Google I/O Extended

What's new in AI & ML

#AI, ML & Research

Dr Martin Andrews

GDE ML & Co-Founder of Red Dragon AI

@mdda123 @mdda

About Me

RED DRAGON AI

- Google Developer Expert for Machine Learning and Deep Learning (2017-2022)

- Deep Learning R&D :
  - Language & Dialogue systems
  - Generative Models
  - Text-to-Speech

- MeetUp Co-organiser:
  - "Machine Learning Singapore"

Martin Andrews

Google Developers Experts

RED DRAGON AI

About Red Dragon AI

- Founded 2017
- Google Partner
- Consulting, Prototyping & Building
- Research - NeurIPS, EMNLP, COLING, NAACL
- Interactive Digital Personas

NEW

Three ML Topics for Today

1. Coral Micro Dev board
2. Models for Language and Images at Google
3. Jax/Flax : A New ML Stack

Coral

On-Device Edge Machine Learning

The problem and a challenge

For deployment of AI at the edge, we need something new...

- Small enough to be easily installed everywhere
- Use very little power to continue monitoring the change in scenes using a ML mode
- Upon the detection of something present in the scene, be able to wake up and perform high-power ML acceleration for another ML model right on the device

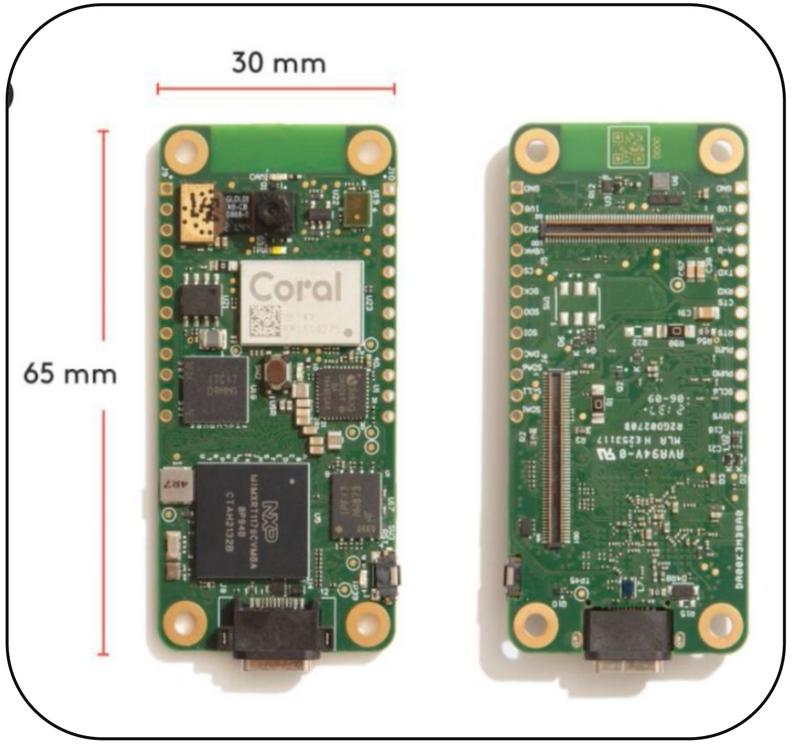
Is there anything existing that can do all of these?

NEW New product launched!

Coral Dev Board Micro

- Microcontroller-class device for low-power consumption with the Coral Accelerator Module (Edge TPU) on board
- **Dual-mode application flexibility:**

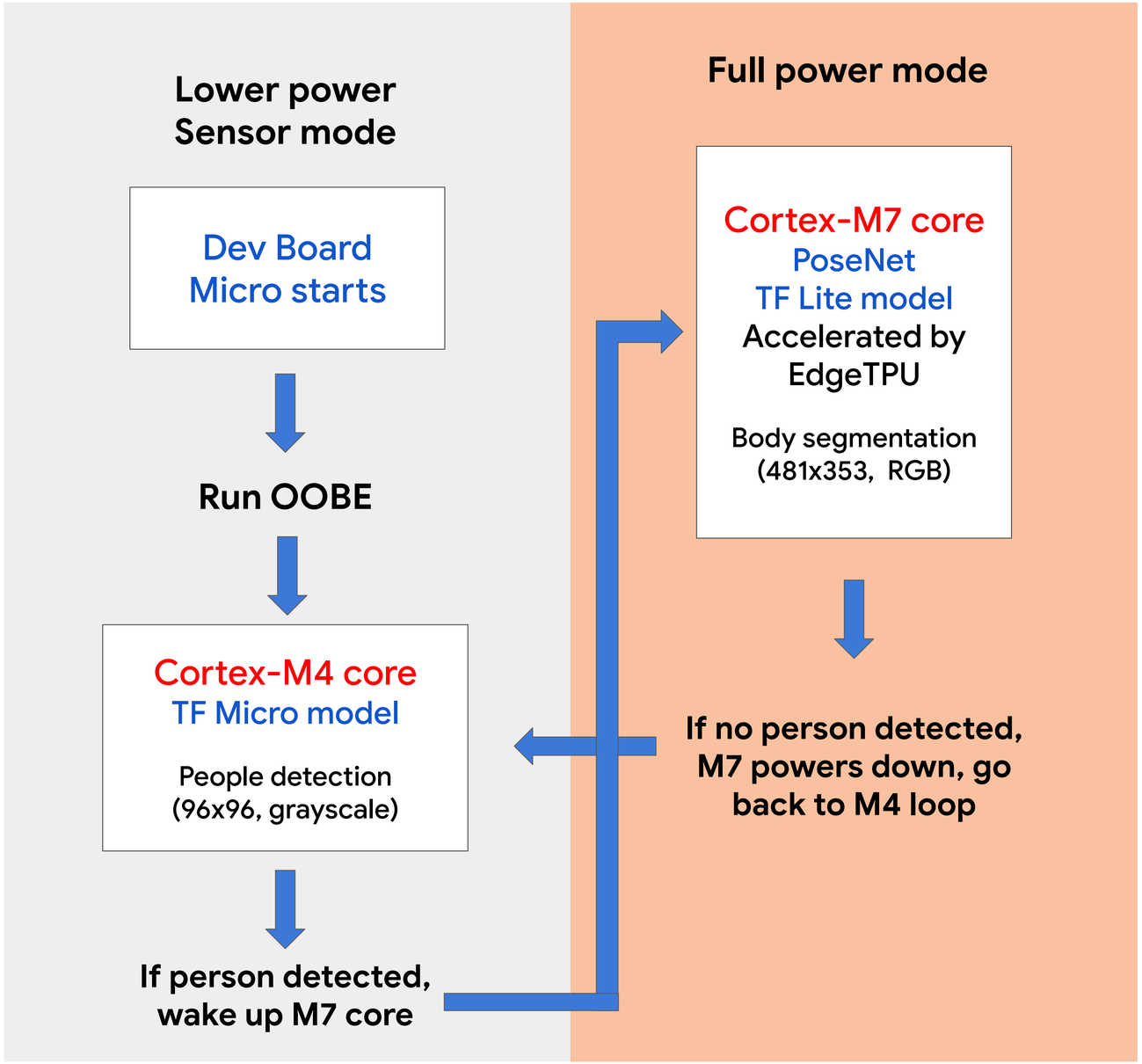
  Includes a NXP i.MX RT 1176 crossover MCU, with ARM Cortex-M7 and M4 cores
  - The M4 core can run lightweight **TF Lite Micro** models for detection & triggering accelerated ML
  - The M7 core to run high-performance **TF Lite** models accelerated by the Edge TPU for enhanced ML performance for application needs



What makes it unique?

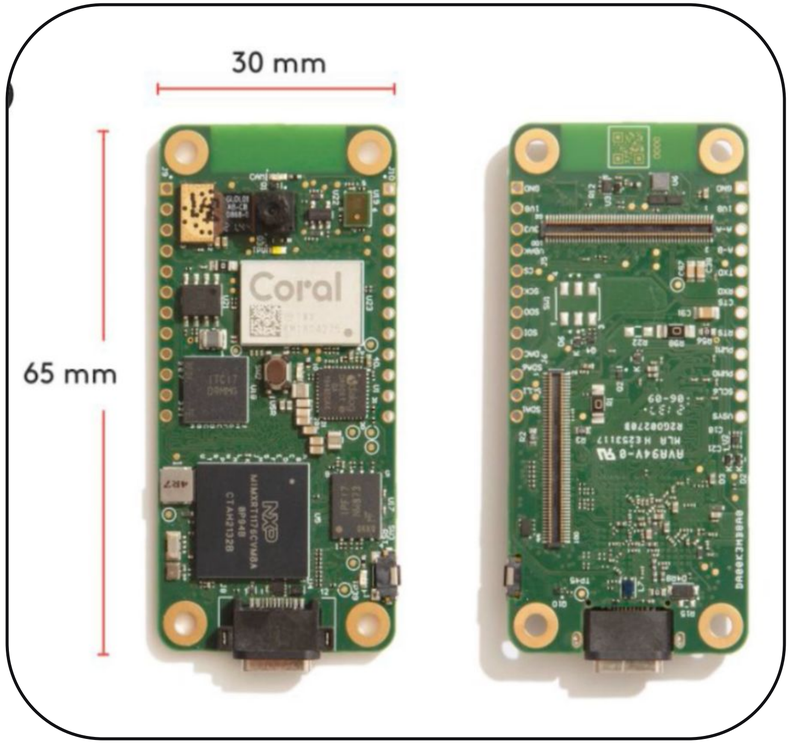
A unique capability only available on this new Coral Dev Board Micro product:

- Dual mode operation and two ML models running in tandem



Coral Dev Board Micro

- **Tiny form factor** for wide deployment - which can be **battery powered**
- Runs **FreeRTOS** with multi-threaded support and compatibility with the **Arduino SDK**
- Support for the new low-powered **TF Lite Micro** models for microcontroller based ML applications
- **On-board camera & microphone** for vision and audio ML applications
  - 324x324 pixel color camera
  - 110° field of view (FoV), aperture f/2.0

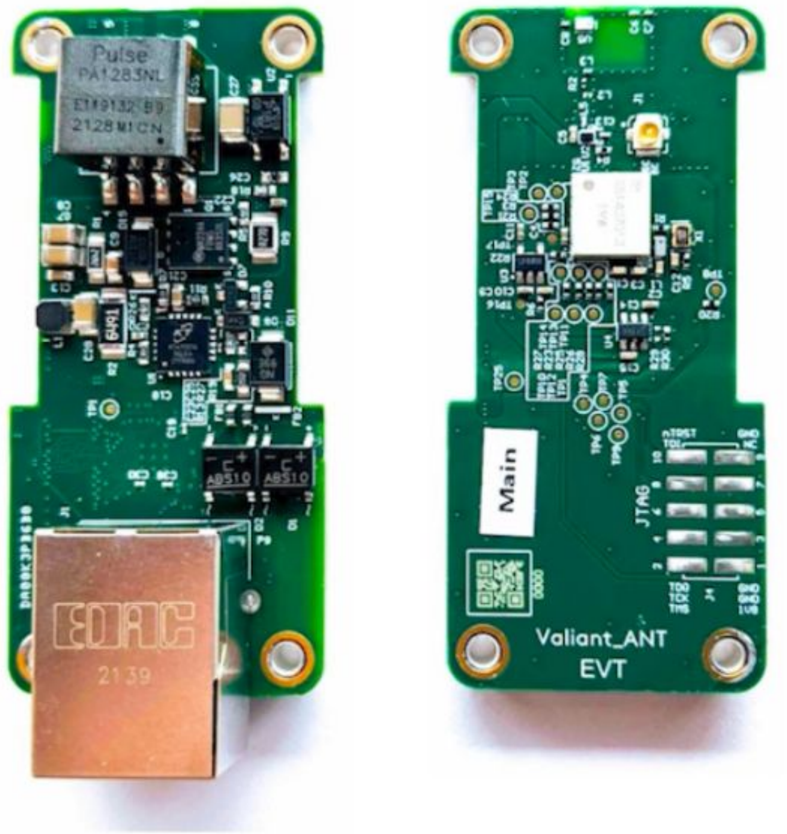


Custom expandability

Support add-on extension boards

- Coral provides a WiFi Accessory expansion board for network & Internet connectivity
- Coral also provides a PoE (power over Ethernet) Accessory board for using network power if connected to a network
- Supports open-source connectivity standards

We encourage developers and businesses to develop many other application specific extension boards and accessories!



Application development & software use

Linux required on the developer's host machine

✔ Windows and Mac support will be added in the future

Two development paths:

✔ CMake scripts for bare C++ running on FreeRTOS

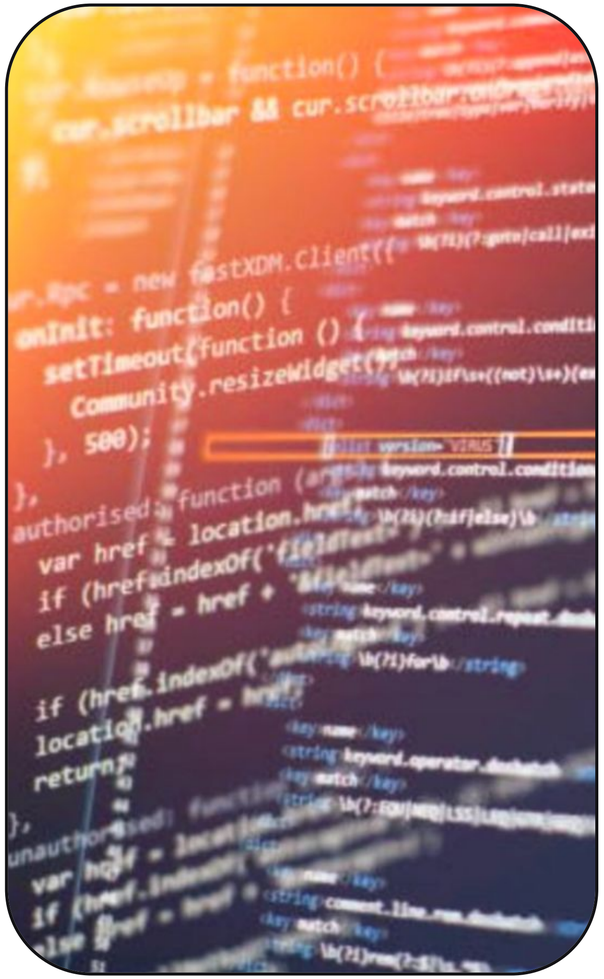
✔ arduino-cli supporting Arduino-style sketches on FreeRTOS

The out-of-the-box experience (OOBE) demo showcase the key capabilities:

✔ Low-power person detector model on M4 core

✔ High-accuracy PoseNet on M7 core

✔ Basic power-state control

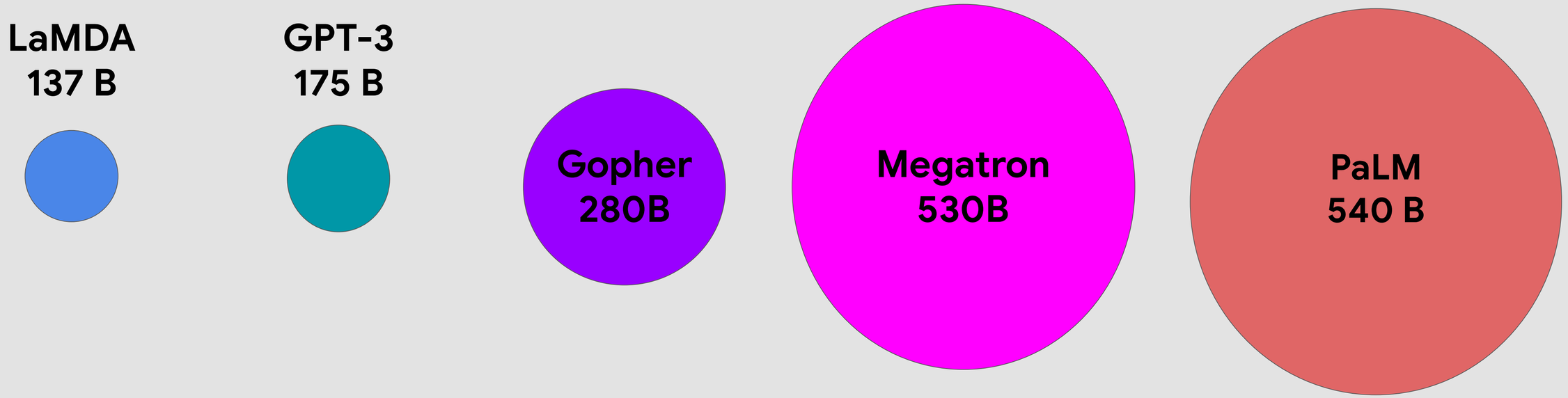


Large Language Models & Research

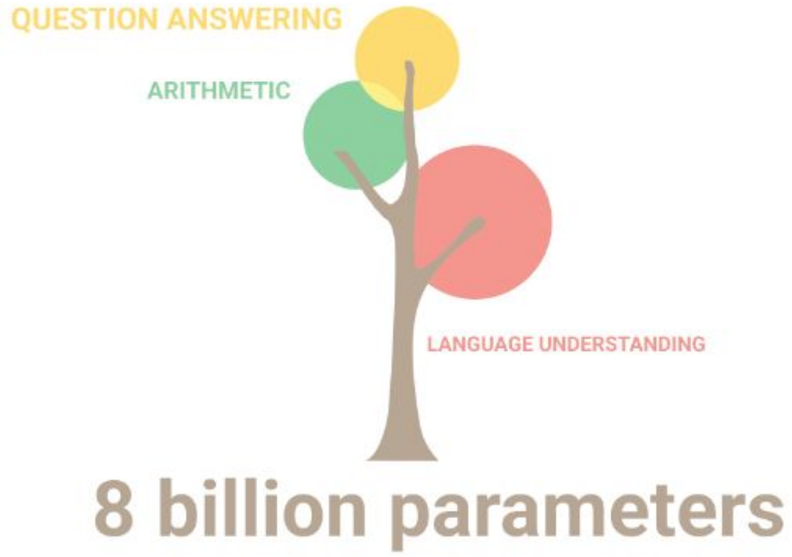
What is a Language Model?

