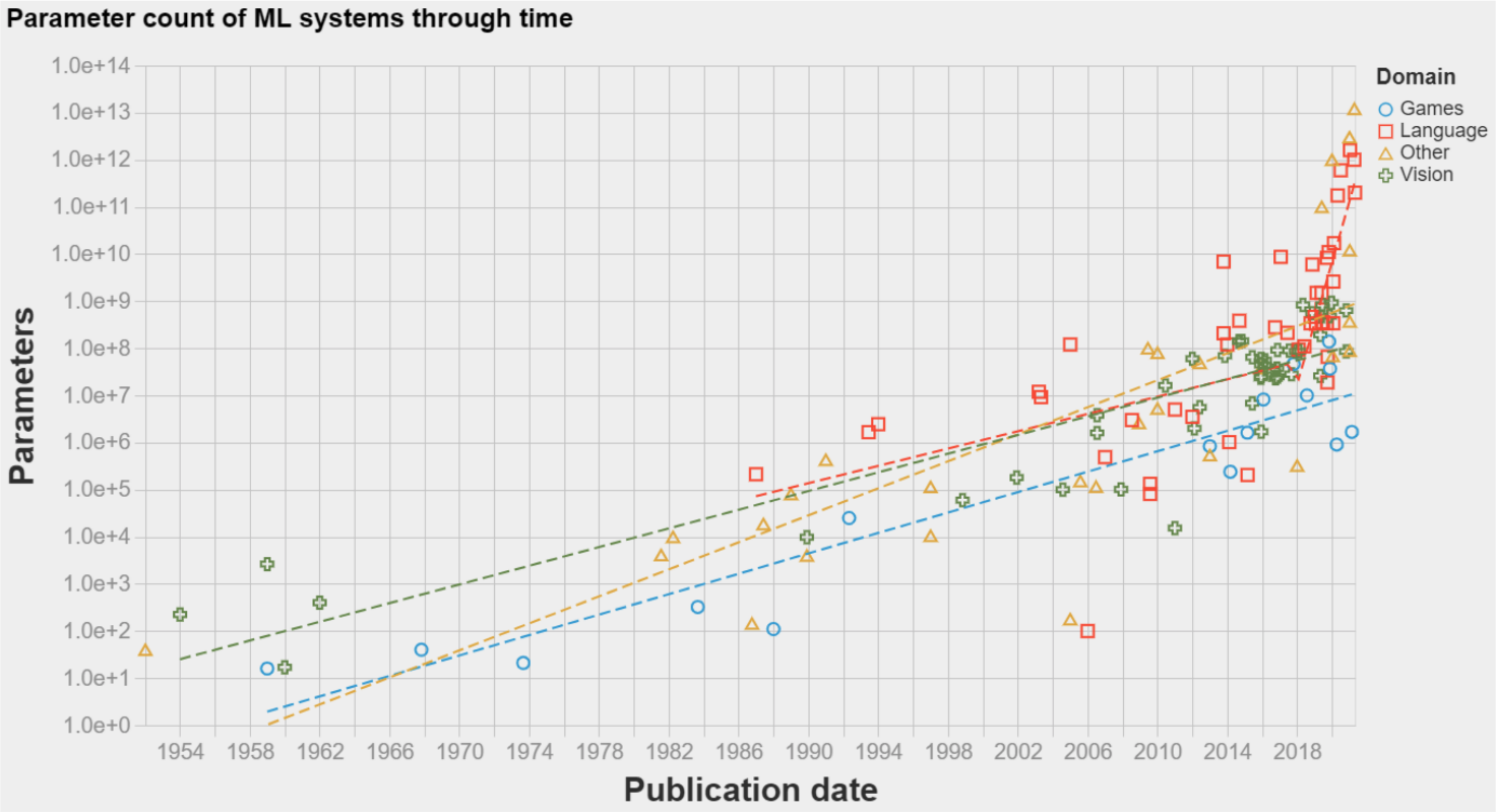
		GPT-3 F1-score	Random Forest F1-score
Training samples	1 ham, 1 spam	**0.90**	0.55
	4 ham, 4 spam	**0.95**	0.79

We would need far more data to learn to detect a spam campaign using traditional methods!

black hat USA 2022

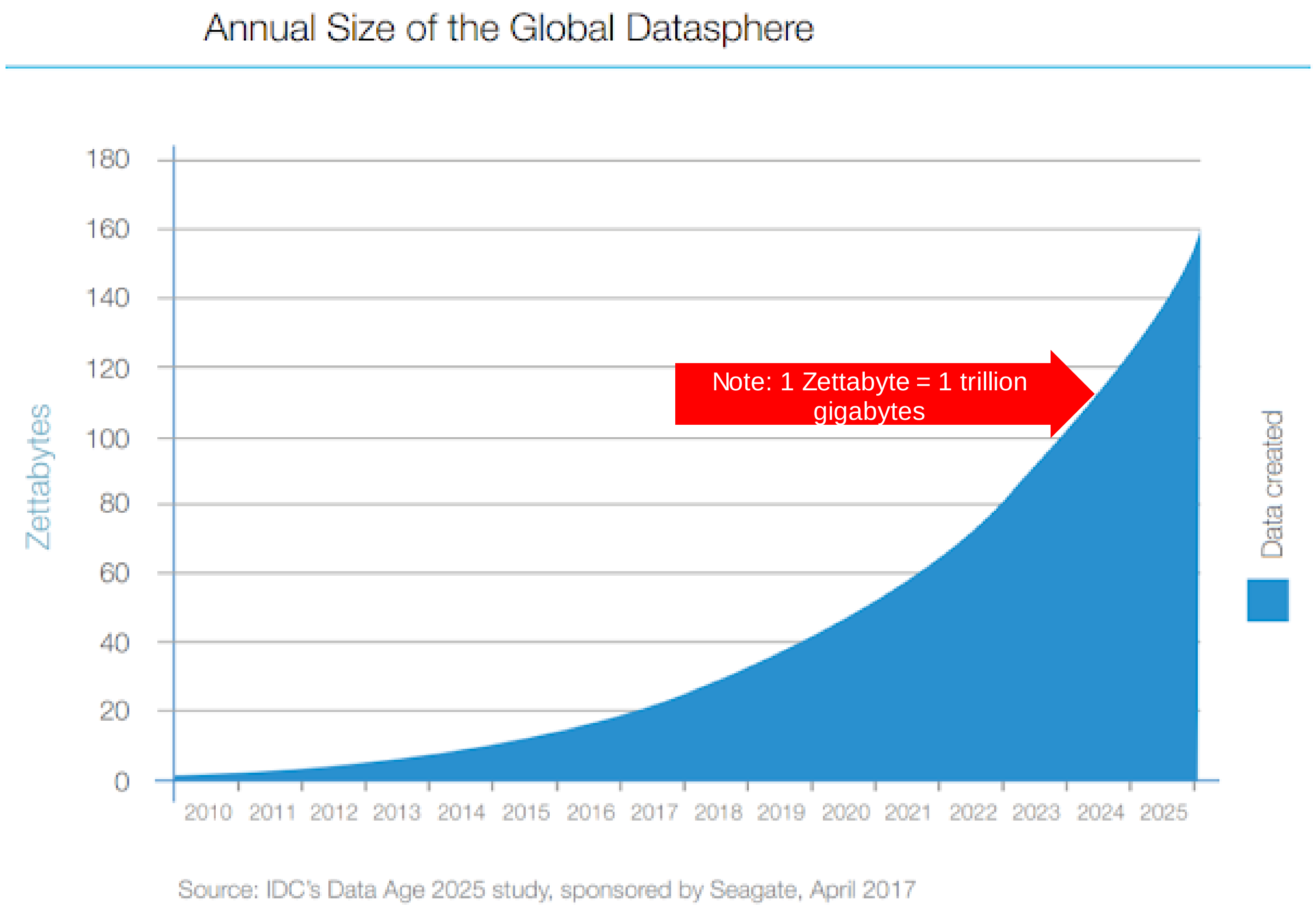
Background: how large language models and self-supervised learning are transforming machine learning

Deep learning models are getting bigger



https://towardsdatascience.com/parameter-counts-in-machine-learning-a312dc4753d0

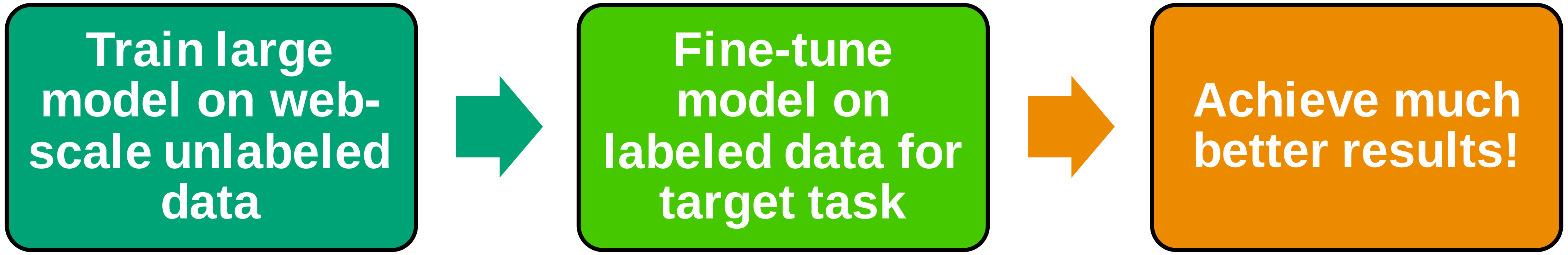
These large models are taking advantage of unlabeled web-scale data



Note: 1 Zettabyte = 1 trillion gigabytes

Source: IDC's Data Age 2025 study, sponsored by Seagate, April 2017

How self-supervised learning works

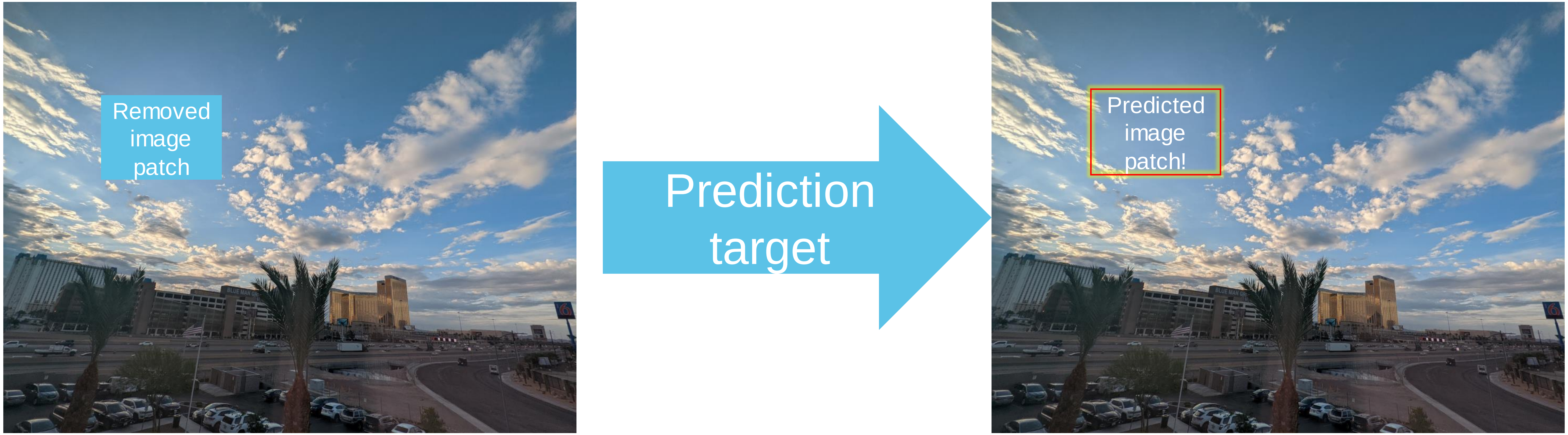


Train large model on web-scale unlabeled data

Fine-tune model on labeled data for target task

Achieve much better results!

