

# Intro to Gemma

The official Train the Trainer deck!



Gus Martins  
he/him  
AI Developer Relations Engineer



Ju-yeong Ji  
he/him  
AI Developer Relations Engineer

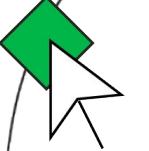




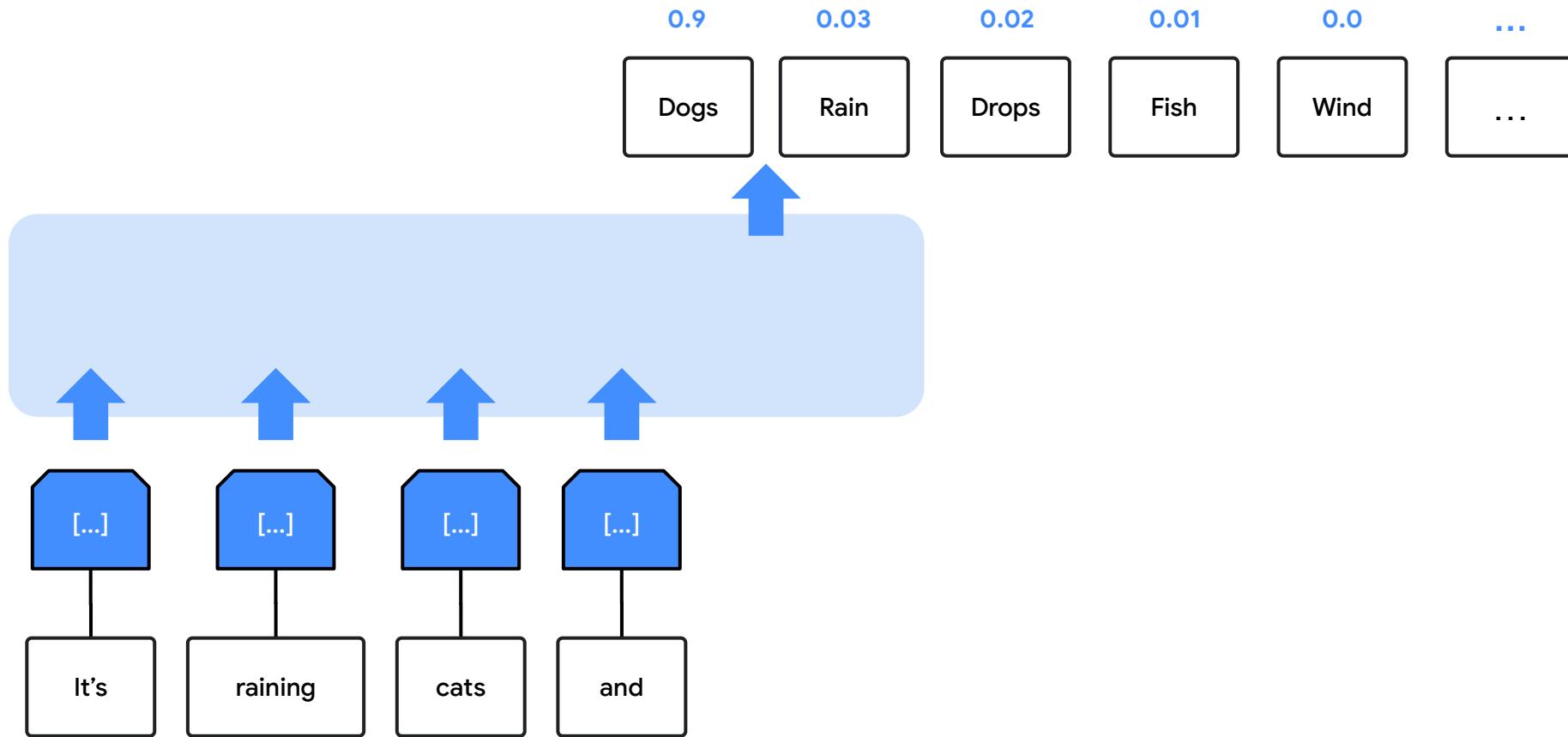
# What is AI?



# **What is generative AI?**



# What is an LLM?



Roses are red,

Roses are red,

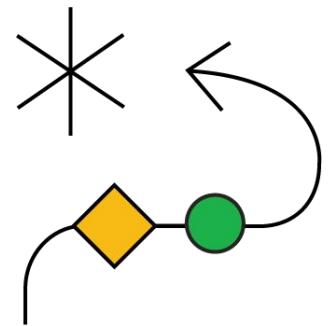
Violets are blue,

Sugar is sweet,

```
for(var i = 0
```

```
for(var i = 0; i < 10; i++) {
```

Modern LLMs are  
large



# Classic Natural Language Problems



Entity extraction



Classification



Summarization



Sentiment Analysis



Translation



...

# User:

Explain this joke: “I used to be addicted to soap, but I'm clean now.”

# Model:

Model: The joke is a pun on the word "clean". It has two meanings:

- To be free of dirt or other unwanted substances.
- To be free of addiction.

The joke is funny because it is unexpected. The listener expects the speaker to be talking about the first meaning of "clean", but they are actually talking about the second meaning. This creates a surprise and a laugh.

# User:

Give me five neat ideas for science fair projects.

# Model:

## 1. \*\*Build a volcano.\*\*

This is a classic science fair project that is both fun and educational. You can learn about the science of volcanoes, earthquakes, and plate tectonics.

## 2. \*\*Make a solar oven.\*\*

This is a great project to learn about renewable energy and how to harness the power of the sun. You can use your solar oven to cook food, dry fruit, or even melt chocolate.

## 3. \*\*Create a 3D printer.\*\*

This is a more challenging project, but it is also very rewarding. You can learn about 3D printing technology and design your own custom objects.

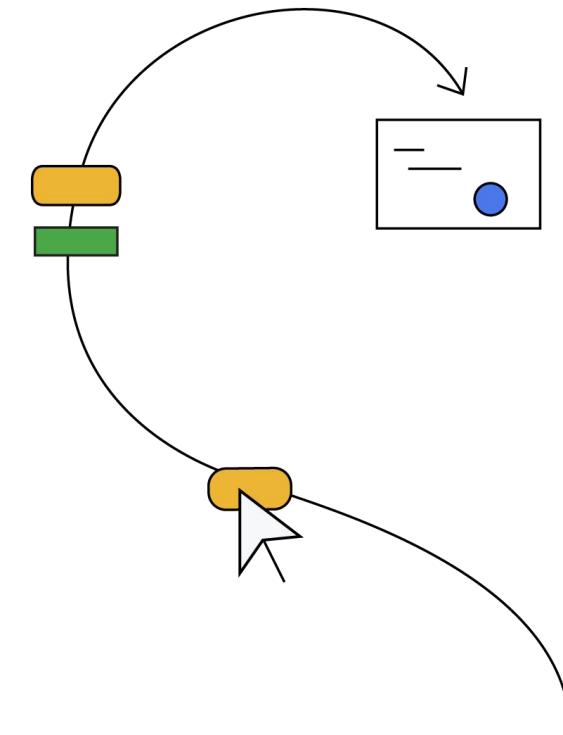
## 4. \*\*Build a robot.\*\*

This is a great project for those who are interested in engineering and robotics. You can learn about basic electronics, mechanics, and programming.

## 5. \*\*Design a water filtration system.\*\*

This is a project that can help you learn about the importance of clean water and how to filter it. You can use your water filtration system to clean water from a stream or river.

LLMs let us  
prototype fast



# Why are large language models different?

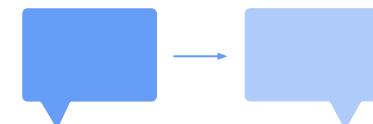
LLMs are characterized by **emergent abilities**, or the ability to perform tasks that were not present in smaller models.

LLMs contextual understanding of human language **changes how we interact** with data and intelligent systems.

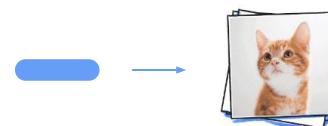
LLMs can find patterns and connections in **massive, disparate data corpora**.



Conversation

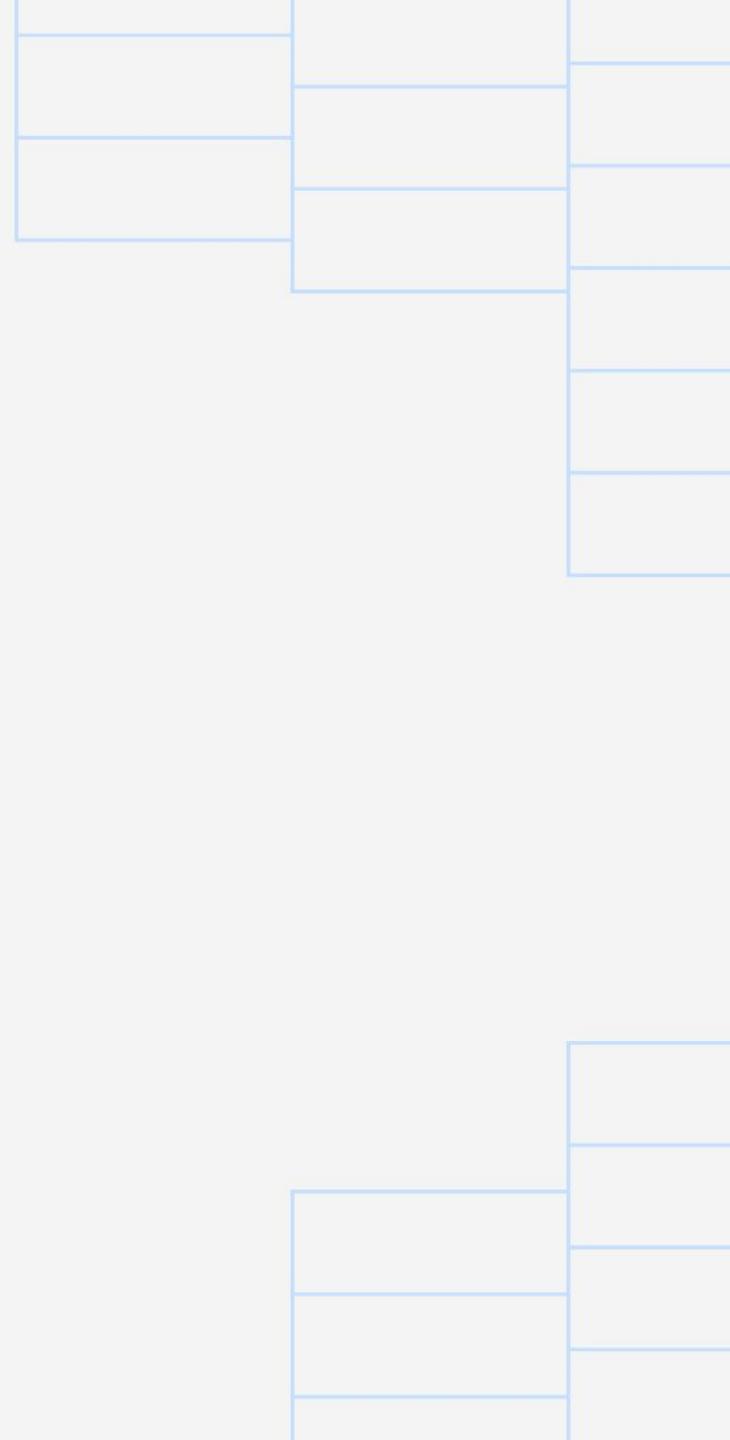
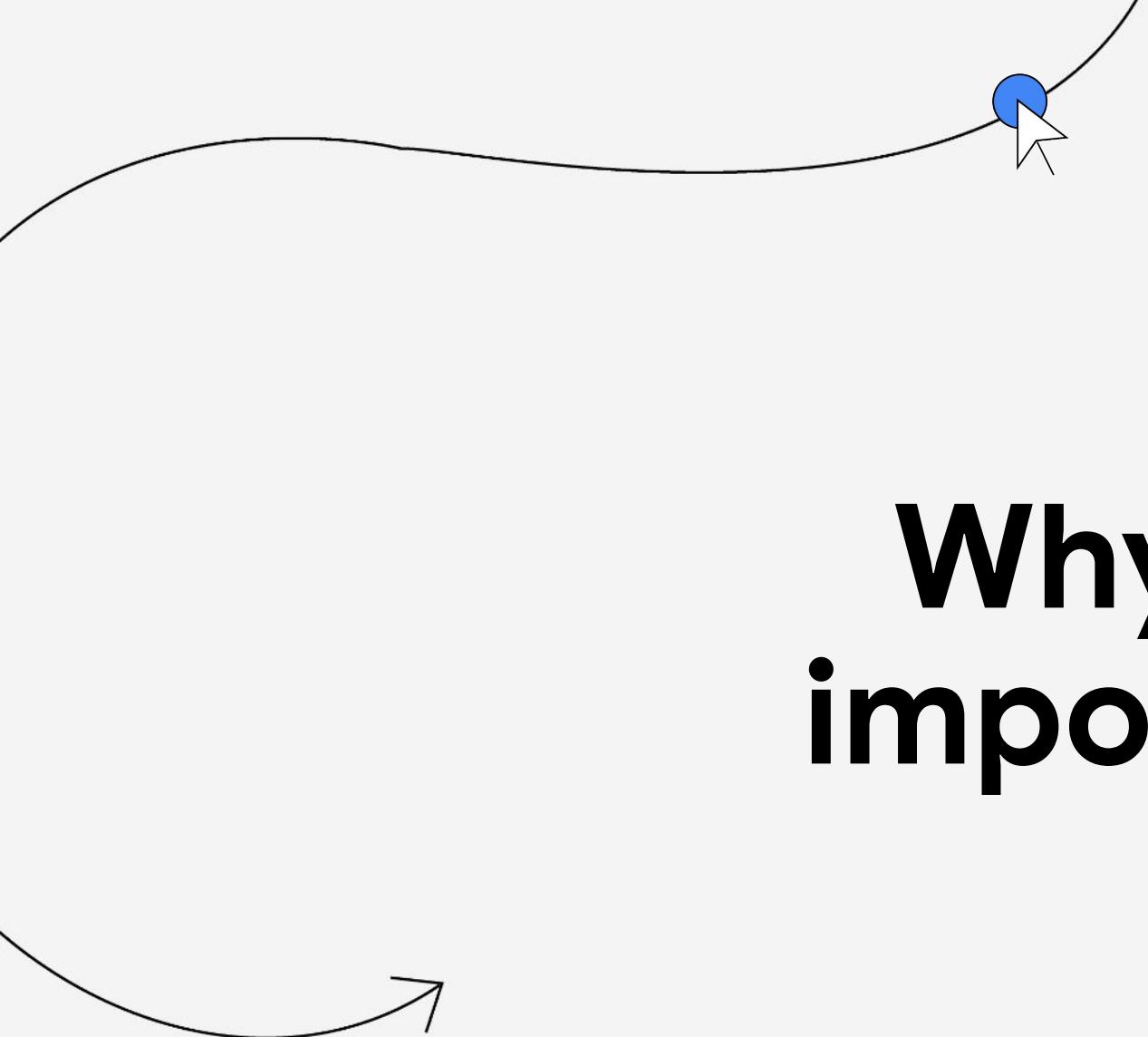


