https://sebastianraschka.com
@rasbt

#ODSC

Machine Learning in 2021

Recent Trends, Technologies, and Challenges

Sebastian Raschka

About Myself

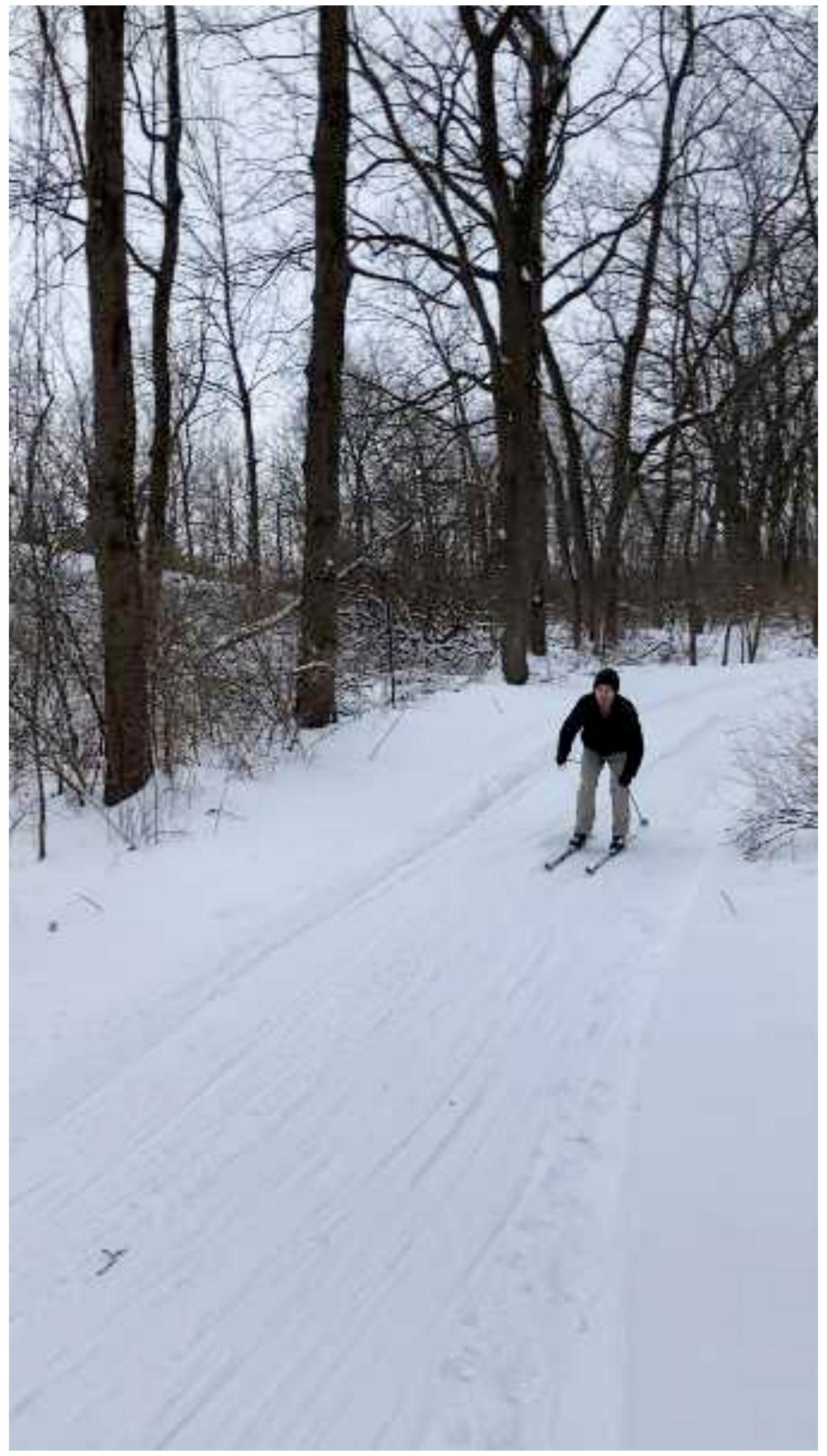
Contact:

https://sebastianraschka.com

@rasbt

Specialties:

- Python
- Machine learning
- Deep learning
- Wisconsin State Parks



Affiliation:

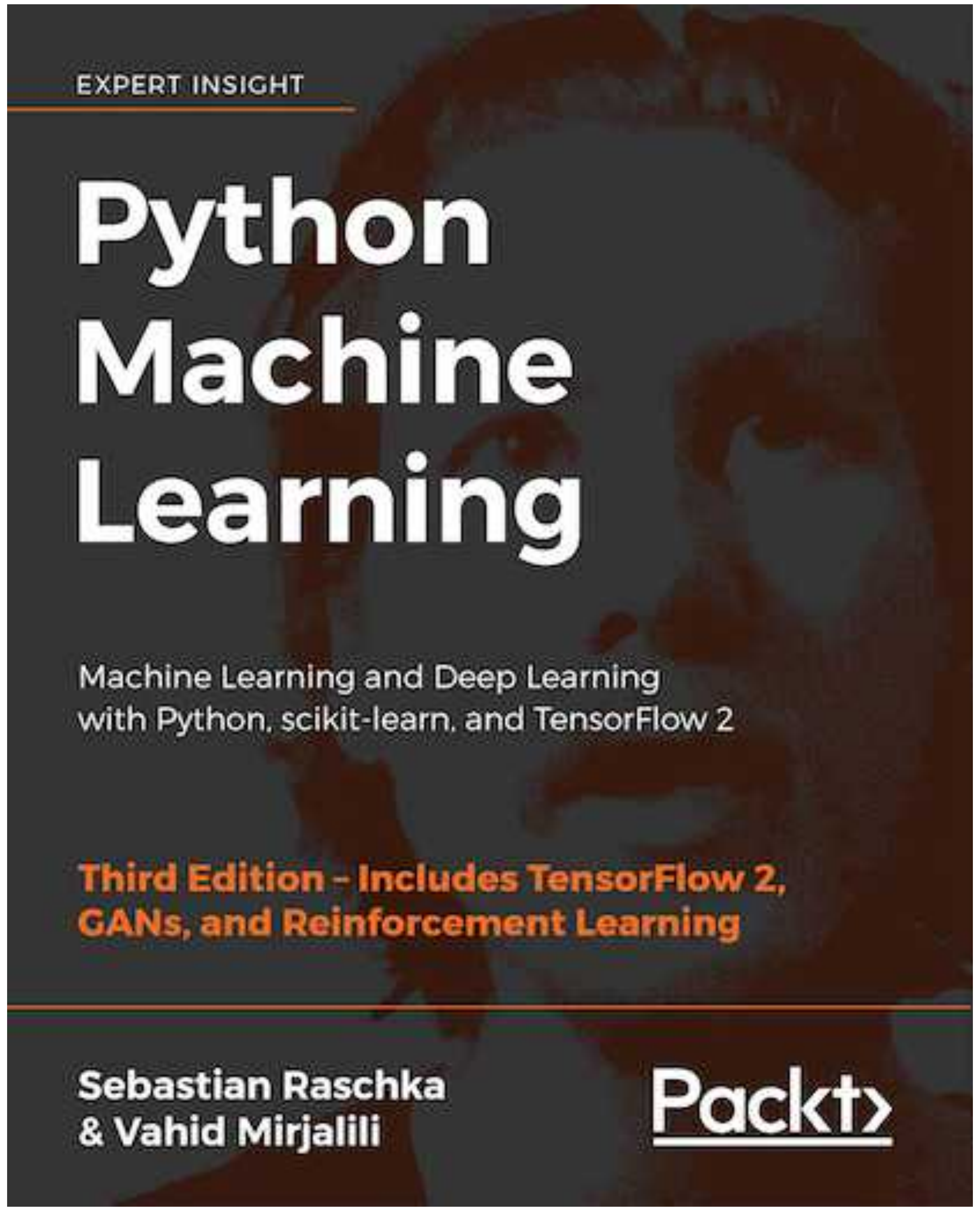
Assistant Professor

Department of Statistics

https://stat.wisc.edu

Meet the Speaker and Book Session

Today 2:00 p.m. - 2:45 p.m. EDT



Topics

(1) Technologies
- Hardware
- Deep Learning Frameworks
- Programming Languages

(2) Challenges
- Small data
- Ordinal data
- Adversarial attacks
- Bias
- Privacy

(3) Research Trends
- Graph neural nets
- GANs
- Self-supervised learning
- Language transformers
- Vision transformers

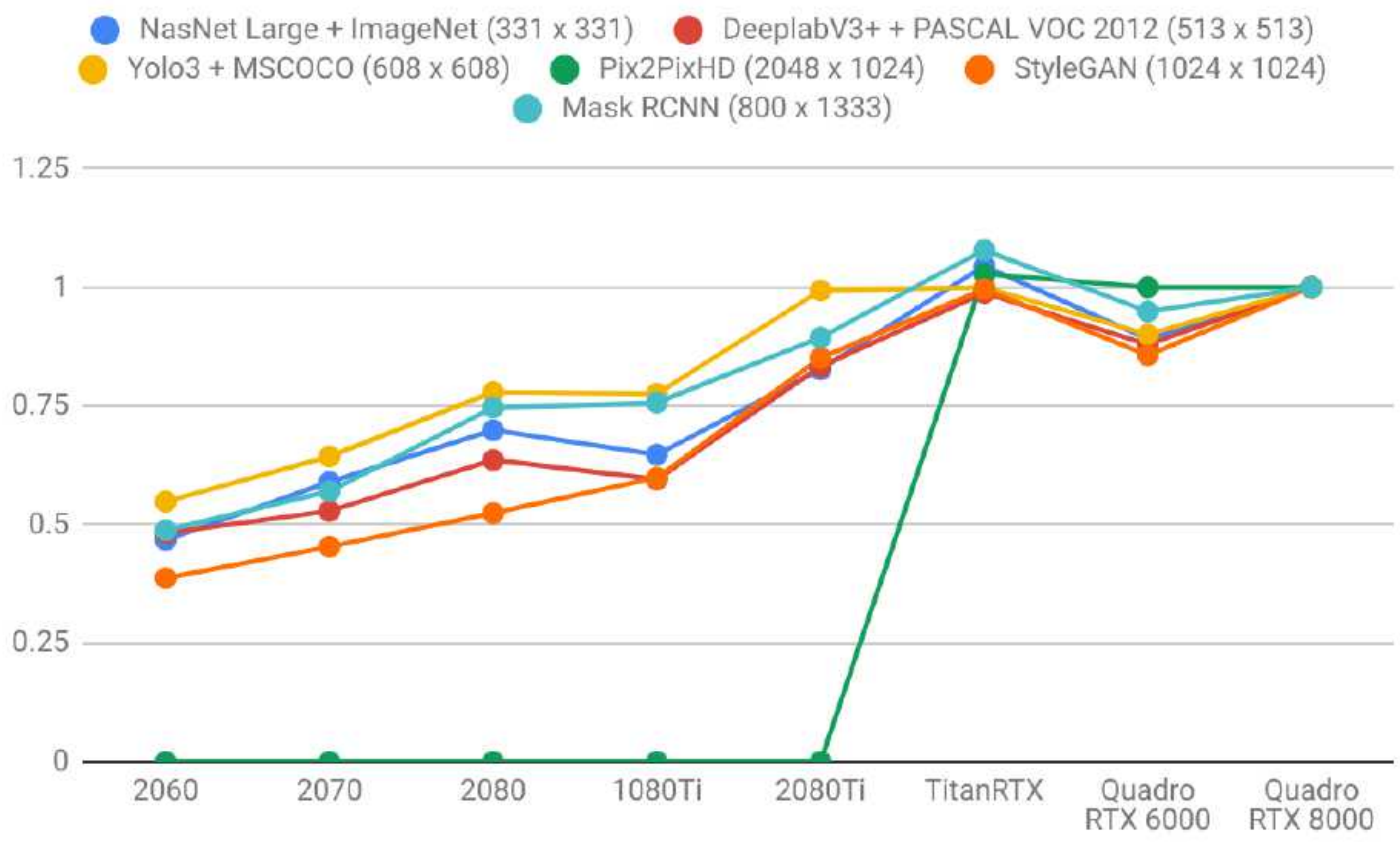
(1) Technologies

Hardware

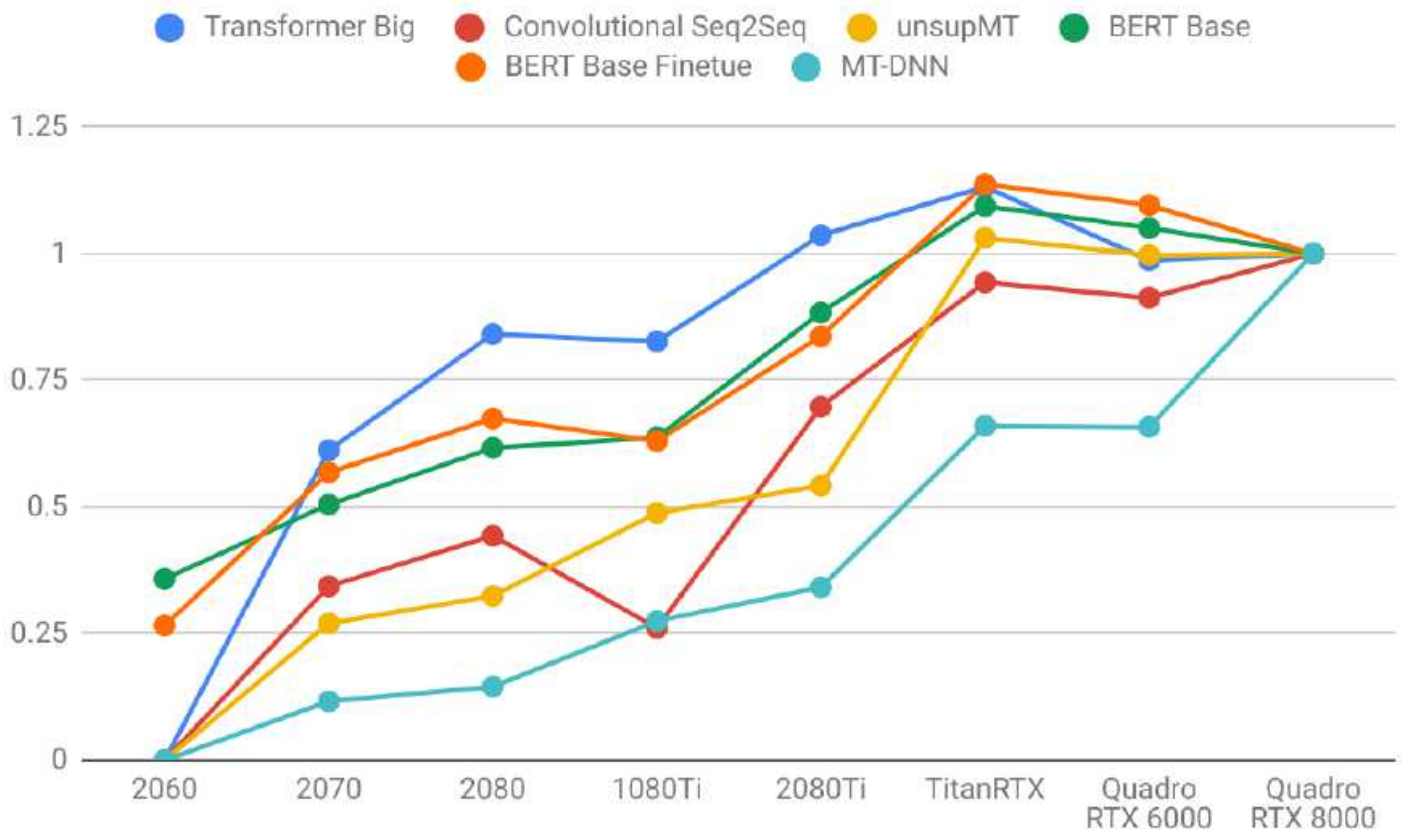
Deep Learning Frameworks

Programming Languages

GPUs for Deep Learning Continue to Improve



Computer vision models



Language models

Source: https://lambdalabs.com/blog/choosing-a-gpu-for-deep-learning

Beyond Words/Images Per Second: Batch Size Matters, Too

Image models

Maximum batch size before running out of memory

Model / GPU	2060	2070	2080	1080 Ti	2080 Ti	Titan RTX	RTX 6000	RTX 8000
NasNet Large	4	8	8	8	8	32	32	64
DeepLabv3	2	2	2	4	4	8	8	16
Yolo v3	2	4	4	4	4	8	8	16
Pix2Pix HD	0*	0*	0*	0*	0*	1	1	2
StyleGAN	1	1	1	4	4	8	8	16
MaskRCNN	1	2	2	2	2	8	8	16

The GPU does not have enough memory to run the model.