Autoregressive Language Model

Dense LM Model Sizes



PaLM - 540 billion parameters



PaLM Pathways

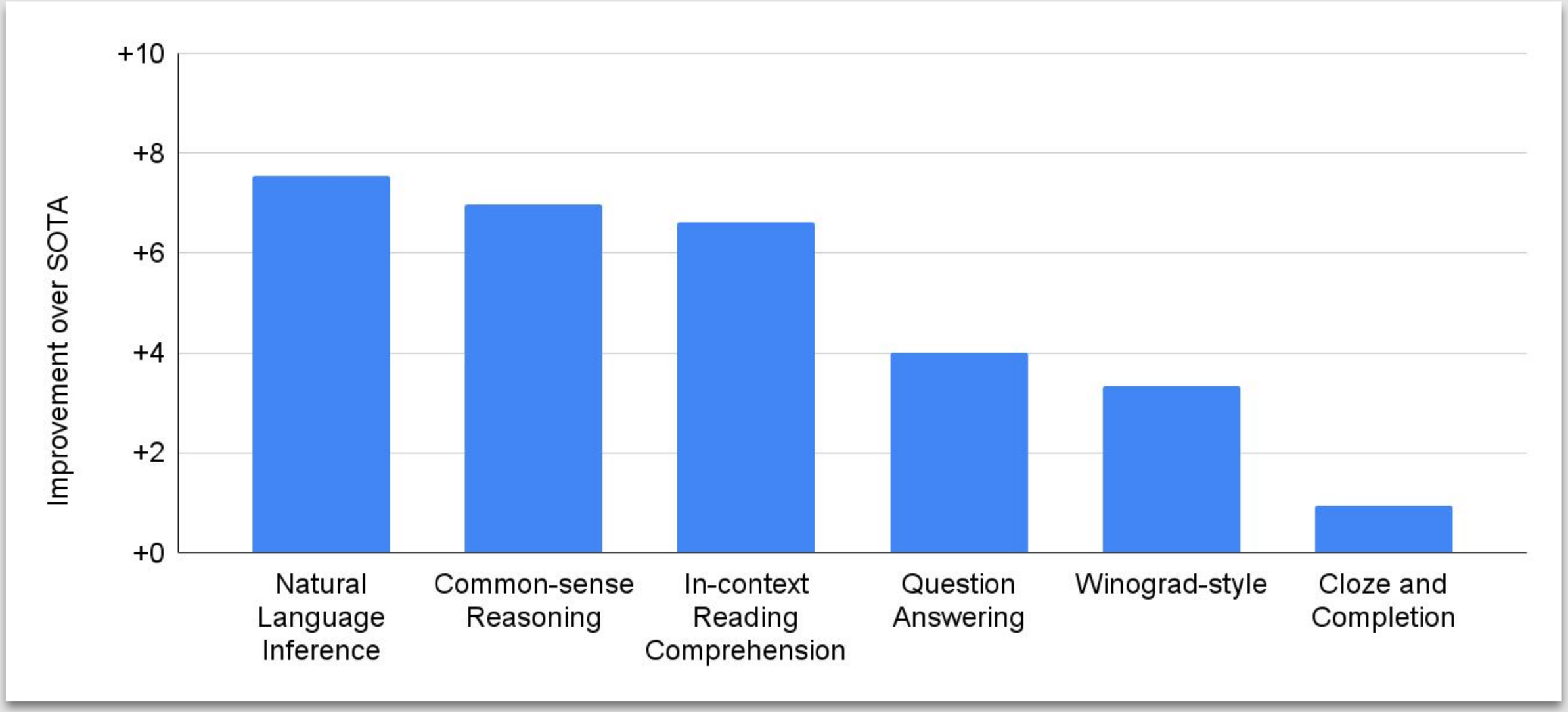
PaLM: Scaling Language Modeling with Pathways

Aakanksha Chowdhery* Sharan Narang* Jacob Devlin*
Maarten Bosma Gaurav Mishra Adam Roberts Paul Barham
Hyung Won Chung Charles Sutton Sebastian Gehrmann Parker Schuh Kensen Shi
Sasha Tsvyashchenko Joshua Maynez Abhishek Rao† Parker Barnes Yi Tay
Noam Shazeer‡ Vinodkumar Prabhakaran Emily Reif Nan Du Ben Hutchinson
Reiner Pope James Bradbury Jacob Austin Michael Isard Guy Gur-Ari
Pengcheng Yin Toju Duke Anselm Levskaya Sanjay Ghemawat Sunipa Dev
Henryk Michalewski Xavier Garcia Vedant Misra Kevin Robinson Liam Fedus
Denny Zhou Daphne Ippolito David Luan‡ Hyeontaek Lim Barret Zoph
Alexander Spiridonov Ryan Sepassi David Dohan Shivani Agrawal Mark Omernick
Andrew M. Dai Thanumalayan Sankaranarayana Pillai Marie Pellat Aitor Lewkowycz
Erica Moreira Rewon Child Oleksandr Polozov† Katherine Lee Zongwei Zhou
Xuezhi Wang Brennan Saeta Mark Diaz Orhan Firat Michele Catasta† Jason Wei
Kathy Meier-Hellstern Douglas Eck Jeff Dean Slav Petrov Noah Fiedel