How self-supervised learning works



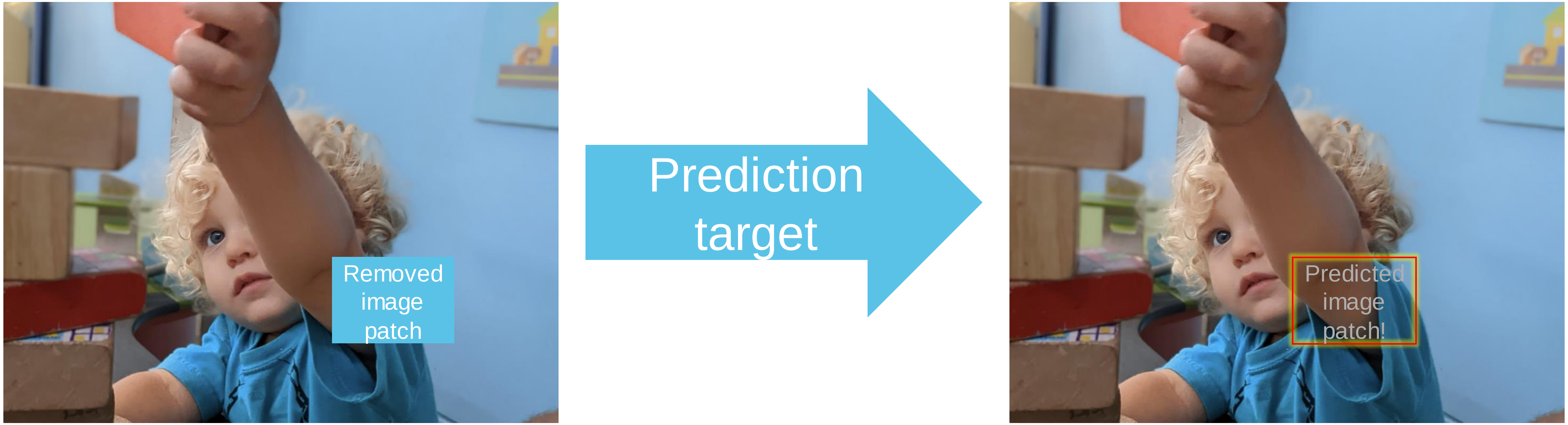
Removed image patch

Prediction target

Predicted image patch!

Information Classification: General

How self-supervised learning works

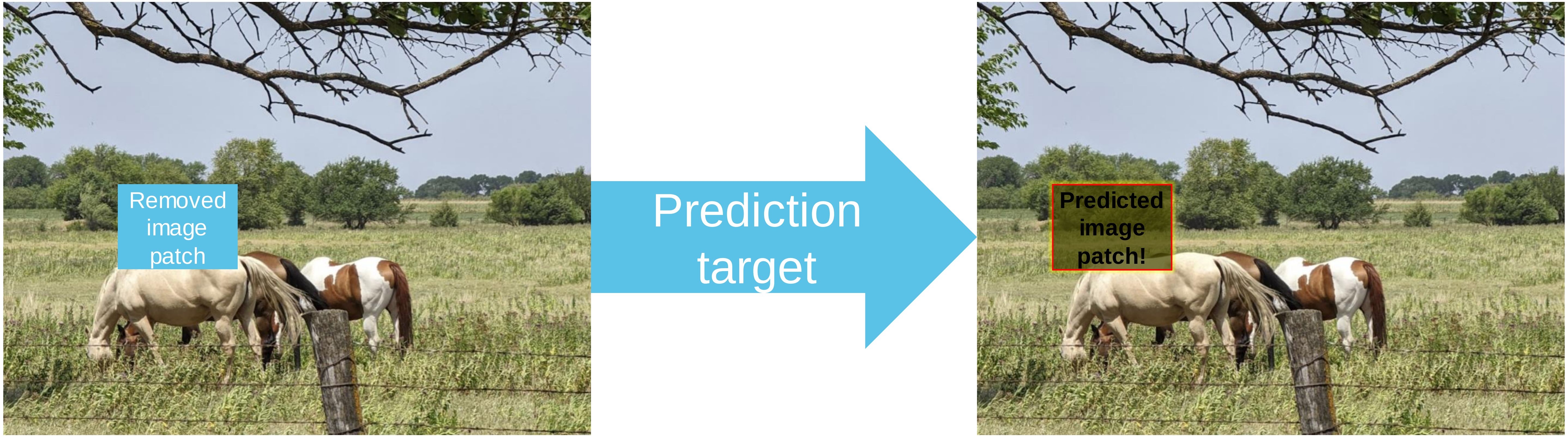


Removed image patch

Prediction target

Predicted image patch!

How self-supervised learning works

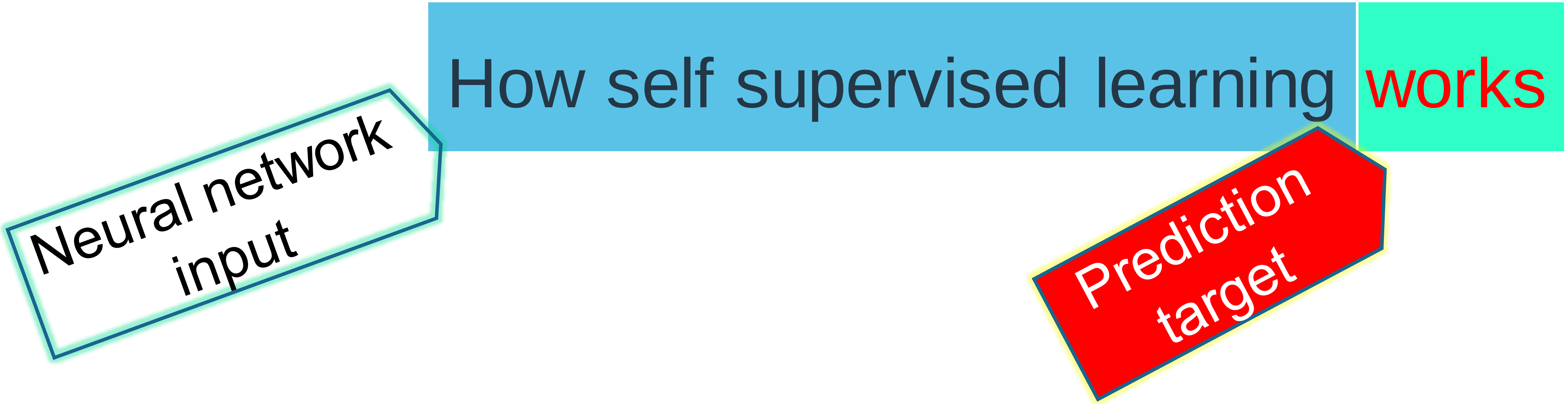


Removed image patch

Prediction target

Predicted image patch!

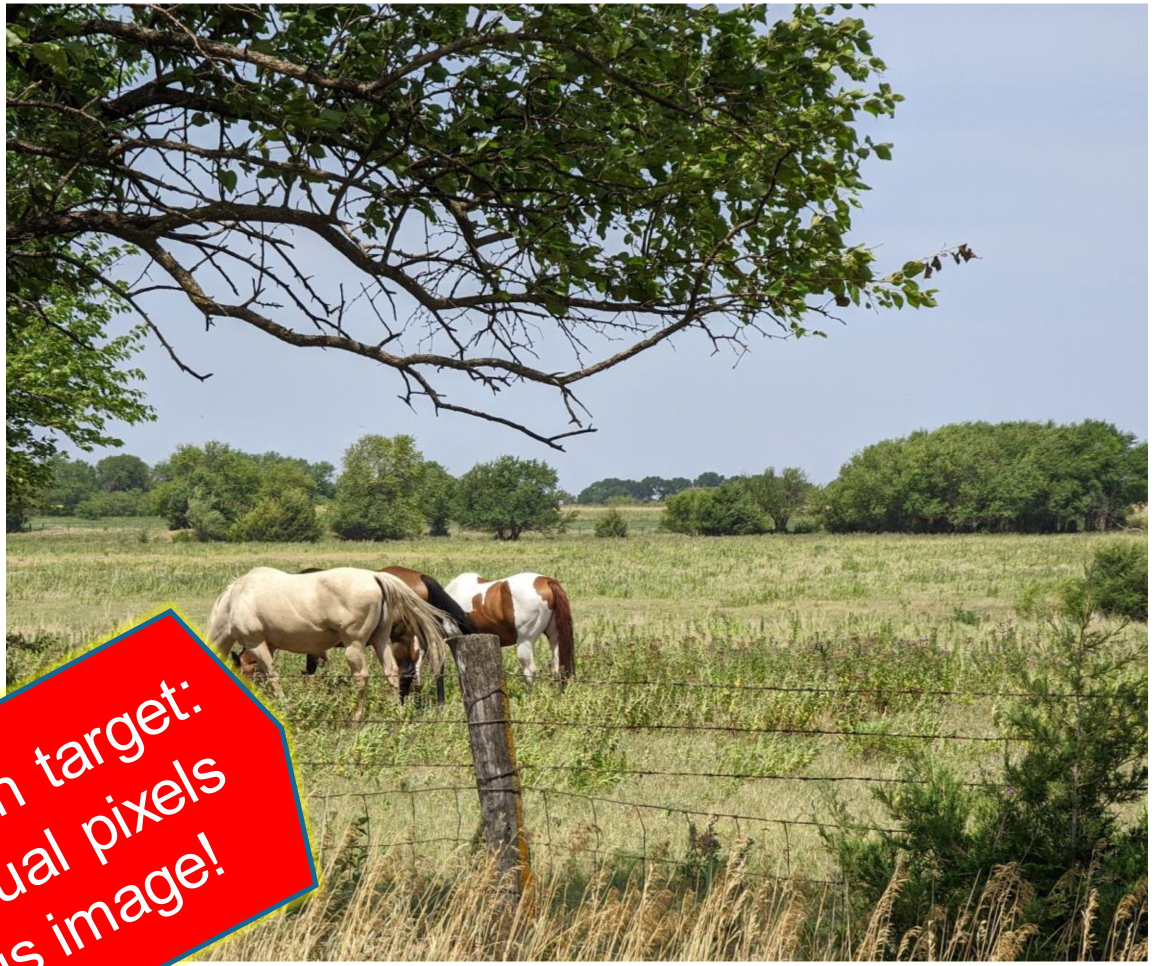
How self-supervised learning works



How self-supervised learning works

Model input: “Horses graze in a bucolic Kansas field, a tree is in the foreground above them,a fence separates the viewer from the horses, and grass and trees stretch into the distance”

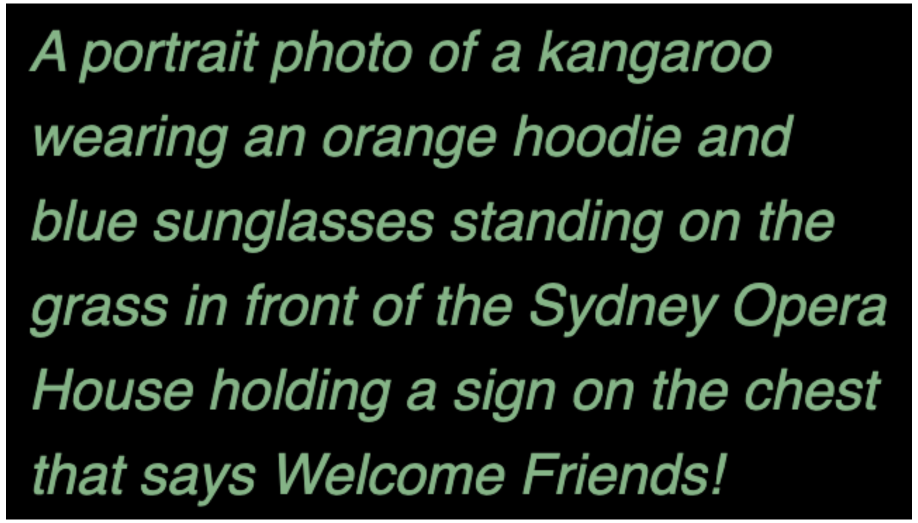
Neural network input



Prediction target: the actual pixels in this image!

Information Classification: General

Capabilities emerge with self-supervised learning + scale

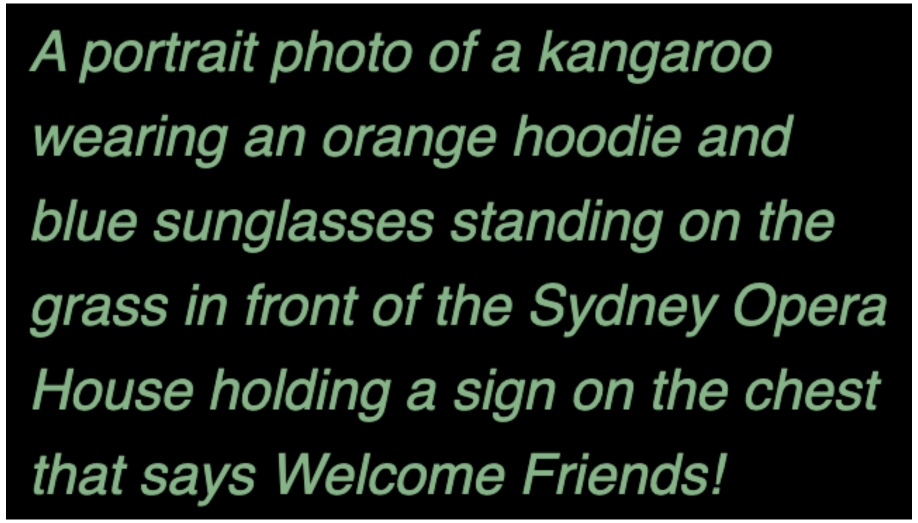


A portrait photo of a kangaroo wearing an orange hoodie and blue sunglasses standing on the grass in front of the Sydney Opera House holding a sign on the chest that says Welcome Friends!