Source: https://lambdalabs.com/blog/choosing-a-gpu-for-deep-learning

Traditionally: Use GPUs for (Gaming and) Deep Learning

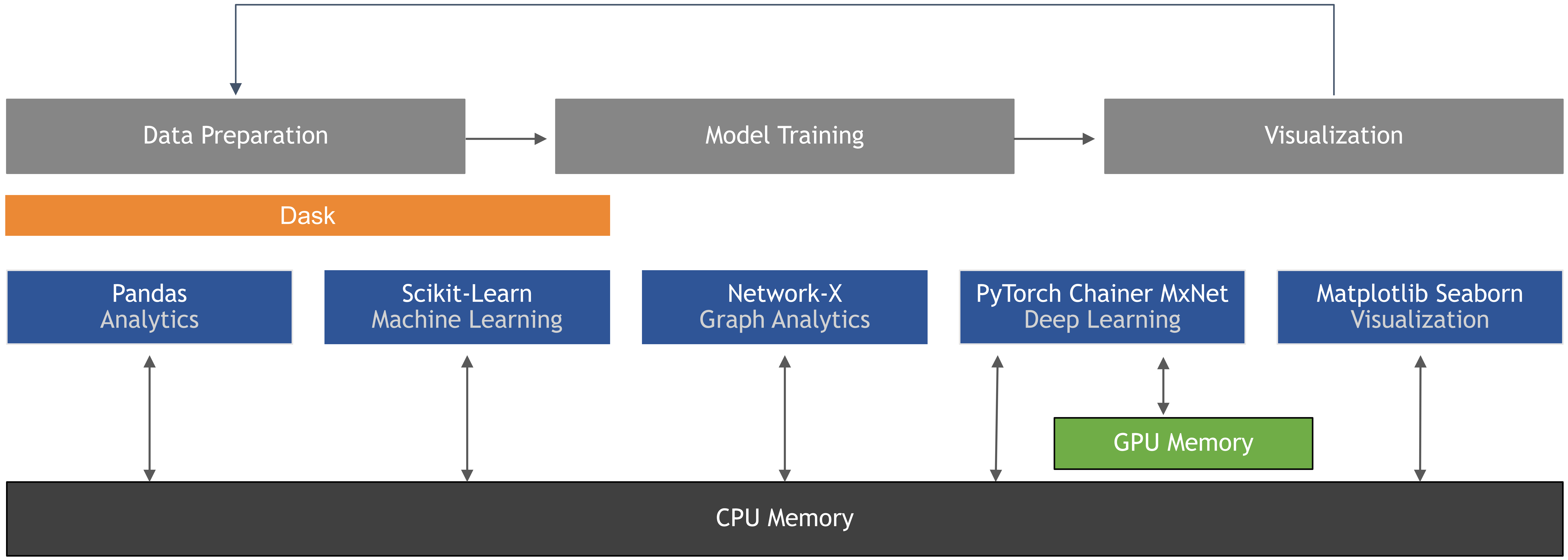


Figure 1. The standard Python ecosystem for machine learning, data science, and scientific computing.

Sebastian Raschka, Joshua Patterson, and Corey Nolet (2020)
Machine Learning in Python: Main developments and technology trends in data science, machine learning, and artificial intelligence
Information 2020, 11, 193

Today: Use GPUs for All ML & Data Science (and Bitcoin Mining)

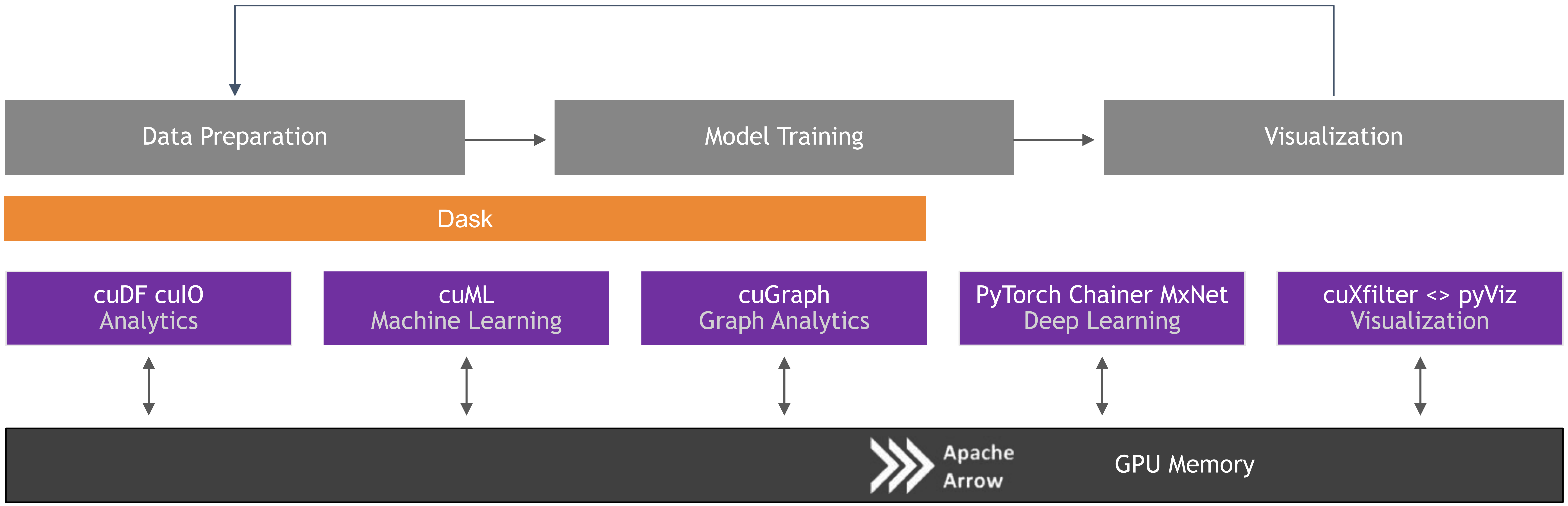
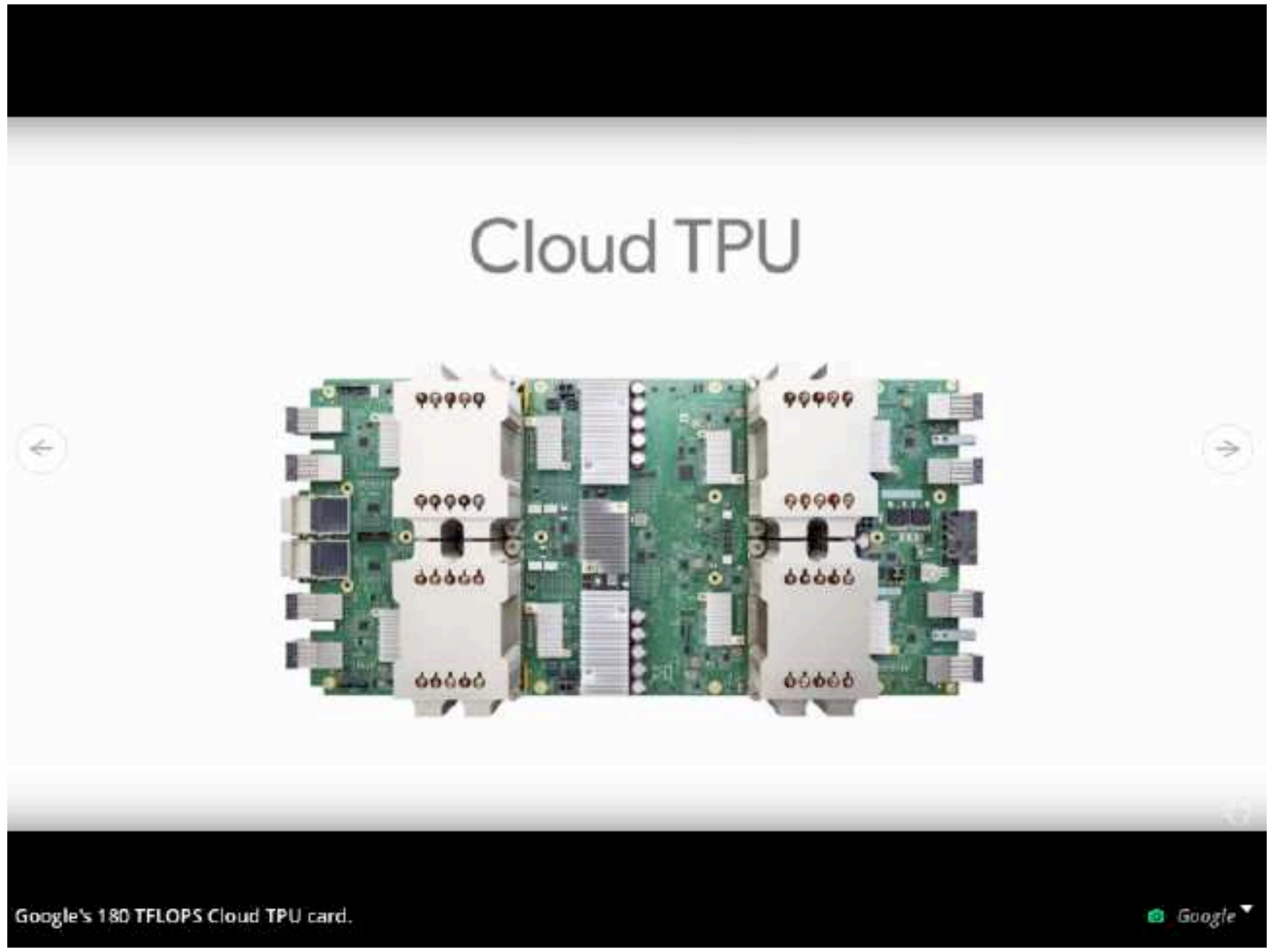


Figure 4. RAPIDS is an open source effort to support and grow the ecosystem of GPU-accelerated Python tools for data science, machine learning, and scientific computing. RAPIDS supports existing libraries, fills gaps by providing open source libraries with crucial components that are missing from the Python community, and promotes cohesion across the ecosystem by supporting interoperability across the libraries.

Sebastian Raschka, Joshua Patterson, and Corey Nolet (2020)
Machine Learning in Python: Main developments and technology trends in data science, machine learning, and artificial intelligence
Information 2020, 11, 193

Besides GPUs, Companies Develop Specialized Hardware



https://arstechnica.com/gadgets/2018/07/the-ai-revolution-has-spawned-a-new-chips-arms-race/

https://www.graphcore.ai

https://developer.arm.com/products/processors/machine-learning/arm-ml-processor

TECHNOLOGY NEWS NOVEMBER 28, 2018 / 2:59 PM / 2 MONTHS AGO

Amazon launches machine learning chip, taking on Nvidia, Intel

https://www.reuters.com/article/us-amazon-com-nvidia/amazon-launches-machine-learning-chip-taking-on-nvidia-intel-idUSKCN1NX2PY

Deep Learning Frameworks: An Abbreviated History

2000s:
- OpenNN, Torch, Matlab

2010s:
- (Multi)-GPU support: Caffe, config files; Chainer imperative; Theano declarative

2015s:
- TensorFlow (Google), declarative
- Caffe2 (FAIR, by TensorFlow dev)
- CNTK (Microsoft)
- DyNet (Carnegie Mellon University)
- Paddle Paddle (Baidu)
- MXNet (Amazon support), declarative & imperative "mix"
- Keras API
- PyTorch (FAIR), imperative (Torch and Chainer)

Things Looks Much Simpler in 2021

2000s:

- OpenNN, ~~Torch~~, Matlab

2010s:

(PyMC3)

- ~~Caffe, config files~~; ~~Chainer imperative~~; ~~Theano declarative~~

2015s:

- TensorFlow (Google), declarative
- Caffe2 (FAIR, by TensorFlow dev)
- ~~CNTK (Microsoft)~~
- ~~MXNet (Amazon support), declarative & imperative "mix"~~

...

- Keras API
- PyTorch (FAIR), imperative (Torch and Chainer)

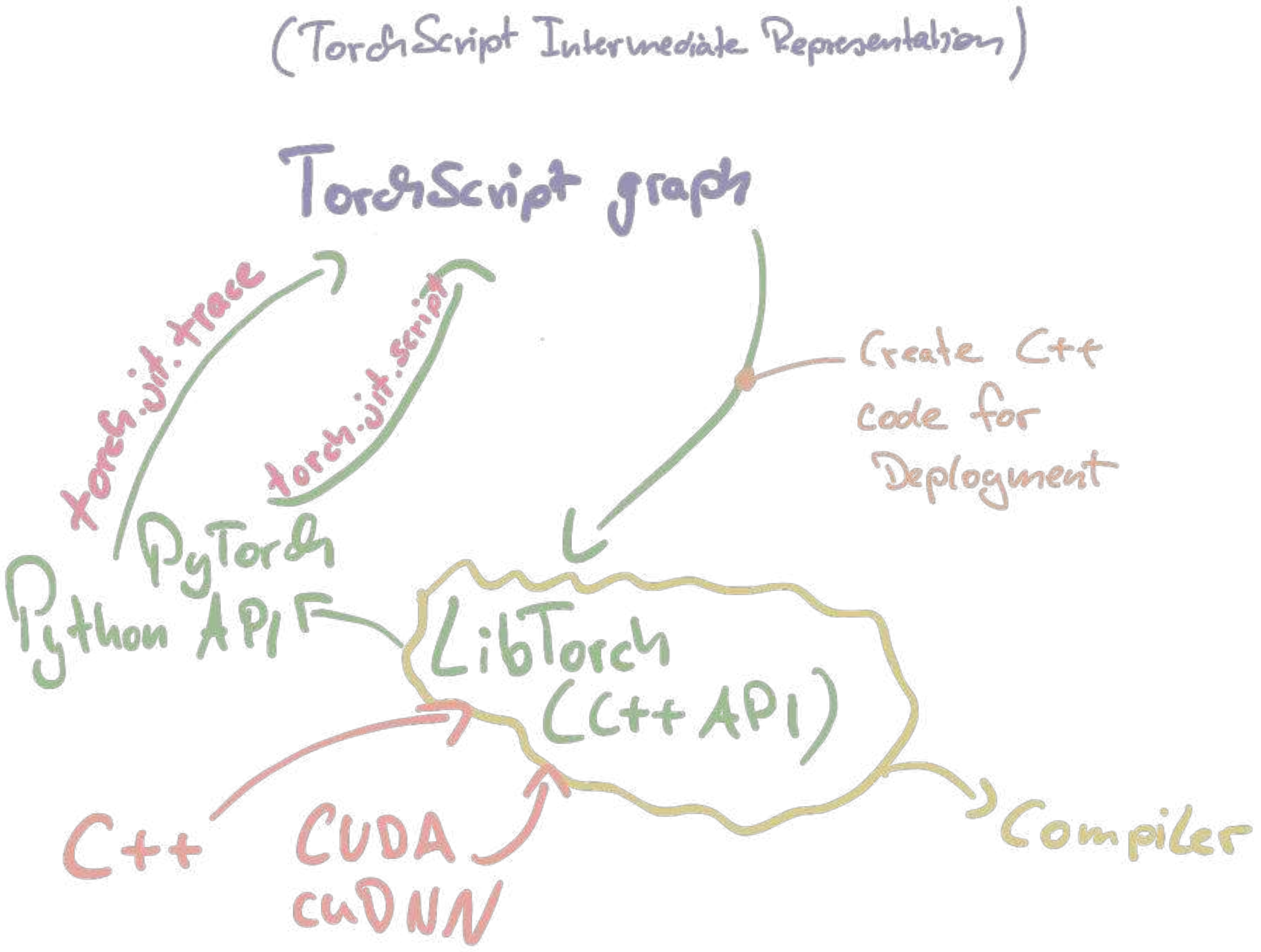
2021:

- TensorFlow
- PyTorch
- JAX

DL Frameworks are Converging

E.g.,

- TensorFlow adds eager mode
- PyTorch adds static graph support



Alternatives?