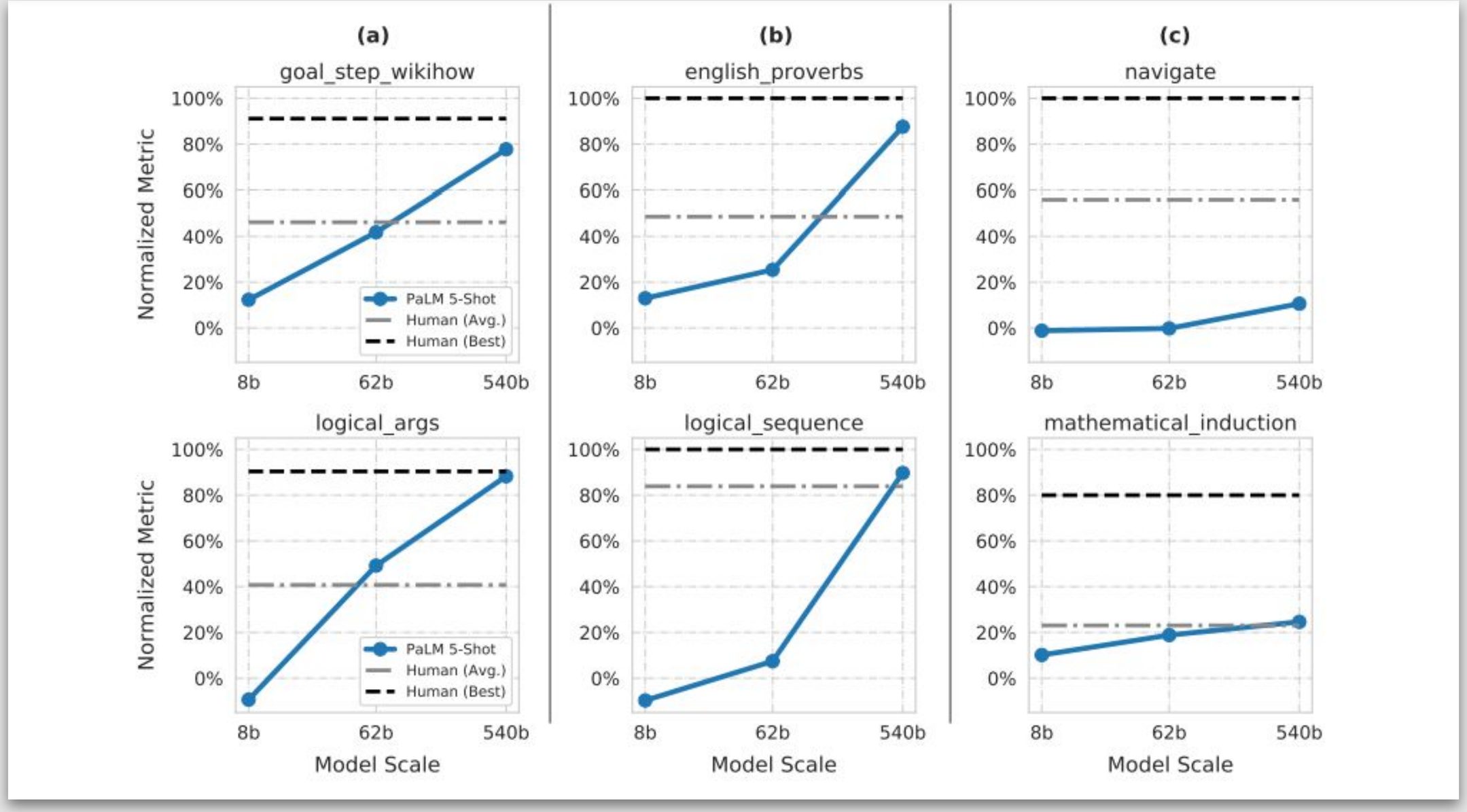
Google Research

https://arxiv.org/pdf/2204.02311.pdf

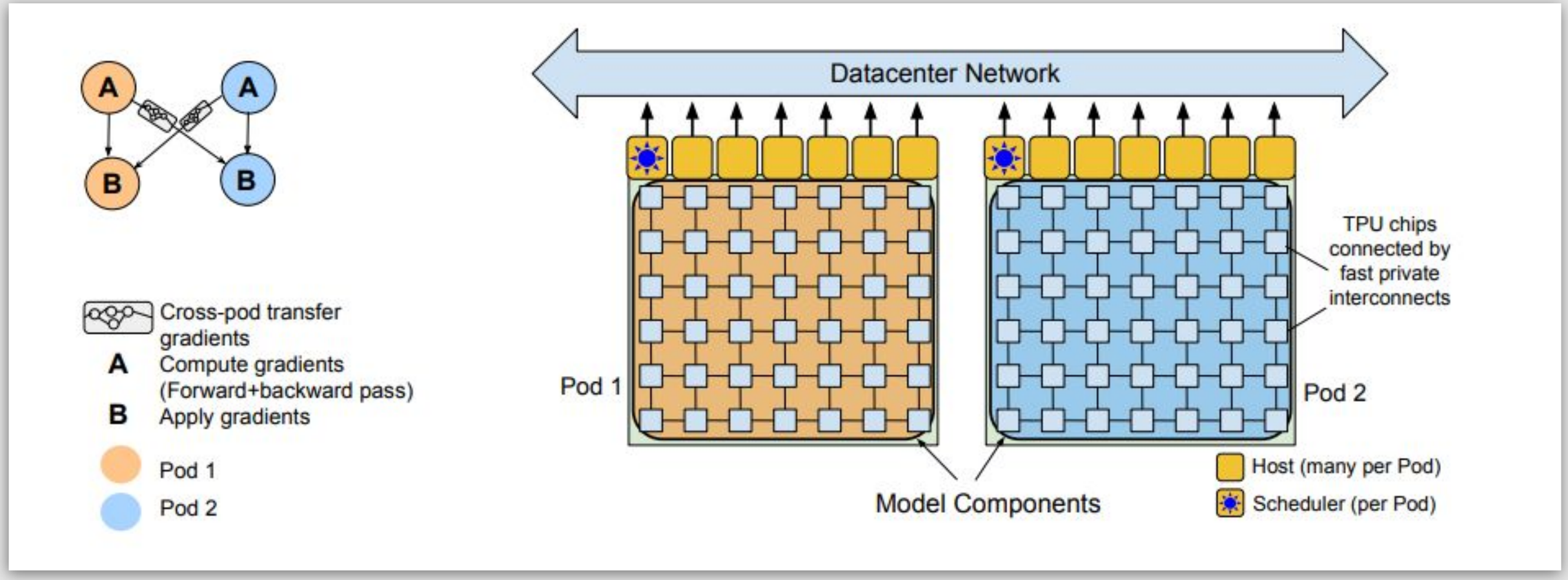
PaLM State of the art results



Suddenly more capable...



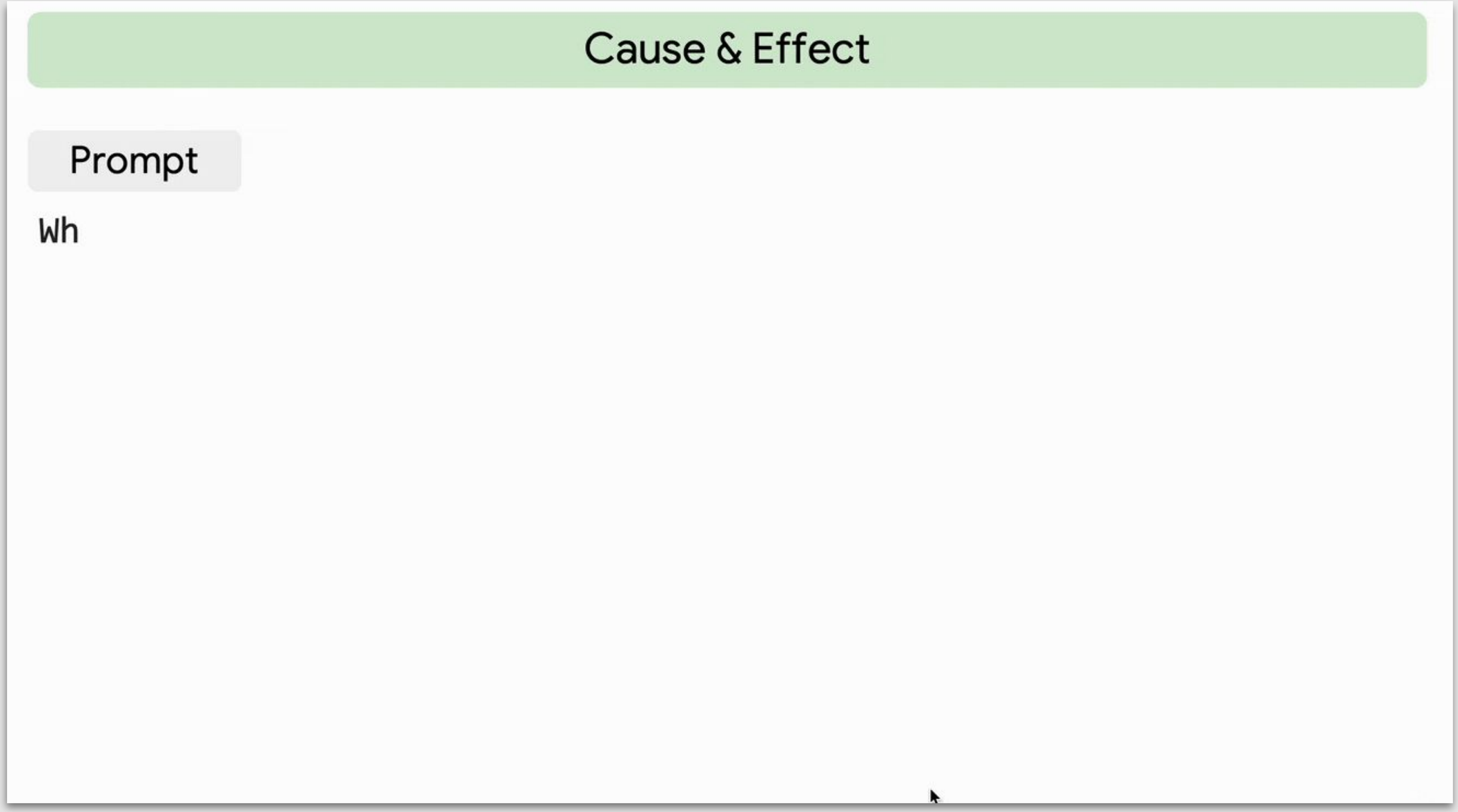
Training PaLM - TPUv4 Pods



6144 - TPUv4 cores

Prompting

Suddenly more capable...