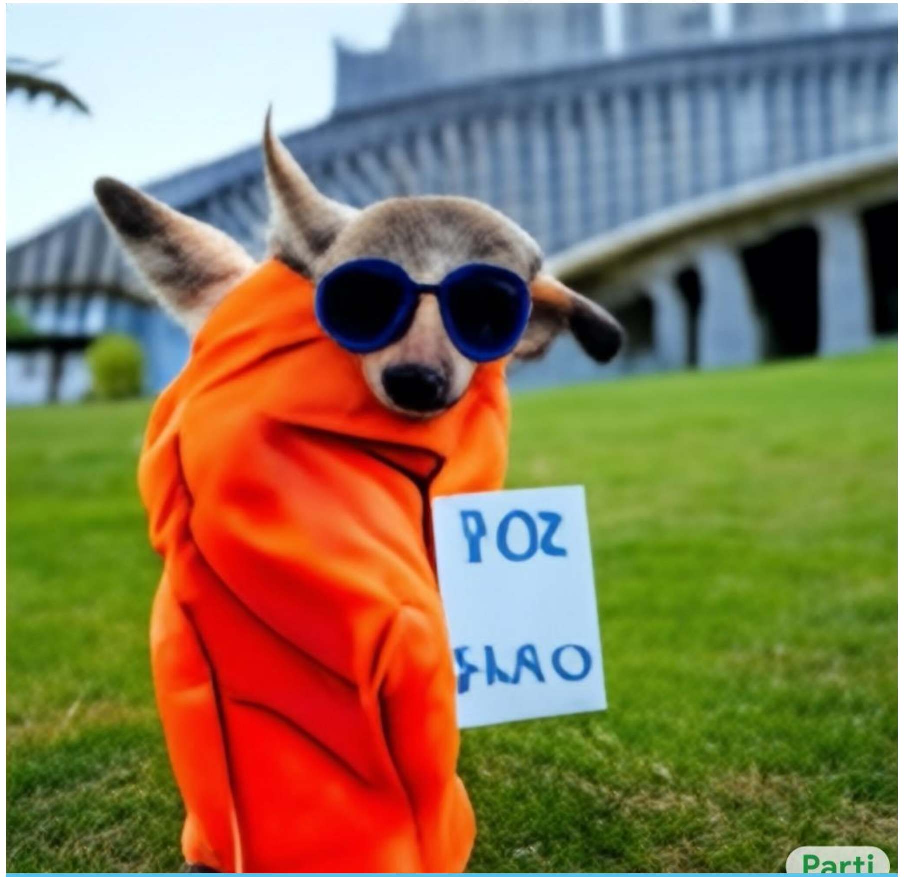
?

Information Classification: General

Capabilities emerge with self-supervised learning + scale



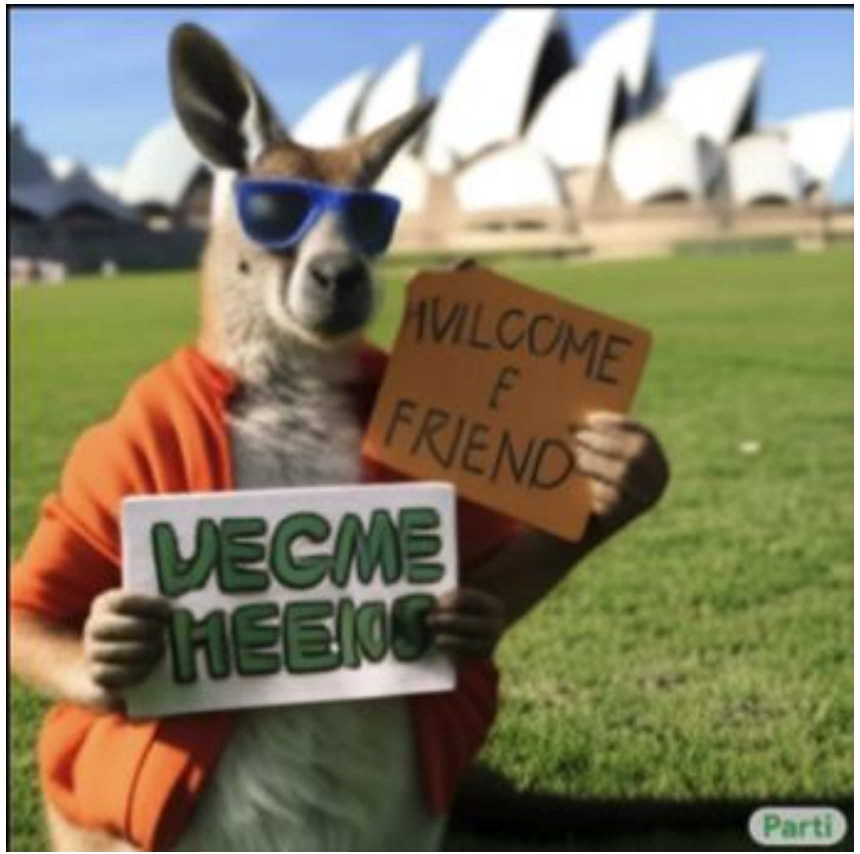
A portrait photo of a kangaroo wearing an orange hoodie and blue sunglasses standing on the grass in front of the Sydney Opera House holding a sign on the chest that says Welcome Friends!



350 million model parameters

Capabilities emerge with self-supervised learning + scale

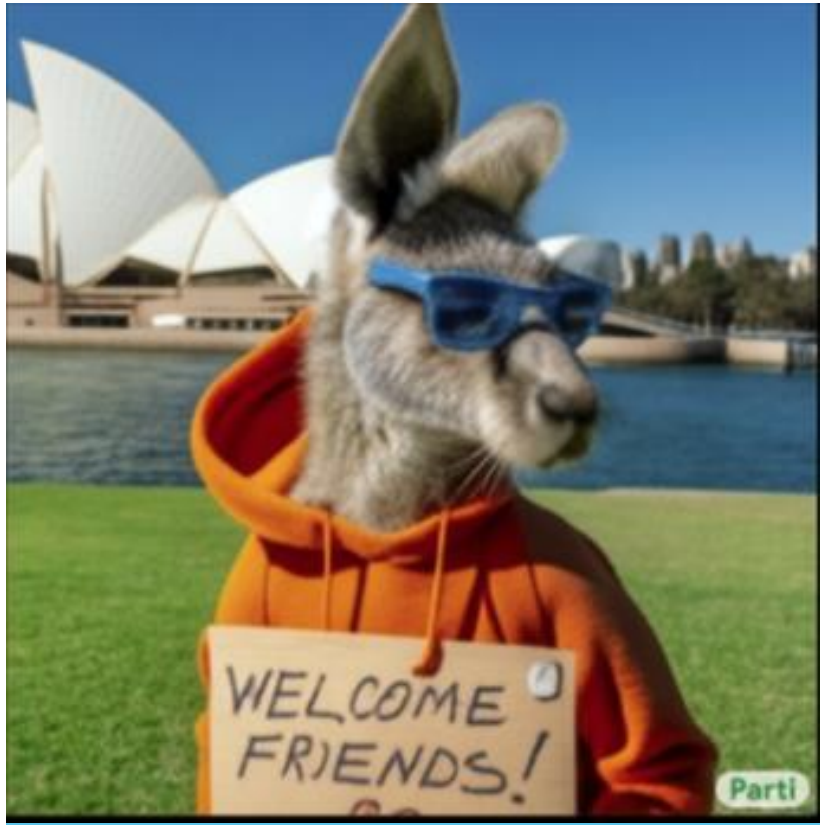
A portrait photo of a kangaroo wearing an orange hoodie and blue sunglasses standing on the grass in front of the Sydney Opera House holding a sign on the chest that says Welcome Friends!



3 billion model parameters

Capabilities emerge with self-supervised learning + scale

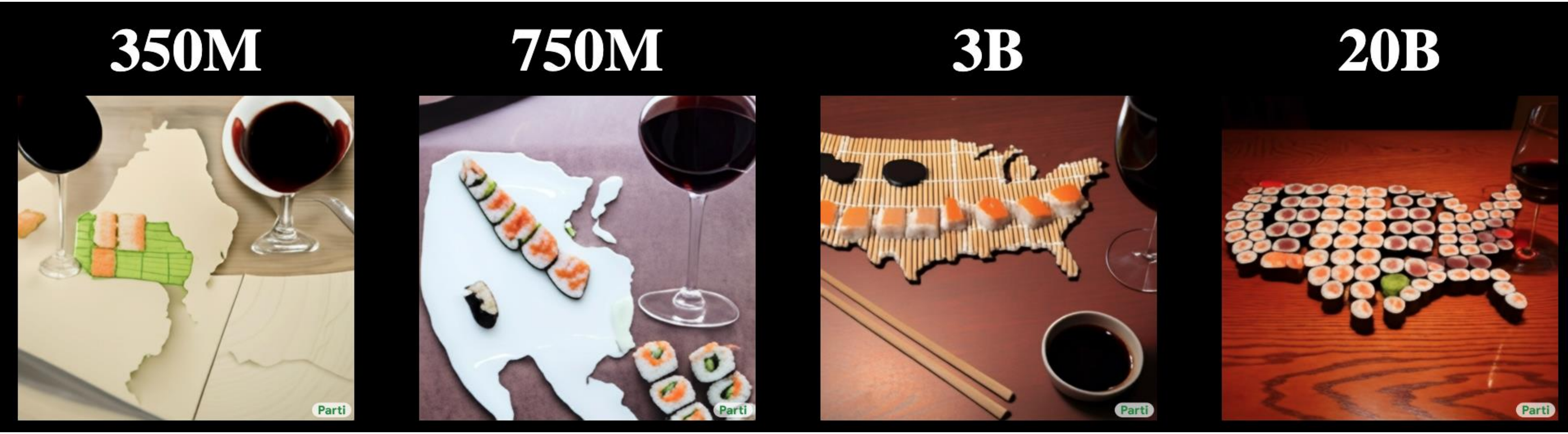
A portrait photo of a kangaroo wearing an orange hoodie and blue sunglasses standing on the grass in front of the Sydney Opera House holding a sign on the chest that says Welcome Friends!



20 billion model parameters

Another example of model scaling

A map of the United States made out of sushi. It is on a table next to a glass of red wine.



Google's "Parti" model in all its 20 billion parameter glory

Toaday!
Parti

A. *A photo of a frog reading the newspaper named "Toaday" written on it. There is a frog printed on the newspaper too.*

B. *A portrait of a statue of the Egyptian god Anubis wearing aviator goggles, white t-shirt and leather jacket. The city of Los Angeles is in the background. Hi-res DSLR photograph.*

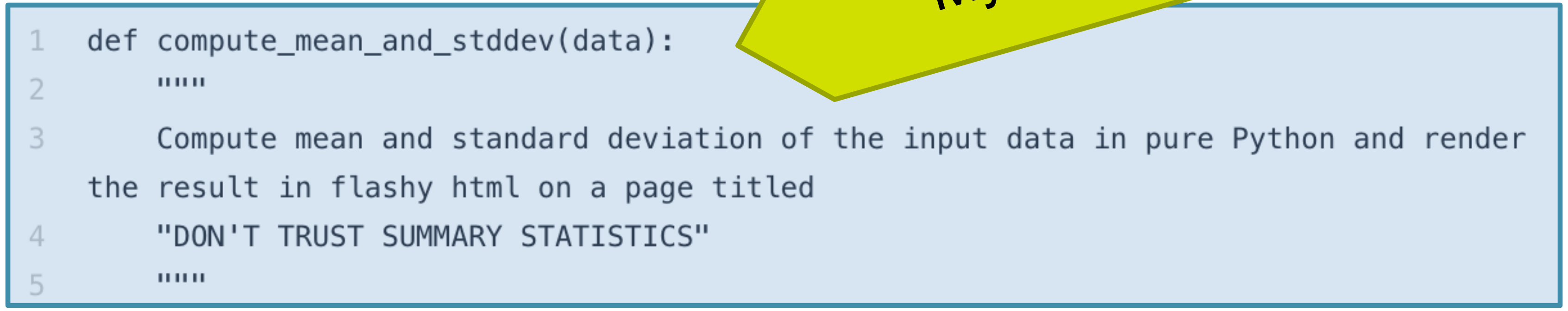
PEACE
Parti

C. *A high-contrast photo of a panda riding a horse. The panda is wearing a wizard hat and is reading a book. The horse is standing on a street against a gray concrete wall. Colorful flowers and the word "PEACE" are painted on the wall. Green grass grows from cracks in the street. DSLR photograph. daytime lighting.*

BE EXCELLENT TO EACH OTHER
BE EXCELLENT TO EACH OTHER..
BE EXCELLENT TO EACH OTHER
Be Excellent to each Other
Parti