Content generation



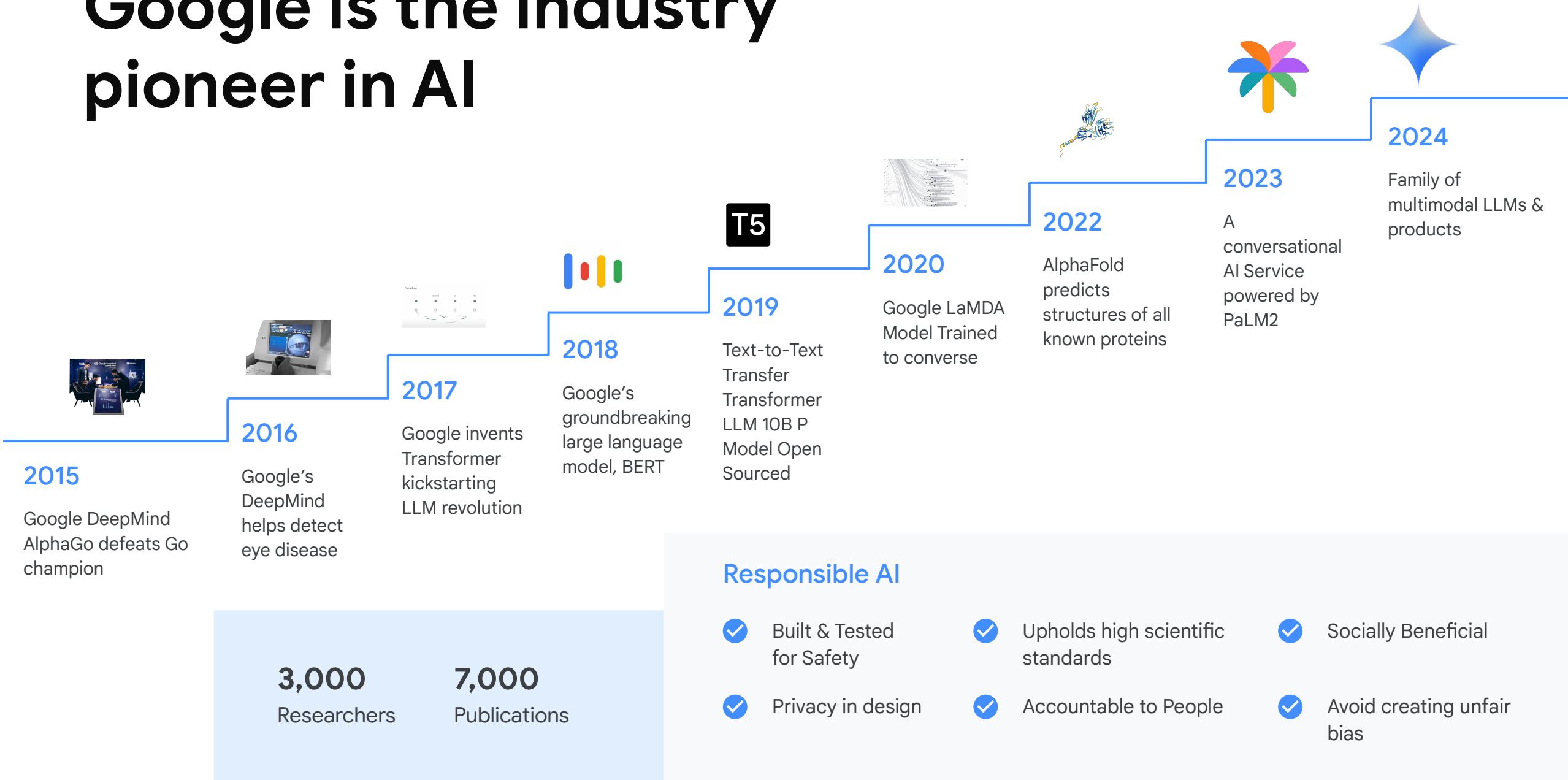
# Google history of Open Source





# **Why is it important?**

# Google is the industry pioneer in AI

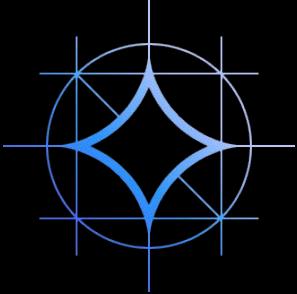


# Background

## A history of disruptive innovations in AI grounded in open development

2012	Distbelief: First distributed system for training neural nets
2013	First DL model in production (Deep Neural Nets for acoustic modeling) “Cats paper”: First network trained in a distributed way
2014	Word2Vec: Proof of viability of word embeddings to understand content Seq2Seq: Foundational framework leading to Transformers and BERT
2015	GoogleNet “Going Deeper with Convolutions” Brain’s first live DL experiments in Ads
2016	Distillation: compression model reducing size of complex ML models RankBrain first launches
2017	TensorFlow released TPUs announced
2018	JAX AutoML: Success in learning to learn
2019	Transformers: Rethinking Seq-to-Seq BERT in Search
2022	AlphaFold LaMDA 2, PaLM, AI Test Kitchen
2024	Imagen, Parti, Imagen-Video, Phenaki Bard , Gemini, Search Labs, and more

Welcome to  
the Gemini era



# Gemma open models

A family of lightweight, state-of-the art open models built from the same research and technology used to create the Gemini models

# Key Gemma goals

Democratizing innovation

Build together, grow together

Accelerating model development

Already 500+ variants

Fostering trust and responsible AI

Safety by design + tools

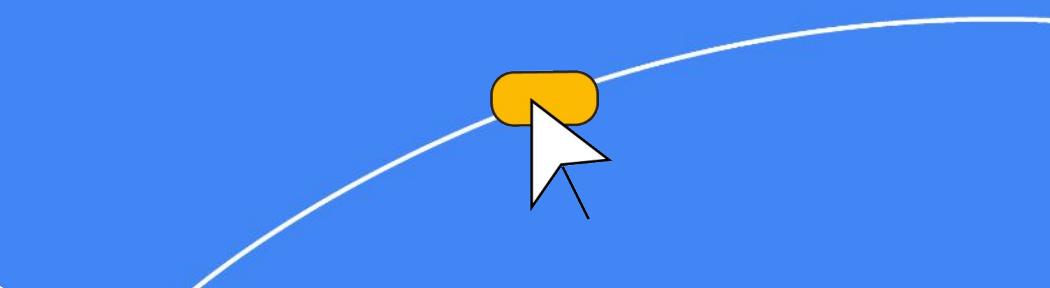
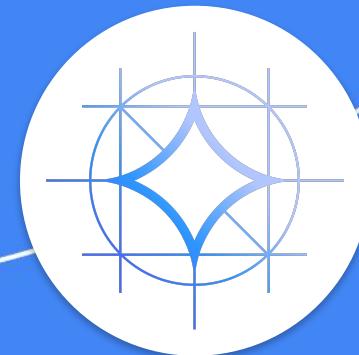
Enabling continuous learning

Events, community contributions

Shaping the AI future

Partner with NVIDIA, Hugging Face, and you!

# Using Gemma - Easy steps



# How do I get access to Gemma?

Gemma models are available across **Kaggle**, **Vertex AI** and **Hugging Face** - choose the model hub that makes the most sense for you.

Just request access on the model hub of your choice, agree to the Gemma Terms of Use and you'll be ready to get started!



## Access Gemma on Kaggle

To access Gemma on Kaggle, you need to first request access.

[Request Access](#)

# How do I get ready to use Gemma?

First, install **KerasNLP** with **Keras 3** into your environment, then import the **keras\_nlp** module.

```
!pip install --upgrade keras-nlp  
!pip install --upgrade keras  
  
import keras_nlp
```

Then, load Gemma from a preset configuration!

```
# https://keras.io/api/keras_nlp/models/gemma/gemma_causal_lm/  
g_lm = keras_nlp.models.GemmaCausalLM.from_preset("gemma_2b_en")
```

Presets are available for the 2B and 7B parameter versions of Gemma.

# Using Gemma

The **generate()** function is how we interact with the model.

For instance, if we wanted to ask Gemma what comes after “it was a dark and stormy night”...

```
txt = g_lm.generate("It was a dark and stormy night.", max_length=64)  
print(txt)
```

Gemma will give us:

*It was a dark and stormy night.*

*The rain was pouring down, and the wind was howling.*

*But that didn't stop a group of friends from going out for a night of fun.*

*They were all dressed up in their best clothes, and they were ready to have a good time.*

# Choosing The Right Gemma

Gemma offers 2B and 7B parameter sizes (larger = more powerful)! Option for instruction-tuned models for specific tasks.

Your use cases and target hardware will help shape your choice of Gemma model to use.

For more information, see <https://ai.google.dev/gemma>



# CodeGemma

An open model that can perform a variety of coding tasks

- Intelligent code completion and generation
- Enhanced accuracy
- Multi-language proficiency
  - Python, JavaScript, Java, Kotlin, C++, C#, Rust, Go, and other languages
- Streamlined workflows

# CodeGemma - Fill-in-the-middle

```
BEFORE_CURSOR = "<|fim_prefix|>"  
AFTER_CURSOR = "<|fim_suffix|>"  
AT_CURSOR = "<|fim_middle|>"  
FILE_SEPARATOR = "<|file_separator|>"
```

## INPUT PROMPT

```
<|fim_prefix|>import <|fim_suffix|>if __name__ == "__main__":\n    sys.exit(0)<|fim_middle|>
```

## OUTPUT

```
<|fim_prefix|>import <|fim_suffix|>if __name__ == "__main__":\n    sys.exit(0)<|fim_middle|>sys\n<|file_separator|>
```

```
import |\nif __name__ == "__main__":\n    sys.exit(0)
```

# CodeGemma - 7B IT model example

```
<start_of_turn>user
```

You are an experienced C and Python programmer. Convert the following Python code into C.

```
```python
def factorial(n):
    result = 1
    for i in range(2, n + 1):
        result *= i
    return result
```

```

```
<end_of_turn>
```

```
<start_of_turn>model
```

# CodeGemma - 7B IT model example

Here is the C code equivalent of the Python code:

```
```c
int factorial(int n) {
    int result = 1;
    for (int i = 2; i <= n; i++) {
        result *= i;
    }
    return result;
}
```

```

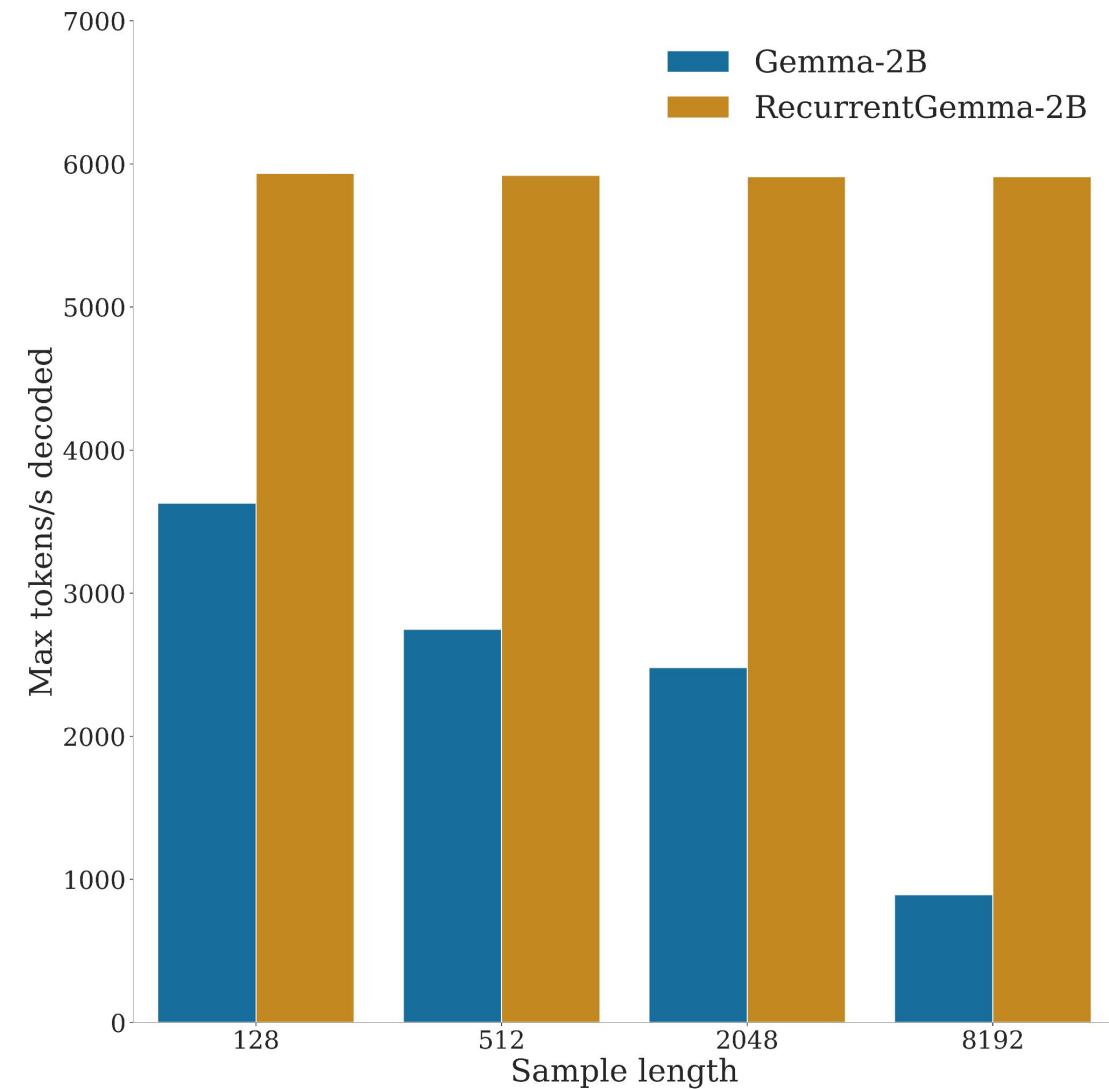
Here is a breakdown of the changes:

- \* The function is declared with the `int` return type, as in Python.
- \* The `for` loop is converted to a `for` loop with an `int` variable `i` initialized to 2 and incremented by 1 in each iteration.
- \* The `range` function is replaced with a simple loop that iterates from 2 to `n` (inclusive).
- \* The `result \*= i` statement is used to multiply `result` by `i` in each iteration.
- \* The `return` statement is used to return the final value of `result`.

# RecurrentGemma

An open model based on Griffin

- Reduced memory usage
- Higher throughput
- High performance



Throughput when sampling from a 2k prompt

# Discover what other people are doing with Gemma

Kaggle is a global hub for data science and machine learning - a place to learn, collaborate, and compete to create cutting-edge AI solutions.

- Gemma-powered Kaggle Notebooks
- Competitions, some with significant prizes
- Leaderboards
- Community discussion forums



# Fine-tune Gemma for your use-cases and data

Easily fine-tuned for your tasks.

## Keras

- High-level APIs and LoRA support for efficiency.
- [https://ai.google.dev/gemma/docs/lora\\_tuning](https://ai.google.dev/gemma/docs/lora_tuning)

## JAX

- in-depth customization by experts
- [https://ai.google.dev/gemma/docs/jax\\_finetune](https://ai.google.dev/gemma/docs/jax_finetune)

# Explore a world of possibilities with on-device integrations

Gemma is a great match for existing technologies such as MediaPipe and TensorFlow Lite. By using Gemma, you can:

- **Summarize:** concisely describe results
- **Classify:** classify results into useful categories, providing feedback on quality
- **Generate Text:** provide a fluent natural language interface

Gemma is available across mobile, web and desktop.

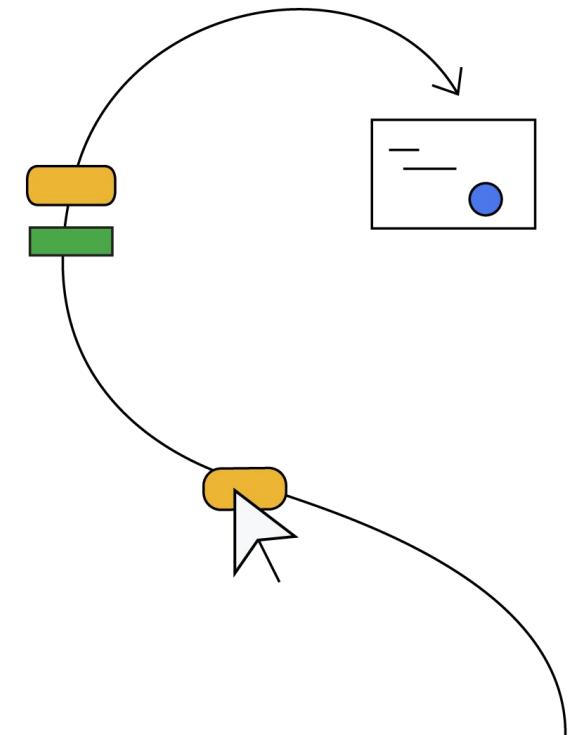
# Productionize Gemma for your team

Gemma is lightweight enough to run locally, but flexible enough to scale - Google Cloud loves Gemma!

- **Vertex AI:** easily deploy and scale Gemma
- **GKE:** integrate Gemma-based solutions with Kubernetes
- **DataFlow:** use Gemma within scalable data pipelines



# Let's jump in!



# Gemma and Kaggle

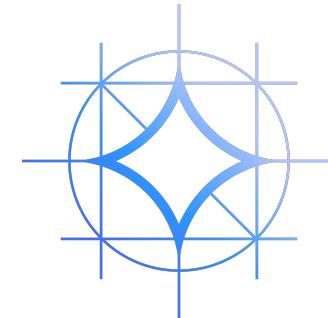


Train the Trainer in AI

The image shows a tablet device with a white frame and a black bezel, displaying the homepage of the Kaggle website. The screen is white and features the following content:

- Kaggle** logo in blue.
- Navigation menu: Competitions, Datasets, Models, Code, Discussions, Courses, and a three-dot menu.
- A search bar with a magnifying glass icon and the text "Search".
- User account links: "Sign In" and "Register".
- Section Header:** "Level up with the largest AI & ML community".
- Text:** "Join over 17M+ machine learners to share, stress test, and stay up-to-date on all the latest ML techniques and technologies. Discover a huge repository of community-published models, data & code for your next project."
- Registration Buttons:** "Register with Google" and "Register with Email".
- Illustration:** A circular illustration of diverse people in lab coats, some holding papers or a brain diagram, with a yellow ribbon banner across the bottom.
- Section Header:** "Who's on Kaggle?"
- Learners:** "Dive into Kaggle courses, competitions & forums." with a person icon.
- Developers:** "Leverage Kaggle's models, notebooks & datasets." with a person icon.
- Researchers:** "Advance ML with our pre-trained model hub & competitions." with a person icon.
- Key Features:** Lists of features for each group, each with an icon:
  - Learners:** Beginner competitions, Practical courses, Public datasets.
  - Developers:** Open-sourced models, Competition solution write-ups, Public notebooks.
  - Researchers:** ML competition hosting, Model training datasets, Open-sourced models.

kaggle



Gemma

Search

Sign In Register

1805 Open in Vertex AI New Notebook :

# Gemma

google/gemma

Gemma is a family of lightweight, open models built from the research and technology that Google used to create the Gemini models.