Explaining a joke

Input

Prompt

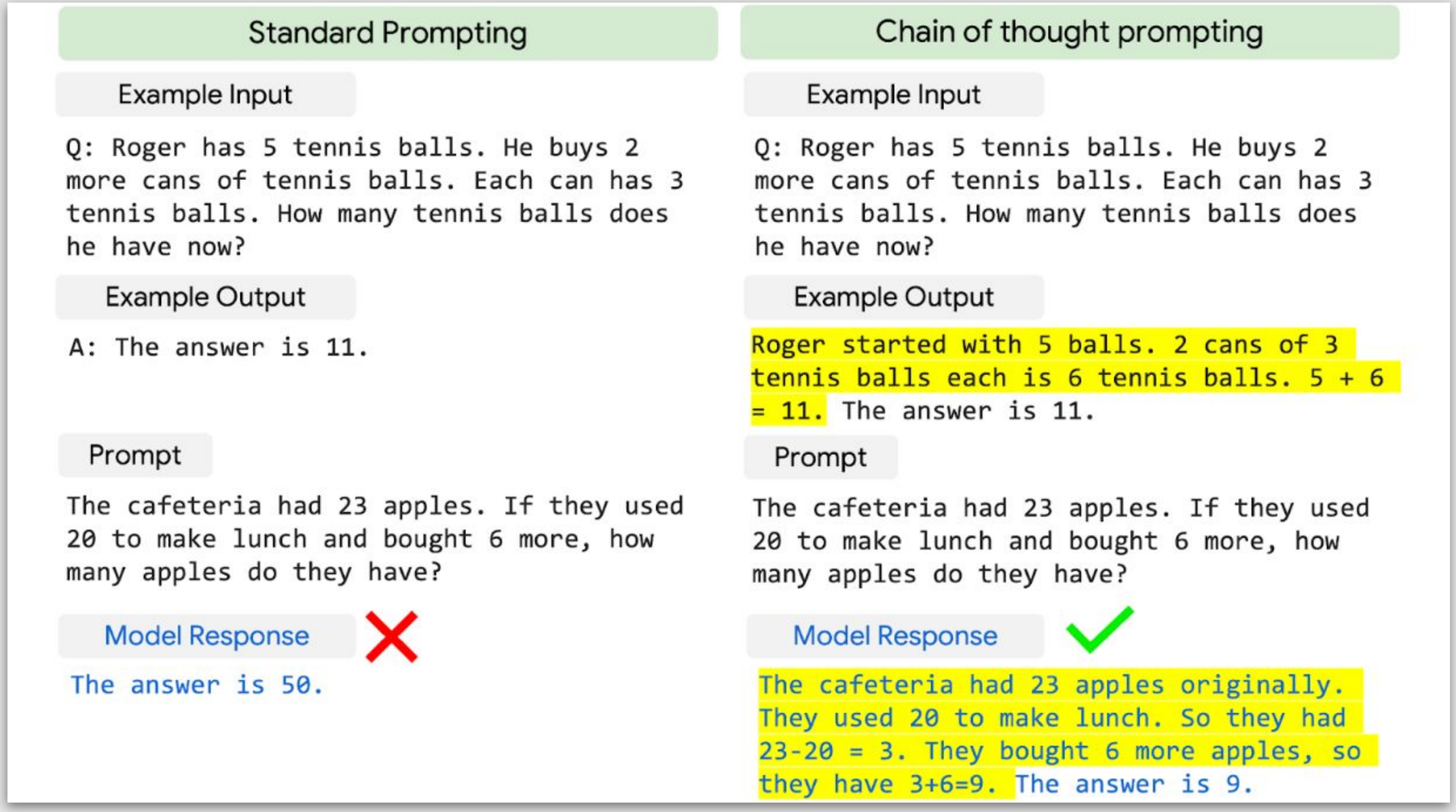
```
Explain this joke:

Joke: Did you see that Google just hired an eloquent whale for
their TPU team? It showed them how to communicate between two
different pods!
```

Output

Model Response

```
Prediction: TPUs are a type of computer chip that Google uses
for deep learning. A "pod" is a group of TPUs. A "pod" is also a
group of whales. The joke is that the whale is able to
communicate between two groups of whales, but the speaker is
pretending that the whale is able to communicate between two
groups of TPUs.
```

Chain of thought prompting



Program Synthesis

Prompt

```
// Translate from C to Python
int add_one ( int x ){
 int m = 1;
 while ( x & m ) {
   x = x ^ m;
   m <<= 1;
 }
 x = x ^ m;
 return x; }
```

Model Response

LaMDA

Conversational A.I.

Google Models:

Meena (2020)

LaMDA (2021)

LaMDA 2 (2022)

LaMDA

- 57 Authors
- Built on a **Transformer Architecture**
- Trained to mimic **Real World Conversations**
- **Retrieval-aided** LM for Dialog
- Ask-an-expert
- Metrics of groundedness - Truth
- 2Bn, 8Bn, 137Bn parameters
- 1.12Bn Dialogs (13.39Bn utterances)

LaMDA

LaMDA: Language Models for Dialog Applications

Romal Thoppilan Daniel De Freitas * Jamie Hall Noam Shazeer * Apoorv Kulshreshtha

Heng-Tze Cheng Alicia Jin Taylor Bos Leslie Baker Yu Du YaGuang Li Hongrae Lee

Huaixiu Steven Zheng Amin Ghafouri Marcelo Menegali Yanping Huang Maxim Krikun

Dmitry Lepikhin James Qin Dehao Chen Yuanzhong Xu Zhifeng Chen Adam Roberts

Maarten Bosma Vincent Zhao Yanqi Zhou Chung-Ching Chang Igor Krivokon Will Rusch

Marc Pickett Pranesh Srinivasan Laichee Man Kathleen Meier-Hellstern

Meredith Ringel Morris Tulsee Doshi Renelito Delos Santos Toju Duke Johnny Soraker

Ben Zevenbergen Vinodkumar Prabhakaran Mark Diaz Ben Hutchinson Kristen Olson

Alejandra Molina Erin Hoffman-John Josh Lee Lora Aroyo Ravi Rajakumar

Alena Butryna Matthew Lamm Viktoriya Kuzmina Joe Fenton Aaron Cohen

Rachel Bernstein Ray Kurzweil Blaise Aguera-Arcas Claire Cui Marian Croak Ed Chi

Quoc Le

https://arxiv.org/pdf/2201.08239.pdf

NEW

LaMDA 2

- **No Paper yet**
  - LaMDA (1) paper came out this year (work was done a while back)
- **Most likely a bigger** PaLM-style model focused on dialog
- (?) Making use of some of the better training techniques ~ Chinchilla
- **Different kinds of prompting tasks**
  - "AI Test Kitchen" available for trying it out... (YMMV)

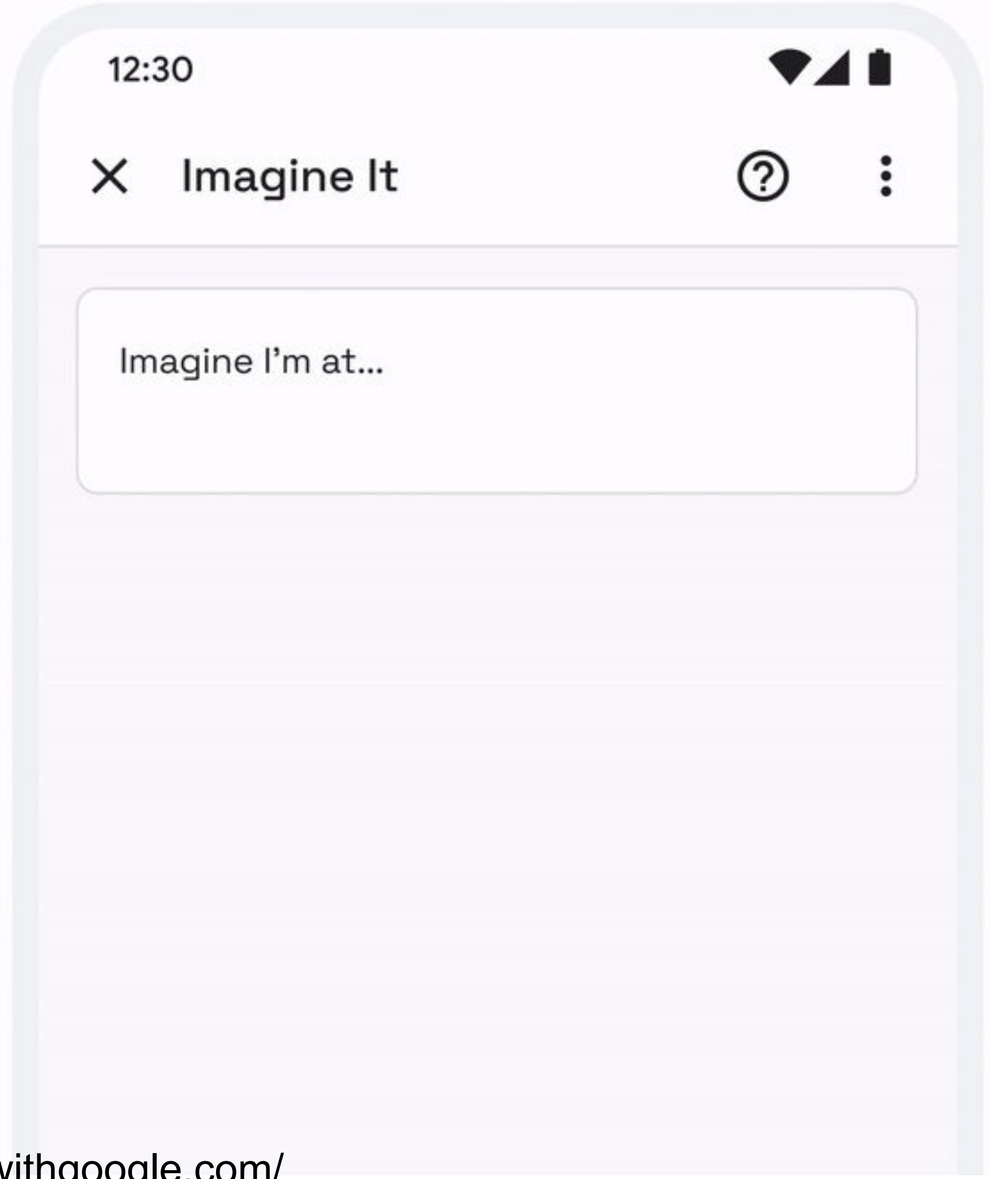
Imagine It

What you can do

Name a place and LaMDA will offer paths to explore your imagination.