Information Classification: General

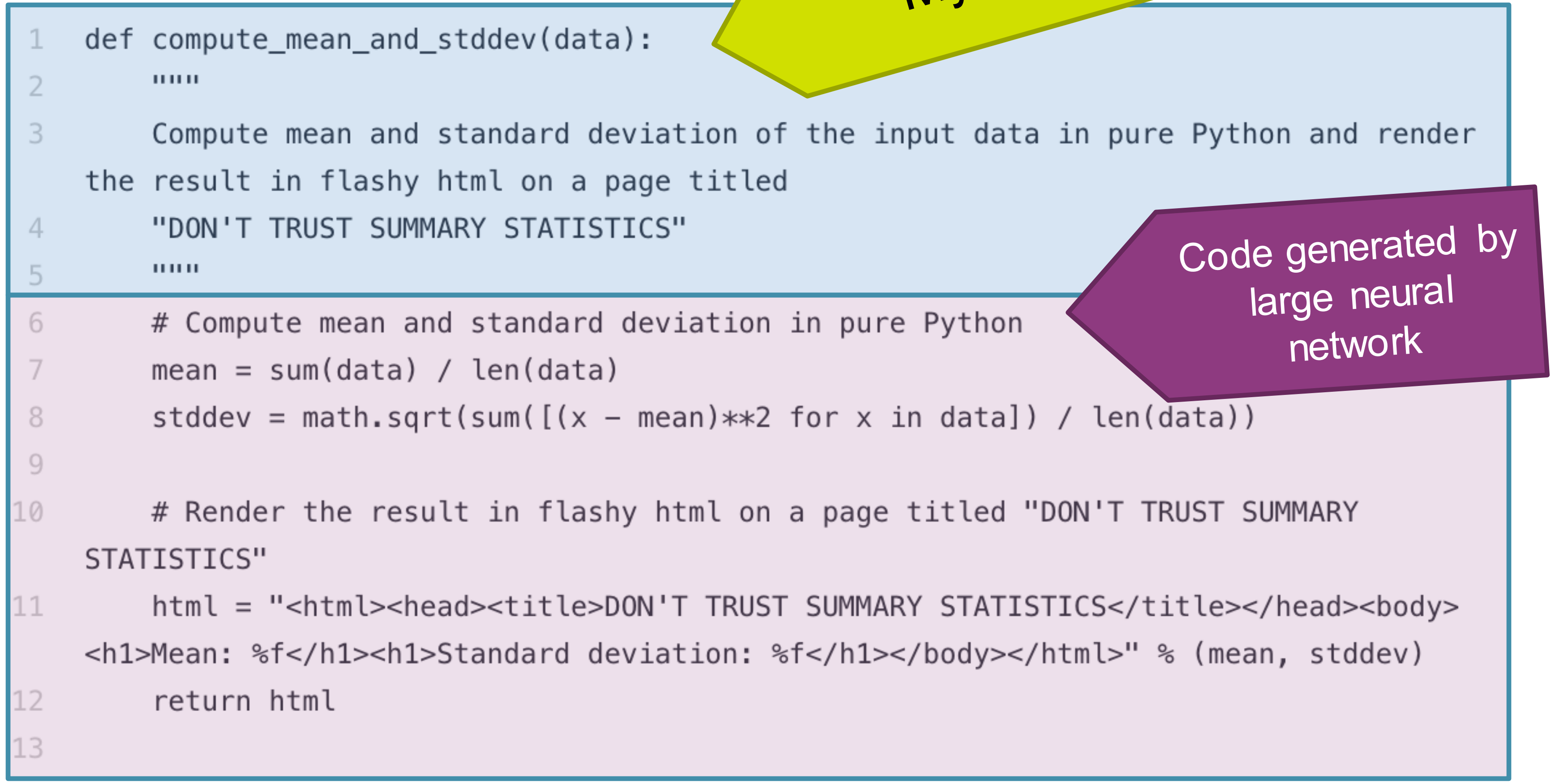
What capability emergence does for images it does for text and code



```
def compute_mean_and_stddev(data):
    """
    Compute mean and standard deviation of the input data in pure Python and render the result in flashy html on a page titled
    "DON'T TRUST SUMMARY STATISTICS"
    """
```

My “prompt”

What capability emergence does for images it does for text and code

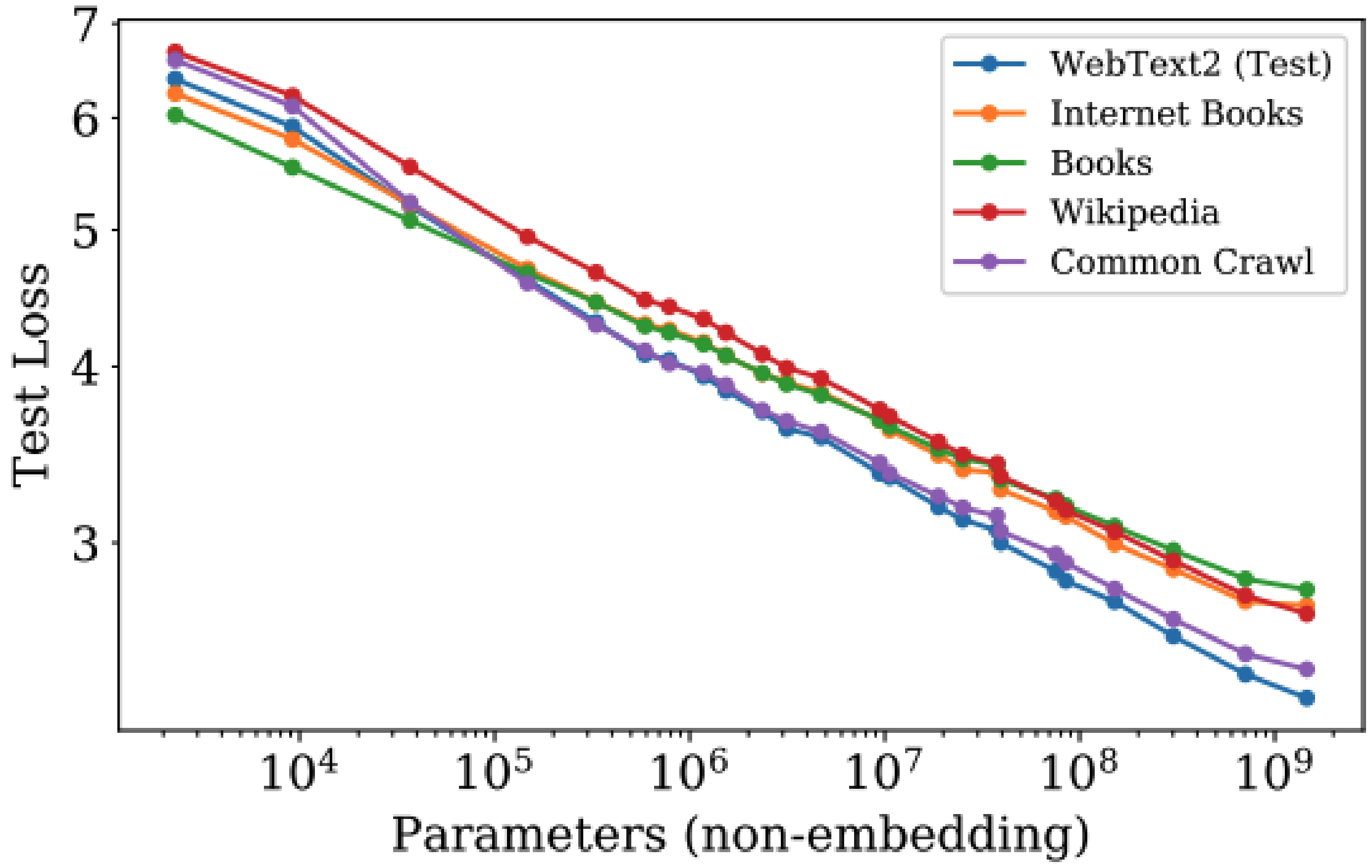


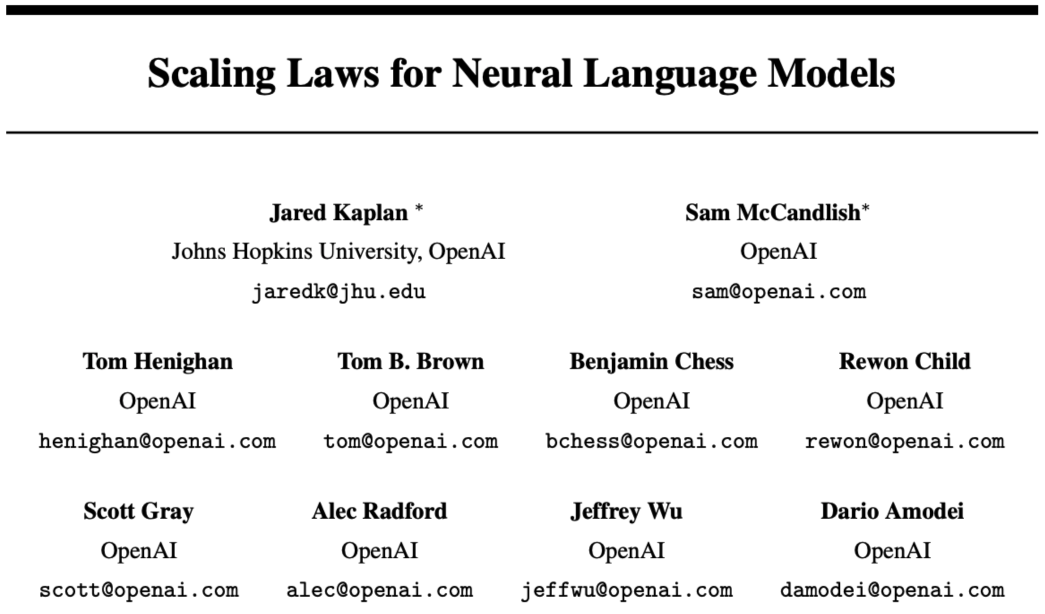
```
def compute_mean_and_stddev(data):
    """
    Compute mean and standard deviation of the input data in pure Python and render the result in flashy html on a page titled
    "DON'T TRUST SUMMARY STATISTICS"
    """
    # Compute mean and standard deviation in pure Python
    mean = sum(data) / len(data)
    stddev = math.sqrt(sum([(x - mean)**2 for x in data]) / len(data))

    # Render the result in flashy html on a page titled "DON'T TRUST SUMMARY STATISTICS"
    html = "<html><head><title>DON'T TRUST SUMMARY STATISTICS</title></head><body><h1>Mean: %f</h1><h1>Standard deviation: %f</h1></body></html>" % (mean, stddev)
    return html

```

My “prompt”