Model Card Code (113) Discussion (32)

Access Gemma on Kaggle Request Access

## Model Details

### Gemma Model Card

Model Page: [Gemma](#)

Resources and Technical Documentation:

- Responsible Generative AI Toolkit
- [Gemma on Kaggle](#)
- [Gemma on Vertex Model Garden](#)

Terms of Use: [Terms](#)

Authors: Google

### Frameworks

PyTorch Transformers  
GemmaCpp TensorRTLLM  
MaxText Pax Flax Keras

### Tags

TASK  
Text Generation  
Text-To-Text Generation  
OTHER



# Accept Terms of Service

 Google

## Gemma Access Request

① Choose Account for Consent

 Verify via Kaggle Account  
Enables seamless access with Keras, JAX, Colab, and the rest of Google's ecosystem of tools.

 Verify via Hugging Face  
Enable access to the Transformers implementation.

OR JUST DOWNLOAD THE MODEL FILES ONLY BY VERIFYING WITH GOOGLE OR EMAIL

 Google

 Email

[Log into Kaggle](#)

② Gemma Consent Form



The screenshot shows the Kaggle AI interface. On the left, a sidebar menu includes options like Home, Competitions, Datasets, Models (which is selected), Code, Discussions, Learn, and More. A central search bar at the top has the word "Search". Below it, the title "Gemma" is displayed, along with a "Model Card" tab (selected), "Code (113)", and "Discussion (32)". To the right of the title are buttons for "1677", "Open in Vertex AI", "New Notebook", and a more options menu. The main content area features an "Inference" section with a note about deployment best practices. It includes a "SUGGESTED PROMPTS" section with four buttons: "Explain quantum physics in an easy way.", "Design a superhero.", "Give me a workout for beginners.", and "Compare old and new movies.". Below this is a text input field containing "Compare old and new movies." and a "Generate" button. The "OUTPUT" section displays the generated text: "Old vs. New Movies: A Tale of Two Reels". It discusses the differences between old and new films, mentioning the predominance of black and white, silent films, and slower pacing. A note at the bottom states "Output may be truncated due to max tokens of 128." At the bottom right, it says "Powered by Google Cloud Vertex AI".

kaggle

Create

Home

Competitions

Datasets

Models

Code

Discussions

Learn

More

View Active Events

Search

Gemma

Model Card Code (113) Discussion (32)

Inference

This widget uses the following settings: Temperature: 0.4, Max output tokens: 128, Top-K: 5. Safety filters ON in accordance with best practice for deploying AI responsibly.

SUGGESTED PROMPTS

Explain quantum physics in an easy way.

Design a superhero.

Give me a workout for beginners.

Compare old and new movies.

Compare old and new movies.

Generate

OUTPUT

**Old vs. New Movies: A Tale of Two Reels**

While the popcorn may be the same, the movies themselves offer vastly different experiences. Let's delve into the differences between old and new films:

**Old Movies:**

- **Black and White:** The predominant color palette was black and white, creating a stark contrast to the vibrant color of today's films.
- **Silent Films:** Many early movies were silent, requiring the audience to imagine the plot through facial expressions and music.
- **Slow Paced:** With a focus on character development and storytelling, these films tend to unfold slowly, often

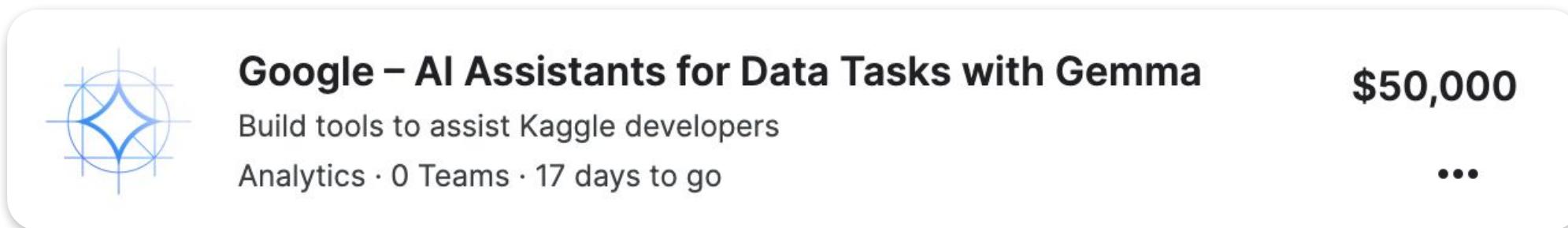
*Output may be truncated due to max tokens of 128.*

Powered by Google Cloud Vertex AI



# Kaggle Competitions: Test Your Skills, Win Prizes, & Learn

Tackle real-world problems with unique datasets! Put your AI skills to the test against the best, with the chance to win cash prizes and gain industry recognition.



**Google – AI Assistants for Data Tasks with Gemma**      **\$50,000**

Build tools to assist Kaggle developers

Analytics · 0 Teams · 17 days to go

...

Competitions are key after the fact as well - entrants often post their solutions, which are a valuable learning resource.

GOOGLE · ANALYTICS COMPETITION · A MONTH TO GO

Join Competition ...

# Google – AI Assistants for Data Tasks with Gemma

Build tools to assist Kaggle developers



Overview Data Code Models Discussion Rules

## Overview

Google recently launched [Gemma](#), a new family of open LLMs built from the same research and technology used to create their Gemini models. In this competition, you're challenged to demonstrate how to use Gemma to accomplish one or more data science oriented tasks.

**Start**  
15 days ago

**Close**  
a month to go

**Description**

**Competition Host**  
Google 

**Prizes & Awards**  
\$50,000  
Does not award Points or Medals

**Tags**

Text Generation  
Artificial Intelligence

Table of Contents

KAGGLE · FEATURED CODE COMPETITION · A MONTH TO GO

Join Competition ...

# LLM Prompt Recovery

Recover the prompt used to transform a given text



Overview Data Code Models Discussion Leaderboard Rules

## Overview

LLMs are commonly used to rewrite or make stylistic changes to text. The goal of this competition is to recover the LLM prompt that was used to transform a given text.

**Start**  
9 days ago

**Close**  
a month to go

Merger & Entry

**Competition Host**  
Kaggle

**Prizes & Awards**  
\$200,000  
Awards Points & Medals

**Participation**  
692 Competitors  
652 Teams  
3,968 Entries

**Description**

Tags

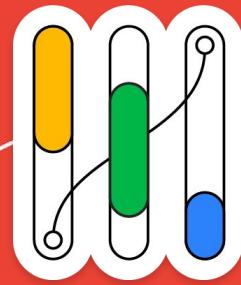
NLP Artificial Intelligence

# Kaggle: Your Gateway to AI Innovation

- Always have the latest Gemma resources and information.
- Easily launch into working with Gemma in notebooks.
- Explore how the community uses Gemma and connect with fellow data enthusiasts.
- Compete for cash prizes, dive into past competitions for cutting-edge techniques.
- Be part of shaping the future of AI alongside Google and the vibrant Kaggle community.

<https://www.kaggle.com/models/google/gemma>

# Integration with Keras



# Why Keras?



Simple

simplifies development,  
empowering focused  
problem-solving.



Flexible

offers scalable complexity  
for evolving needs.

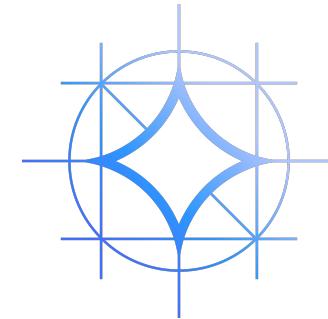


Powerful

industry-strength  
performance and scalability



Keras



Gemma

# Keras

```
import keras_nlp

gemma =
keras_nlp.models.GemmaCausalLM.from_preset("gemma_instruct_7b_en")

...
gemma.generate("Hi there!")
```

API documentation: [keras.io/api/keras\\_nlp/models/gemma/](https://keras.io/api/keras_nlp/models/gemma/)

# Fine-tune Gemma models in Keras using LoRA

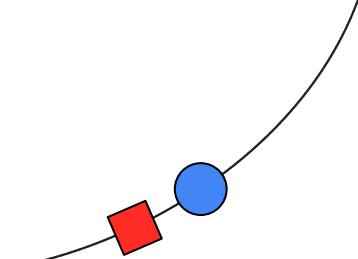
```
# Enable LoRA for the model and set the LoRA rank to 4.  
gemma_lm.backbone.enable_lora(rank=4)  
# Use AdamW (a common optimizer for transformer models).  
optimizer = keras.optimizers.AdamW(  
    learning_rate=5e-5,  
    weight_decay=0.01,  
)  
# Exclude layernorm and bias terms from decay.  
optimizer.exclude_from_weight_decay(var_names=["bias", "scale"])  
  
gemma_lm.compile(  
    loss=keras.losses.SparseCategoricalCrossentropy(from_logits=True),  
    optimizer=optimizer,  
    weighted_metrics=[keras.metrics.SparseCategoricalAccuracy()],  
)  
gemma_lm.fit(data, epochs=1, batch_size=1)
```

# Save / Load pre-trained LoRA

```
# Save LoRA weights after gemma_lm.fit(), filename must end in ".lora.h5"
gemma_lm.backbone.save_lora_weights("my_model.lora.h5")

gemma = keras_nlp.models.GemmaCausalLM.from_preset("gemma_instruct_2b_en")
# Use the same LoRA rank that you trained
gemma_lm.backbone.enable_lora(rank=4)

# Load pre-trained LoRA weights
gemma_lm.backbone.load_lora_weights("my_model.lora.h5")
gemma_lm.compile(sampler=keras_nlp.samplers.TopKSampler(k=3,
temperature=0.7))
```



# Alternatively: Distributed Tuning

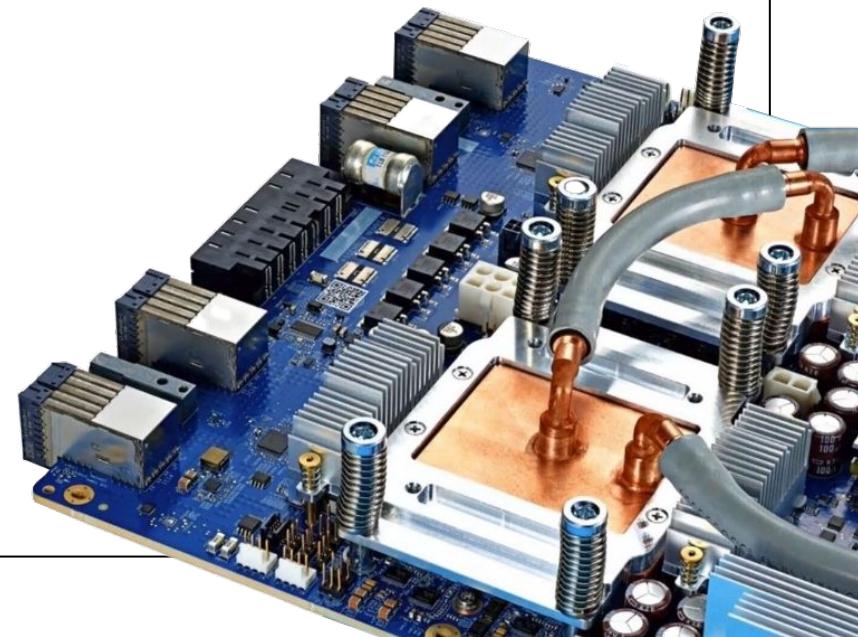
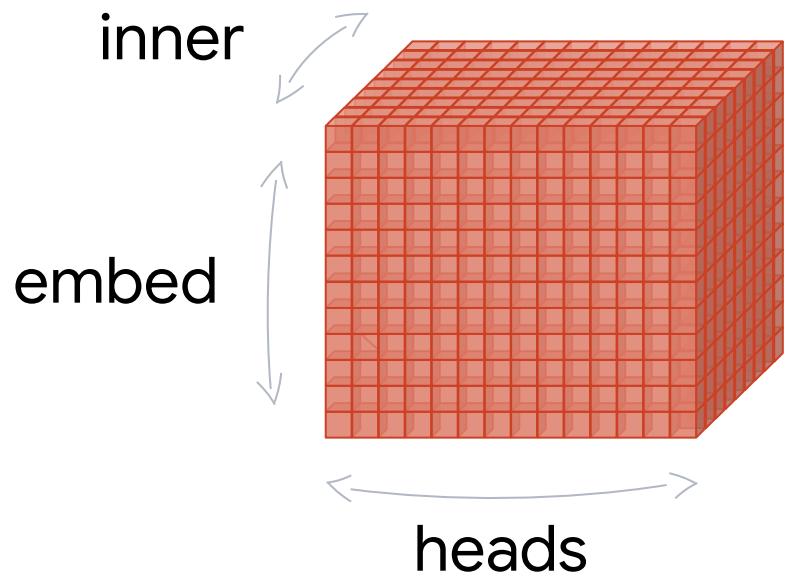
Training a single model across multiple devices (GPUs or TPUs) for faster results.

Why?

- **Large Models:** Complex models with many parameters require significant time to train on a single device.
- **Massive Datasets:** Distributed training accelerates the process of working through extremely large datasets.
- **Time Constraints:** Distributing training can drastically reduce the time needed to reach optimal model performance.

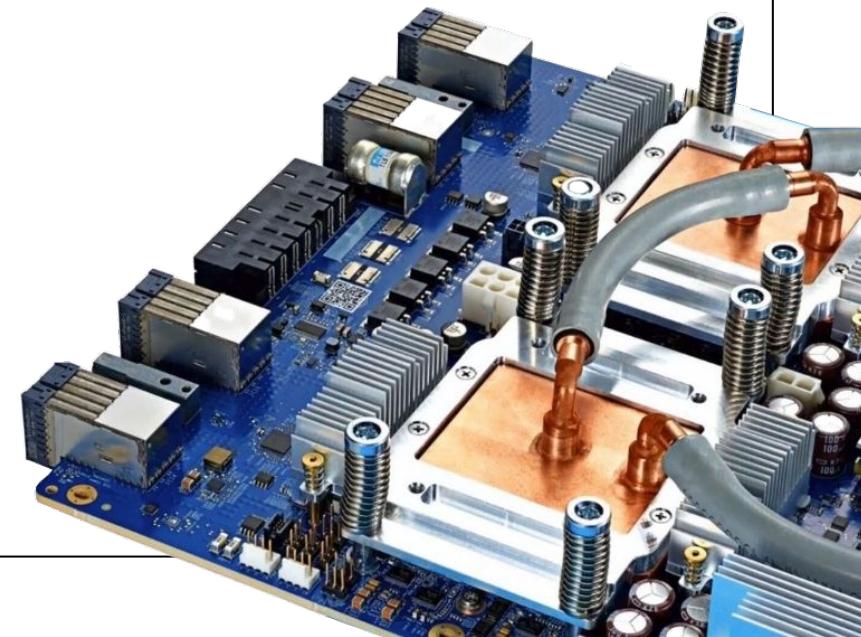
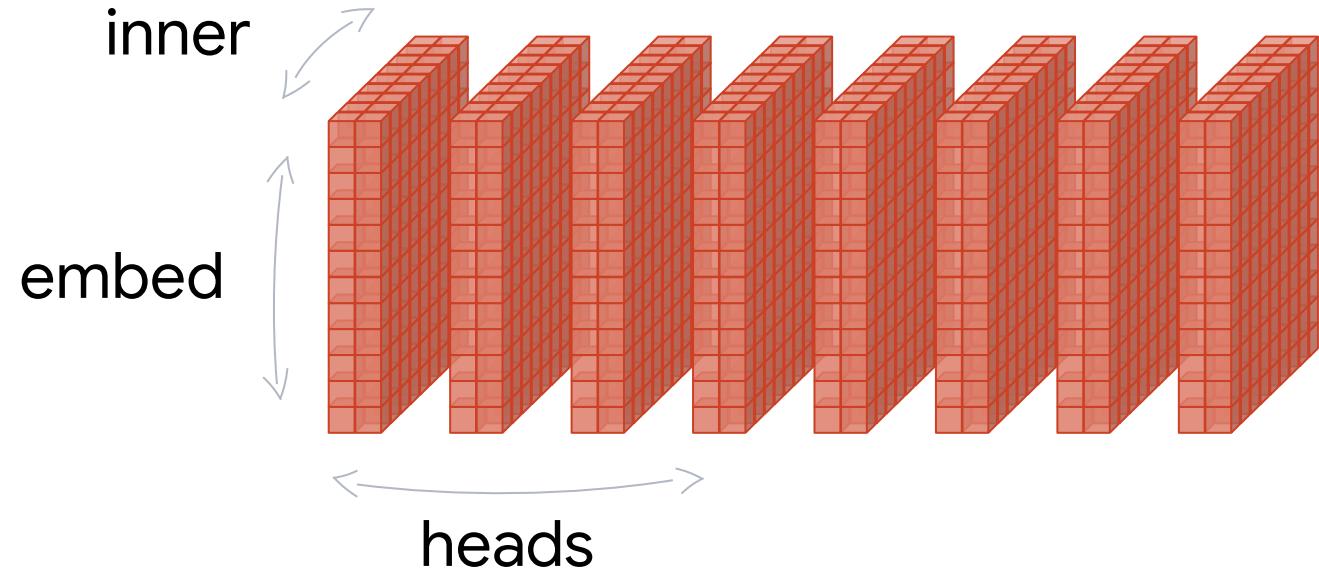
# Transformer partitioning

Powered by  and 



# Transformer partitioning (for TPU)

Powered by  and 



# Distributed Tuning

Defining the structure of our distributed hardware

```
// Set the configuration options for the LLM Inference task
val options = LlmInferenceOptions.builder()
    .setModelPath('/data/local/.../')
    .setMaxTokens(1000)
    .setTopK(40)
    .setTemperature(0.8)
    .setRandomSeed(101)
    .build()

// Create an instance of the LLM Inference task
llmInference = LlmInference.createFromOptions(context, options)
```

# Distributed Tuning (cont'ed)

Specifying how the weights and tensors should be distributed

```
model_dim = "model"
layout_map = keras.distribution.LayoutMap(device_mesh)

# Weights that match 'token_embedding/embeddings' will be sharded on 8 TPUs
layout_map["token_embedding/embeddings"] = (model_dim, None)
# Regex to match against the query, key and value matrices in the decoder
# attention layers
layout_map["decoder_block.*attention.*(query|key|value).*kernel"] = (
    model_dim, None, None)

layout_map["decoder_block.*attention_output.*kernel"] = (
    model_dim, None, None)
layout_map["decoder_block.*ffw_gating.*kernel"] = (None, model_dim)
layout_map["decoder_block.*ffw_linear.*kernel"] = (model_dim, None)
```

# Distributed Tuning (cont'ed)

## Setting up the distribution

```
model_parallel = keras.distribution.ModelParallel(  
    device_mesh, layout_map, batch_dim_name="batch")  
  
keras.distribution.set_distribution(model_parallel)
```

After this, the **generate()** and **fit()** methods will work just like non-distributed code.