A more functional approach
but requires more obj. oriented add-on libraries
for deep learning (e.g., Haiku, Flax)

Swift for TensorFlow: promising but ...
Google canned it in February 2021
https://www.tensorflow.org/swift/guide/overview

~fast as Fortran, ~easy as Python
everyone loves it, not many use it
https://julialang.org

JAX & Haiku

```python
import haiku as hk
import jax.numpy as jnp

def softmax_cross_entropy(logits, labels):
  one_hot = jax.nn.one_hot(labels, logits.shape[-1])
  return -jnp.sum(jax.nn.log_softmax(logits) * one_hot, axis=-1)

def loss_fn(images, labels):
  mlp = hk.Sequential([
      hk.Linear(300), jax.nn.relu,
      hk.Linear(100), jax.nn.relu,
      hk.Linear(10),
  ])
```

https://github.com/deepmind/dm-haiku

TensorFlow 2

```python
model = models.Sequential()
model.add(layers.Conv2D(32, (3, 3), activation='relu', input_shape=(32, 32,
3)))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Conv2D(64, (3, 3), activation='relu'))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Conv2D(64, (3, 3), activation='relu'))
```

https://www.tensorflow.org/tutorials/images/cnn

PyTorch

```python
import torch.nn as nn


class MLP(nn.Module):

    def __init__(self, num_features, num_classes):
        super().__init__()

        self.my_network = torch.nn.Sequential(
            nn.Linear(num_features, 50),
            nn.ReLU(),
            nn.Linear(50, 25),
            nn.ReLU(),
            nn.Linear(25, num_classes)
        )

    def forward(self, x):
        logits = self.my_network(x)
        return logits
```

https://github.com/rasbt/stat453-deep-learning-ss21/blob/main/L09/code/mlp-pytorch_softmax-crossentr.ipynb

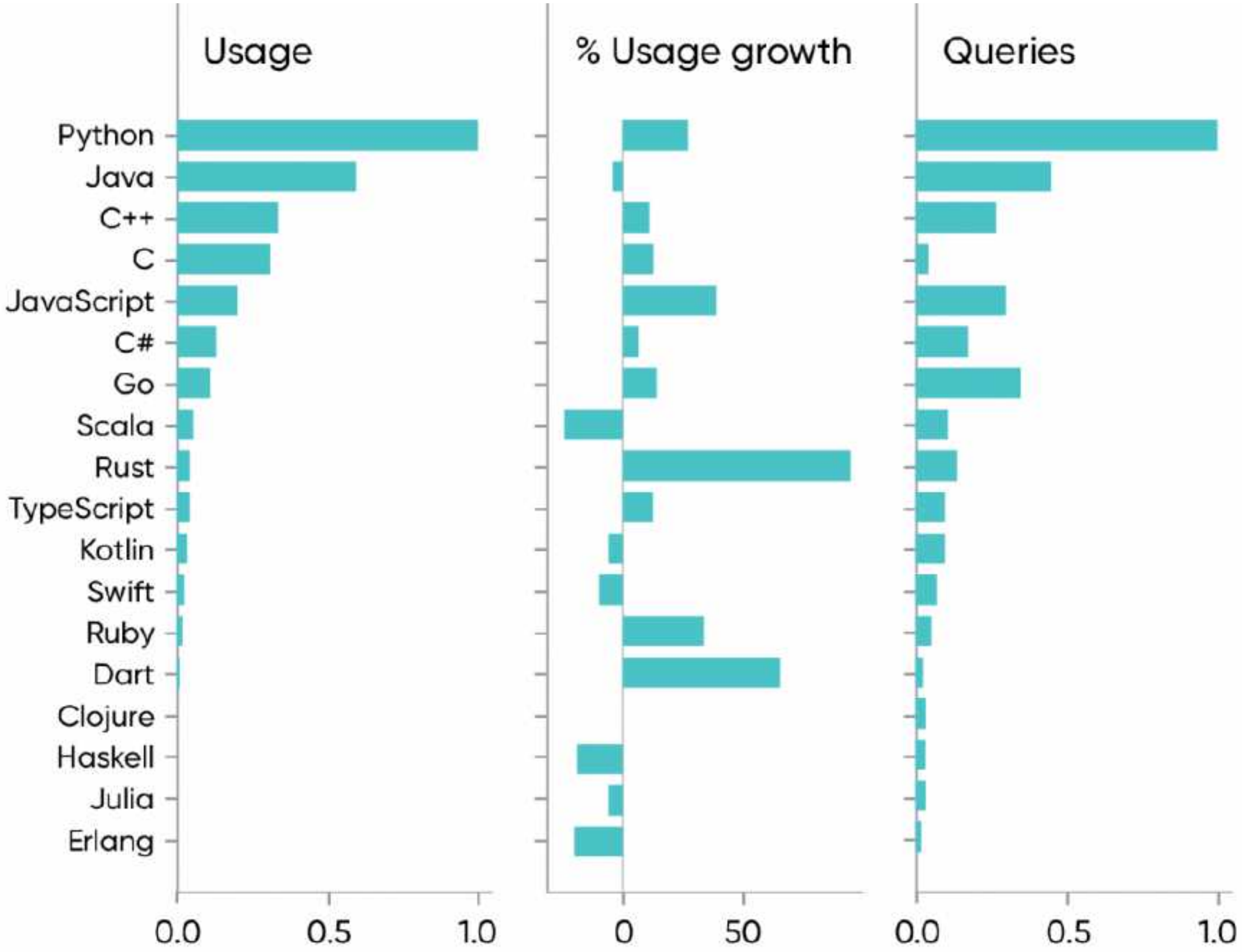
JAX & Flax

```python
from flax import linen as nn


class CNN(nn.Module):
  """A simple CNN model."""

  @nn.compact
  def __call__(self, x):
    x = nn.Conv(features=32, kernel_size=(3, 3))(x)
    x = nn.relu(x)
    x = nn.avg_pool(x, window_shape=(2, 2), strides=(2, 2))
    x = nn.Conv(features=64, kernel_size=(3, 3))(x)
    x = nn.relu(x)
    x = nn.avg_pool(x, window_shape=(2, 2), strides=(2, 2))
    x = x.reshape((x.shape[0], -1))  # flatten
    x = nn.Dense(features=256)(x)
    x = nn.relu(x)
    x = nn.Dense(features=10)(x)
    x = nn.log_softmax(x)
    return x
```

https://github.com/google/flax/blob/master/examples/mnist/train.py

Python seems to be here to stay

https://www.oreilly.com/radar/where-programming-ops-ai-and-the-cloud-are-headed-in-2021/



"[...] the speedup gained by taking Python out of the computation is 10% or less."

-- Stevens E, Antiga L, Viehmann T. Deep learning with PyTorch. Manning; 2020.

(2) Challenges

- Small data
- Ordinal data
- Adversarial attacks
- Bias
- Privacy

Tackling Small Data Problems

Transfer learning
Pre-train on larger related dataset with labels

Active learning
Optimize data order and labeling

Semi-supervised learning
Incorporate unlabeled data into the training

Few-shot learning
Special cases with very few examples per class (incl. transfer learning, metric learning, semi-supervised, meta-learning)

Self-supervised learning
Pre-train on unlabeled dataset by creating leveraging data structure to create labels

Academia Vs Industry

Model-Centric Approach

Primary focus is on tuning and developing models to improve performance on a fixed benchmark set

Data-Centric Approach

Primary focus is on how one can improve the dataset (collect more, select, relabel) to improve model performance

Source: Andrej Karpathy, Andrew Ng

Ordinal Data: Integrating Label Order Info

- **Ranking:** Predict Correct order
  (0 loss if order is correct, e.g., rank a collection of movies by "goodness")

$\prec$

$\prec$

- **Ordinal regression:** Predict correct (ordered) label
  (E.g., age of a person in years; here, regard aging as a non-stationary process)

Excerpt from the UTKFace dataset
https://susanqq.github.io/UTKFace/


18

$\prec$

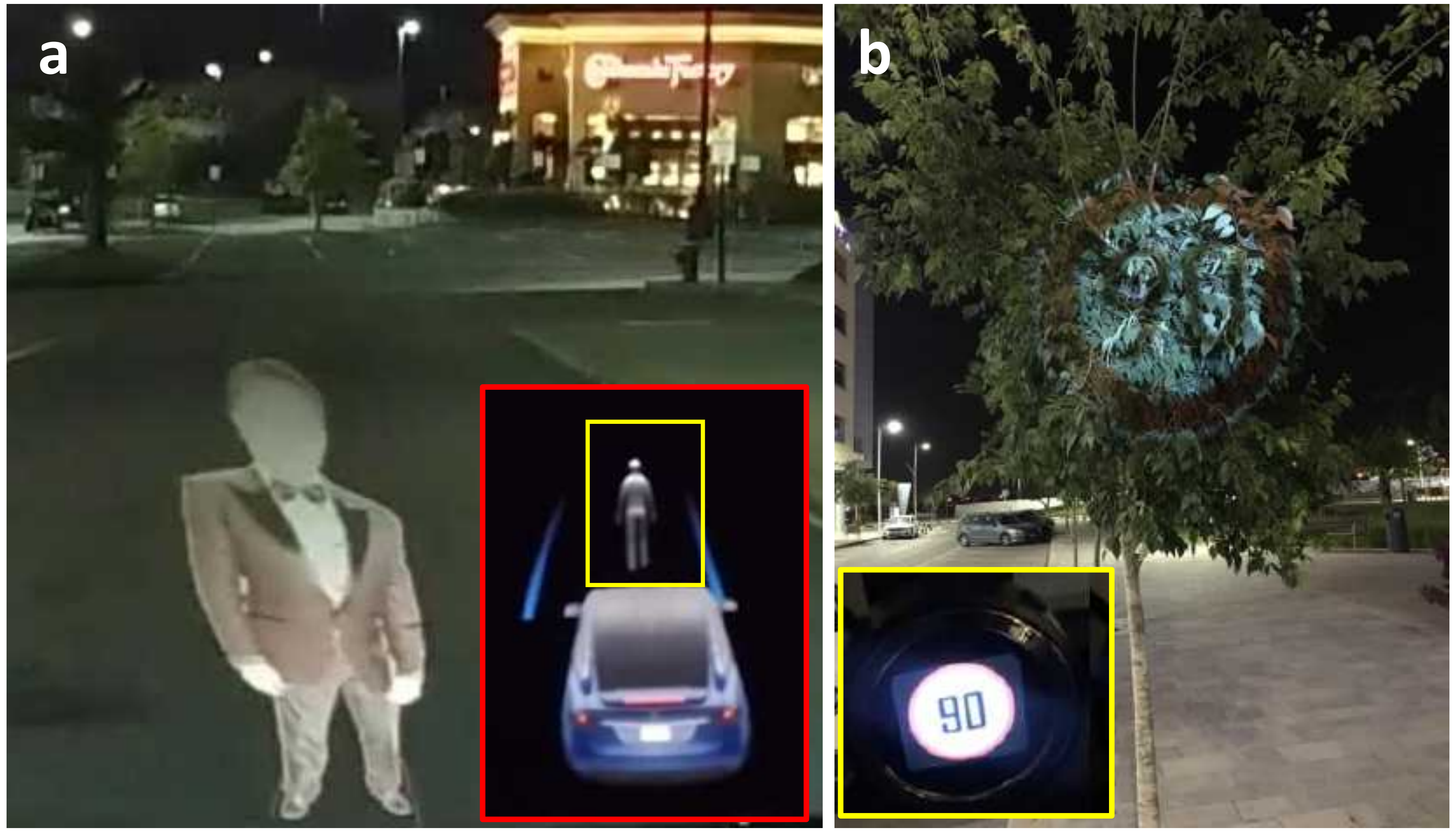

29

$\prec$


41

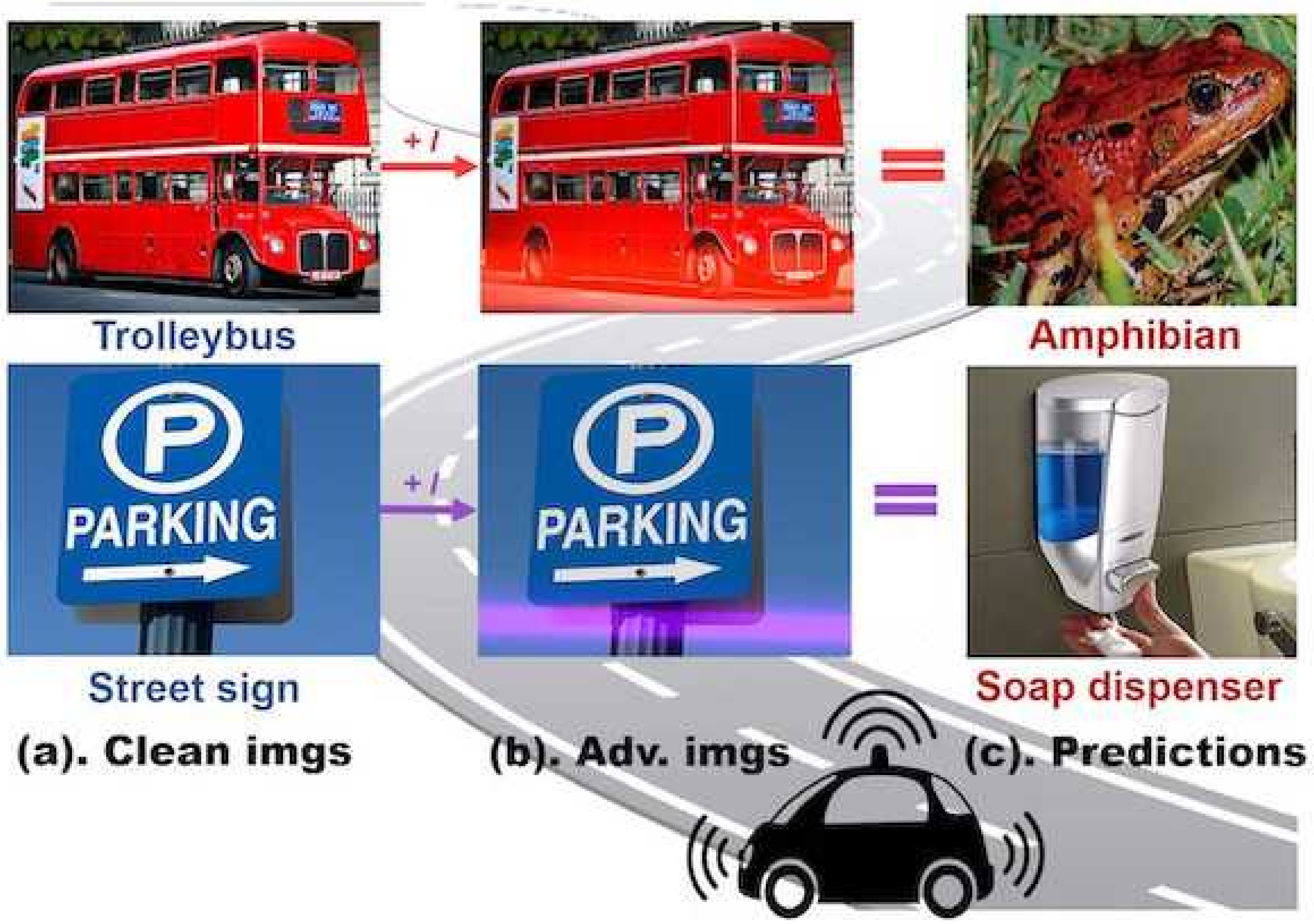
Cao, Mirjalili, Raschka (2020)
Rank Consistent Ordinal Regression for Neural Networks with Application to Age Estimation
Pattern Recognition Letters. 140, 325-331

Beyond Pandas & Gibbons: Real-World Adversarial Attacks



Tesla Autopilot considers (a) as a real person and (b) as a real road sign

Nassi, Mirsky, Nassi, Ben-Netanel, Drokin, Elovici. *Phantom of the ADAS: Securing Advanced Driver-Assistance Systems from Split-Second Phantom Attacks*. ACM SIGSAC Conference on Computer and Communications Security, 2020



Laser beams turn buses into amphibians and street signs into soap dispensers

Duan, Mao, Qin, Yang, Chen, Ye, He. *Adversarial Laser Beam: Effective Physical-World Attack to DNNs in a Blink*. arXiv preprint arXiv:2103.06504. 2021 Mar 11.