Code generated by large neural network

Neural scaling laws: power-law relationship between scale and accuracy





Scaling Laws for Neural Language Models

Jared Kaplan *
Johns Hopkins University, OpenAI
`jaredk@jhu.edu`

Sam McCandlish*
OpenAI
`sam@openai.com`

Tom Henighan
OpenAI
`henighan@openai.com`

Tom B. Brown
OpenAI
`tom@openai.com`

Benjamin Chess
OpenAI
`bchess@openai.com`

Rewon Child
OpenAI
`rewon@openai.com`

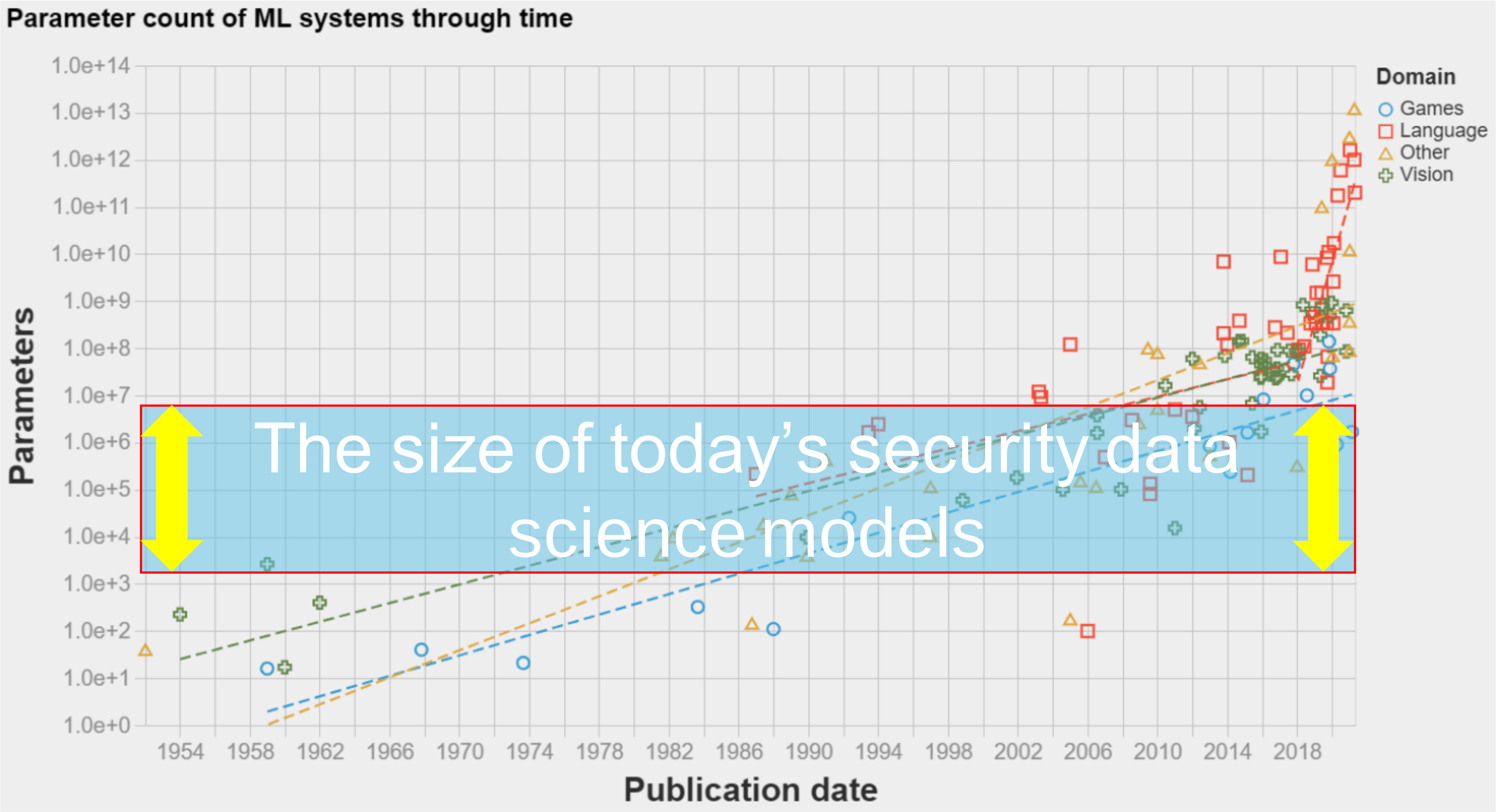
Scott Gray
OpenAI
`scott@openai.com`

Alec Radford
OpenAI
`alec@openai.com`

Jeffrey Wu
OpenAI
`jeffwu@openai.com`

Dario Amodei
OpenAI
`damodei@openai.com`

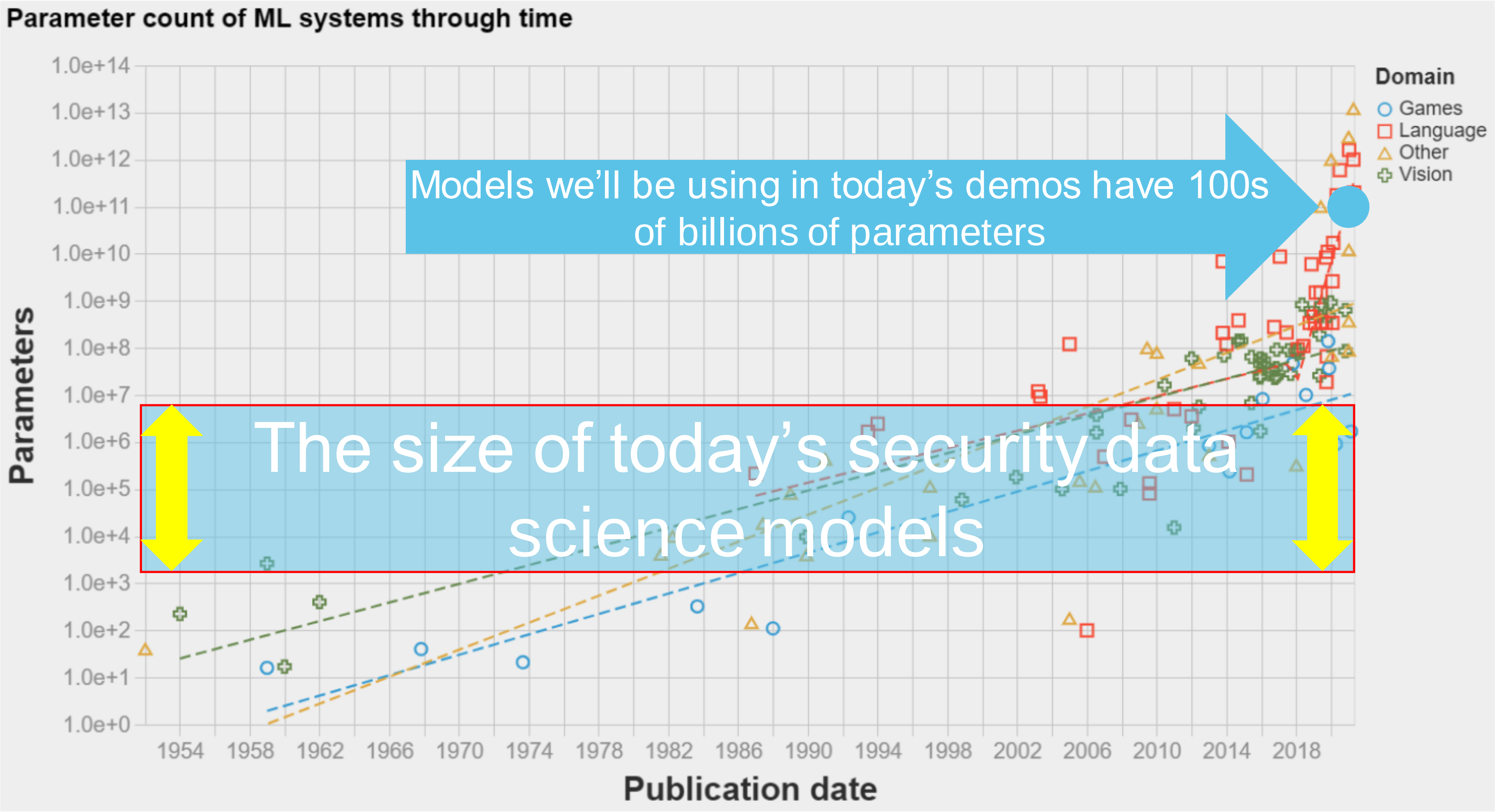
Worth noting: Security machine learning models are still "small" and are not self-supervised



The size of today's security data science models

https://towardsdatascience.com/parameter-counts-in-machine-learning-a312dc4753d0

Worth noting: Security machine learning models are still "small" and are not self-supervised

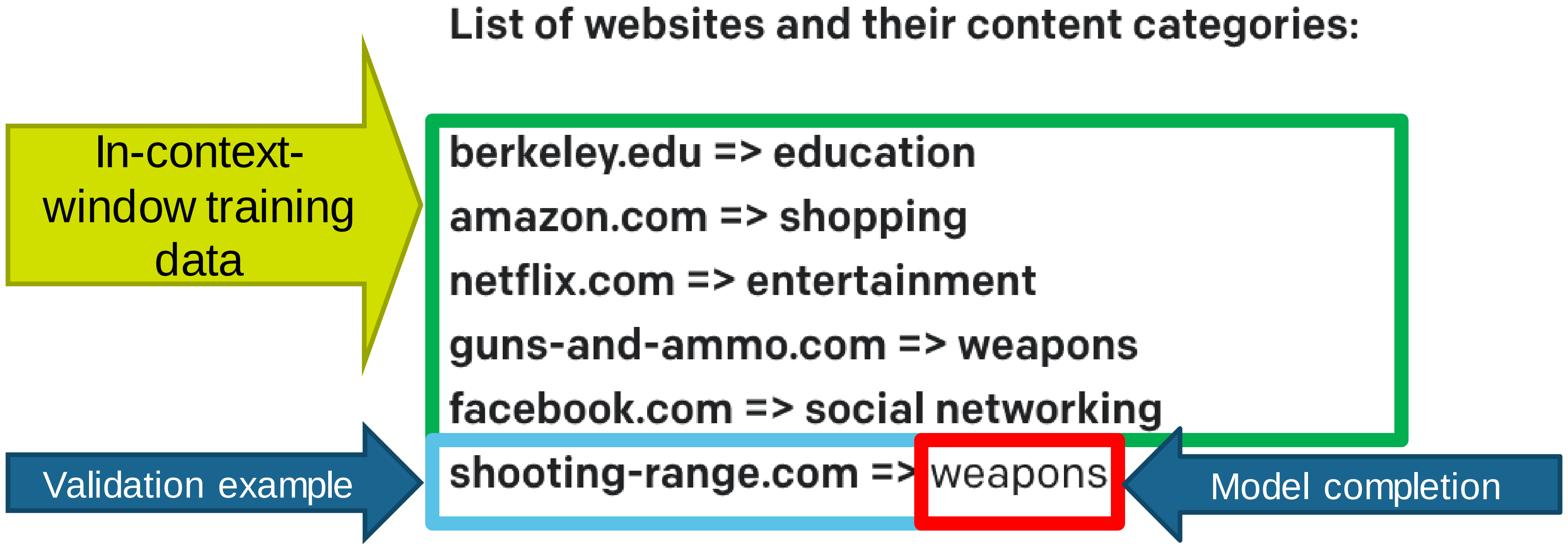


Models we'll be using in today's demos have 100s of billions of parameters

The size of today's security data science models

https://towardsdatascience.com/parameter-counts-in-machine-learning-a312dc4753d0

Why large-scale self-supervised learning matters for detection problems

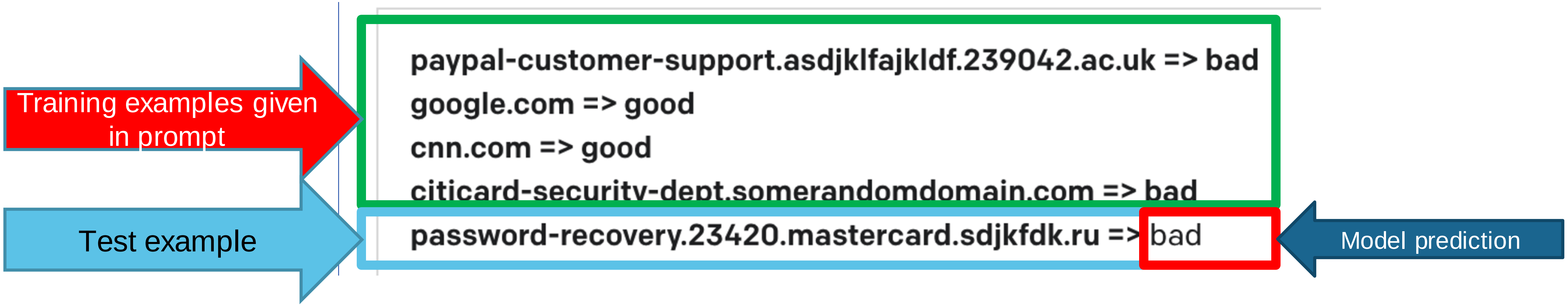


List of websites and their content categories:

berkeley.edu => education
amazon.com => shopping
netflix.com => entertainment
guns-and-ammo.com => weapons
facebook.com => social networking
shooting-range.com => weapons

Source: OpenAI model playground, Davinci model

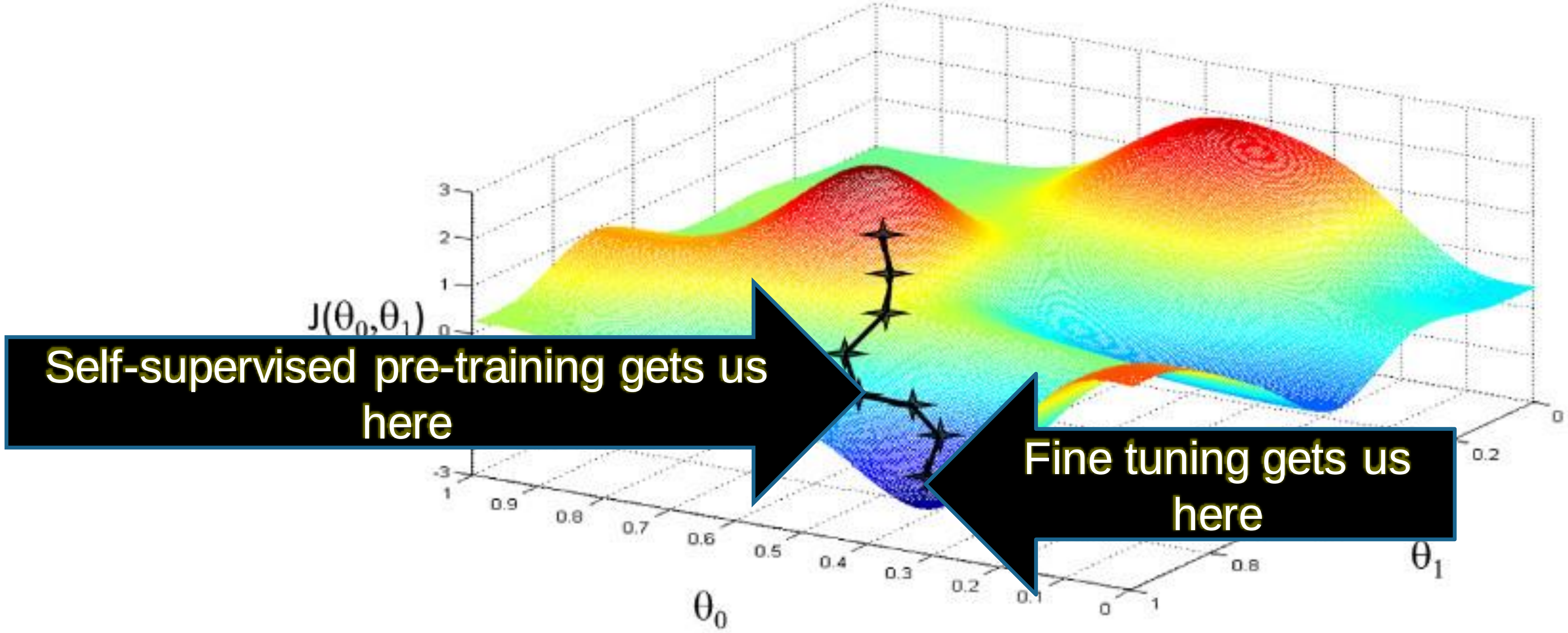
Why large-scale self-supervised learning matters for detection problems



```
paypal-customer-support.asdjklfajkldf.239042.ac.uk => bad
google.com => good
cnn.com => good
citicard-security-dept.somerandomdomain.com => bad
password-recovery.23420.mastercard.sdjkfdk.ru => bad
```

Source: OpenAI model playground, Davinci model

Beyond in-context learning: fine-tuning



Self-supervised pre-training gets us here

Fine tuning gets us here

https://zitaoshen.rbind.io/project/optimization/1-min-of-machine-learning-gradient-decent/

What might the applications of large models be for cybersecurity?