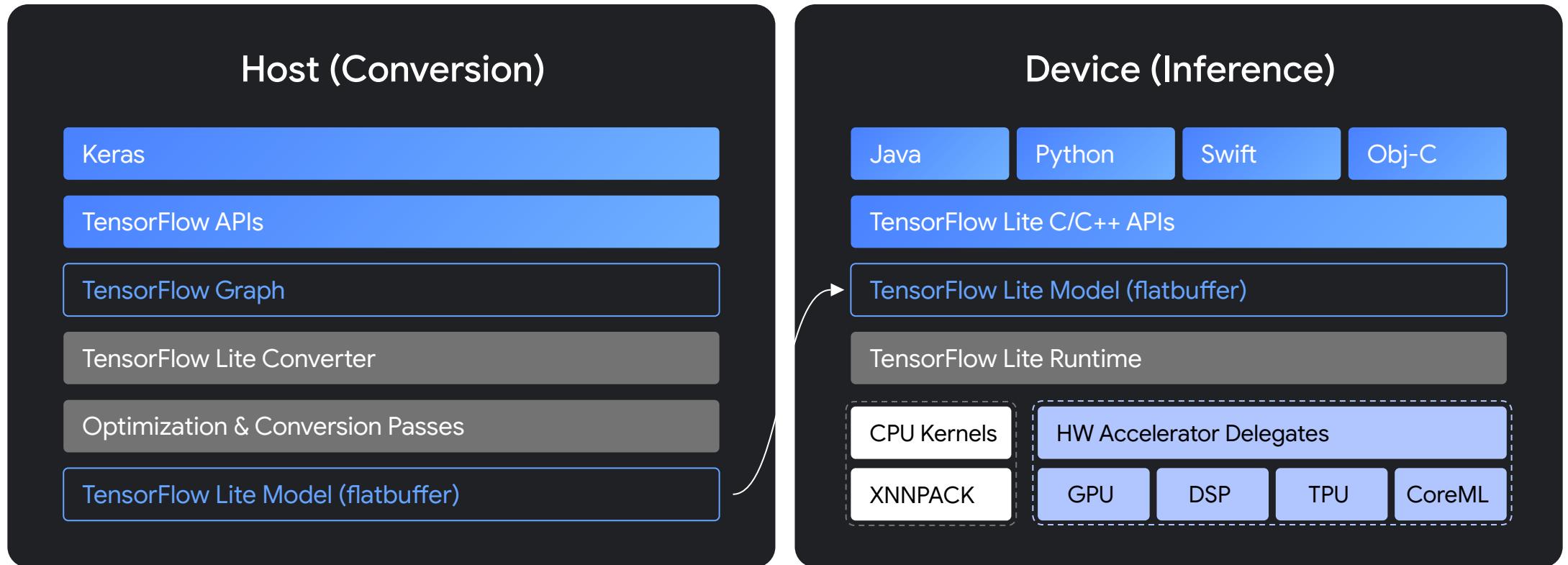
# Gemma on-device with MediaPipe and TensorFlow Lite



Train the Trainer in AI

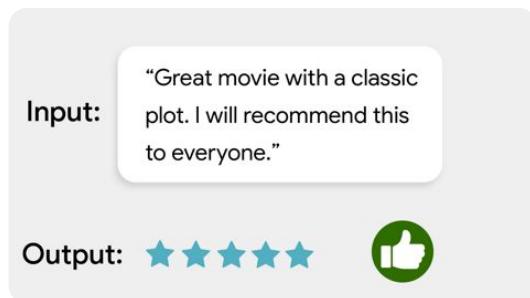
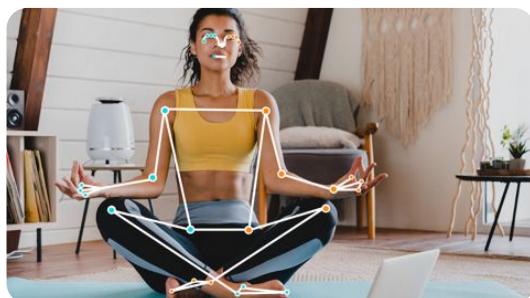
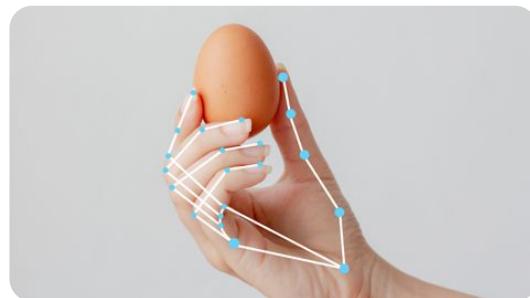
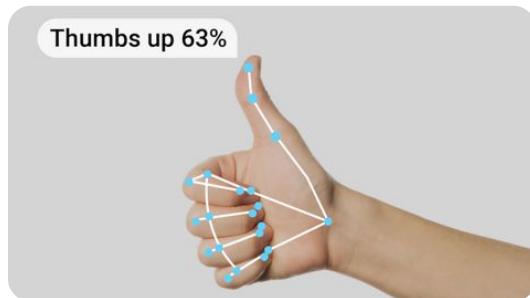
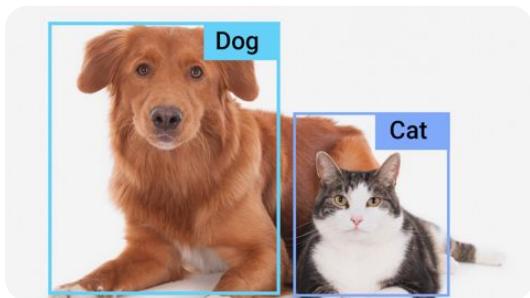
# TensorFlow Lite

Framework for deploying ML on web, mobile, and embedded systems



# MediaPipe

On-Device framework & solutions for pipelining ML models with pre/post processing



# What can you do with Gemma on-device?

- Text generation  
(smart replies, email compose, etc)
- Text rewrite
- Summarization
- Document / Corpus Q&A
- Text classification  
(toxicity, sentiment, etc)



Without  
cloud bills



Without  
privacy concerns



Without  
connectivity

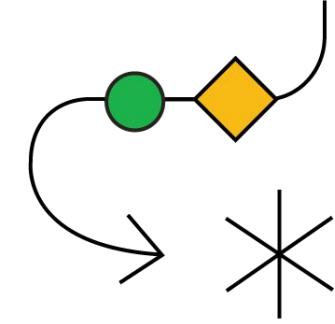
# Gemma 2B running on web

The screenshot shows the MediaPipe Studio interface with a focus on the LLM Inference section. At the top, there's a navigation bar with three circles, the MediaPipe logo, and the text "MediaPipe Studio". Below the navigation bar, a yellow banner displays the message "All outputs generated by user created models.". The main area is titled "LLM Inference". On the left, a text block explains the API's purpose: "The LLM Inference API lets you run large language models (LLMs) completely on-device, which you can use to perform a wide range of tasks, such as generating text, retrieving information in natural language form, and summarizing documents. The task provides built-in support for multiple text-to-text large language models, so you can apply the latest on-device generative AI models to your apps and products. .". Below this text is a button labeled "Code examples" with links to "Web | Android | iOS". A note below the text says, "The sample parameters below can be changed. See [documentation](#) for more details". On the right, there's a text input field with placeholder text "Enter some text... Write me an email to my friend Chris about how cake is so much better than pie. Use arguments like how good frosting is and how much better cakes look" and a cursor icon. Below the input field is a label "Inference time (ms)". At the bottom left, there's a "LLM model:" dropdown menu set to "gemma-2b-lt-gpu-int4.bin". A yellow circular button with a checkmark is located at the bottom right corner of the window.

# Integration on Device

- Media Pipe Studio:  
<https://bit.ly/mediapipe-studio>
- Documentation:  
<https://bit.ly/mediapipe-doc>





# Code Samples

- Android:  
<https://bit.ly/odml-android>
- iOS:  
<https://bit.ly/odml-ios>
- Web:  
<https://bit.ly/odml-web>

Clone the git repository

```
git clone https://github.com/googlesamples/mediapipe
```

# Download a model

[Download](#)

## Model Variations

Keras PyTorch Transformers Gemma C++ TensorRT-LLM **TensorFlow Lite** MaxText Pax Flax

VARIATION

gemma-2b-it-gpu-int4

gemma-2b-it-gpu-int4  
gemma-2b-it-cpu-int4  
gemma-2b-it-cpu-int8  
gemma-2b-it-gpu-int8

VERSIONS

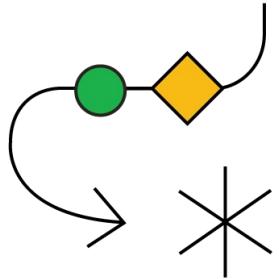
Version 1

+ New Notebook

LICENSE

Gemma

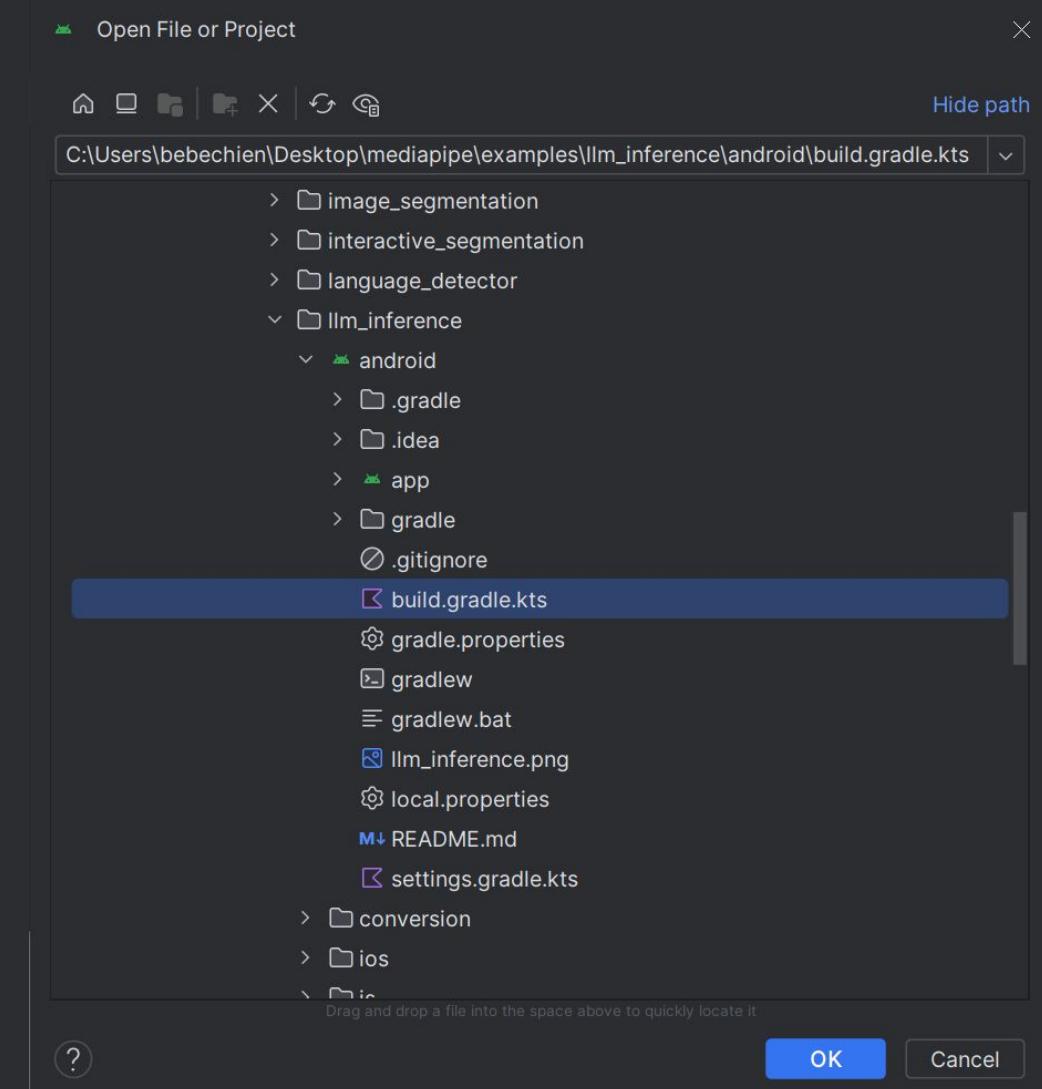
A lite implementation of the Gemma 2B instruction-tuned model, optimized for on-device inference. The model has been instruction-tuned to respond to prompts in a conversational manner.



# Open “build.gradle” under `llm_inference`

`mediapipe/examples/llm_inference/android`

If Android Studio requests a Gradle Sync,  
choose OK.



# Android Code Snippet

```
// Set the configuration options for the LLM Inference task  
val options = LlmInferenceOptions.builder()  
    .setModelPath('/data/local/.../')  
    .setMaxTokens(1000)  
    .setTopK(40)  
    .setTemperature(0.8)  
    .setRandomSeed(101)  
    .build()
```

[Create the task](#)

```
// Create an instance of the LLM Inference task  
llmInference = LlmInference.createFromOptions(context, options)
```

```
val result = llmInference.generateResponse(inputPrompt)  
logger.atInfo().log("result: $result")
```

[Run the task](#)

# Push model to the device

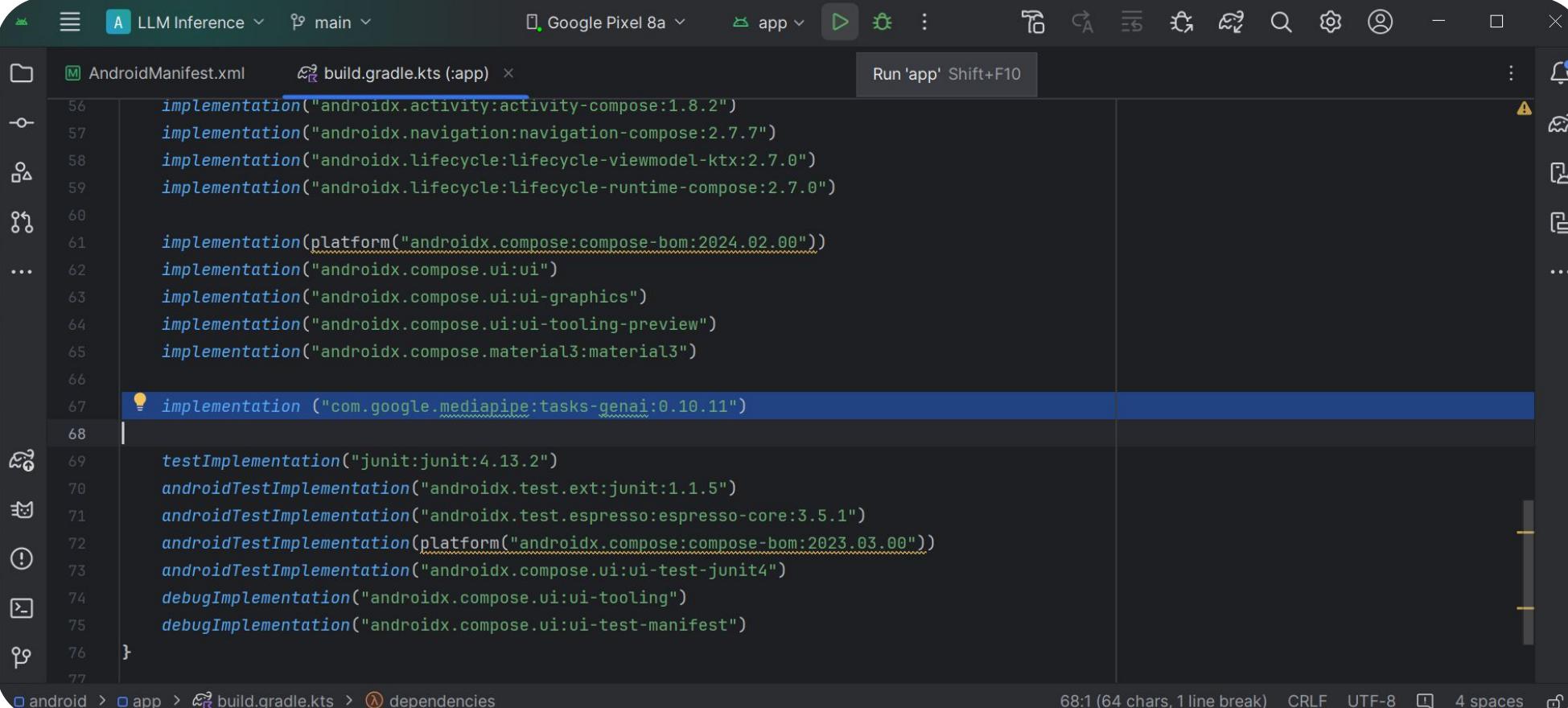
```
>adb shell mkdir -p /data/local/tmp/llm
```

```
>adb push gemma-2b-it-gpu-int4.bin /data/local/tmp/llm/model.tflite  
gemma-2b-it-gpu-int4.bin: 1 file pushed, 0 skipped. 36.6 MB/s  
(1354301440 bytes in 35.292s)
```

Ensure that your Android device is Connected to your computer and developer mode is enabled.