Some Common Adversarial Attacks & Defenses

	Cleverhans v3.0.1	FoolBox v2.3.0	ART v1.1.0	DEEPSEC (2019)	AdvBox v0.4.1
Supported frameworks					
TensorFlow	yes	yes	yes	no	yes
MXNet	yes	yes	yes	no	yes
PyTorch	no	yes	yes	yes	yes
PaddlePaddle	no	no	no	no	yes
(Evasion) attack mechanisms					
BLB [163]	yes	no	no	yes	no
AMD [170]	yes	no	no	no	no
ZOO [171]	no	no	yes	no	no
VA [172]	yes	yes	yes	no	no
AP [173]	no	no	yes	no	no
STA [174]	no	yes	yes	no	no
DTA [175]	no	no	yes	no	no
FGSM [176]	yes	yes	yes	yes	yes
R+FGSM [177]	no	no	no	yes	no
R+LLC [177]	no	no	no	yes	no
U-MI-FGSM [178]	yes	yes	no	yes	no
T-MI-FGSM [178]	yes	yes	no	yes	no
BIM [179]	no	yes	yes	yes	yes
LLC / ILLC [179]	no	yes	no	yes	no
UAP [180]	no	no	yes	yes	no
DeepFool [181]	yes	yes	yes	yes	yes
NewtonFool [182]	no	yes	yes	no	no
JSMA [183]	yes	yes	yes	yes	yes
CW/CW2 [184]	yes	yes	yes	yes	yes
PGD [185]	yes	no	yes	yes	yes
OM [186]	no	no	no	yes	no
EAD [187]	yes	yes	yes	yes	no
Boundary Attack [188]	no	yes	yes	no	no
HopSkipJumpAttack [189]	yes	yes	yes	no	no
MaxConf [190]	yes	no	no	no	no
Inversion attack [191]	yes	yes	no	no	no
SparseL1 [192]	yes	yes	no	no	no
SPSA [193]	yes	no	no	no	no
HCLU [194]	no	no	yes	no	no
ADef [195]	no	yes	no	no	no
DDNL2 [196]	no	yes	no	no	no
Local Search [197]	no	yes	no	no	no
Pointwise attack [198]	no	yes	no	no	no
GenAttack [199]	no	yes	no	no	no
Defense mechanisms					
Feature Squeezing [200]	no	no	yes	no	yes
Spatial Smoothing [200]	no	no	yes	no	yes
Label Smoothing [200]	no	no	yes	no	yes
Gaussian Augmentation [201]	no	no	yes	no	yes
Adversarial Training [185]	no	no	yes	yes	yes
Thermometer Encoding [202]	no	no	yes	yes	yes
NAT [203]	no	no	no	yes	no
EAT [177]	no	no	no	yes	no
DD [204]	no	no	no	yes	no
IGR [205]	no	no	no	yes	no
EIT [206]	no	no	yes	yes	no
RT [207]	no	no	no	yes	no
PixelDefend [208]	no	no	yes	yes	no
Regr.-based classfication [209]	no	no	no	yes	no
JPEG compression [210]	no	no	yes	no	no

Machine Learning in Python: Main developments and technology trends in data science, machine learning, and artificial intelligence (2020). Sebastian Raschka, Joshua Patterson, and Corey Nolet

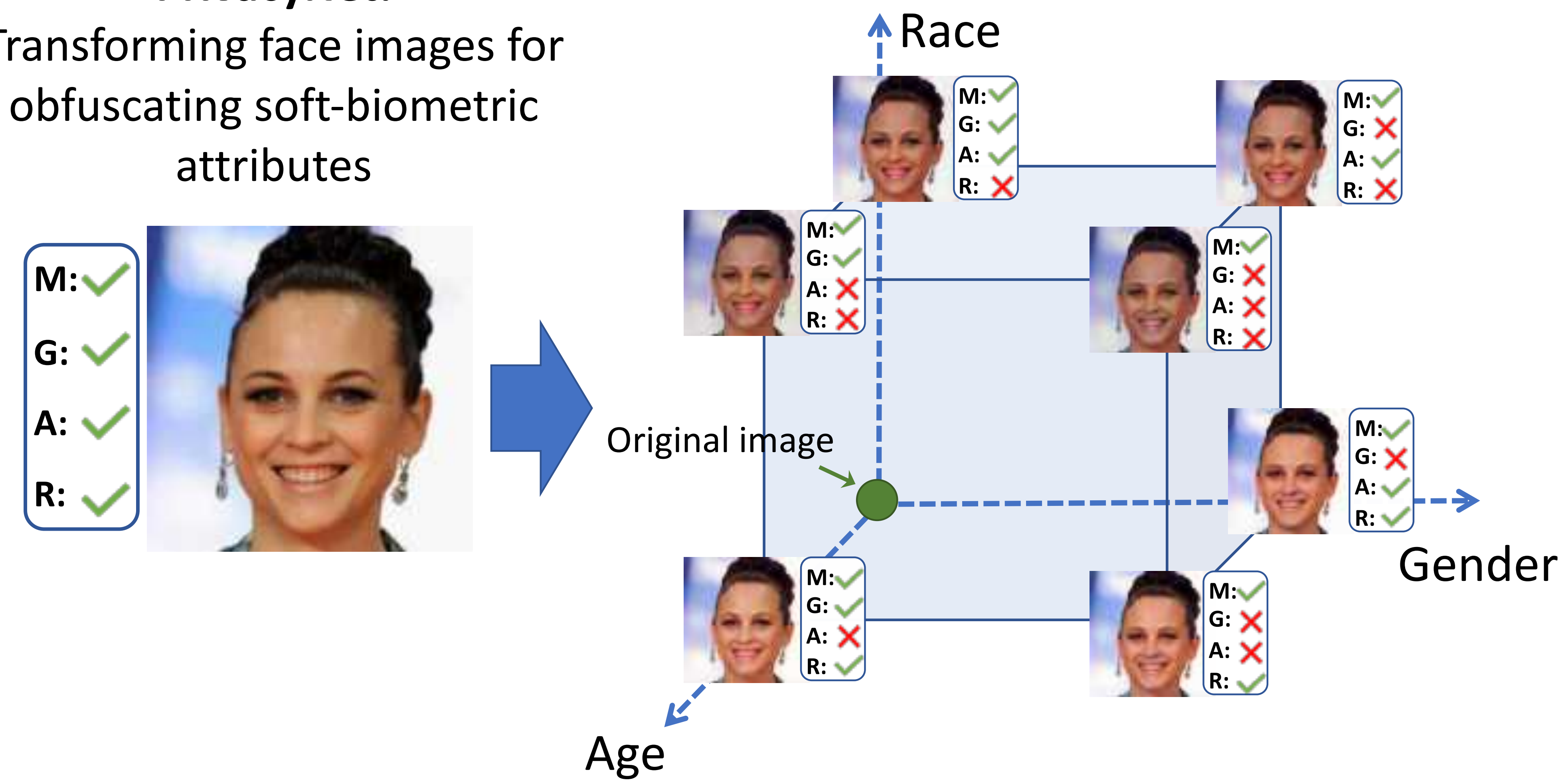
APRIL 14, 2020

Half of Americans have decided not to use a product or service because of privacy concerns

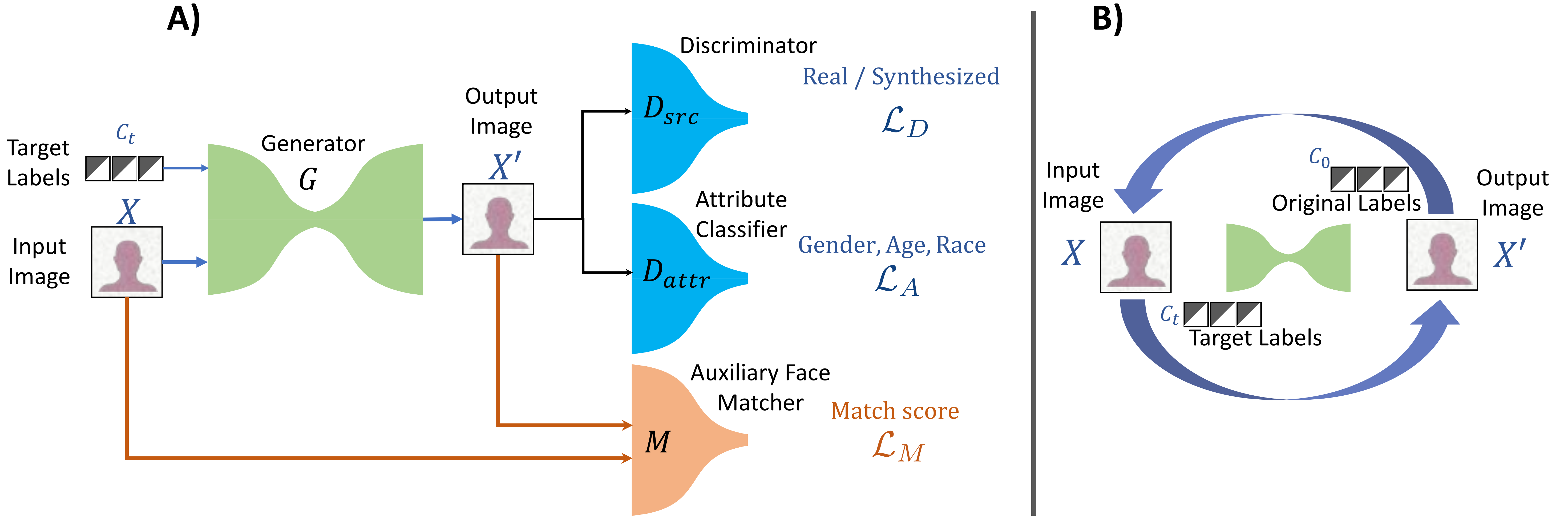
https://www.pewresearch.org/fact-tank/2020/04/14/half-of-americans-have-decided-not-to-use-a-product-or-service-because-of-privacy-concerns/

Enhancing Privacy:
(1) Hiding Information by Modifying Data

PrivacyNet:
Transforming face images for obfuscating soft-biometric attributes



Vahid Mirjalili, Sebastian Raschka, and Arun Ross (2020)
PrivacyNet: Semi-Adversarial Networks for Multi-attribute Face Privacy
IEEE Transactions in Image Processing. Vol. 29, pp. 9400-9412, 2020



Enhancing Privacy:
(2) Differential Privacy via Synthetic Datasets

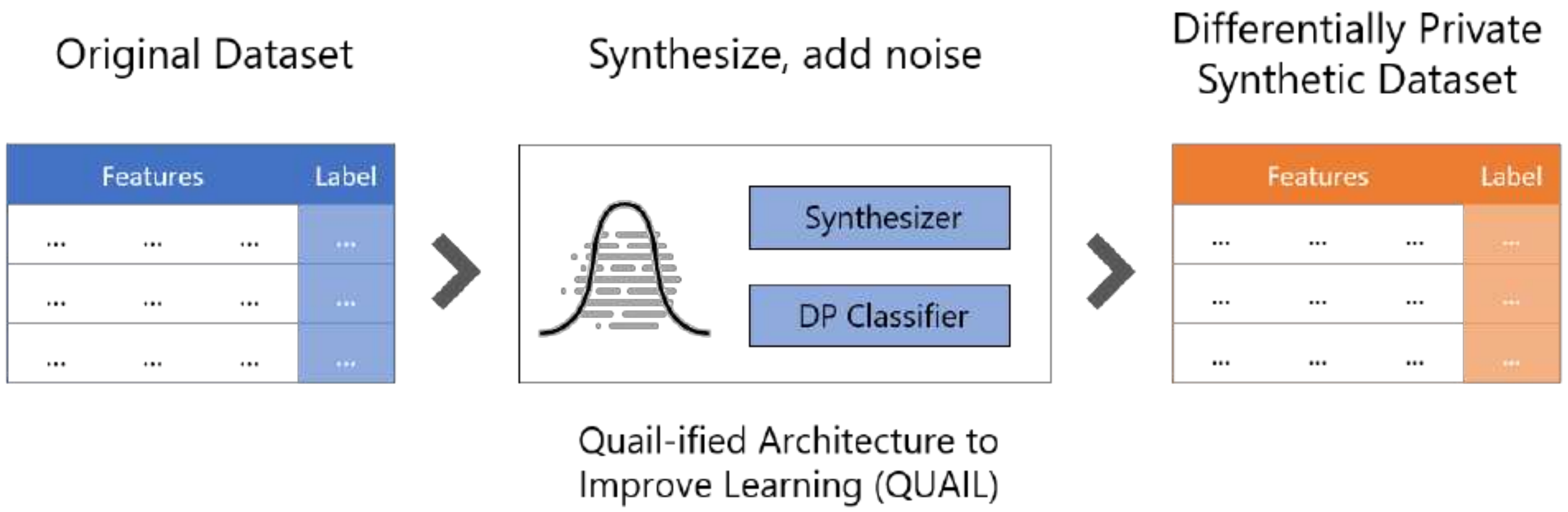
Microsoft Open Source Blog

Create privacy-preserving synthetic data for machine learning with SmartNoise

February 18, 2021

Share

https://cloudblogs.microsoft.com/opensource/2021/02/18/create-privacy-preserving-synthetic-data-for-machine-learning-with-smartnoise/



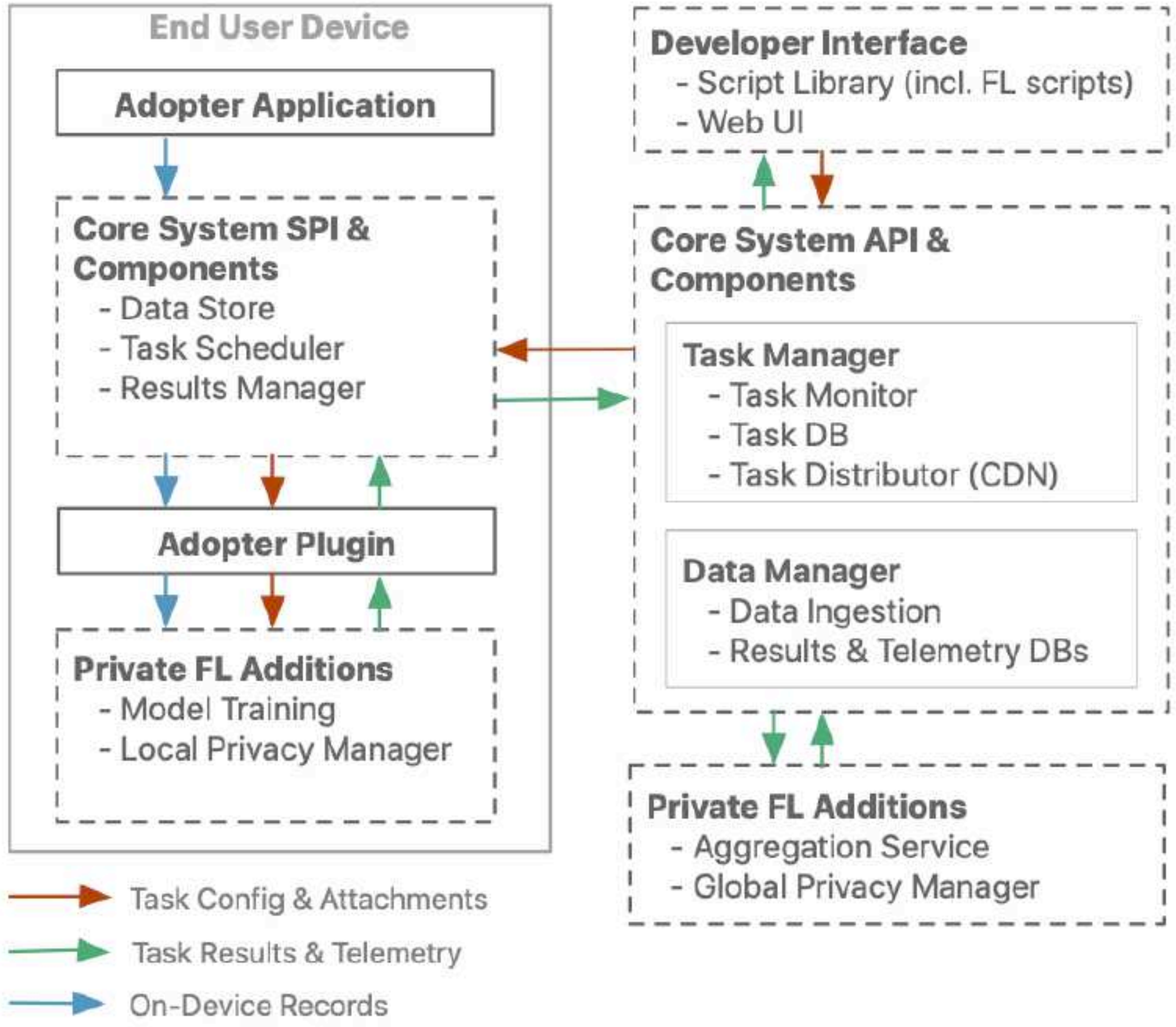
Whitepaper: https://azure.microsoft.com/en-us/resources/microsoft-smartnoisedifferential-privacy-machine-learning-case-studies/