- Detection technology that detects previously unseen attack types better
- User interfaces that understand natural language user requests from security operators
- ”Auto-complete on steroids” models for DevSecOps
- Models that aid reverse engineering

Information Classification: General

black hat USA 2022

Spam detection with GPT-3

What if you have only a few examples of a new spam campaign?

Small scale machine learning with no self-supervision needs more training data to learn new tricks!

https://www.istockphoto.com/photos/pile-of-dog-treats

Large scale self-supervised models need far less training data to learn new tricks!

https://www.youtube.com/watch?v=tyunqxGGD9k

GPT-3 learns to detect spam using a tiny number of training examples, because of self-supervised pretraining derived “background knowledge”

		GPT-3 F1-score	Random Forest F1-score
Training samples	1 ham, 1 spam	**0.90**	0.55
	4 ham, 4 spam	**0.95**	0.79

We would need far more data to learn to detect a spam campaign using traditional methods!

Information Classification: General

Evaluation data and models

Data source

- Training and test datasets were randomly sampled from the following Spam dataset.
- https://archive.ics.uci.edu/ml/datasets/sms+spam+collection

Training datasets

- Sample size for 2, 8, 32, 512, 1024

Test dataset

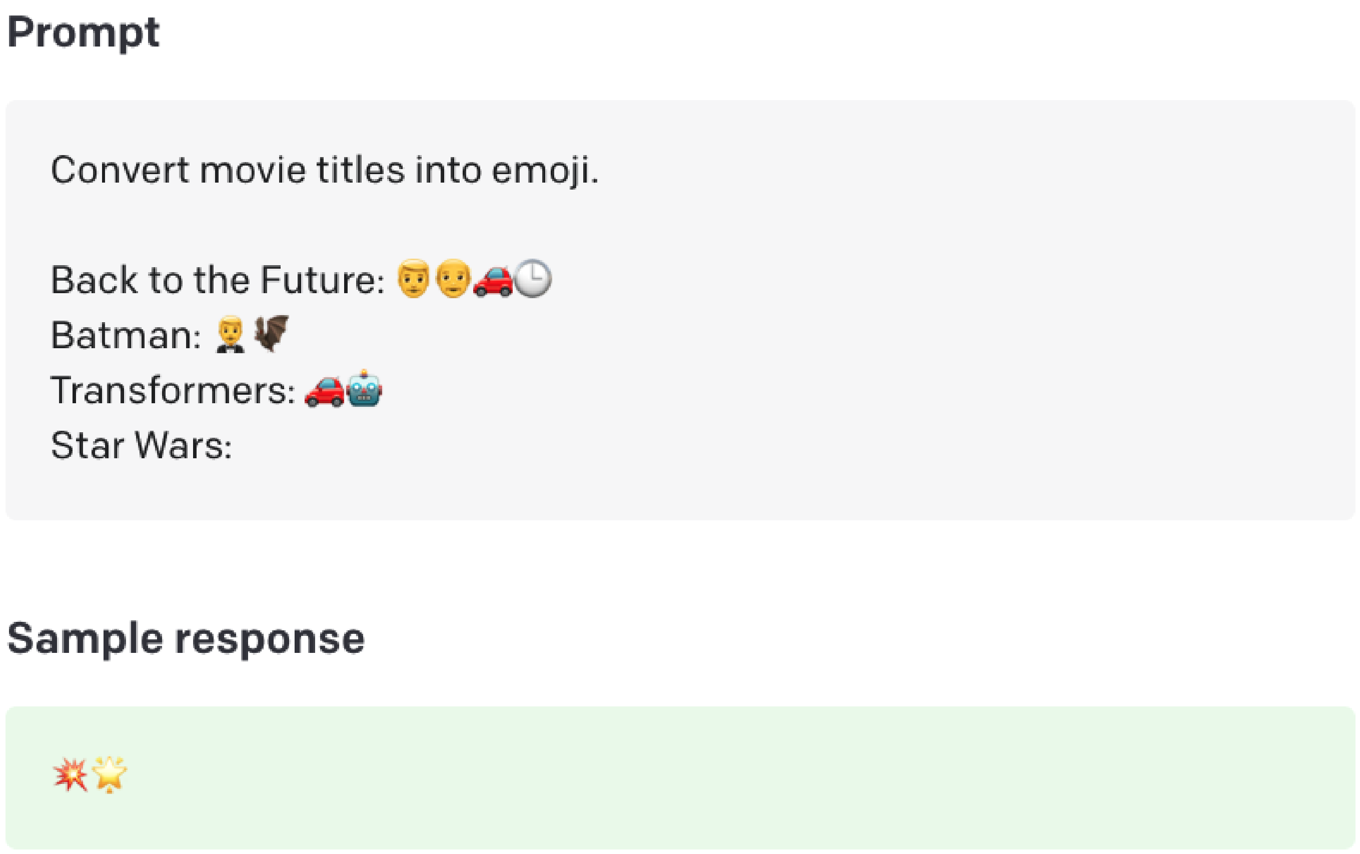
- Sample size for 256

ML models

- Tree-based Random Forest model with token frequency-based ML features.
- GPT-3’s Davinci model with few-shot learning.

Information Classification: General

Teaching GPT-3 to solve a prediction problems can be as simple as designing a prompt for it to auto-complete



Prompt

Convert movie titles into emoji.

Back to the Future: 👨👴🚗🕒
Batman: 🤵🦇
Transformers: 🚗🤖
Star Wars:

Sample response

💥🌟

”Prompt” showing GPT-3 examples of what you want

GPT-3’s “auto-complete” response for Star Wars

https://beta.openai.com/docs/guides/completion/prompt-design

Prompting GPT-3 to detect spam

## Prompt: Input data

Classify the Message as Spam or Ham.

1. The part of the prompt where we tell GPT-3 what we want it to do

Message: FreeMSG You have been awarded . . .

Label: Spam

Message: Going to join tomorrow

Label: Ham

2. The part of the prompt where we show GPT-3 some training examples

Message: Free Top ringtone -sub to weekly

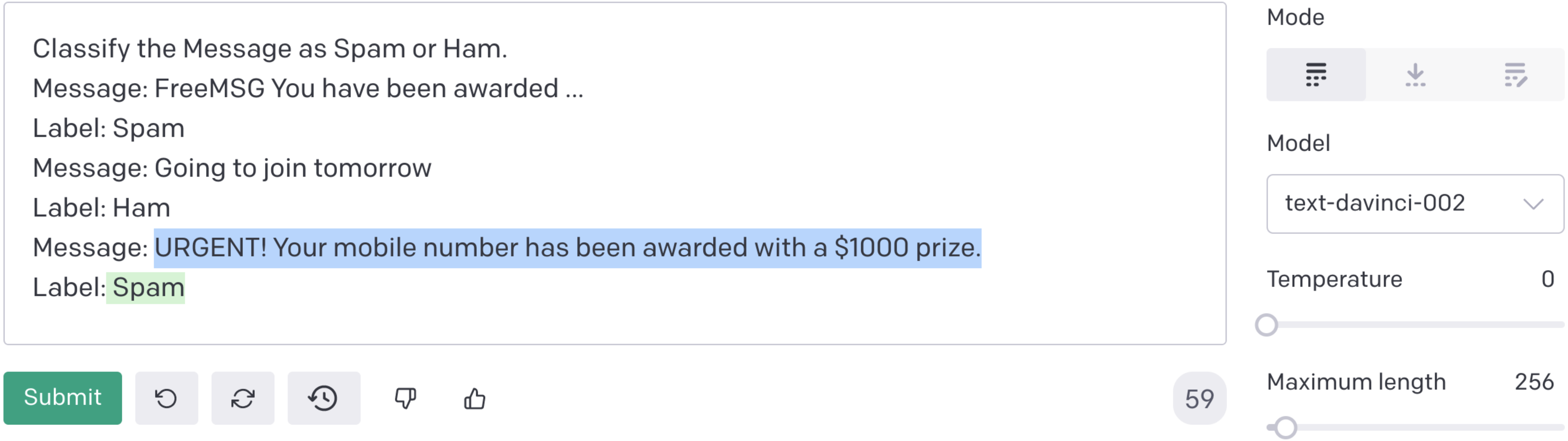
Label:

3. The part of the prompt where we ask GPT-3 to classify our target example

Spam

4. GPT-3's correct classification of the new message!

Examples of Spam detection



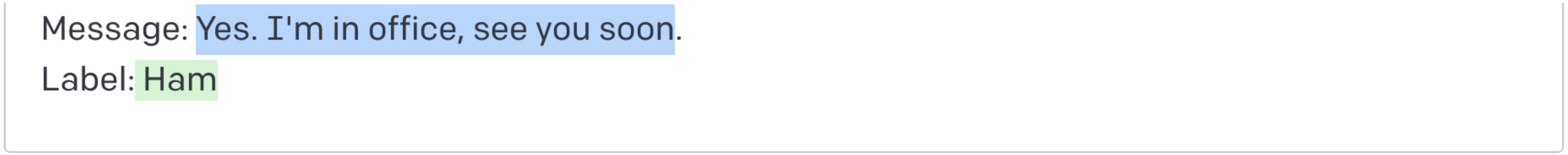
Classify the Message as Spam or Ham.
Message: FreeMSG You have been awarded ...
Label: Spam
Message: Going to join tomorrow
Label: Ham
Message: URGENT! Your mobile number has been awarded with a $1000 prize.
Label: Spam

Mode

Model
text-davinci-002

Temperature 0

Maximum length 256



Message: Yes. I'm in office, see you soon.
Label: Ham

Information Classification: General

black hat USA 2022

Generating human-readable explanations of malicious command lines

Analyzing thousands of suspicious commands over a large enterprise is exhausting for SOC analysts.

Research question: Can large language models make this easier by describing them in simple language?

Real-world suspicious command line. Our goal: automatically describe this in simple language to reduce cognitive load

```
"C:\\WINDOWS\\system32\\cmd.exe /Q /c echo dir \"C:\\Users\\admin\\OneDrive -
ADMINISTRATORS INC\" ^> \\\\127.0.0.1\\C$\\__output 2^>^&1 >
C:\\WINDOWS\\TEMP\\execute.bat & C:\\WINDOWS\\system32\\cmd.exe /Q /c
C:\\WINDOWS\\TEMP\\execute.bat & del C:\\WINDOWS\\TEMP\\execute.bat
```

Information Classification: General

Takeaway up front: GPT-3 can often do a good job at this task

## Command

```
C:\\WINDOWS\\system32\\cmd.exe /Q /c echo dir \"C:\\Users\\admin\\OneDrive ADMINISTRATORS INC\" ^>
\\\\127.0.0.1\\C$\\__output 2^>^&1 > C:\\WINDOWS\\TEMP\\execute.bat &
C:\\WINDOWS\\system32\\cmd.exe /Q /c C:\\WINDOWS\\TEMP\\execute.bat & del
C:\\WINDOWS\\TEMP\\execute.bat
```

Real suspicious command observed on a real network!

## Description

The command will create a file called \"execute.bat\" in the C:\\WINDOWS\\TEMP folder. It will then run the command \"dir C:\\Users\\admin\\OneDrive ADMINISTRATORS INC\" and output the results to the __output file on the local machine. The batch file will then execute itself and delete itself afterwards.

Real, accurate description generated by GPT-3!

Information Classification: General

"GPT-3" actually refers to two types of OpenAI large language models

https://www.digitaltrends.com/computing/openai-gpt-3-text-generation-ai/

Text-GPT3 for text generation

These models can write interesting stories and poems.

https://www.datasciencecentral.com/open-ai-codex-challenge-seen-by-the-participants/

Code-GPT3(Codex) for code generation

These models are proficient in over a dozen of languages including Python, JavaScript, PHP and even Shell.