What you can give feedback on:

If LaMDA generates interesting scene descriptions that are relevant to your idea.



https://aitestkitchen.withgoogle.com/

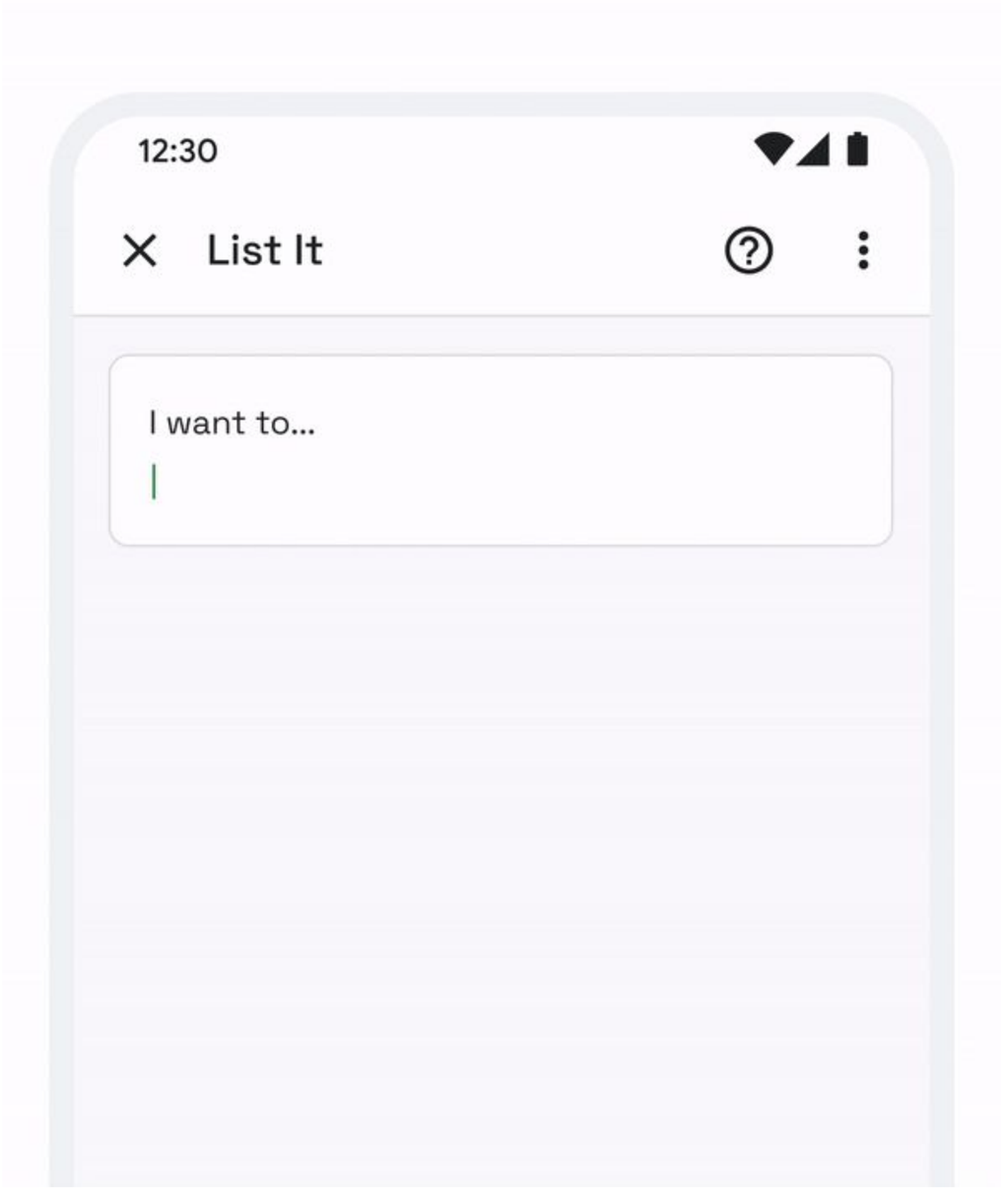
List It

What you can do

Name a goal or topic and see how much LaMDA can break it down into multiple lists of subtasks.

What you can give feedback on:

If LaMDA generates useful lists of subtasks, some of which you might not have thought of.



https://aitestkitchen.withgoogle.com/

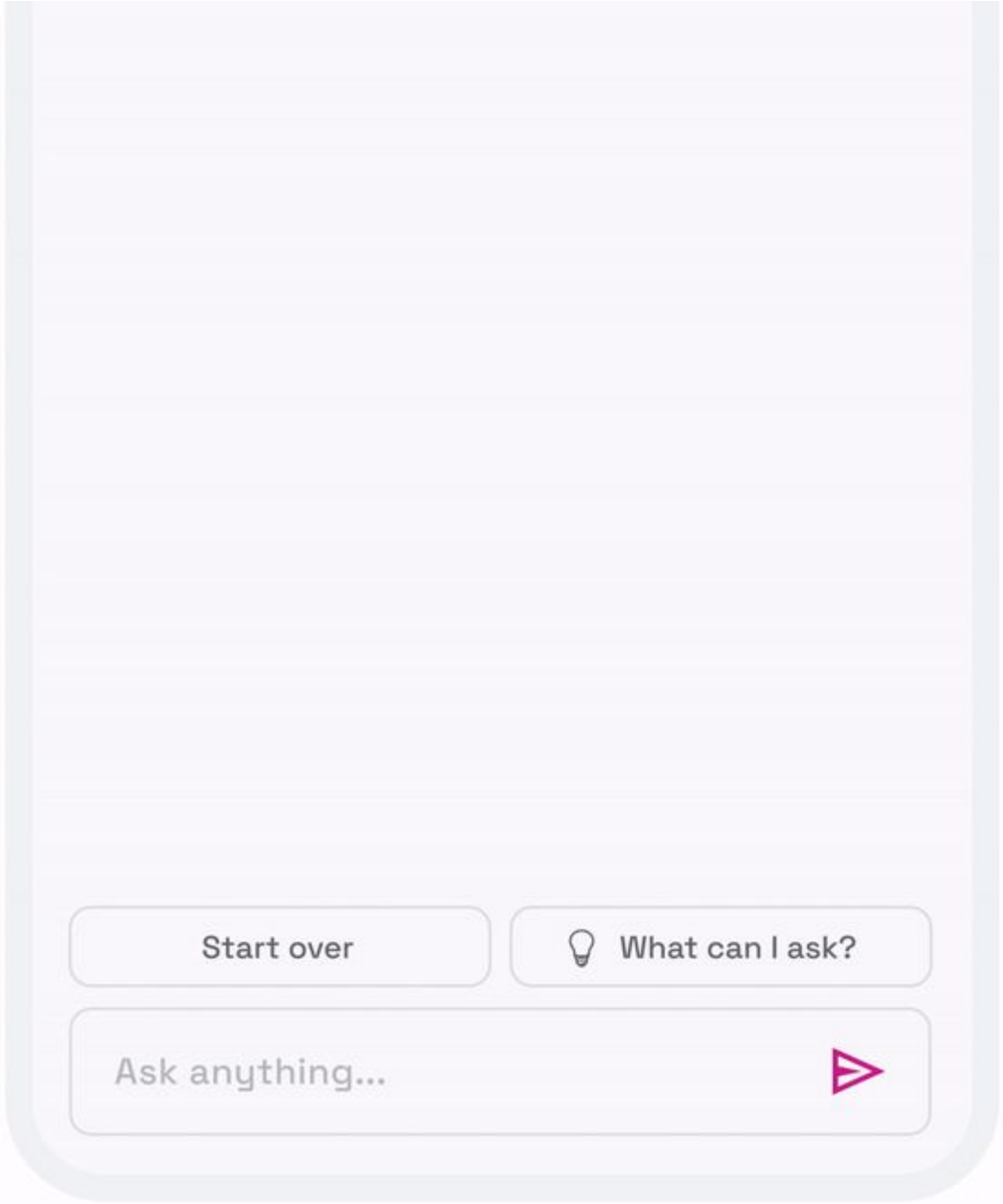
Talk About It (Dogs Edition)

What you can do

Roll with the conversation and see where it goes. It's just a fun, kinda-weird, open-ended chat.