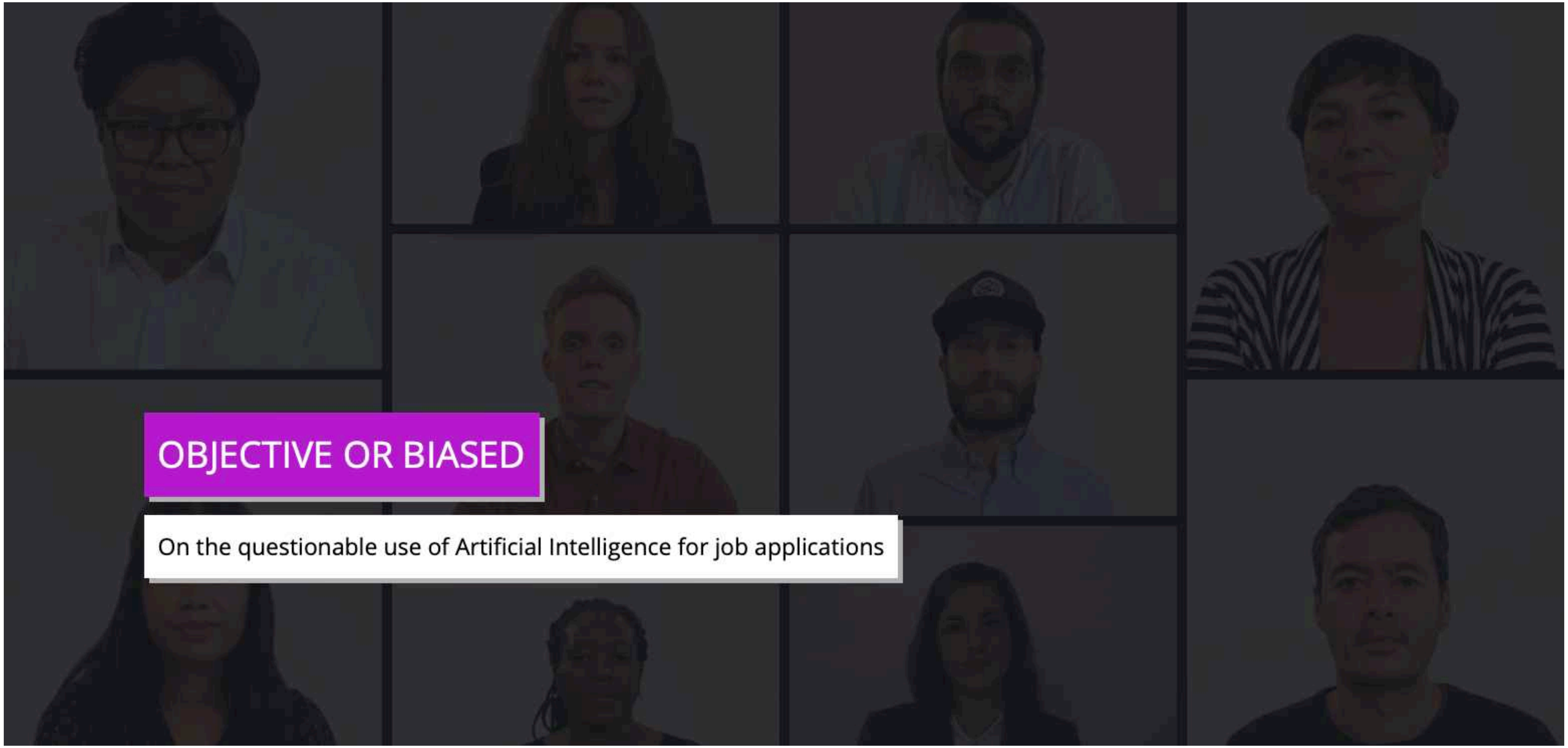
Enhancing Privacy:
(3) User Data Stays on the Device

Paulik, Seigel, Mason, Telaar, Kluivers, van Dalen, Lau, Carlson, Granqvist, Vandevelde, Agarwal. *Federated Evaluation and Tuning for On-Device Personalization: System Design & Applications*. arXiv preprint arXiv:2102.08503. 2021 Feb 16.

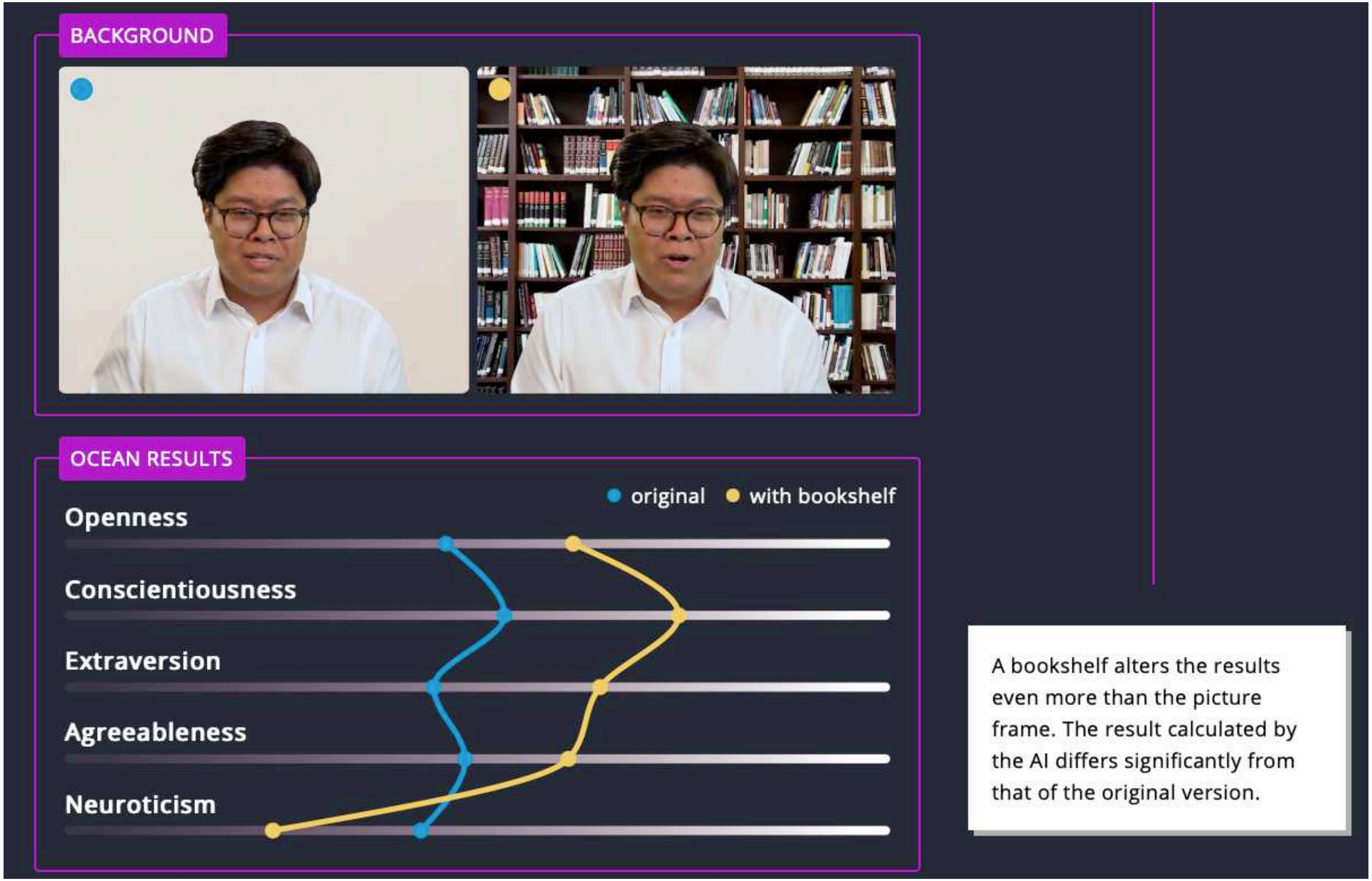
- **Other approaches:** use federated learning to tune a global neural network
- **Apple**:
  - Use global parameters but train local model
  - User data remains inaccessible to server-side

Apple's On-Device ML System for Federated Evaluation and Tuning

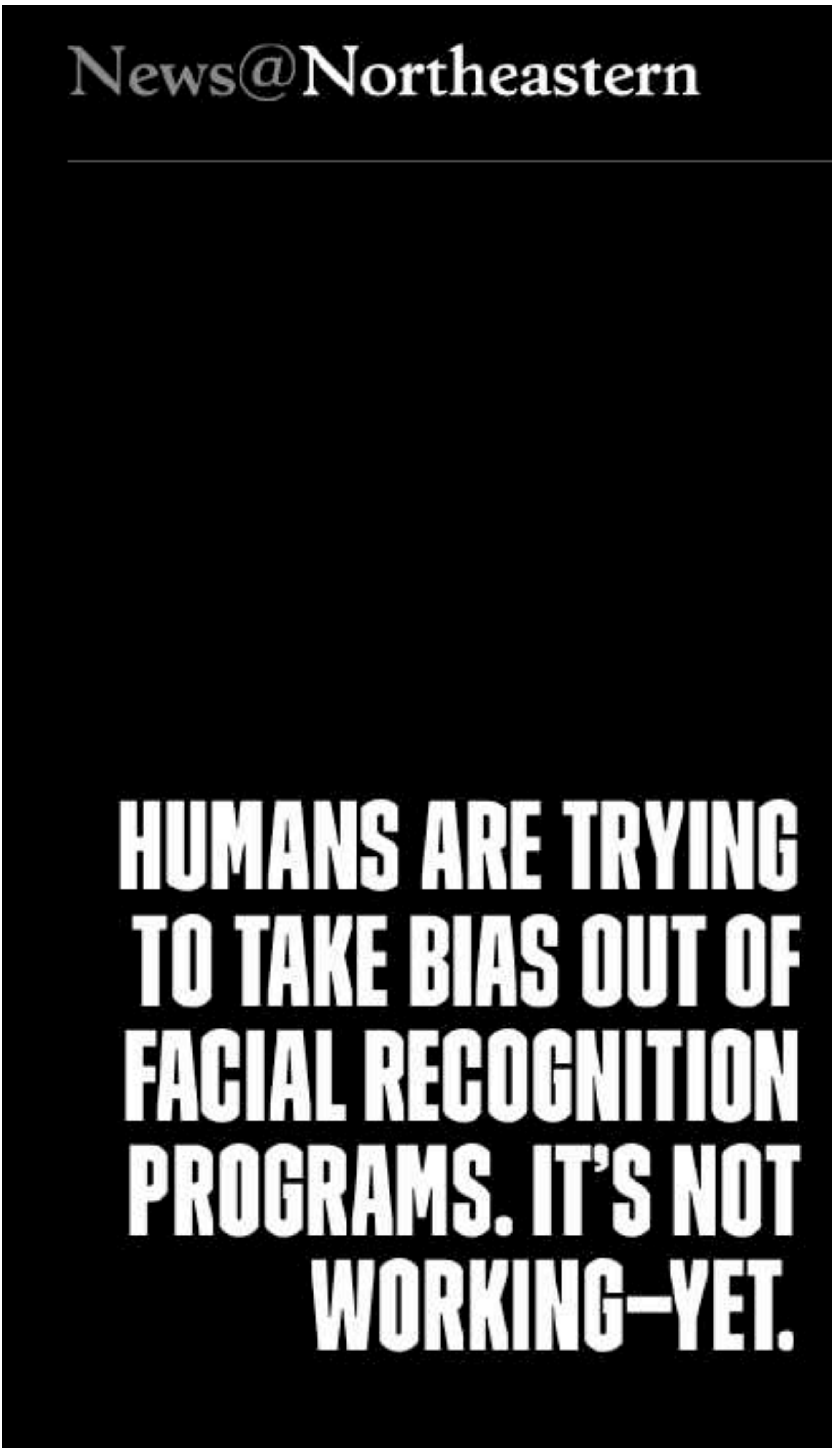




https://web.br.de/interaktiv/ki-bewerbung/en/



https://web.br.de/interaktiv/ki-bewerbung/en/



Common approach: Address lack of diversity in datasets.
--> provide algorithms with datasets that represent all groups equally and fairly

Does it work? Only for a stereotypical sense of fairness according to Zaid Khan:

> "The people in the images appeared to fit racial stereotypes.
> For example, an algorithm was more likely to label an individual in an image as 'white' if that person had blond hair."

https://news.northeastern.edu/2021/02/22/humans-are-trying-to-take-bias-out-of-facial-recognition-programs-its-not-working-yet/

Paper:
Khan Z, Fu Y.
One Label, One Billion Faces: Usage and Consistency of Racial Categories in Computer Vision.
ACM Conference on Fairness, Accountability, and Transparency 2021 Mar 3
https://dl.acm.org/doi/abs/10.1145/3442188.3445920

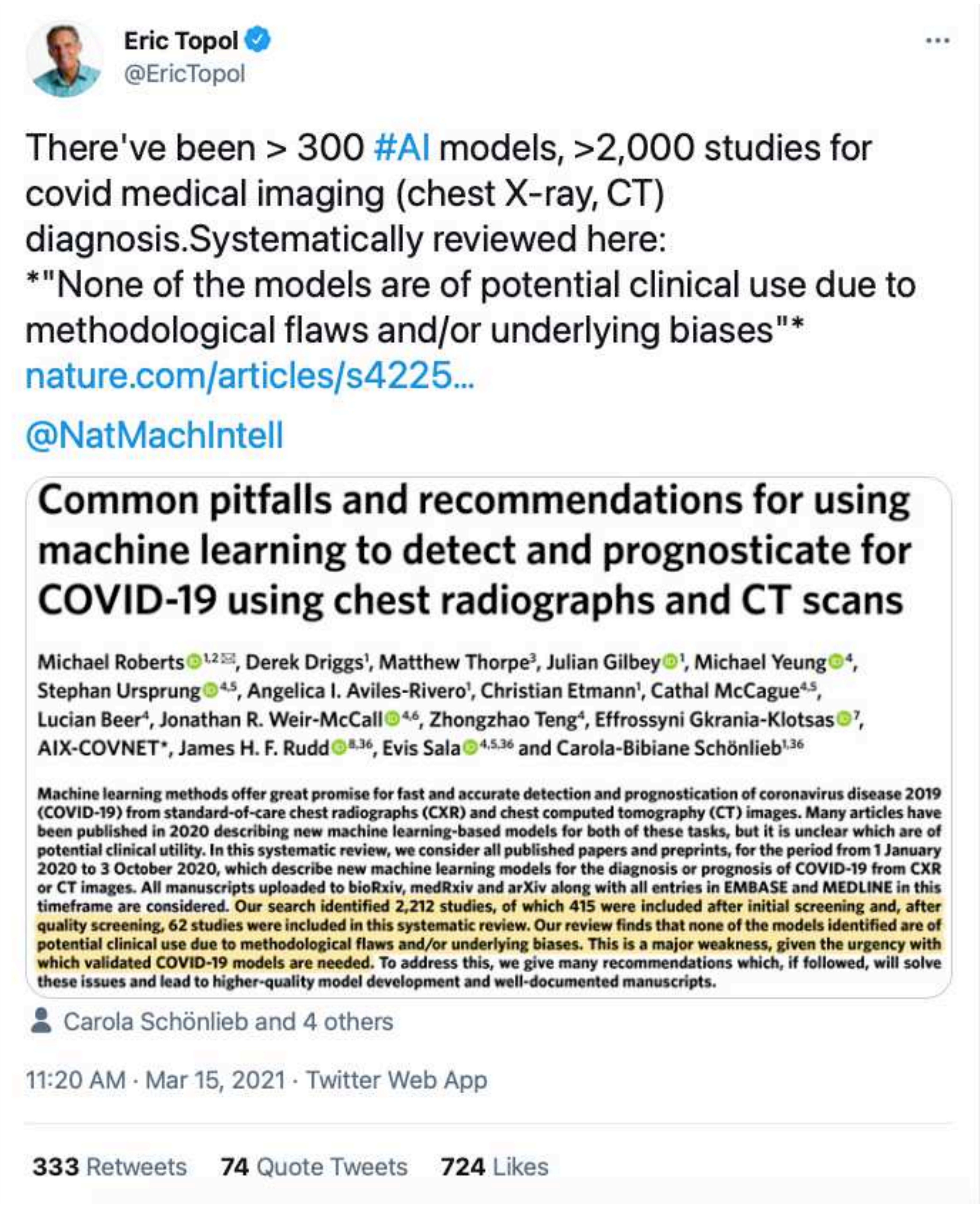
Don't Have to Drive a Car Off a Cliff to Learn What Happens

Schölkopf B, Locatello F, Bauer S, Ke NR, Kalchbrenner N, Goyal A, Bengio Y. **Toward Causal Representation Learning**. Proceedings of the IEEE. 2021 Feb 26.