We use "prompts" to get GPT-3 to auto-complete us a good command line description

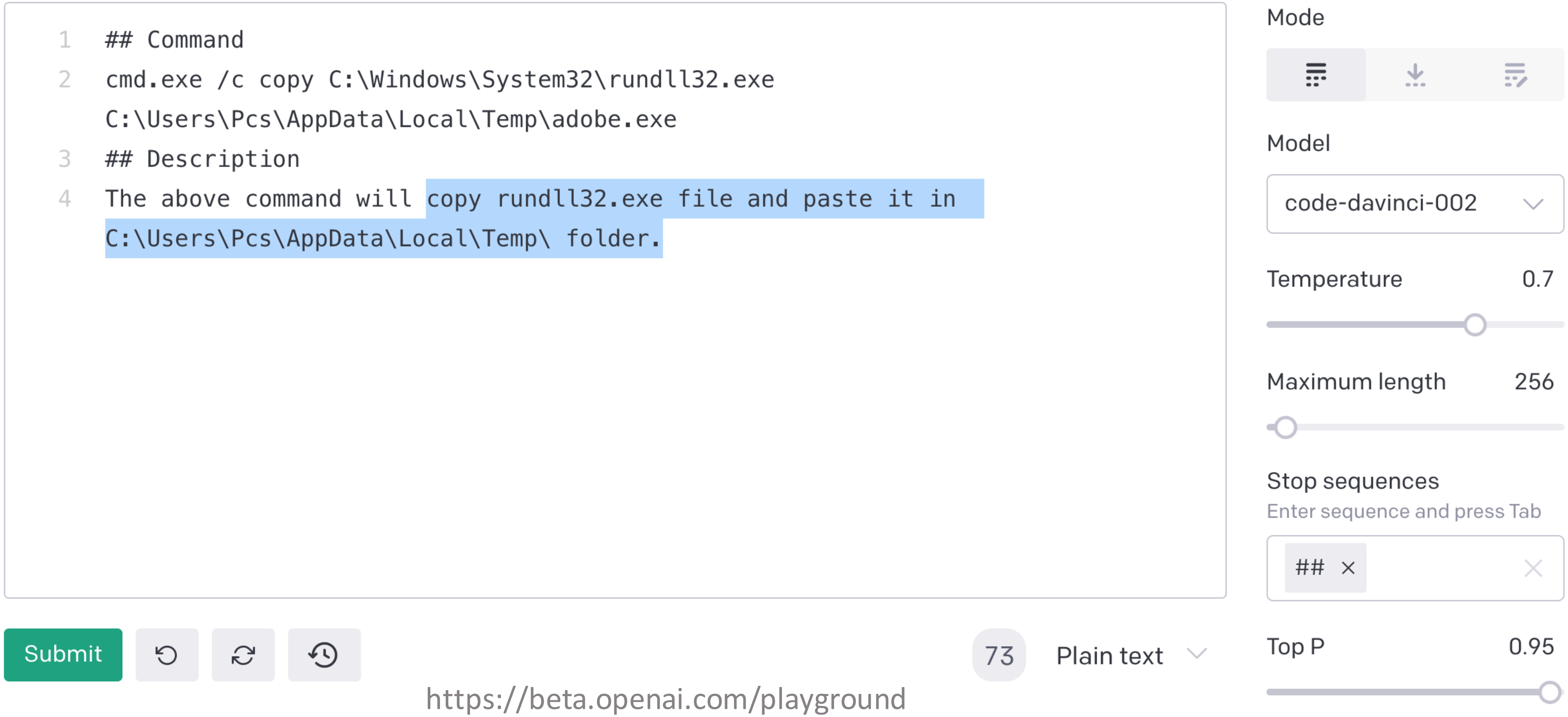
```
## Command

cmd.exe /c copy C:\Windows\System32\rundll32.exe C:\Users\Pcs\AppData\Local\Temp\adobe.exe

## Description

The above command will
```

code

Here's where GPT-3 begins auto-completing, hopefully with an accurate description

Information Classification: General

An example auto-completion that describes a command line



```
## Command
cmd.exe /c copy C:\Windows\System32\rundll32.exe
C:\Users\Pcs\AppData\Local\Temp\adobe.exe
## Description
The above command will copy rundll32.exe file and paste it in
C:\Users\Pcs\AppData\Local\Temp\ folder.
```

Mode

Model: code-davinci-002

Temperature: 0.7

Maximum length: 256

Stop sequences: ##

Top P: 0.95

https://beta.openai.com/playground

Improving command line descriptions by using signature match context data

SIGMA

Sigma

Generic Signature Format for SIEM Systems

https://github.com/SigmaHQ/sigma

yara

The pattern matching swiss knife for malware researchers

https://virustotal.github.io/yara/

cmd.exe /c copy C:\Windows\System32\rundll32.exe C:\Users\Pcs\AppData\Local\Temp\adobe.exe => was detected by win_suspicious_copy_system32 signature.

Suspicious commands can be detected by signature-based rules and the signature names often provide useful context information.

Better prompts by adding in signature match information

```
## Command

cmd.exe /c copy C:\Windows\System32\rundll32.exe
C:\Users\Pcs\AppData\Local\Temp\adobe.exe

## Tags

win_suspicious_copy_system32

## Description

The above command will
```

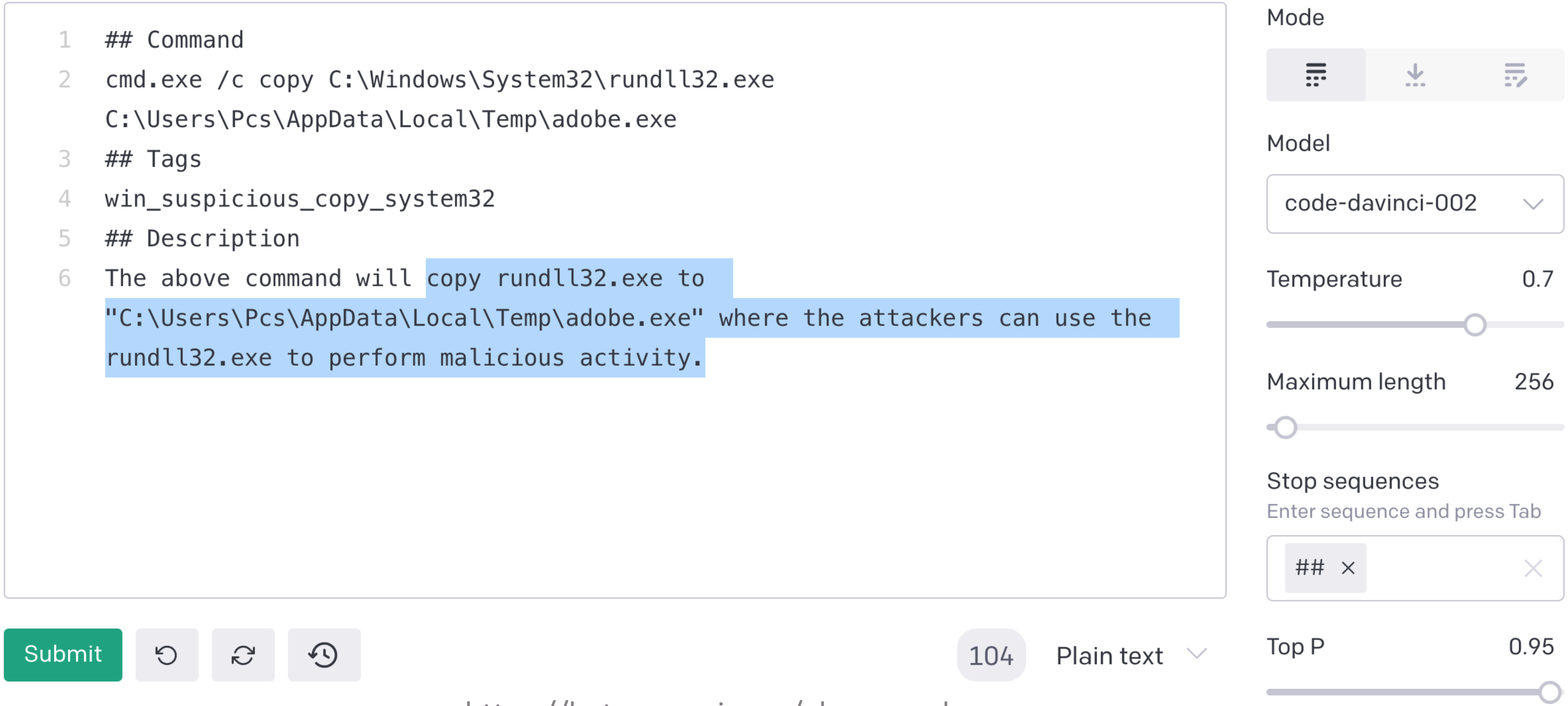
code

tags

description

Information Classification: General

Description from a command and tags



```
## Command
cmd.exe /c copy C:\Windows\System32\rundll32.exe C:\Users\Pcs\AppData\Local\Temp\adobe.exe
## Tags
win_suspicious_copy_system32
## Description
The above command will copy rundll32.exe to "C:\Users\Pcs\AppData\Local\Temp\adobe.exe" where the attackers can use the rundll32.exe to perform malicious activity.
```

Mode

Model: code-davinci-002

Temperature: 0.7

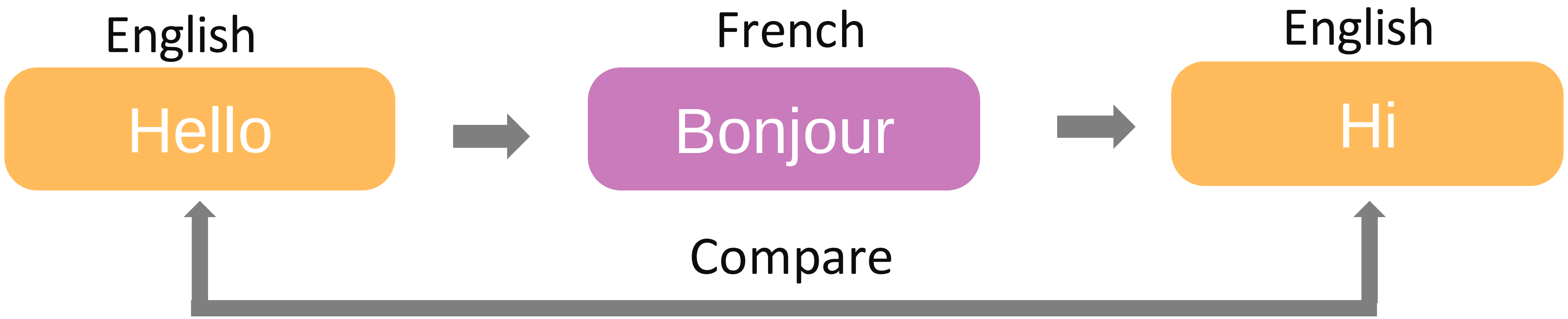
Maximum length: 256

Stop sequences: ##

Top P: 0.95

https://beta.openai.com/playground

Our clever idea: evaluating descriptions for quality using "back-translation"



English

Hello

French

Bonjour

English

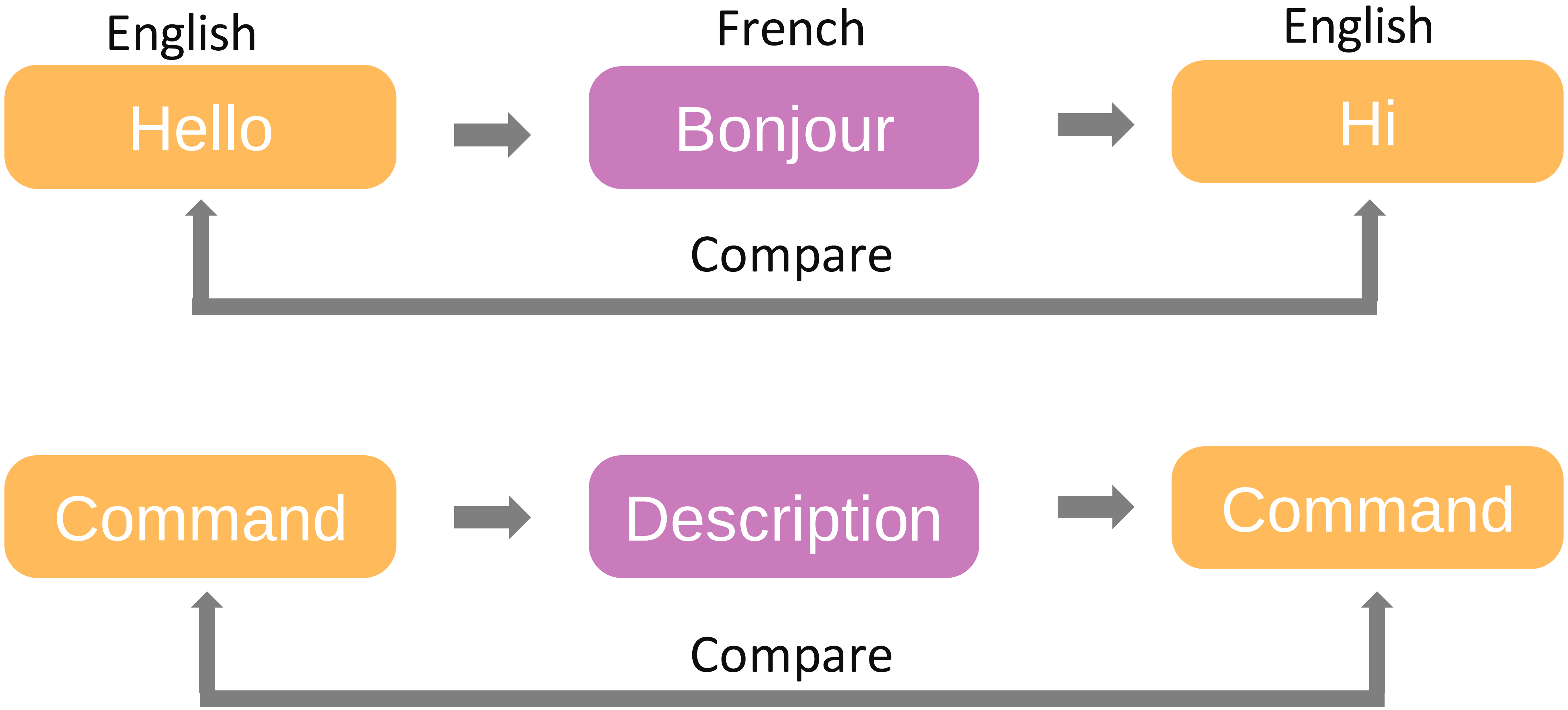
Hi

Compare

Language translation task from English to French

Information Classification: General

Our clever idea: evaluating descriptions for quality using “back-translation”