What you can give feedback on:

If LaMDA, no matter what you ask it, keeps the conversation going while bringing the topic back to dogs.



https://aitestkitchen.withgoogle.com/

Imagen

NEW

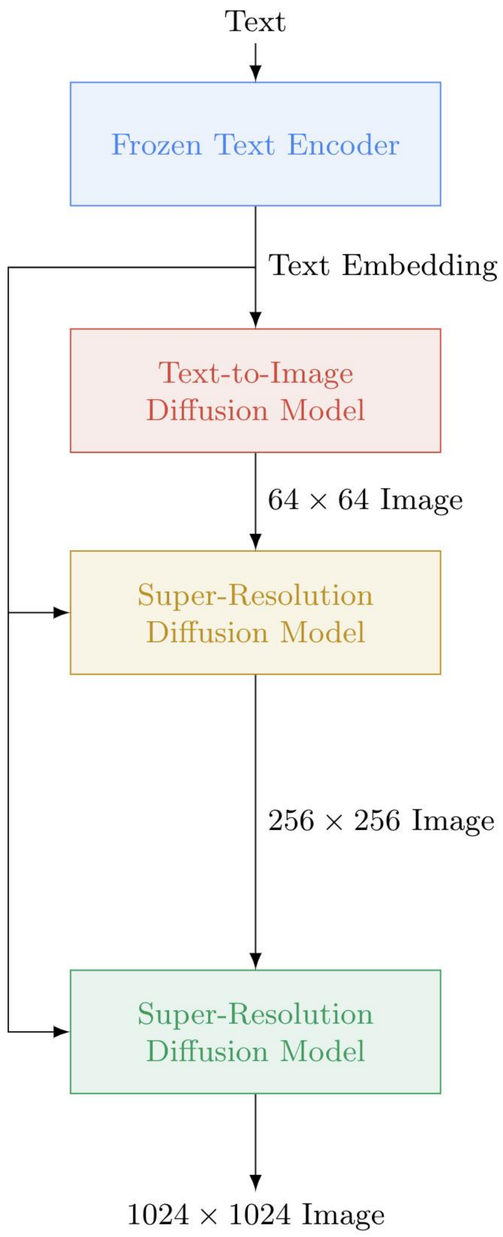
Imagen

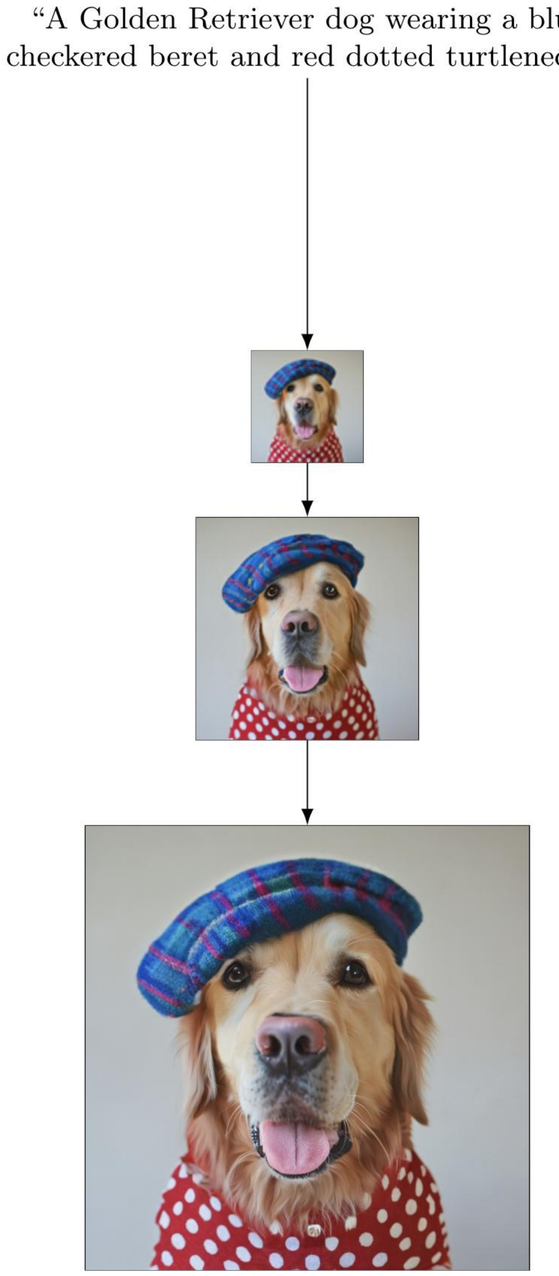
- **Imagen** is an AI system that creates **photorealistic** images from input text
- Uses **Text Encoders** + **Diffusion models** for generation
- Language models use as Text Encoders

A photo of a Corgi dog riding a bike in
Times Square. It is wearing sunglasses and a beach hat.

NEW

Imagen





https://imagen.research.google/

Imagen

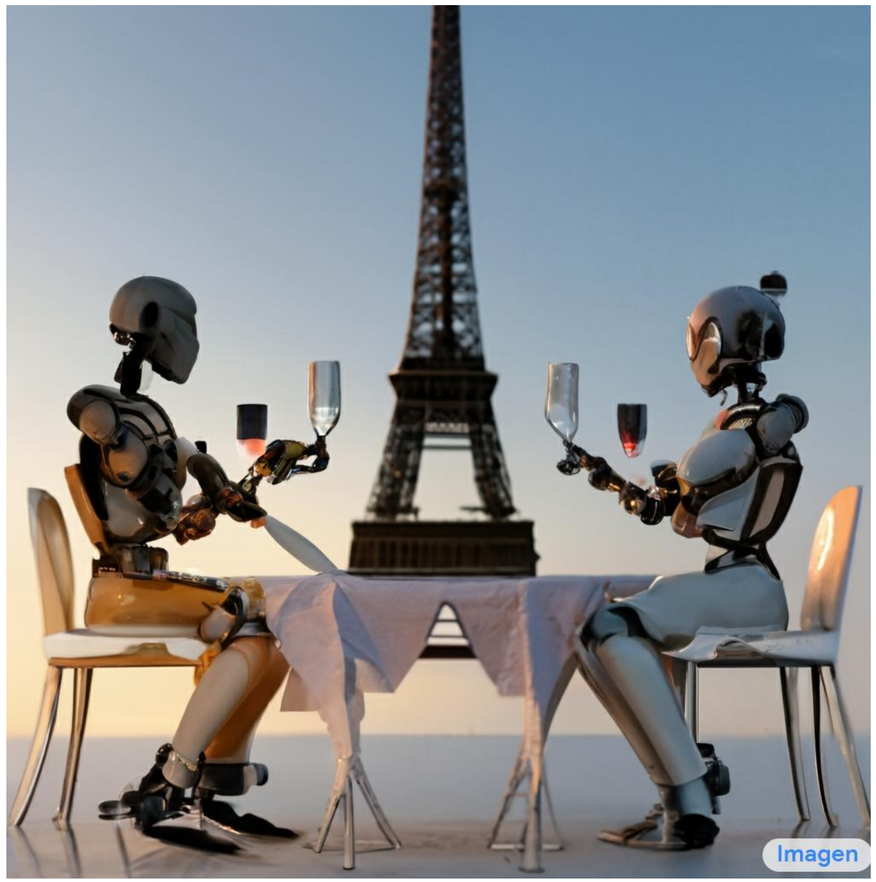
A single beam of light enter the room from the ceiling. The beam of light is illuminating an easel. On the easel there is a Rembrandt painting of a raccoon.

A majestic oil painting of a raccoon Queen wearing red French royal gown. The painting is hanging on an ornate wall decorated with wallpaper.

A photo of a raccoon wearing an astronaut helmet, looking out of the window at night.

The Toronto skyline with Google brain logo written in fireworks.

A marble statue of a Koala DJ in front of a marble statue of a turntable. The Koala has wearing large marble headphones.



A robot couple fine dining with Eiffel Tower in the background.

Jax and Flax

Deep Learning Frameworks: Timeline

1995

Numpy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays

2002

Torch is an open-source machine learning library, a scientific computing framework, and a script language based on the Lua programming language

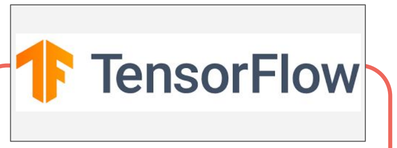
2007

Theano is a Python library and optimizing compiler for manipulating and evaluating mathematical expressions, especially matrix-valued ones.

2015

The initial release was March 2015, Keras is an open-source software library that provides a Python interface for artificial neural networks.

Deep Learning Frameworks: Timeline



2015

Released on November 2015, TensorFlow is a free and open-source software library for machine learning and artificial intelligence

2016

PyTorch is an open source machine learning framework based on the Torch library, used for applications such as computer vision and natural language processing, primarily developed by Meta AI

2017

Caffe (Convolutional Architecture for Fast Feature Embedding) is a deep learning framework, originally developed at University of California, Berkeley

2019

JAX is Autograd and XLA, brought together for high-performance machine learning research.

2020

Flax was originally started by engineers and researchers within the Brain Team in Google Research (in close collaboration with the JAX team), and is now developed jointly with the open source community.