- Deep learning is currently largely based on statistical correlations from i.i.d. data
- Learning causal relationships can make models more robust to unexpected situations
- Can make training cheaper -- fewer examples like objects from different angles required
- Enable transfer learning beyond fine-tuning

The challenges: *Does the data reveal causal relationships?*

How do we infer abstract causal variables?


Eric Topol
@EricTopol

There've been > 300 #AI models, >2,000 studies for covid medical imaging (chest X-ray, CT) diagnosis.Systematically reviewed here:
"None of the models are of potential clinical use due to methodological flaws and/or underlying biases"
nature.com/articles/s4225...

@NatMachIntell

Common pitfalls and recommendations for using machine learning to detect and prognosticate for COVID-19 using chest radiographs and CT scans

Michael Roberts[1,2], Derek Driggs[1], Matthew Thorpe[3], Julian Gilbey[1], Michael Yeung[4], Stephan Ursprung[4,5], Angelica I. Aviles-Rivero[1], Christian Etmann[1], Cathal McCague[4,5], Lucian Beer[4], Jonathan R. Weir-McCall[4,6], Zhongzhao Teng[4], Effrossyni Gkrania-Klotsas[7], AIX-COVNET*, James H. F. Rudd[8,36], Evis Sala[4,5,36] and Carola-Bibiane Schönlieb[1,36]

Machine learning methods offer great promise for fast and accurate detection and prognostication of coronavirus disease 2019 (COVID-19) from standard-of-care chest radiographs (CXR) and chest computed tomography (CT) images. Many articles have been published in 2020 describing new machine learning-based models for both of these tasks, but it is unclear which are of potential clinical utility. In this systematic review, we consider all published papers and preprints, for the period from 1 January 2020 to 3 October 2020, which describe new machine learning models for the diagnosis or prognosis of COVID-19 from CXR or CT images. All manuscripts uploaded to bioRxiv, medRxiv and arXiv along with all entries in EMBASE and MEDLINE in this timeframe are considered. Our search identified 2,212 studies, of which 415 were included after initial screening and, after quality screening, 62 studies were included in this systematic review. Our review finds that none of the models identified are of potential clinical use due to methodological flaws and/or underlying biases. This is a major weakness, given the urgency with which validated COVID-19 models are needed. To address this, we give many recommendations which, if followed, will solve these issues and lead to higher-quality model development and well-documented manuscripts.

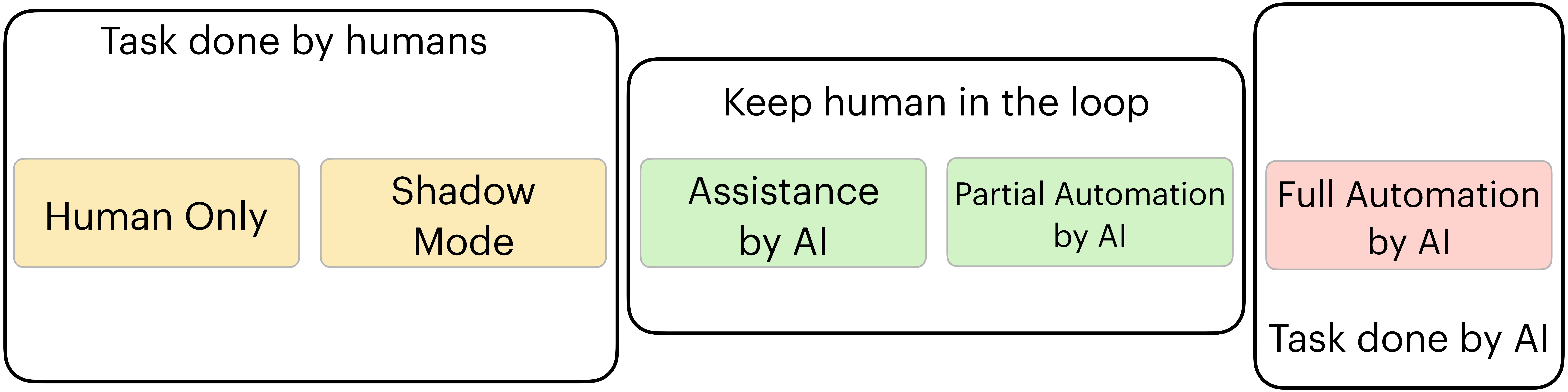
Carola Schönlieb and 4 others

11:20 AM · Mar 15, 2021 · Twitter Web App

333 Retweets 74 Quote Tweets 724 Likes

https://www.nature.com/articles/s42256-021-00307-0

Finding Middle Ground

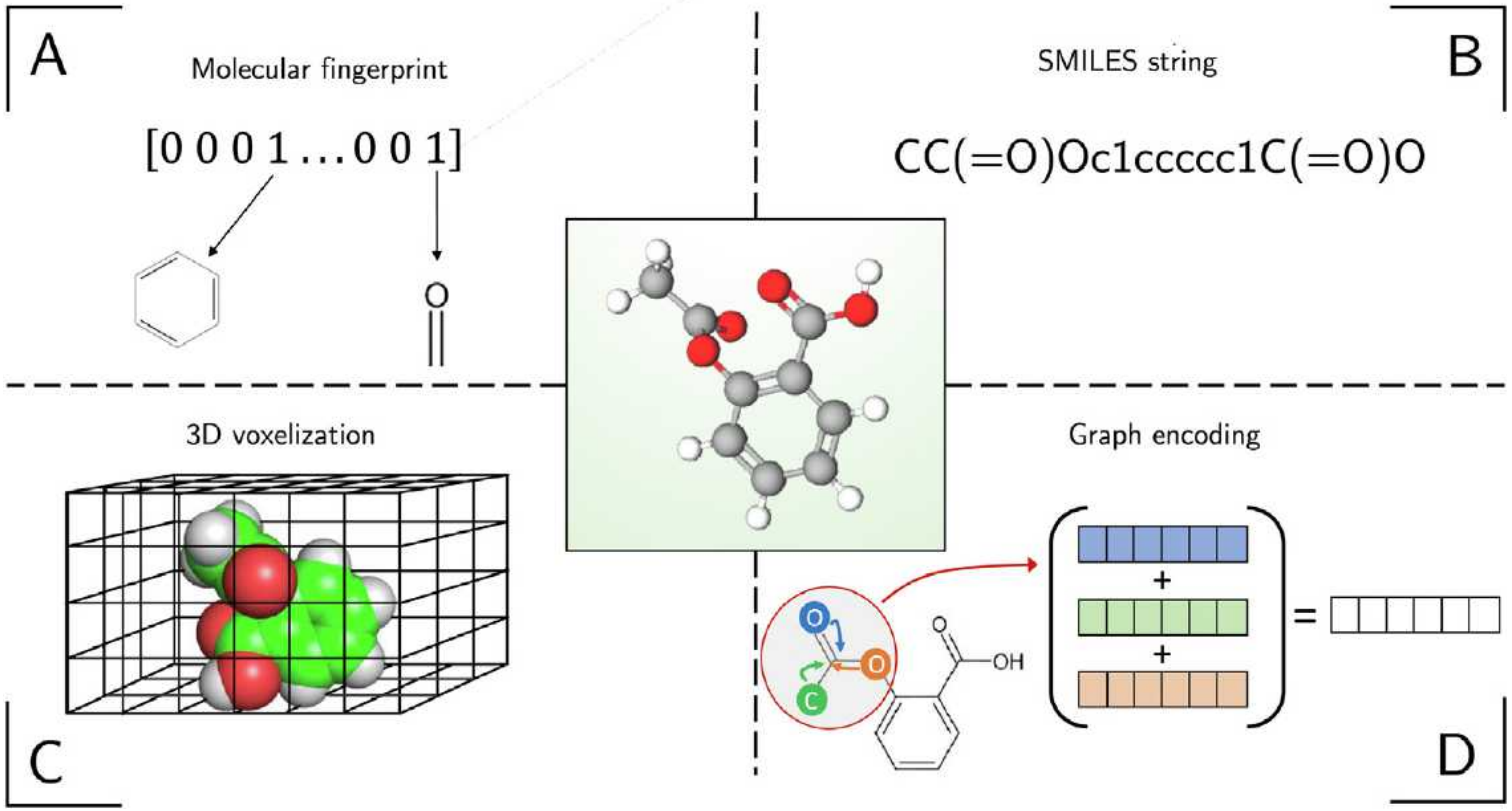


Source: Andrew Ng

(3) Research Trends

- Graph neural nets
- GANs
- Self-supervised learning
- Language transformers
- Vision transformers

Why Are Graph Neural Nets Interesting?



Sebastian Raschka and Benjamin Kaufman (2020)
Machine Learning and AI-based Approaches for Bioactive Ligand Discovery and GPCR-ligand Recognition
Elsevier Methods, 180, 89–110

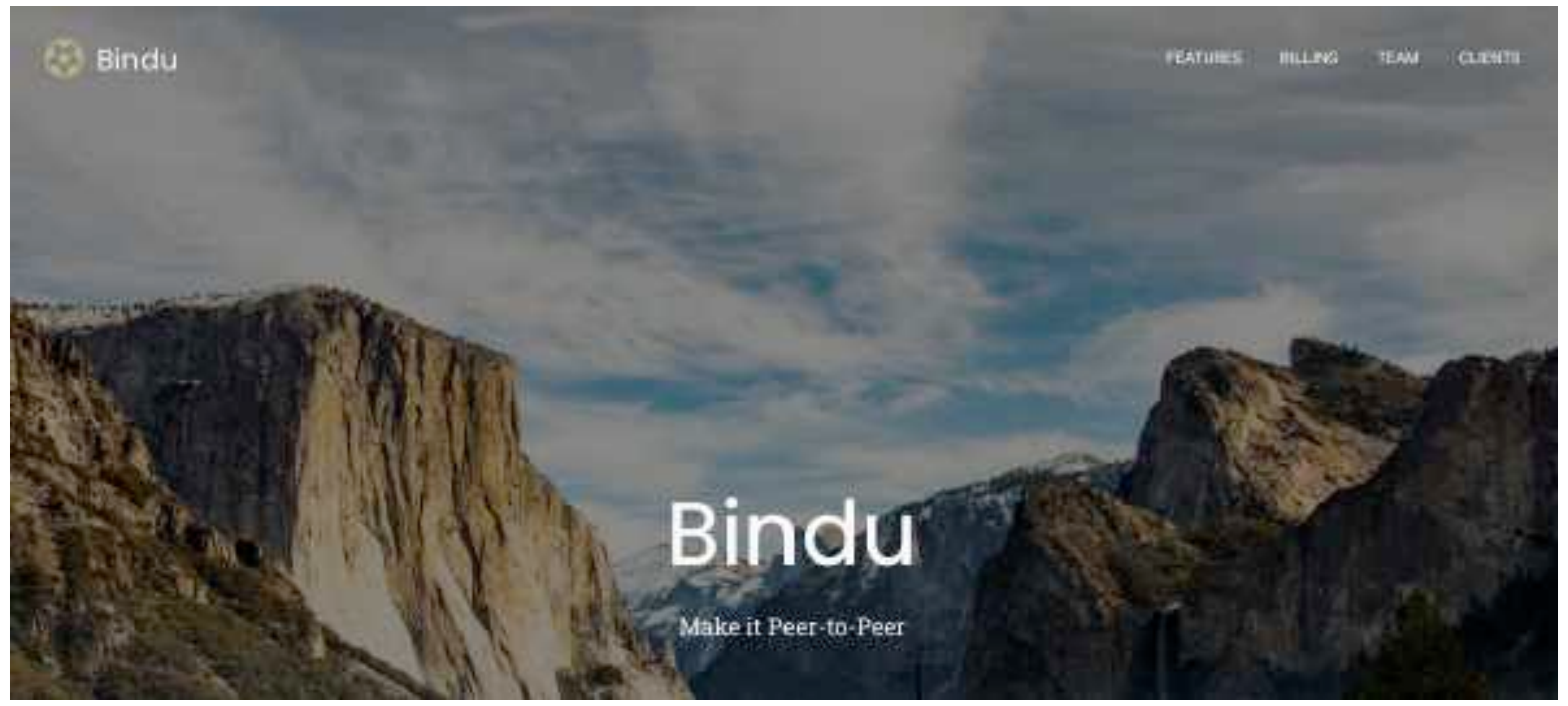
PyTorch geometric

https://github.com/rusty1s/pytorch_geometric

As of this writing: 82 graph neural net methods already implemented

Generative Adversarial Networks Have Come A Long Way

https://thiscatdoesnotexist.com



https://thisstartupdoesnotexist.com

https://thispersondoesnotexist.com

https://thisponydoesnotexist.net

Qualitative assessment in a class-conditional setting (class: *goldfinch*)

Vector Quantized GAN + Transformer (**Visual Transformer**)	Vector Quantized Variational Autoencoder 2 (**VQVAE-2**)	**BigGAN**	Masked Self-Prediction (**MSP**)
Esser, Rombach, Ommer. Taming Transformers for High-Resolution Image Synthesis, 2021	Razavi, van den Oord, Vinyals. Generating Diverse High-Fidelity Images with VQ-VAE-2, 2019	Brock, Donahue, and Simonyan. Large Scale GAN Training for High Fidelity Natural Image Synthesis, 2019	De Fauw, Dieleman, Simonyan. Hierarchical Autoregressive Image Models with Auxiliary Decoders, 2019.

Image source: Esser P, Rombach R, Ommer B. Taming Transformers for High-Resolution Image Synthesis. arXiv:2012.09841. 2020 Dec 17.

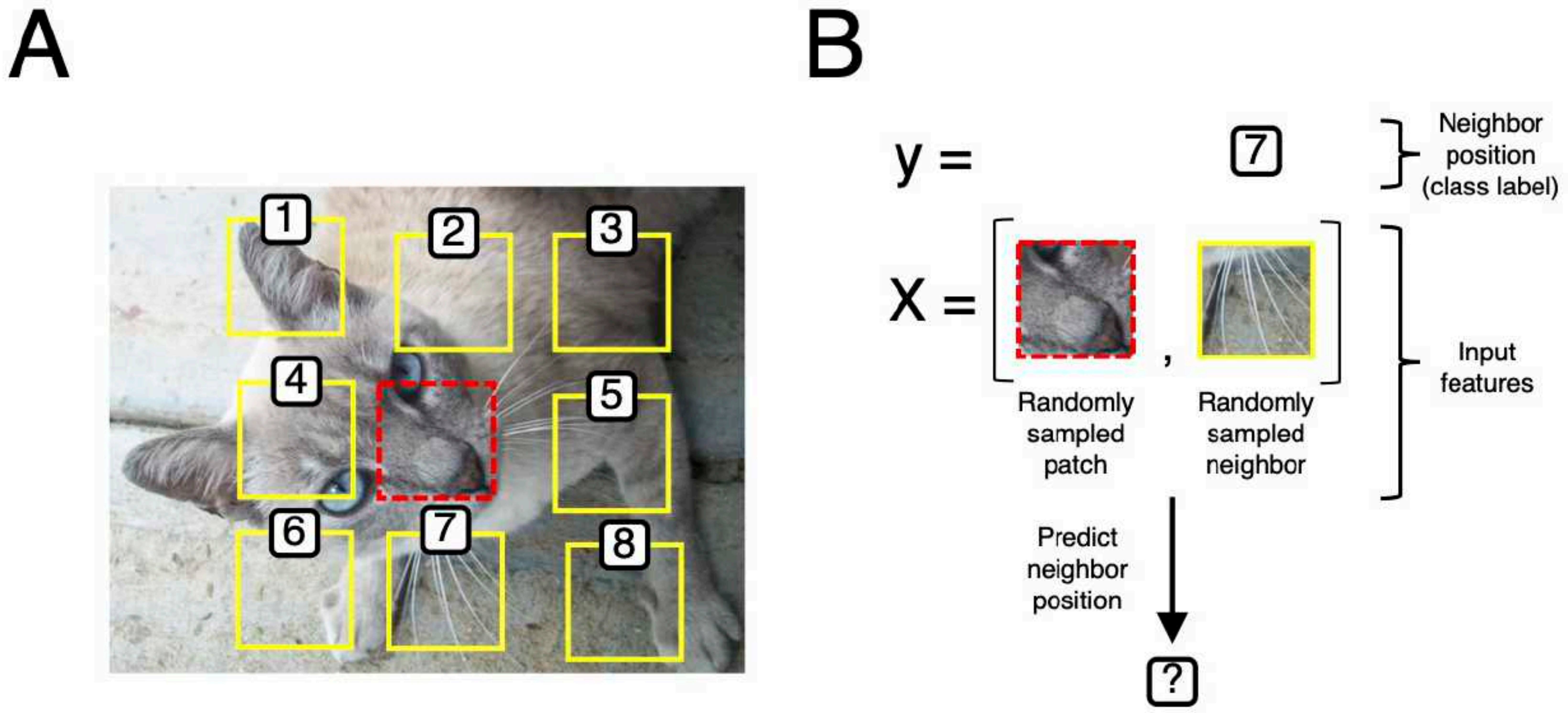
Self-Supervised Learning

Leverage structure of data to create labels for supervised learning,
to utilize large amounts of unlabeled data