# Run the sample application

Click the green Run arrow



The screenshot shows the Android Studio interface with the code editor open to the `build.gradle.kts` file of a sample application. The file contains Gradle dependency declarations for AndroidX Compose and MediaPipe. A green run arrow icon is positioned in the top right corner of the code editor window.

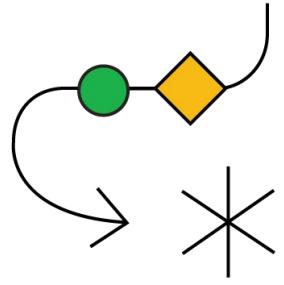
```
implementation("androidx.activity:activity-compose:1.8.2")
implementation("androidx.navigation:navigation-compose:2.7.7")
implementation("androidx.lifecycle:lifecycle-viewmodel-ktx:2.7.0")
implementation("androidx.lifecycle:lifecycle-runtime-compose:2.7.0")

implementation(platform("androidx.compose:compose-bom:2024.02.00"))
implementation("androidx.compose.ui:ui")
implementation("androidx.compose.ui:ui-graphics")
implementation("androidx.compose.ui:ui-tooling-preview")
implementation("androidx.compose.material3:material3")

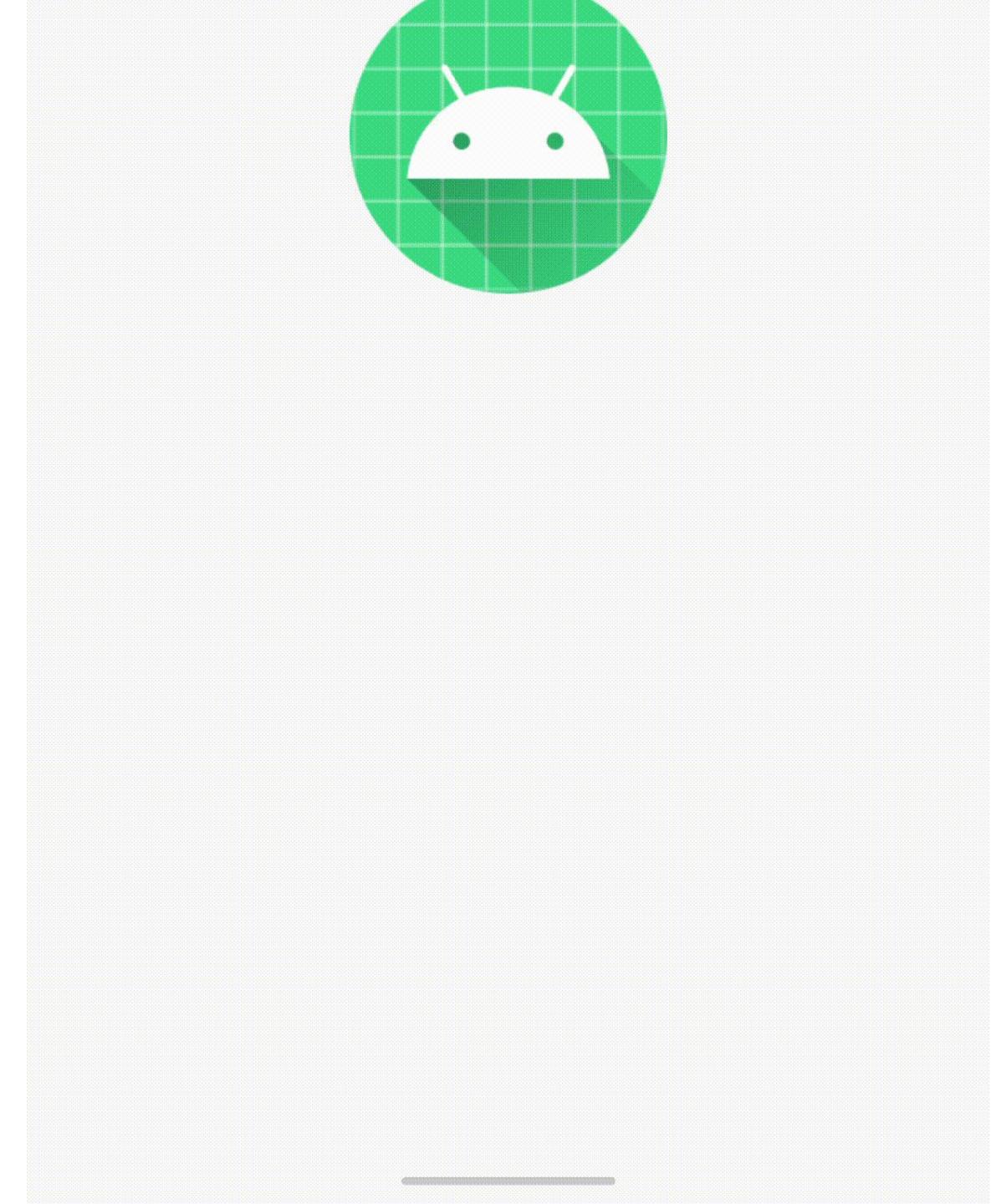
implementation ("com.google.medaiapipe:tasks-genai:0.10.11")

testImplementation("junit:junit:4.13.2")
androidTestImplementation("androidx.test.ext:junit:1.1.5")
androidTestImplementation("androidx.test.espresso:espresso-core:3.5.1")
androidTestImplementation(platform("androidx.compose:compose-bom:2023.03.00"))
androidTestImplementation("androidx.compose.ui:ui-test-junit4")
debugImplementation("androidx.compose.ui:ui-tooling")
debugImplementation("androidx.compose.ui:ui-test-manifest")
```

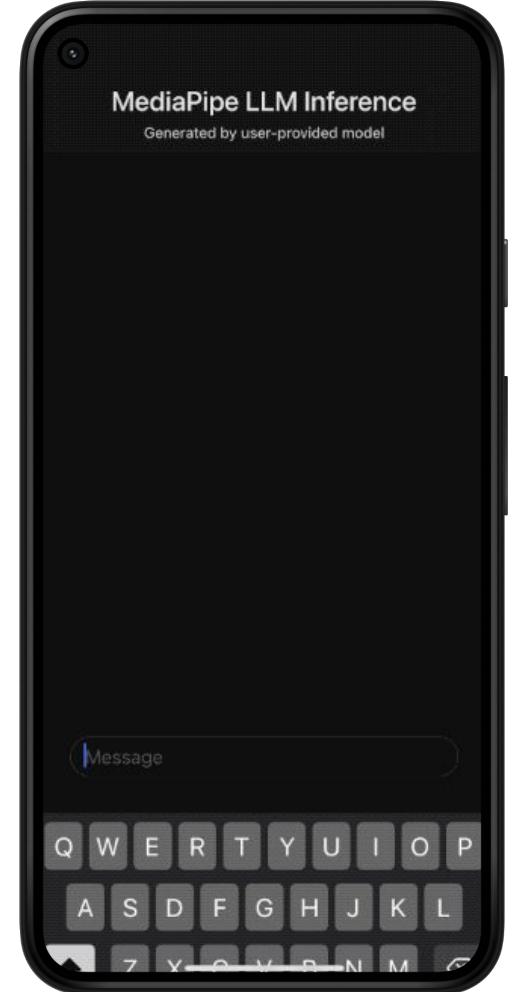
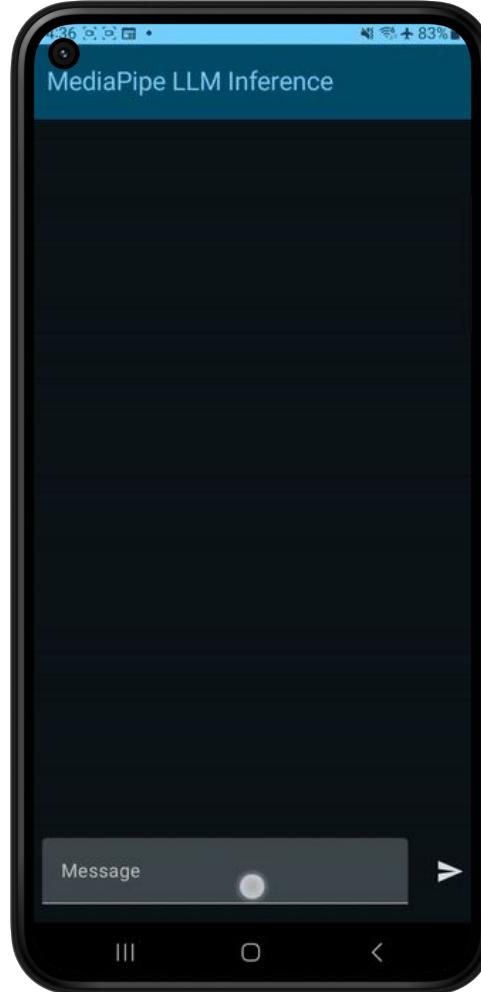
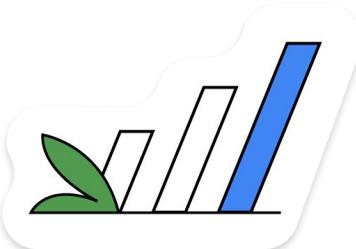
The status bar at the bottom indicates the file is 68:1 (64 chars, 1 line break), and the encoding is CRLF, UTF-8, with 4 spaces.



# Demo of Android Sample App

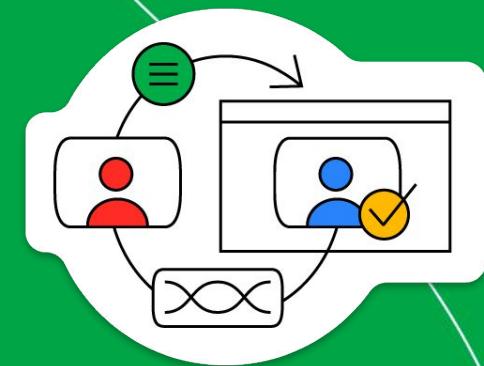


# Gemma 2B running on Android & iOS



- Video sped up for demo purposes
- May display inaccurate or offensive information that doesn't represent Google's views

# gemma.cpp

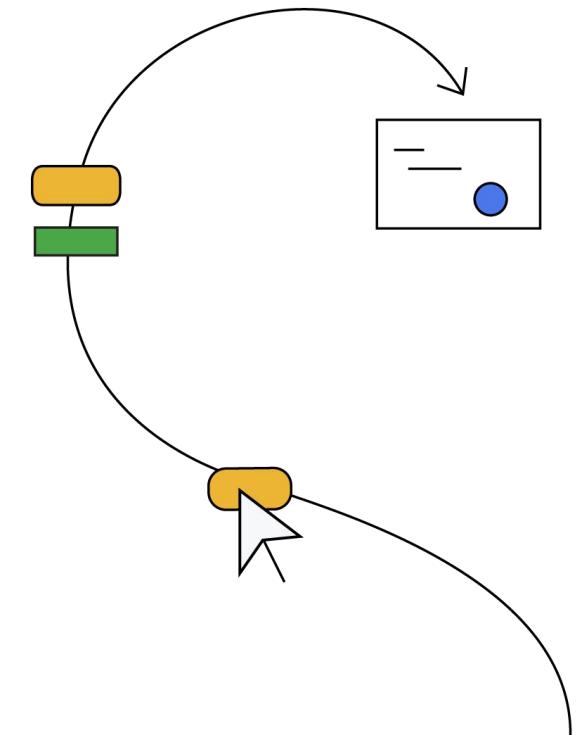


Train the Trainer in AI

# What is gemma.cpp?

gemma.cpp is a lightweight, standalone C++ inference engine for Google's Gemma series of foundation models.

It prioritizes simplicity and portability, and being built with C++, can be easily integrated into other native projects.



# Who is this project for?

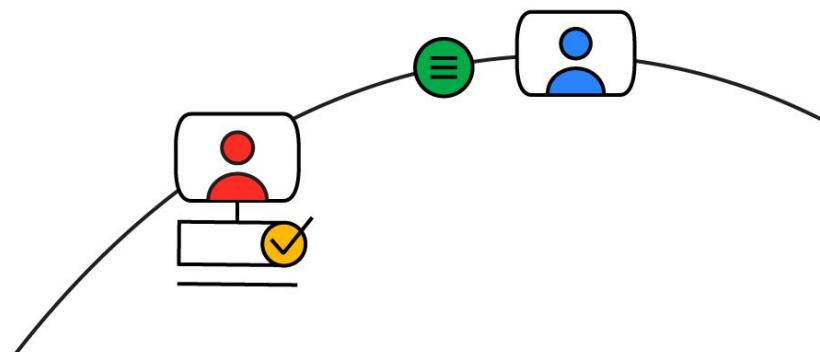
- **AI researchers**

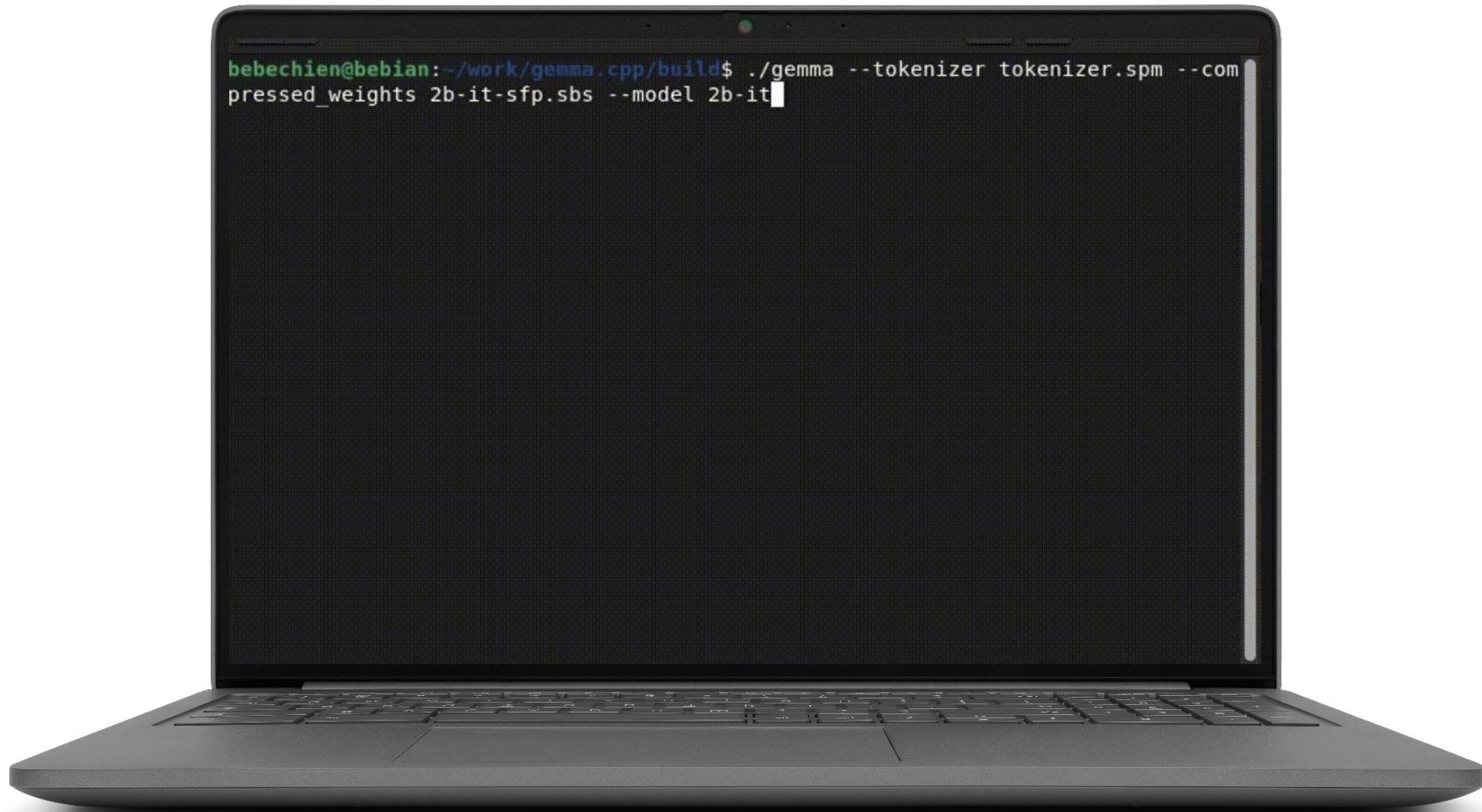
Because gemma.cpp is a C++ implementation instead of a black box, researchers can easily experiment with direct code modifications

- **C++ developers with LLM needs**

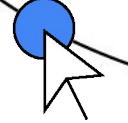
Developers building or maintaining existing native apps that want to add LLM capabilities

- Anyone who thinks it's cool! (Maybe that's you.)