English

Hello

French

Bonjour

English

Hi

Compare

Command

Description

Command

Compare

Language translation task from Command to Description

Steps to choose the best description

Generating descriptions

- input: a command, tags

Back-translating commands

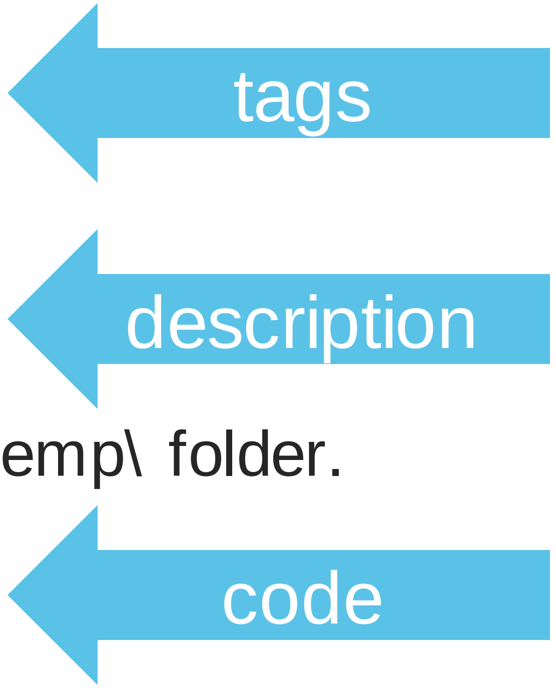
- input: a description, tags

Ranking descriptions by similarity scores

- input: the query and back-translated commands

Prompt design for Back-translation

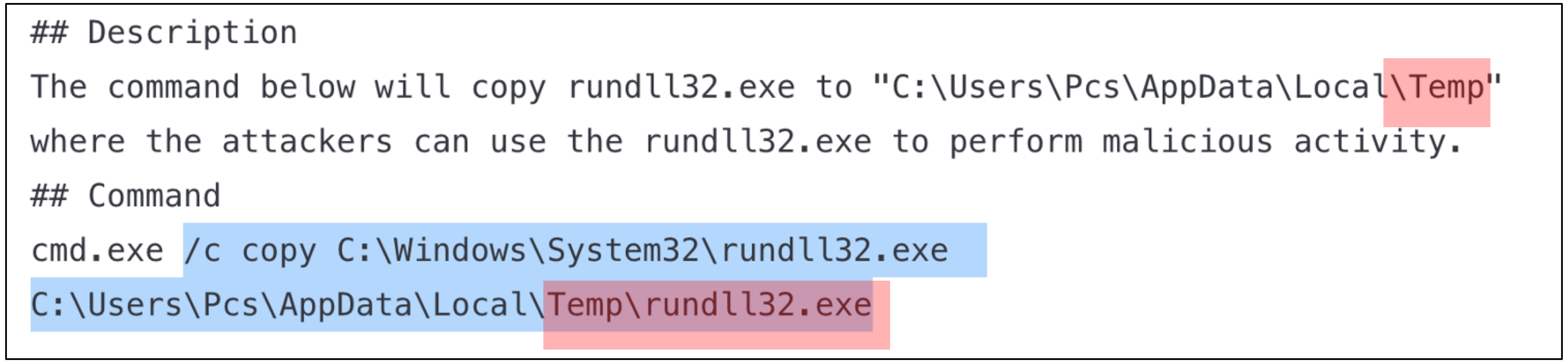
```
## Tags
win_suspicious_copy_system32
## Description
The command below will copy the rundll32.exe file from the
C:\Windows\System32 folder to the C:\Users\PCs\AppData\Temp\ folder.
## Command
cmd.exe
```



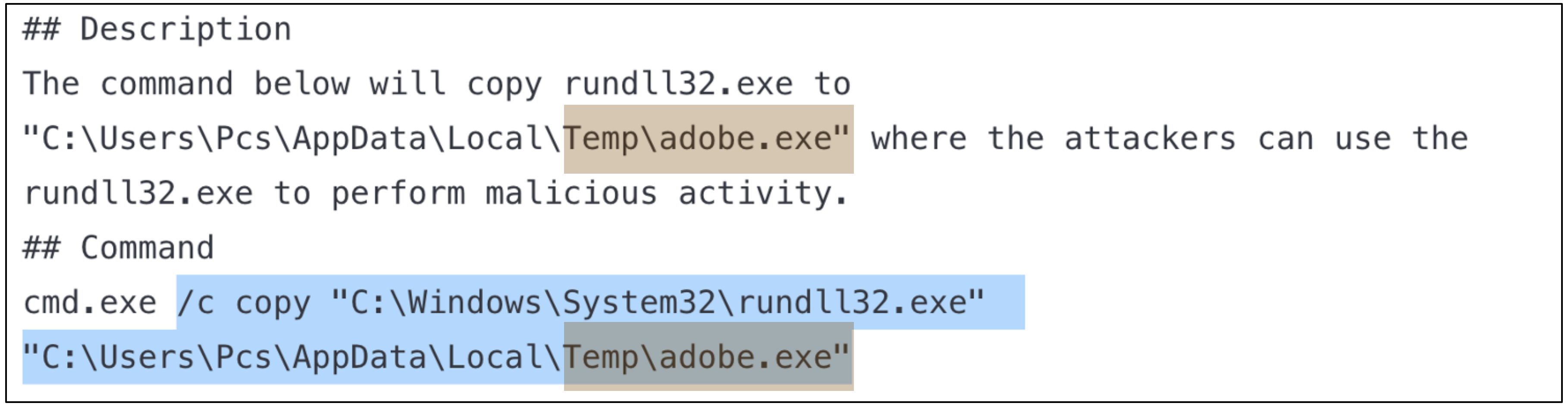
The first word, cmd.exe in Command is provided from the query code to guide GPT-3 to generate relevant code.

Information Classification: General

Improved description with Back-translation



```
## Description
The command below will copy rundll32.exe to "C:\Users\Pcs\AppData\Local\Temp"
where the attackers can use the rundll32.exe to perform malicious activity.
## Command
cmd.exe /c copy C:\Windows\System32\rundll32.exe
C:\Users\Pcs\AppData\Local\Temp\rundll32.exe
```



```
## Description
The command below will copy rundll32.exe to
"C:\Users\Pcs\AppData\Local\Temp\adobe.exe" where the attackers can use the
rundll32.exe to perform malicious activity.
## Command
cmd.exe /c copy "C:\Windows\System32\rundll32.exe"
"C:\Users\Pcs\AppData\Local\Temp\adobe.exe"
```

GPT-3 can generate multiple descriptions but the second one is better as it has higher information coverage. Back-translation step allows us to choose the best candidate.

Information Classification: General

Description from Command and Tags

## Command

```
cmd.exe dir /b /s \"C:\\Users\\Pcs\\Desktop\\*.*\" | findstr /i password
```

## Tags

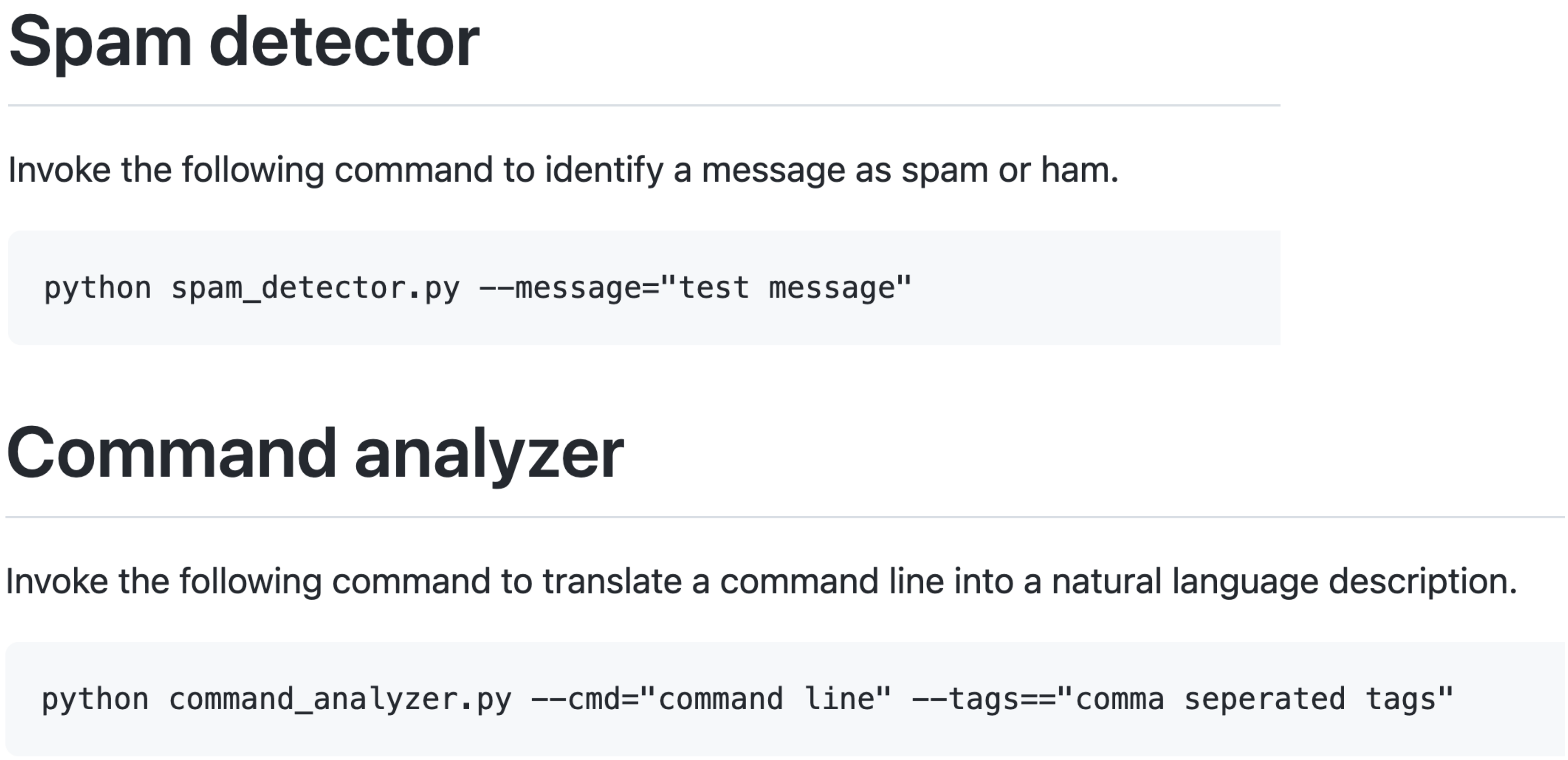
win_pc_suspicious_dir,win_suspicious_findstr

## Description

The command will recursively list all files in the "C:\Users\Pcs\Desktop" directory and all subdirectories and will search the output for files containing the word "password".

Source code

Source code is available in https://github.com/sophos/gpt3-and-cybersecurity.



Spam detector

Invoke the following command to identify a message as spam or ham.

```
python spam_detector.py --message="test message"
```

Command analyzer

Invoke the following command to translate a command line into a natural language description.

```
python command_analyzer.py --cmd="command line" --tags=="comma seperated tags"
```

Q&A