1. Create labels (pre-text task) by leveraging structure of the data
2. Pre-train in self-supervised fashion to learn embeddings
3. Fine-tune in transfer learning fashion

Self-Supervised Learning

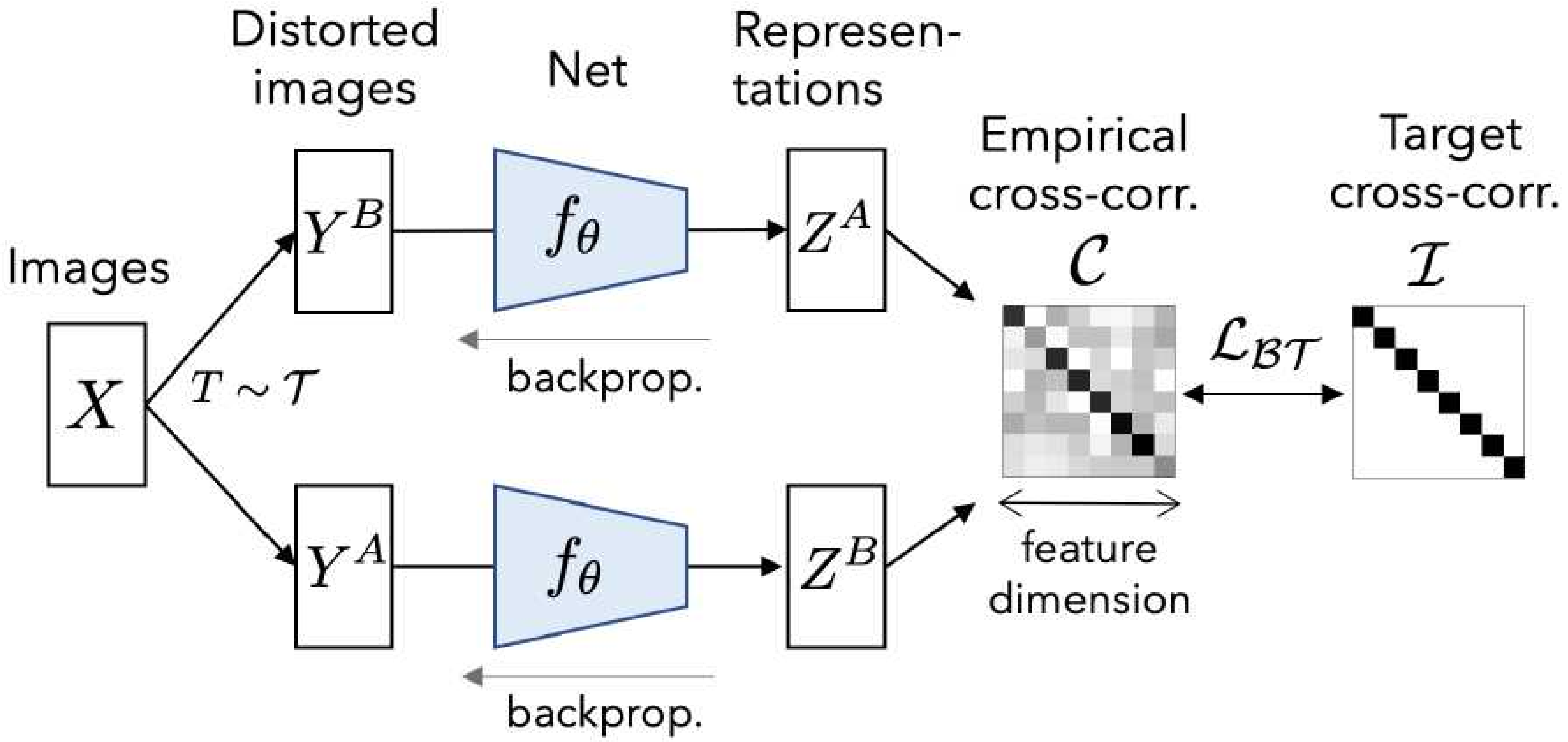
Leverage structure of data to create labels for supervised learning, to utilize large amounts of unlabeled data



https://sebastianraschka.com/blog/2020/intro-to-dl-ch01.html

Based on: Doersch, C., Gupta, A., & Efros, A. A.. *Unsupervised visual representation learning by context prediction*. CVPR 2015

Zbontar, Jing, Misra, LeCun, Deny.
Barlow Twins: Self-Supervised Learning via Redundancy ReductionarXiv:2103.03230, 2021 Mar 4.



1. Run original and distorted image through same network
2. Compute correlation matrix
3. Add objective to make correlation matrix close to identity matrix

↓

Forces representation vectors of similar samples to be similar

Goyal, Caron, Lefaudeux, Xu, Wang, Pai, Singh, Liptchinsky, Misra, Joulin, Bojanowski. **Self-supervised Pretraining of Visual Features in the Wild**. arXiv:2103.01988, 2021 Mar 2.

- SEER = SElf-supERvised
- new billion-parameter self-supervised computer vision model
- pretraining on a **billion** random, **unlabeled** and uncurated public Instagram images
- self-supervised SOTA: reaching 84.2 percent top-1 accuracy on ImageNet
- SwAV (https://arxiv.org/abs/2006.09882) uses online clustering to rapidly group images with similar visual concepts and leverage their similarities (doesn't need pair-wise comparisons; fast)

Self-Supervised Learning (Text Example)

Input sentence: A quick brown fox jumps over the lazy dog

15% randomly masked: A quick brown [MASK] jumps over the lazy dog

BERT

Possible classes (all words)

0.2%	ant
...	...
11%	fox
...	...
0.01%	zoo

"Old" Language Transformer Models

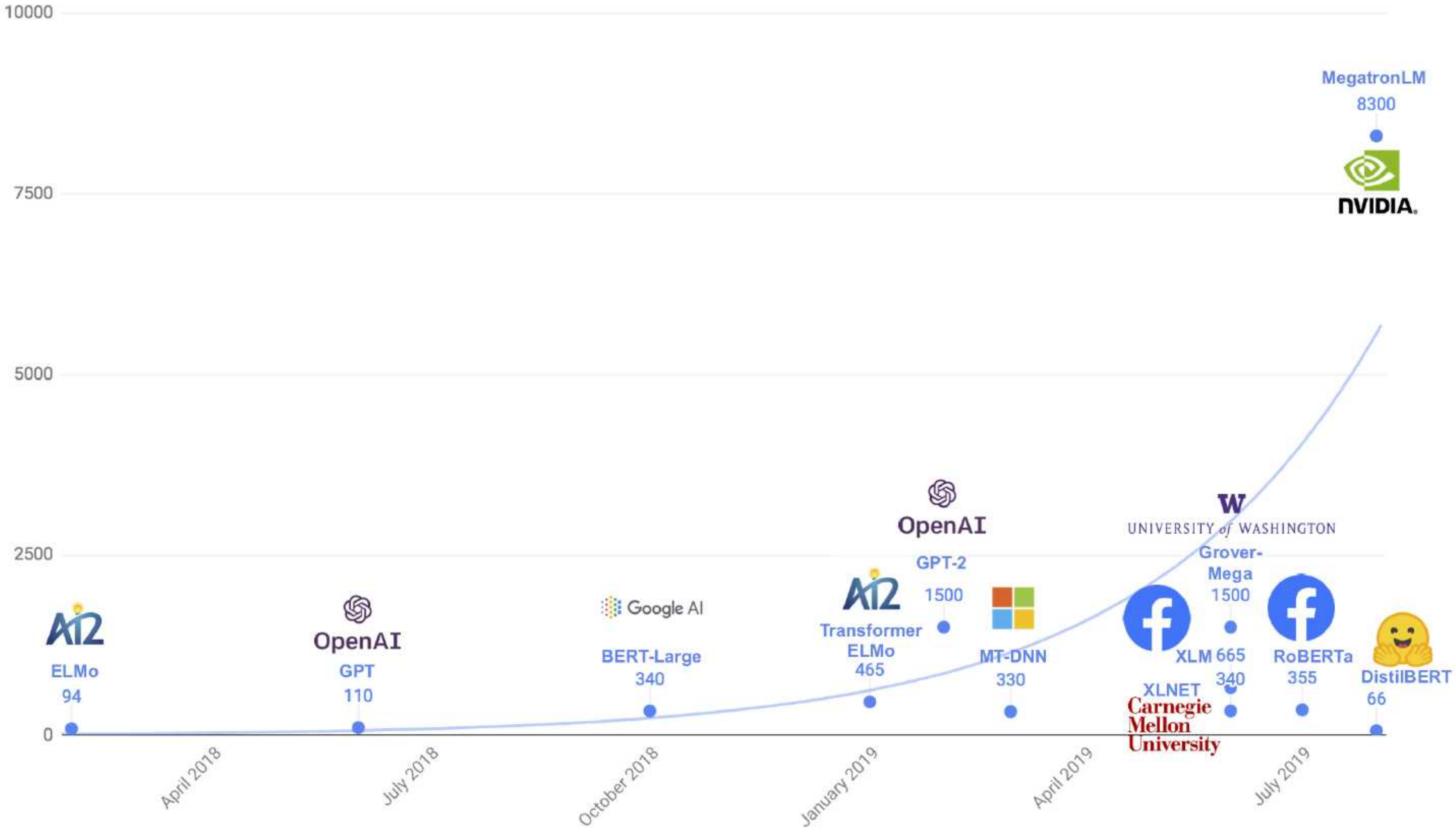


Image Source: https://medium.com/huggingface/distilbert-8cf3380435b5

2

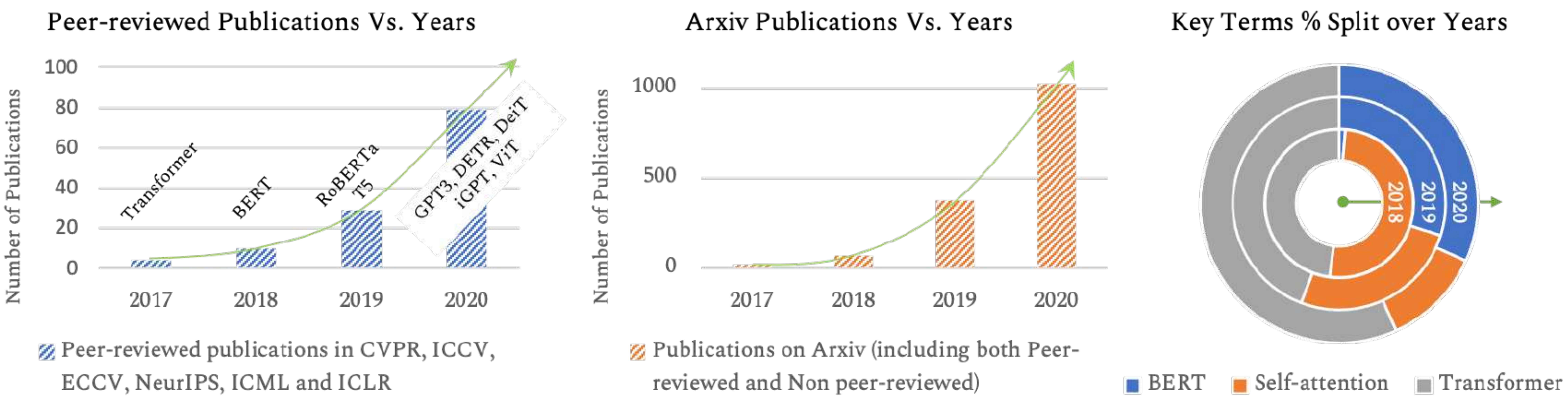


Fig. 1. Statistics on the number of times keywords such as BERT, Self-Attention, and Transformers appear in the titles of Peer-reviewed and arXiv papers over the past few years. The plots show consistent growth in recent literature. We cover this progress in the computer vision domain.

TECH \ ARTIFICIAL INTELLIGENCE

OpenAI's text-generating system GPT-3 is now spewing out 4.5 billion words a day

Robot-generated writing looks set to be the next big thing

By James Vincent | Mar 29, 2021, 8:24am EDT

https://www.theverge.com/2021/3/29/22356180/openai-gpt-3-text-generation-words-day

THE COST OF TRAINING NLP MODELS

A CONCISE OVERVIEW

Or Sharir
AI21 Labs
ors@ai21.com

Barak Peleg
AI21 Labs
barakp@ai21.com

Yoav Shoham
AI21 Labs
yoavs@ai21.com

April 2020

http://arxiv.org/abs/2004.08900

Costs: Not for the faint hearted

- $2.5k - $50k (110 million parameter model)
- $10k - $200k (340 million parameter model)
- $80k - $1.6m (1.5 billion parameter model)

HOME > AI > The Billion Dollar AI Problem That Just Keeps Scaling

THE BILLION DOLLAR AI PROBLEM THAT JUST KEEPS SCALING

February 11, 2021 Nicole Hemsoth

https://www.nextplatform.com/2021/02/11/the-billion-dollar-ai-problem-that-just-keeps-scaling/

MEGATRON SCALING ON NVIDIA'S DGX-A100 CLUSTER

SELENE

- Batch size: 3072
- 2048 tokens sequences
- 48-way data parallel
- Vocabulary size: 51200

Case	Hidden Size	Number of Layers	Model Parallel Size	Number of GPUs
174.6B (GPT-3)	12288	96	64	3072
86.1B	10240	68	32	1536
40.7B	8192	50	16	768
19.4B	6144	42	8	384
8.3B	4096	40	4	192

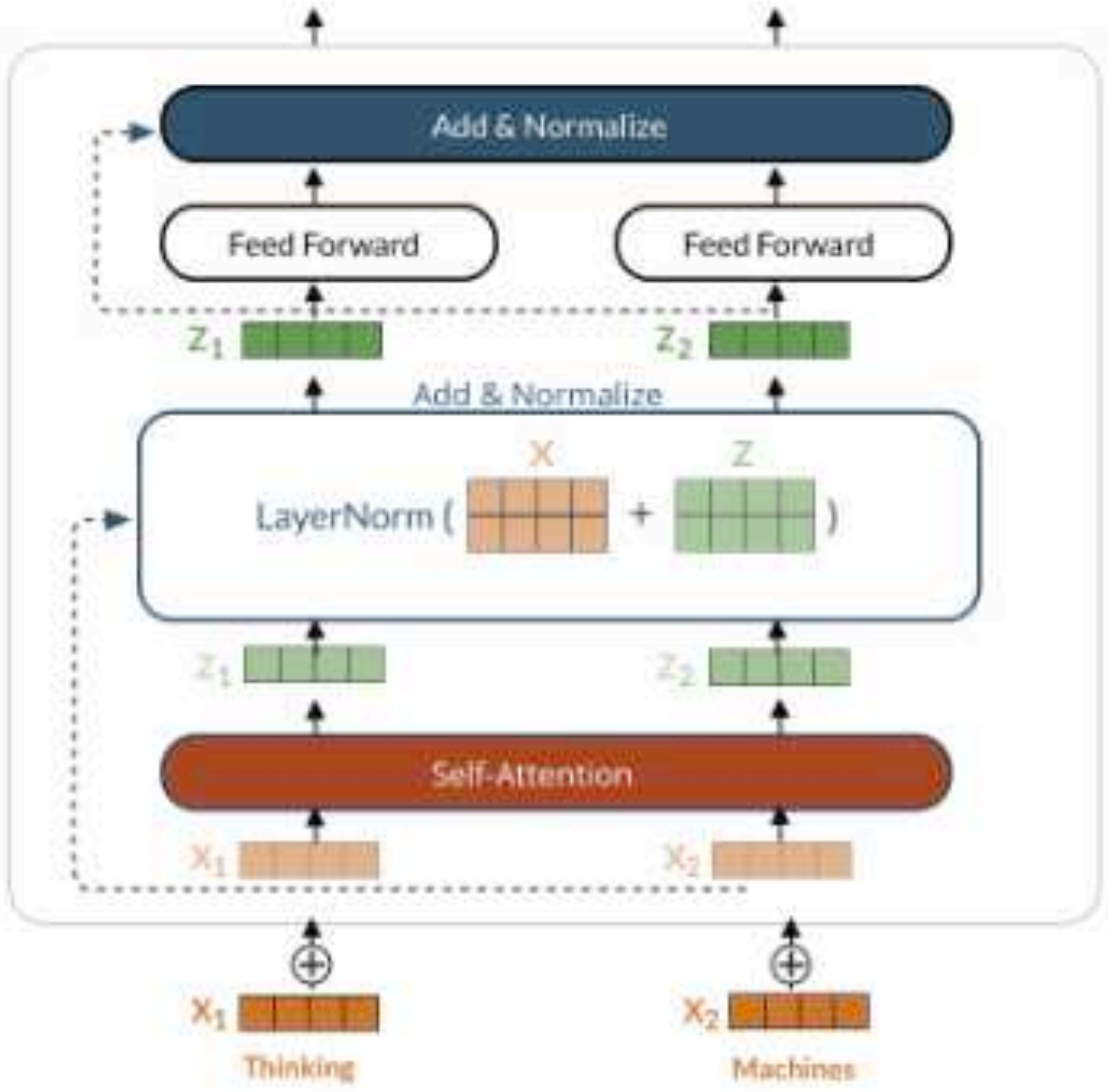
280 DGX-A100 systems, which cost $199,000 each