# How to use gemma.cpp?



## System requirements

- [CMake](#)
- [Clang C++ compiler](#), supporting at least C++17

# How to use gemma.cpp?

Clone [the git repo](#) or Download

```
$ git clone https://github.com/google/gemma.cpp.git
```



# Download a model

[Download](#)

## Model Variations

Keras PyTorch Transformers Gemma C++ TensorRT-LLM TensorFlow Lite MaxText Pax Flax

VARIATION  VERSIONS Version 2 [+ New Notebook](#)

|           |         |       |                     |
|-----------|---------|-------|---------------------|
| 2b-it-sfp | LICENSE | Gemma | BASE MODEL          |
| 2b-it     |         | gemma | · Gemma C++ · 2b-pt |
| 2b-pt     |         |       |                     |
| 2b-pt-sfp |         |       |                     |
| 7b-it     |         |       |                     |
| 7b-it-sfp |         |       |                     |
| 7b-pt     |         |       |                     |
| 7b-pt-sfp |         |       |                     |

gemma 2B. It is easy to download, build, and tinker with. This is a compressed base model that has been instruction-tuned. These will load, run, and download more quickly. These base models for further development, while instruction-tuned (IT) variants can be used



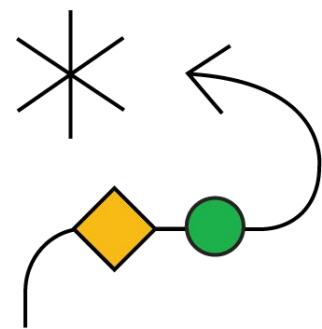
# How to use gemma.cpp?

## Extract Files

```
$ tar -xf archive.tar.gz
```

This can take a few minutes, should produce a file containing model weights such as “**2b-it-sfp.sbs**” and a tokenizer file “**tokenizer.model**”.

You may want to move these files to a convenient directory (e.g. **gemma.cpp/build/** directory)



# How to use gemma.cpp?

## Build

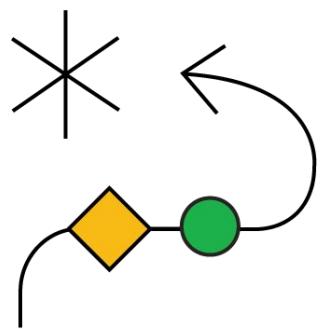
Generate the build files using cmake

```
$ cd gemma.cpp/build && cmake ..
```

Then run make to build the gemma executable

```
$ make gemma
```

If this is successful, you should now have a **gemma** executable in the **gemma.cpp/build/** directory.



# How to use gemma.cpp?



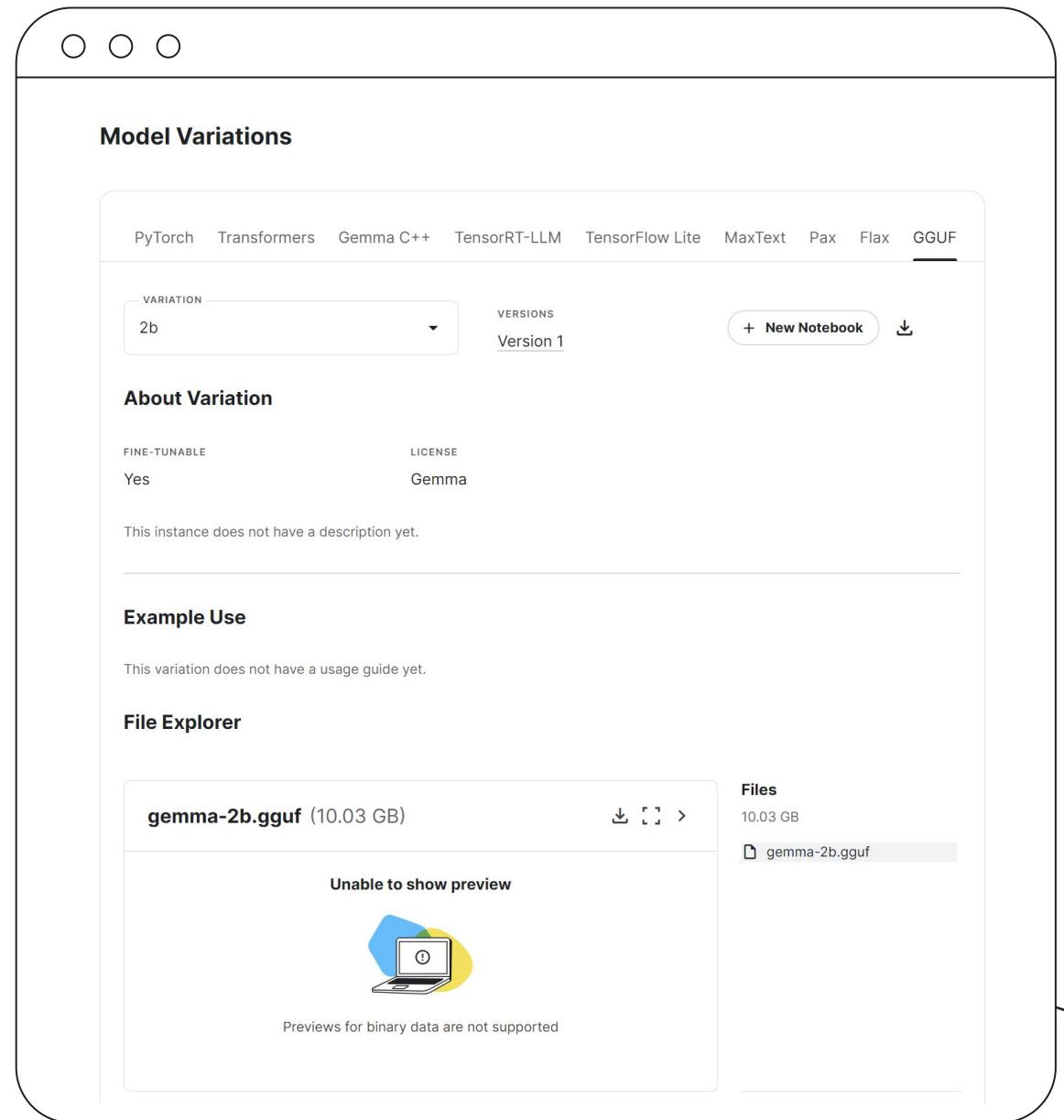
## Run

gemma has the following required arguments:

| Argument             | Description             | Example value                 |
|----------------------|-------------------------|-------------------------------|
| --model              | model type              | 2b-it   2b-pt   7b-it   7b-pt |
| --compressed_weights | compressed weights file | 2b-it-cpp-sfp.sbs             |
| --tokenizer          | tokenizer file          | tokenizer.spm                 |

# Gemma and llama.cpp

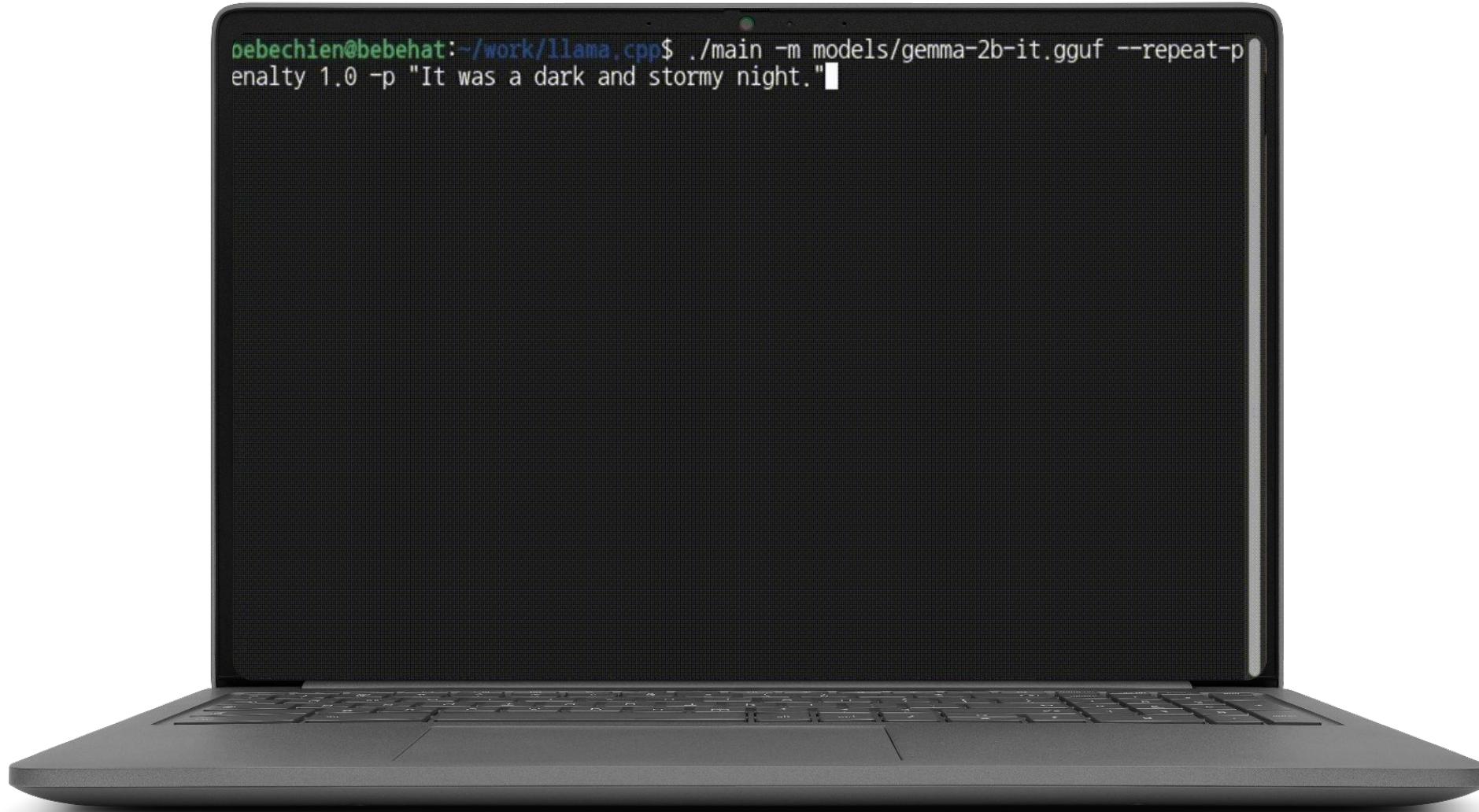
Gemma also provides  
[GGUF model variation](#)



# Gemma and llama.cpp: Usage

```
# Clone the Repo  
$ git clone https://github.com/ggerganov/llama.cpp.git  
$ cd llama.cpp  
  
# Build  
$ make  
  
# Inference!  
$ ./main -m ./models/gemma_7b-instruct.gguf --repeat-penalty 1.0
```