Kinda NEW

JAX

- **Introduced in 2019 -** Took off in 2021
- **Open source framework** - Moreso than TensorFlow
- **Autograd** - Automatic differentiation
- **Numpy** on accelerators (CPU/GPU/TPU)
- **XLA** - Compiles native Python and Numpy code

JAX vs Numpy

```python
import jax.numpy as jnp

array = jnp.arange(5, dtype=jnp.int32).reshape(-1, 1)
print(f"Shape of array: {array.shape} Array type: {type(array)}")
print("Minimum element in array: ", array.min())
print("Maximum element in array: ", array.max())
print("Dot product:")
print(jnp.dot(array, array.T))
```

```
Shape of array: (5, 1) Array type: <class 'jaxlib.xla_extension.DeviceArray'>
Minimum element in array:  0
Maximum element in array:  4
Dot product:
[[ 0  0  0  0  0]
 [ 0  1  2  3  4]
 [ 0  2  4  6  8]
 [ 0  3  6  9 12]
 [ 0  4  8 12 16]]
```

```python
import numpy as np

array = np.arange(5, dtype=np.int32).reshape(-1, 1)
print(f"Shape of array: {array.shape} Array type: {type(array)}")
print("Minimum element in array: ", array.min())
print("Maximum element in array: ", array.max())
print("Dot product:")
print(np.dot(array, array.T))
```

```
Shape of array: (5, 1) Array type: <class 'numpy.ndarray'>
Minimum element in array:  0
Maximum element in array:  4
Dot product:
[[ 0  0  0  0  0]
 [ 0  1  2  3  4]
 [ 0  2  4  6  8]
 [ 0  3  6  9 12]
 [ 0  4  8 12 16]]
```

JIT : Just-in-Time compilation via Tracing

```python
@jit
def squared(x):
    return x ** 2

scalar_input = 2
vector_input = jnp.arange(5, dtype=jnp.int32)

print(f"Scalar input: {scalar_input} => {squared(scalar_input)}")
print(f"Vector input: {vector_input} => {squared(vector_input)}")
```

```
Scalar input: 2 => 4
Vector input: [0 1 2 3 4] => [ 0  1  4  9 16]
```

Tracing a function

- Output depends only on inputs
- Same input always results in same output
- Optimization
- Fusing of Ops

NEW

Flax

- **Library on top of JAX - built for Flexibility**
- A high level Neural Network API for JAX
- Same kind 'level' as PyTorch or Keras
- Allows for **full reproducibility** of NN models
  - Careful handling of PRNGs

NEW

Linen API

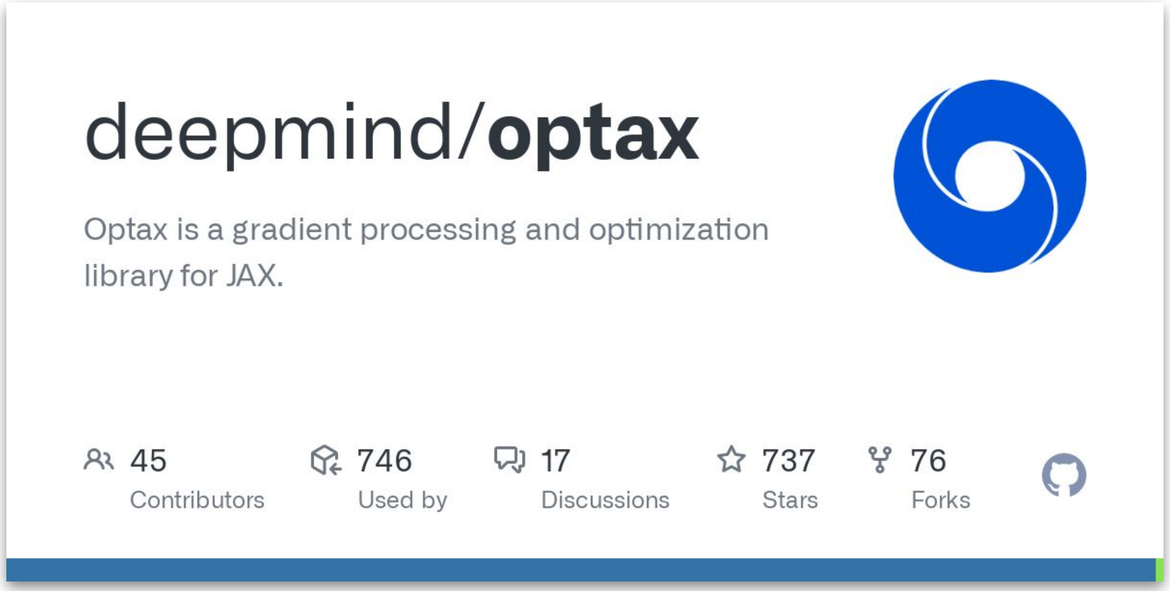
- **PyTorch-style Sub-classing API**
- Full set of of NN layers
- Allows for easy building and training of ML models

```
class CiFarCNN(nn.Module):
  """A simple CNN model for CiFar10/100"""

  @nn.compact
  def __call__(self, x):
    x = nn.Conv(features=32, kernel_size=(3, 3))(x)
    x = nn.gelu(x)
    x = nn.max_pool(x, window_shape=(2, 2), strides=(1, 1))
    x = nn.Conv(features=64, kernel_size=(3, 3))(x)
    x = nn.gelu(x)
    x = nn.max_pool(x, window_shape=(2, 2), strides=(1, 1))
    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
```

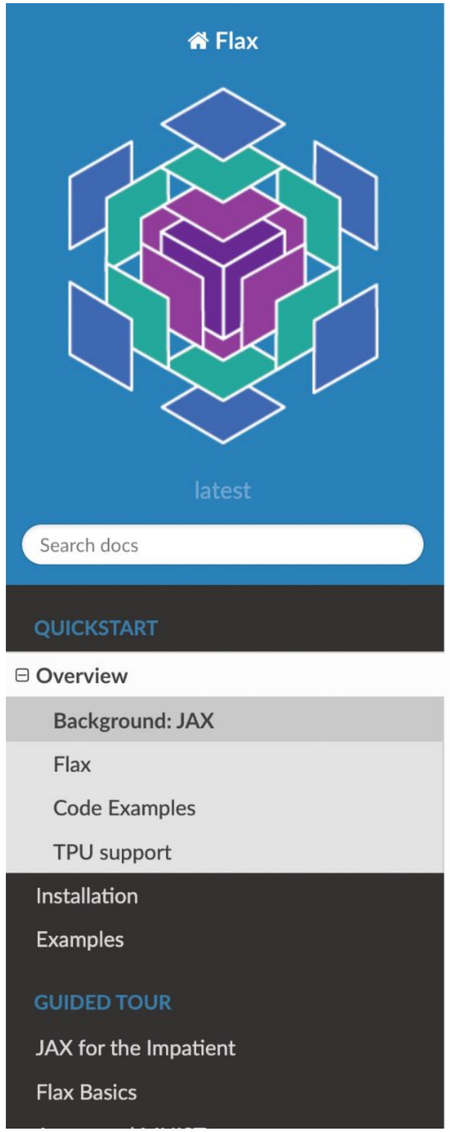
NEW

Optax

- **Optimizers & Losses**
- Made by **DeepMind**
- Standard Losses and Optimizers written and optimized for JAX



Getting started with Flax



Background: JAX

JAX is NumPy + autodiff + GPU/TPU

It allows for fast scientific computing and machine learning with the normal NumPy API (+ additional APIs for special accelerat

JAX comes with powerful primitives, which you can compose arbitrarily:

- Autodiff (`jax.grad`): Efficient any-order gradients w.r.t any variables
- JIT compilation (`jax.jit`): Trace any function ⟶ fused accelerator ops
- Vectorization (`jax.vmap`): Automatically batch code written for individual samples
- Parallelization (`jax.pmap`): Automatically parallelize code across multiple accelerators (including across hosts, e.g. for TPU p

If you don't know JAX but just want to learn what you need to use Flax, you can check our JAX for the impatient notebook.

Flax

Flax is a high-performance neural network library for JAX that is **designed for flexibility**: Try new forms of training by forking a training loop, not by adding features to a framework.

Flax is being developed in close collaboration with the JAX team and comes with everything you need to start your research, i

- **Neural network API** (`flax.linen`): Dense, Conv, {Batch|Layer|Group} Norm, Attention, Pooling, {LSTM|GRU} Cell, Dropout
- **Utilities and patterns**: replicated training, serialization and checkpointing, metrics, prefetching on device
- **Educational examples** that work out of the box: MNIST, LSTM seq2seq, Graph Neural Networks, Sequence Tagging
- **Fast, tuned large-scale end-to-end examples**: CIFAR10, ResNet on ImageNet, Transformer LM1b

Code Examples

See the What does Flax look like section of our README.

https://flax.readthedocs.io/en/latest/overview.html

ML Topics Covered

1. Coral Micro Dev board
2. Models for Language and Images at Google
3. Jax/Flax : Cool New ML Stack

Google I/O Extended

Thank you!

Martin Andrews

Head of AI : Red Dragon AI
Google Developer Expert Machine Learning & Deep Learning

@mdda123 @mdda

Resources

https://imagen.research.google/

https://ai.googleblog.com/2022/04/pathways-language-model-palm-scaling-to.html

https://coral.ai/

https://github.com/google/flax