+15% networking cost of the total cost

+20% storage

List price: 75 million (electricity not included)

EleutherAI Home About Blog Publications Get Involved



GPT-Neo

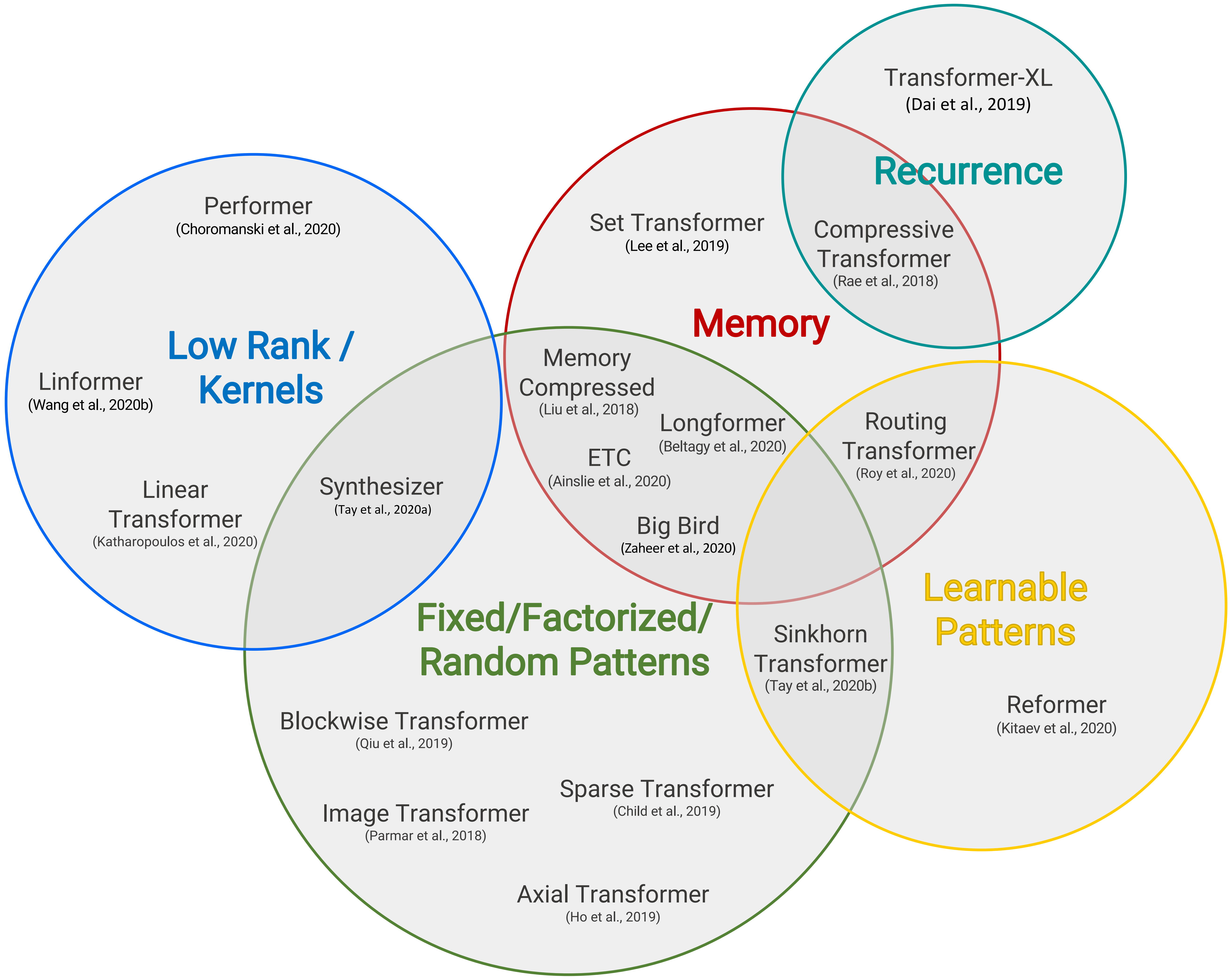
GPT-Neo is the code name for a series of transformer-based language models loosely styled around the GPT architecture that we plan to train and open source. Our primary goal is to replicate a GPT-3 sized model and open source it to the public, for free.

Along the way we will be running experiments with alternative architectures and attention types, releasing any intermediate models, and writing up any findings on our blog.

Our models are built in Tensorflow-mesh, which will allow us to scale up to GPT-3 sizes and beyond using simultaneous model and data parallelism.

https://www.eleuther.ai/projects/gpt-neo/

Training on "The Pile," an **825 GB** language modeling dataset from various sources (YouTube, PubMed, etc.)



Tay, Dehghani, Bahri, Metzler. **Efficient Transformers**: A Survey. arXiv:2009.06732, 2020

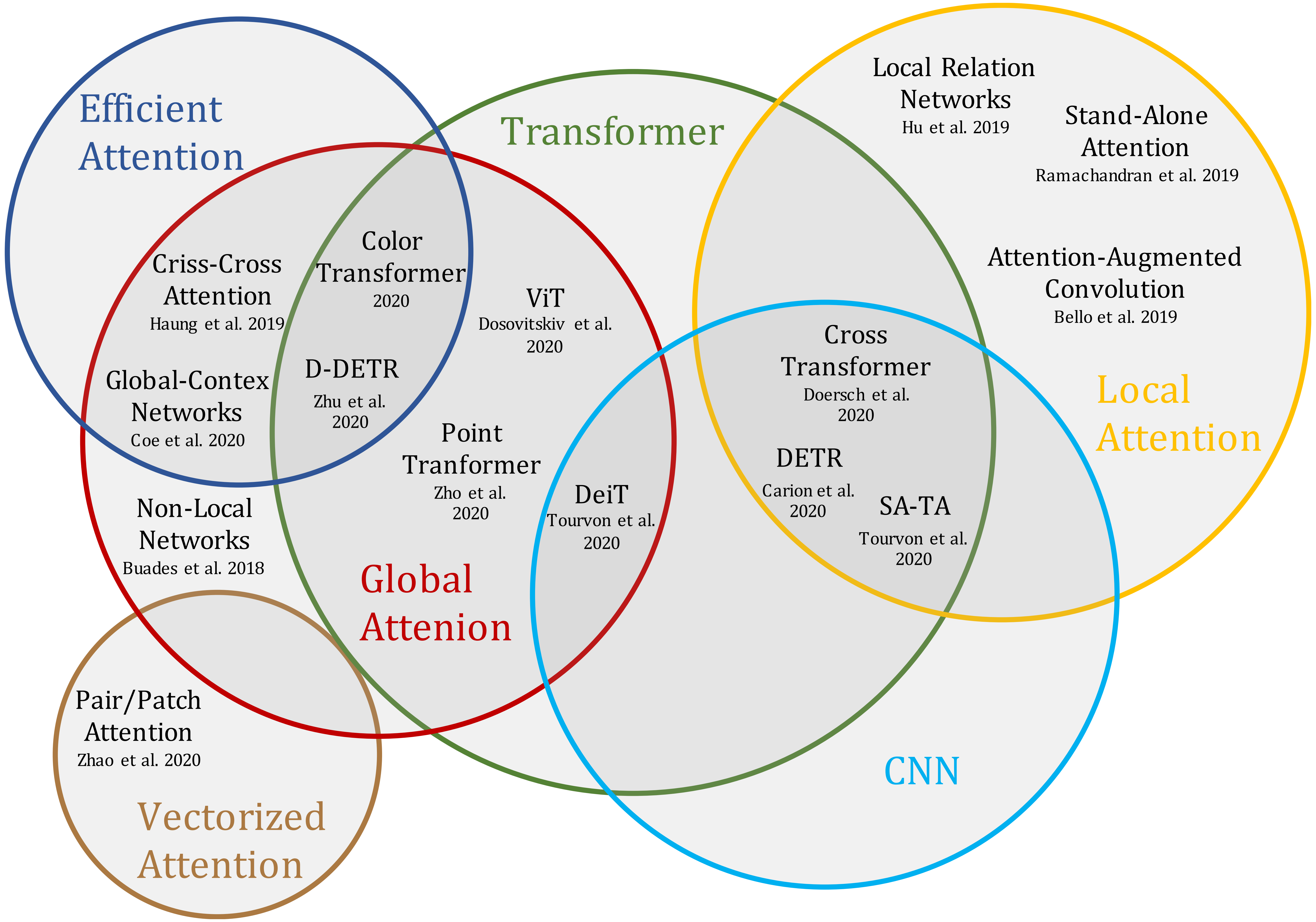


Fig. 3. A taxonomy of self-attention design space.

Khan, Naseer, Hayat, Zamir, Khan, Shah. **Transformers in Vision: A Survey**. arXiv preprint arXiv:2101.01169. 2021 Jan.

arXiv.org > cs > arXiv:2103.01209

Computer Science > Computer Vision and Pattern Recognition

[Submitted on 1 Mar 2021 (v1), last revised 2 Mar 2021 (this version, v2)]

Generative Adversarial Transformers

Drew A. Hudson, C. Lawrence Zitnick

https://arxiv.org/abs/2103.01209

https://github.com/dorarad/gansformer

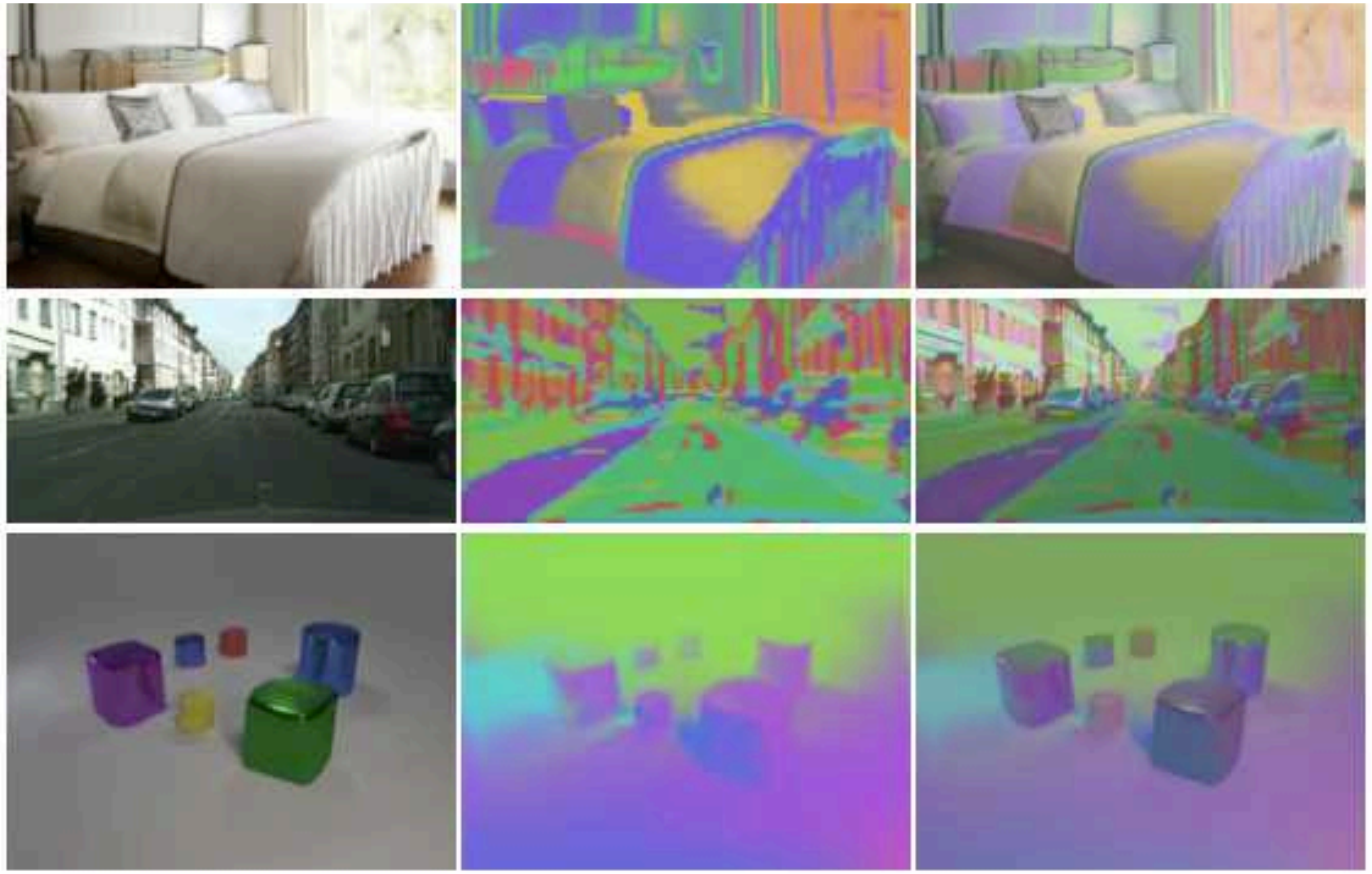
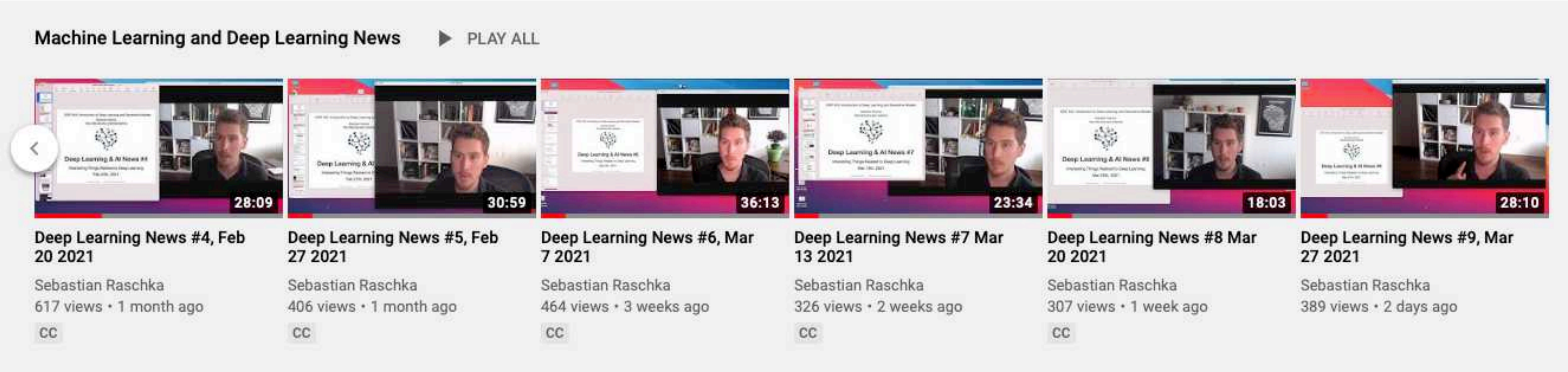


Figure 1. Sample images generated by the GANsformer, along with a visualization of the model attention maps.



https://www.youtube.com/channel/UC_CzsS7UTjcxJ-xXp1ftxtA

https://tinyurl.com/rry9jamd

Contact:

https://sebastianraschka.com

@rasbt

Sebastian Raschka