# Gemma and llama.cpp: Demo



# Google Cloud



Train the Trainer in AI



# Vertex AI

Duet AI

Build your own generative AI-powered agents

#### AI Solutions

Contact Center | Document | Discovery | Risk | Healthcare

Search

Conversation

#### AI Platform

Extensions | Connectors | Grounding Prompt | Serve | Tune | Distill | Eval

Gemini

and 130+  
models

#### Model Garden

Google | OSS | Partner Models



Gemma

Google Cloud Infrastructure (GPU/TPU) | Google Data Cloud

# Best of Predictive and Generative AI in one platform



## Discover

**Foundation Models**

**Task Specific Models**

**Open Models**

**Data & Algorithms**



## Prototype

**Colab & Workbench Notebooks**

**AI Studio**

**Developer SDKs**



## Develop

**Customize**  
Training on GPUs & TPUs  
Tuning: SFT, RLHF, & Distillation

**Augment**  
Grounding & Extensions  
Embeddings & Vector Search

**Evaluate**  
Experiments & Tensboard  
AutoMetrics & AutoSxS



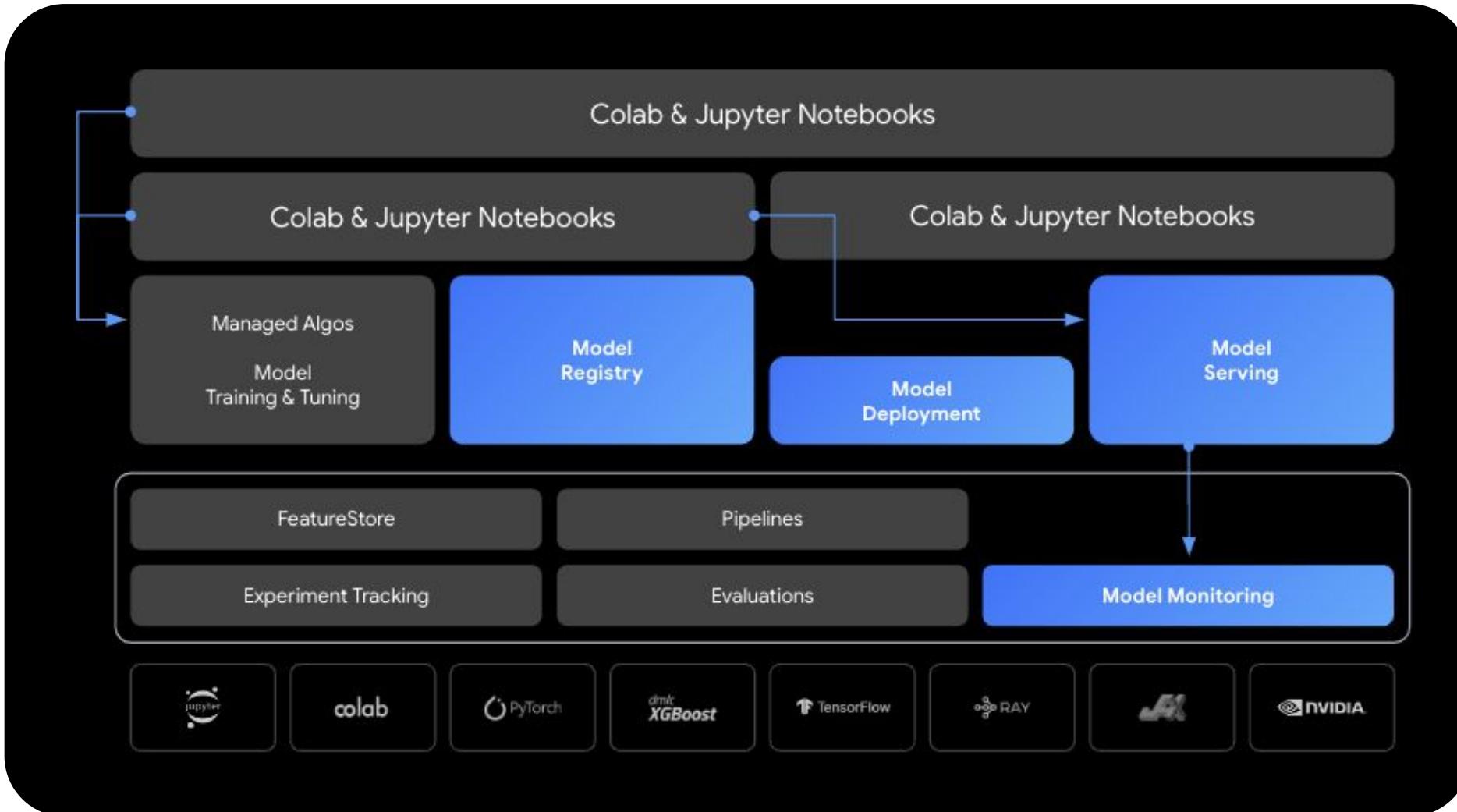
## Deploy

**Orchestrate**  
Pipelines & Metadata

**Govern**  
FeatureStore & Model Registry

**Predict**  
Endpoints on GPUs & TPUs  
Monitoring & Safety

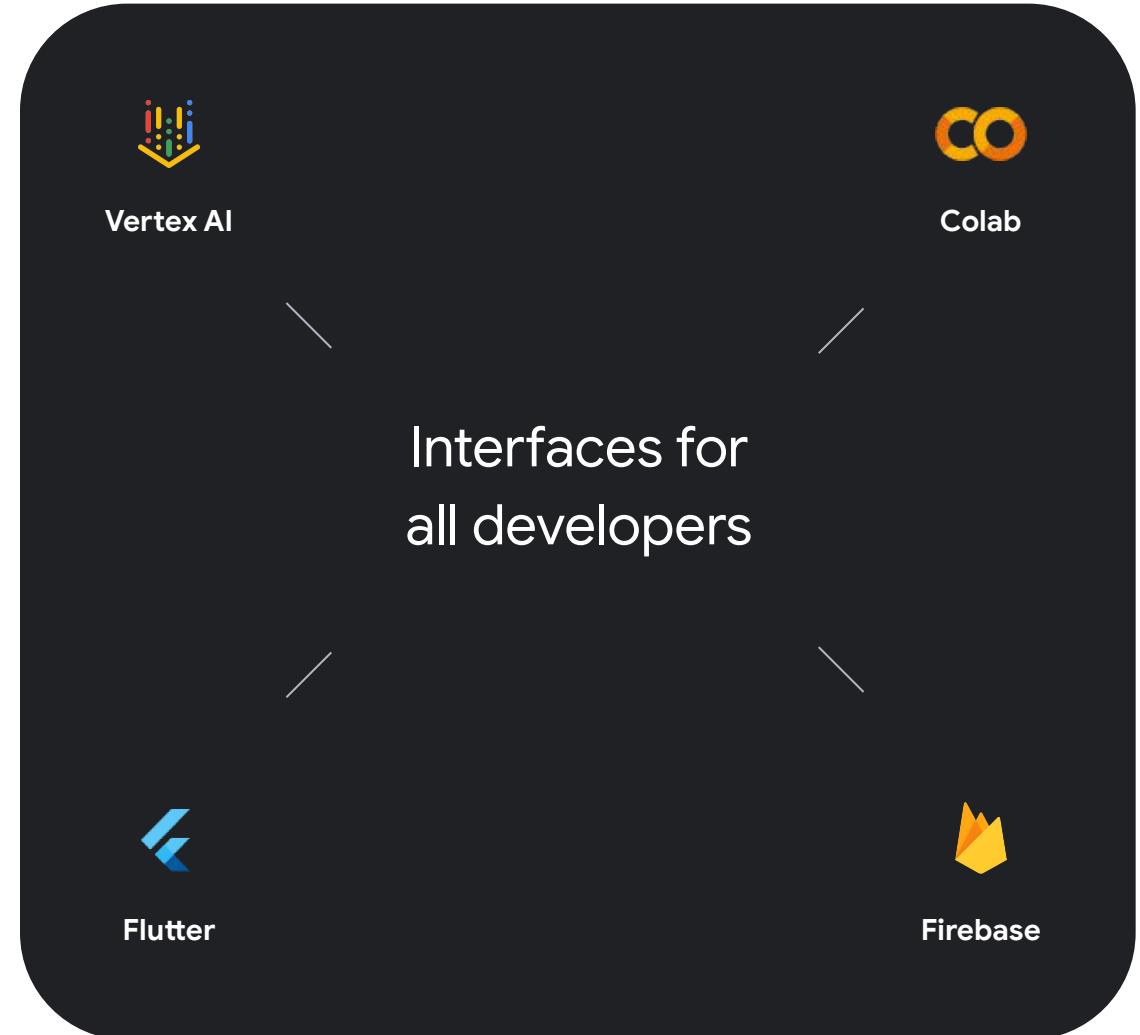
# Managed AND open, unified AND modular



# Vertex AI is built for developers

Making it easy for any developer to also be an AI developer:

- Extensive **quick start library** with code samples
- Free developer labs and training resources at <https://cloudskillsboost.google/>
- Robust integrations with popular third party developer tools
- Packages and extensions to natively support Google Cloud foundation models



# Gemma in the Vertex AI Model Garden

The screenshot shows the Vertex AI Model Garden interface. On the left, there's a large white callout box containing the text "Vertex AI" and a stylized logo composed of vertical bars in red, yellow, green, and blue. Below the logo, there are three menu items: "Feature Store", "Datasets", and "Labeling tasks". At the top of the main interface, there are navigation links: "EXPLORE GENERATIVE AI", "VIEW MY MODELS", and "DEPLOY FROM HUGGING FACE". A search bar labeled "Search models" is also present. The main content area is titled "Foundation models" and contains a list of pre-trained multi-task models. The models listed are:

- Gemini Pro**: Pre-trained multi-task model. Description: "The best performing Gemini model with features for a wide range of tasks". Checkpoint: "google/gemini-1.0-pro". "VIEW DETAILS" button.
- Gemini Pro Vision**: Multimodal model. Description: "Created from the ground up to be multimodal (text, images, videos) and to scale across a wide range of tasks". Checkpoint: "google/gemini-1.0-pro-vision". "VIEW DETAILS" button.
- Gemini Ultra Vision**: Large-scale model. Description: "The most capable model for large-scale, highly complex text and image reasoning tasks". Checkpoint: "google/gemini-1.0-ultra-vision". "VIEW DETAILS" button.
- Gemini Ultra**: Large-scale model. Description: "The most capable Gemini model for large-scale, highly complex text tasks". Checkpoint: "google/gemini-1.0-ultra". "VIEW DETAILS" button.
- Gemma**: Open checkpoint variants of Google Deepmind's Gemini model. Description: "Suited for a variety of text generation tasks". Checkpoint: "google/gemma-2b". "VIEW DETAILS" button.
- Imagen for Image Generation and Editing**: Model for image generation and editing. Description: "Use text prompts to generate novel images, edit existing ones, edit parts of an image with a mask and more.". Checkpoint: "google/imagegeneration-005". "VIEW DETAILS" button.

# Gemma's Model Card

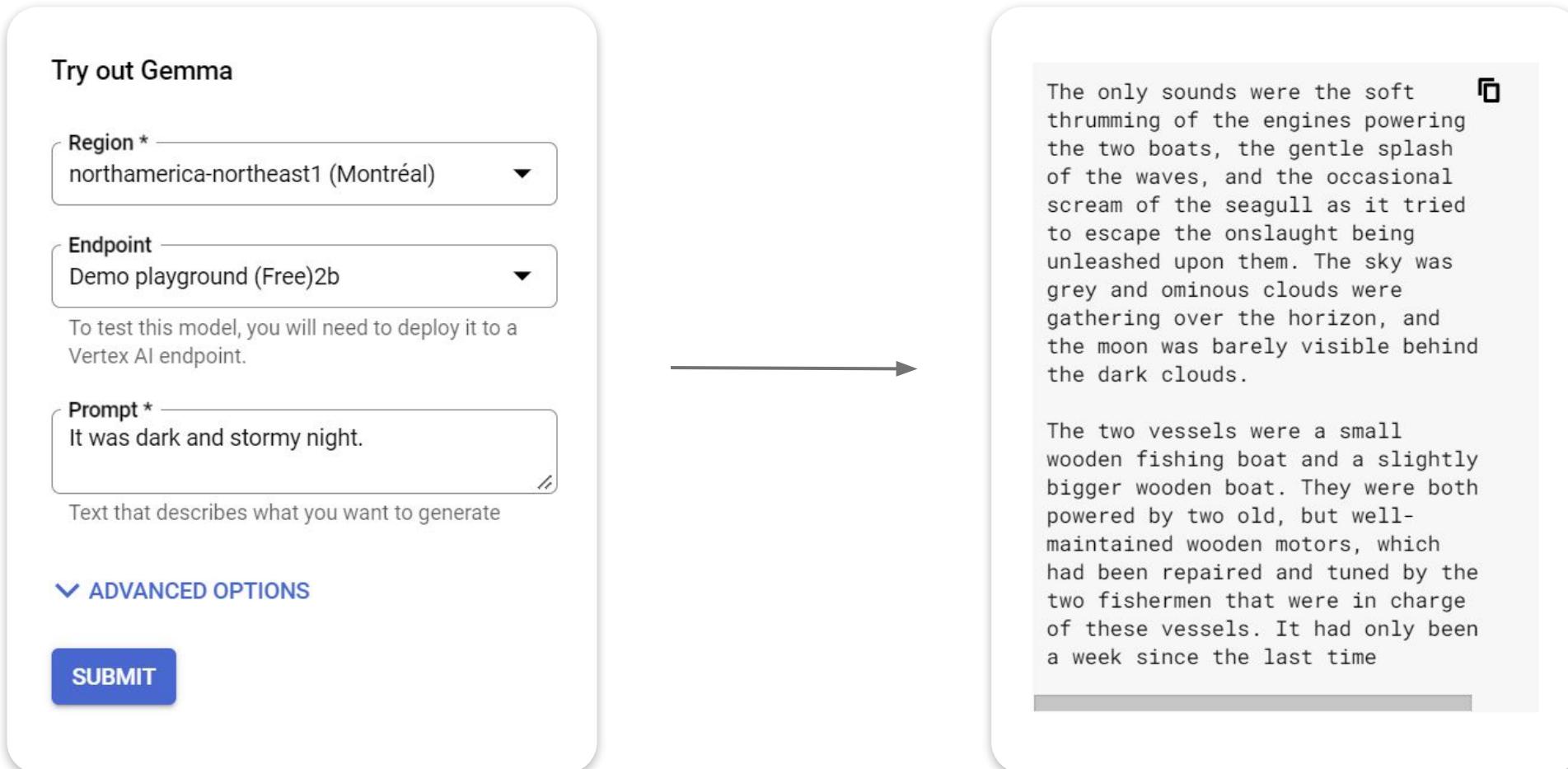
The image shows a laptop displaying the Google Cloud Platform interface for a model named "Gemma". The main page provides an overview of the Gemini model, mentioning it is a family of lightweight, state-of-the-art open models built from research and technology used to create Google Gemini models. It highlights that they are text-to-text, decoder-only large language models, available in English, with open weights, pre-trained variants, and instruction-tuned variants.

The "Overview" section notes that Gemma models are well-suited for a variety of text generation tasks, including question answering, summarization, and reasoning. Their relatively small size makes it possible to deploy them in environments with limited resources such as a laptop, desktop or your own cloud infrastructure, democratizing access to state of the art AI models and helping foster innovation for everyone.

A floating modal window titled "API is not enabled" is overlaid on the screen. It informs the user that the Vertex AI API is not enabled and provides a button to "ENABLE" it. There is also a link to "Visit the Marketplace" and a "CLOSE" button.

On the left side of the laptop screen, a modal window titled "Review the Terms of Use" is visible. It contains the "Gemma Terms of Use" and specifies the "Gemma Version Release Date: February 21, 2024". The terms of use include several legal definitions and conditions, such as the meaning of "Gemma", "Distribution", and "Model Derivatives". At the bottom of this modal are "CANCEL" and "I HAVE READ AND ACCEPT THE LICENSE FOR GEMMA" buttons.

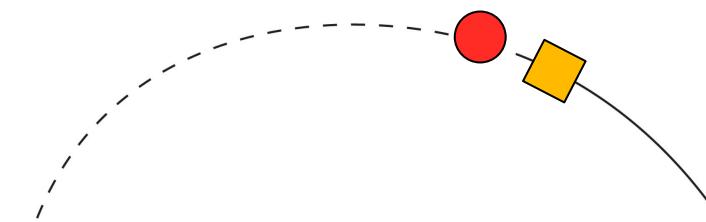
# Quick start - Trying out Gemma



# Migrating from local Gemma to Vertex AI

- Assumptions
  - You have a Google Cloud Project (GCP) with Vertex AI enabled.
  - You have **fine-tuned Gemma Model** in local
- Steps
  1. Export Gemma Model:
  2. Prepare Model for Vertex AI:
  3. Upload Model to Google Cloud Storage (GCS):
  4. Create a Vertex AI Model Resource:
  5. Import Model to Vertex AI:
  6. Deploy as an Endpoint (Optional):
  7. Test and Integrate:

<https://cloud.google.com/vertex-ai/docs/model-registry/import-model>



# Kubernetes as the foundation for Open LLMs

## Flexibility

Choice of frameworks and ecosystem tools that are container portable

## Performance

Orchestrate AI models at massive scale across specialized compute

## Efficiency

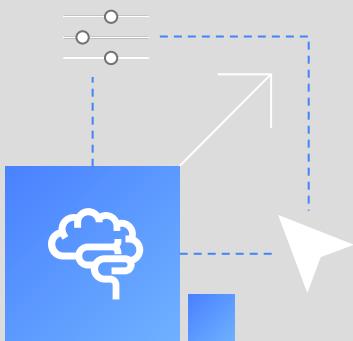
Optimize valuable compute resources while reducing operational complexity



## Kubernetes

# Harness the full potential of AI/ML with GKE

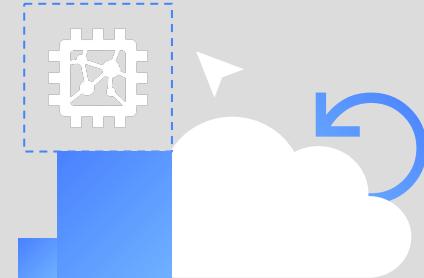
GKE is a robust, production ready AI and ML platform with all the benefits of managed Kubernetes



Train, fine-tune and serve AI models, with portability, scalability, and easy administration

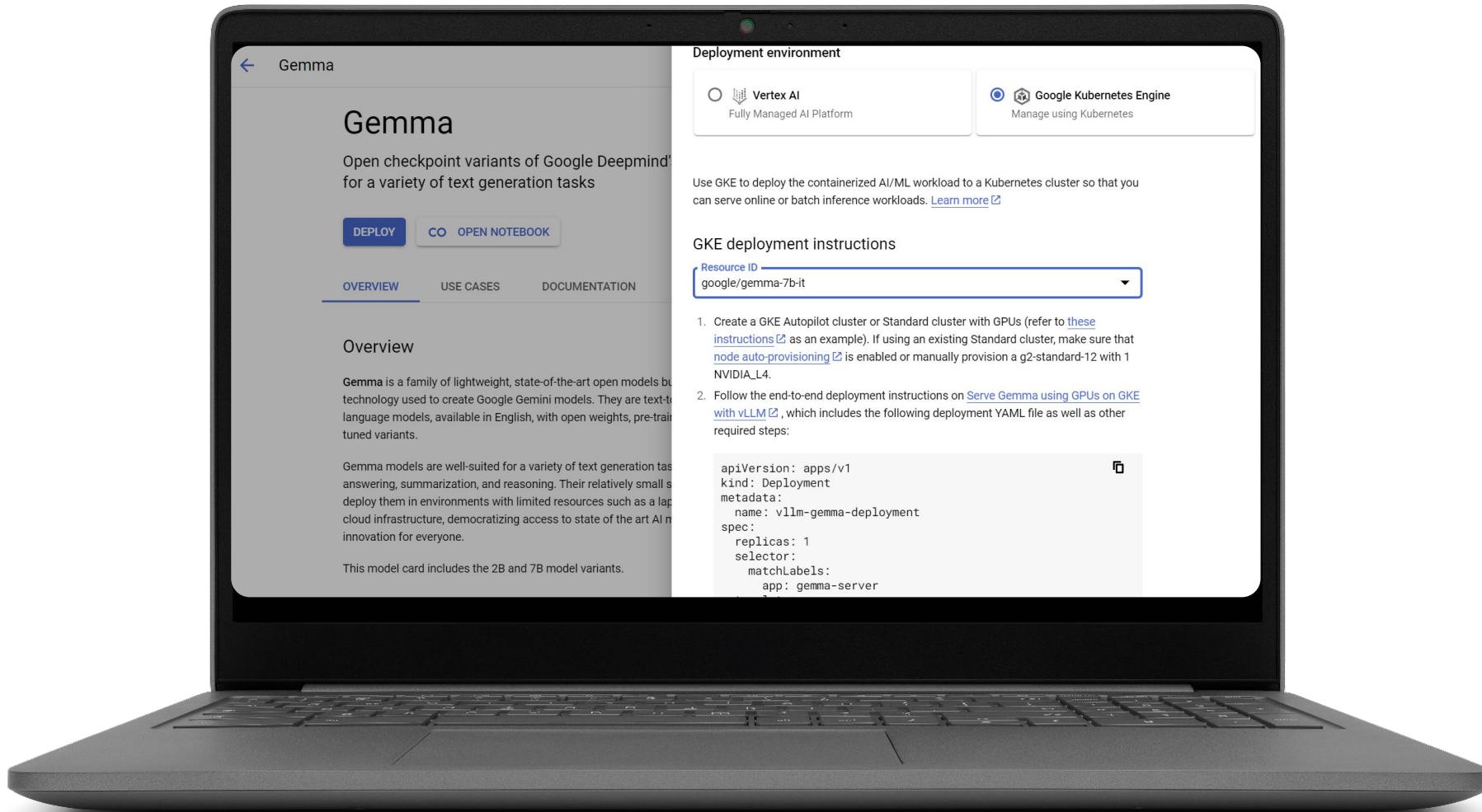


Efficient resource management, consistent ops environments, and autoscaling

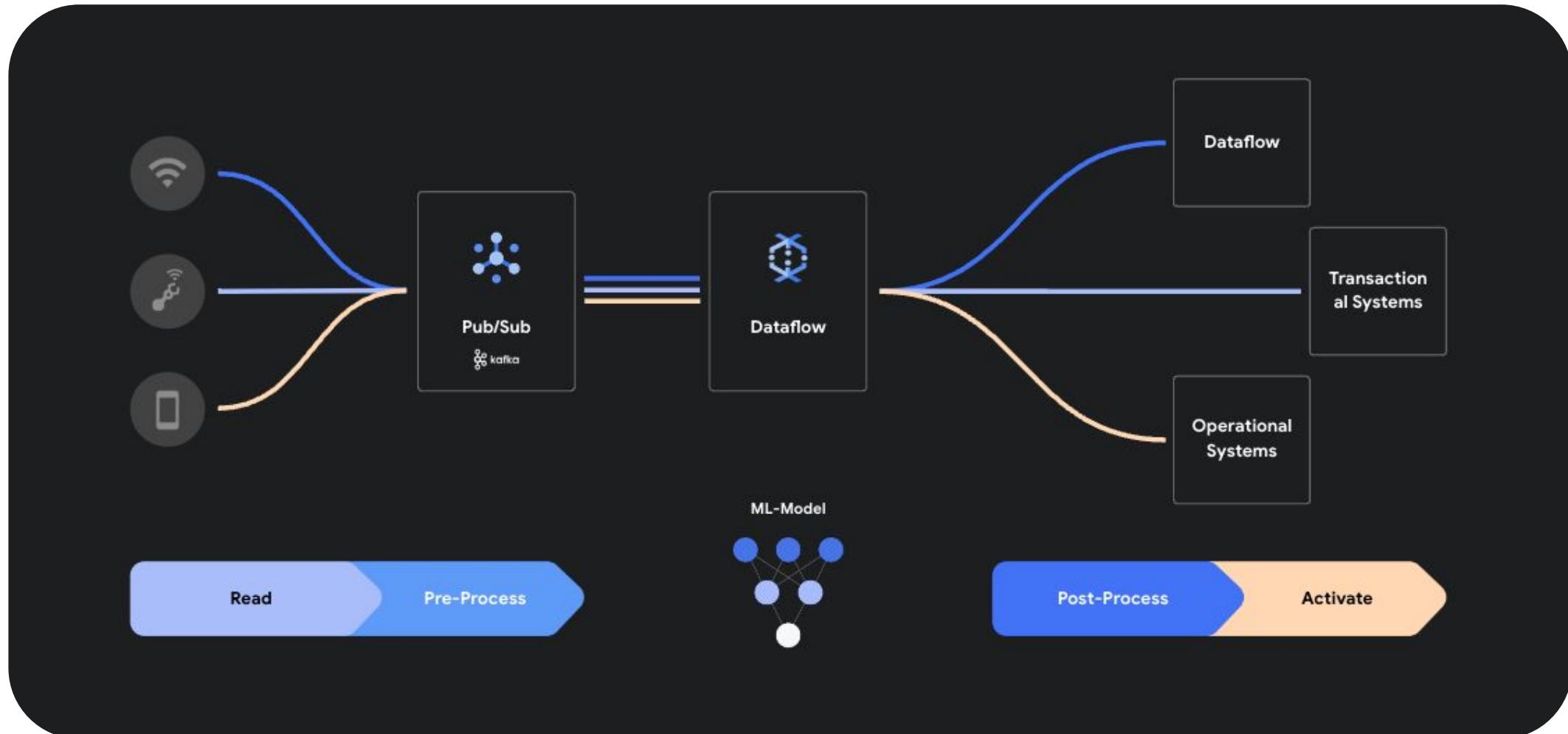


Easy orchestration of Google Cloud AI accelerators, including GPUs and TPUs, for faster training and inference

# Deploy Gemma with GKE



# Streaming ML - a with Dataflow ML



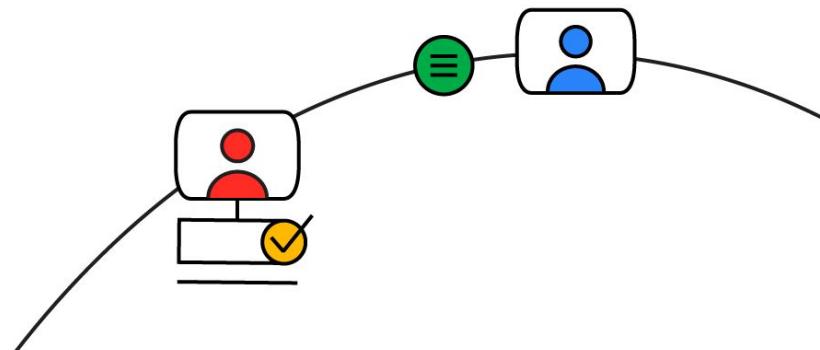
# Streaming ML - Gemma with Dataflow ML

- Low Code pipelines
- Single line of code for multiple sources and destinations
- Pre and post processing steps for real world use cases
- Simple to complex ensambles with multiple models / frameworks

```
model_handler = TFModelHandlerNumpy(model_path,
inference_fn=gemma_inference_function)
with beam.Pipeline() as p:
    _ = (p | beam.io.ReadFromPubSub(topic=topic) # Read from Source
          | beam.Map(lambda x:x.decode("utf-8")) # Decode the source data
          | RunInference(model_handler,
                         inference_args={'max_length': 32}) # Send prompt to Gemma.
          | beam.ParDo(FormatOutput()) # Format the output.
          | beam.Map(print) # Print the formatted output.
    )
```

# Useful Resources

- Serve Gemma using GPUs
  - [on GKE with vLLM](#)
  - [on GKE with Hugging Face TGI](#)
  - [on GKE with Triton and TensorRT-LLM](#)
- Serve Gemma using TPUs
  - [on GKE with Saxml](#)
- [Gemma deployment to GKE using TGI on GPU](#)
- [Use Gemma with Dataflow](#)



# Responsible AI



Train the Trainer in AI



# Gemma

The responsible way to build

# Responsible AI

# Google's AI Principles



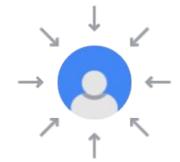
Be socially  
beneficial



Avoid creating or  
reinforcing unfair bias



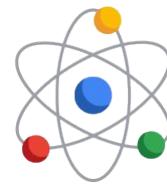
Be built and  
tested for safety



Be accountable  
to people



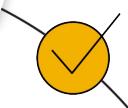
Incorporate privacy  
design principles

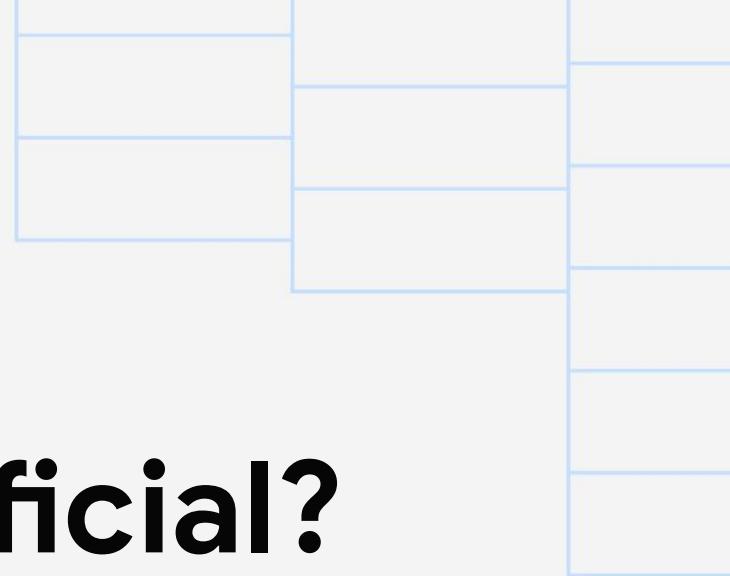
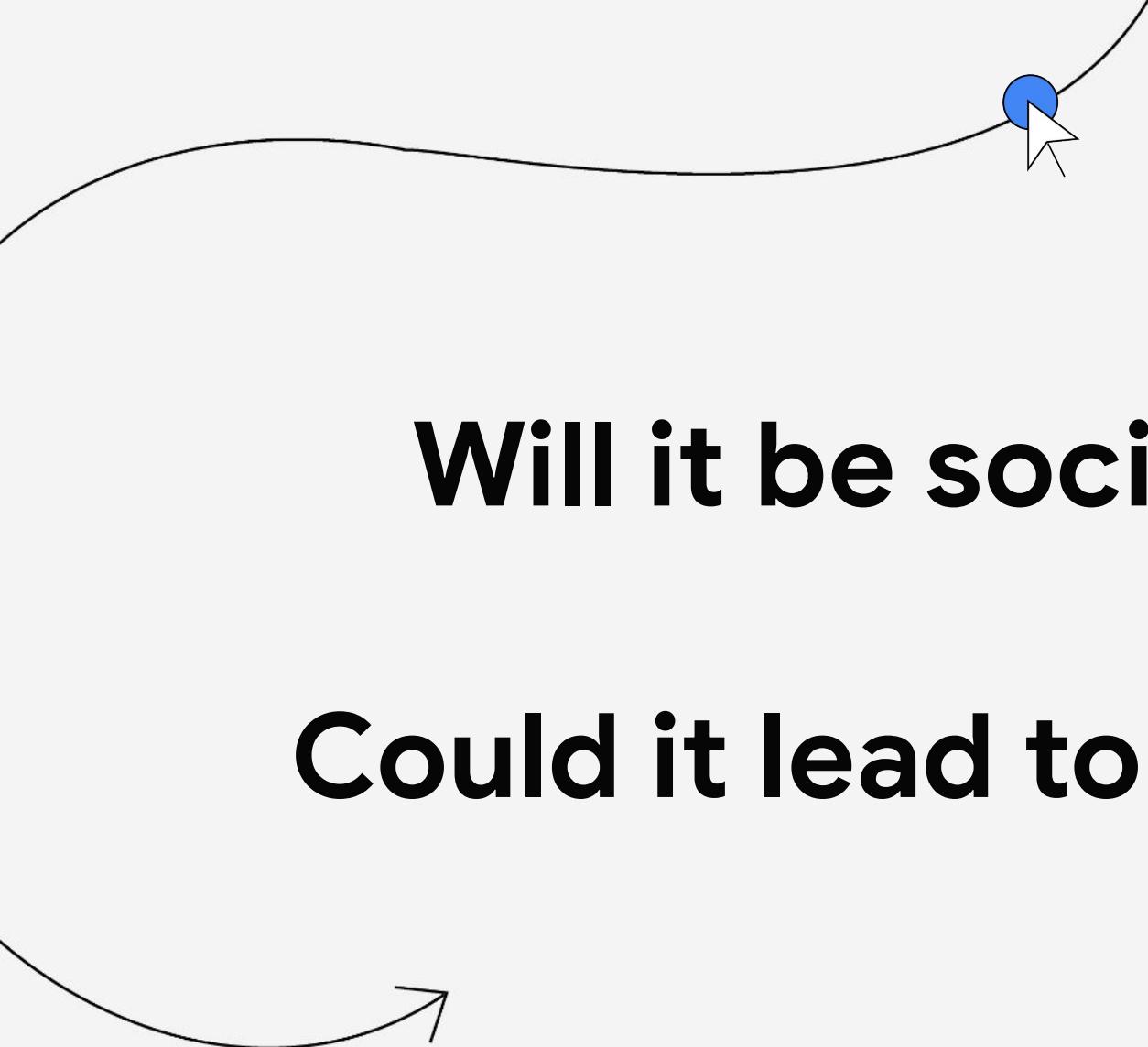


Uphold high standards  
of scientific excellence



Be made available for  
uses that accord with  
these principles





**Will it be socially beneficial?**

**Could it lead to harm in any way?**

# Gemma's approach to responsible AI



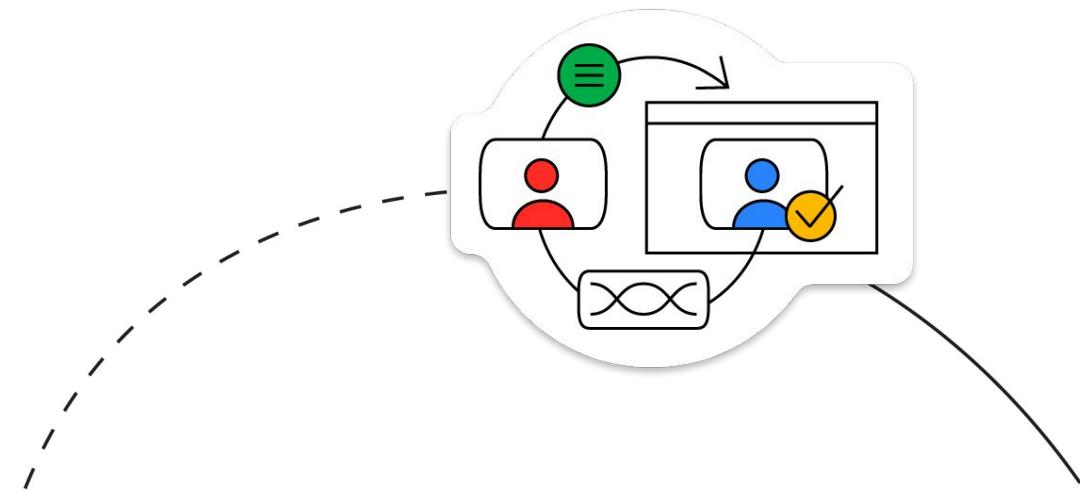
Safety by Design



Transparent and Robust Evaluations



Responsible Generative AI Toolkit



# Safety by design



## Filtered pre-training data

Filter out certain personal information and other sensitive data



## Safety tuning

Supervised Fine Tuning & Reinforcement Learning with Human Feedback for safety

# Transparent & Robust Evaluations

Tested for safety, privacy, societal risks, data memorization and dangerous capabilities.



## Human SxS

Human rater have compared Gemma and Mistral over 400 prompts testing basic for safety protocols.



## Academic benchmarks

Gemma results on 9 academic authoritative safety and fairness benchmarks.



## Internal safety evaluations

Advanced red teaming.  
Manual testing for advanced capabilities (chemical, biological risks).

# Transparent & Robust Evaluations

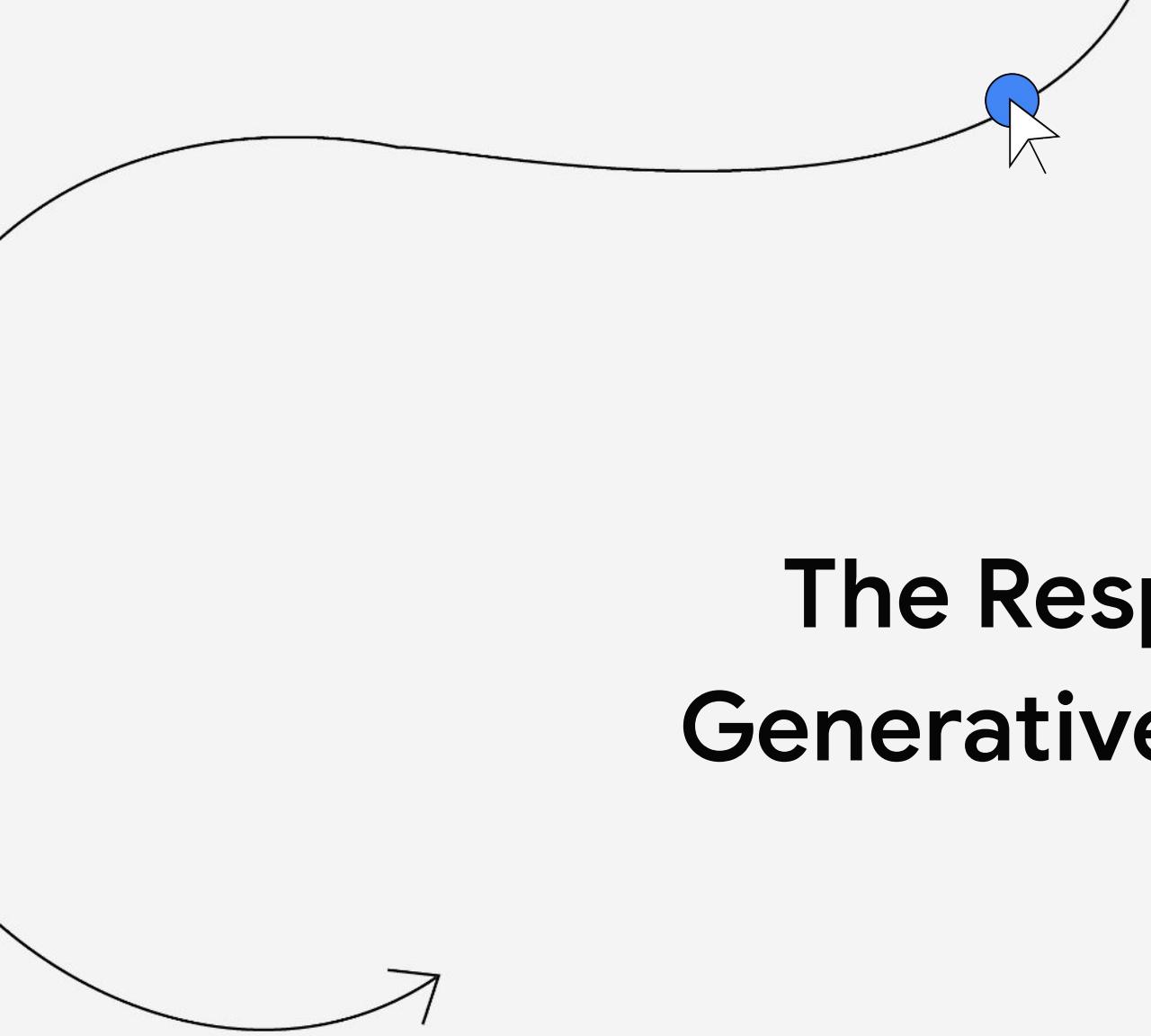
Side by Side safety testing with Mistral 7B v0.2

| Model              | Safety                |
|--------------------|-----------------------|
| <b>Gemma 7B IT</b> | <b>58%</b>            |
| 95% Conf. Interval | [55.9%, 60.1%]        |
| Win / Tie / Loss   | 42.9% / 30.2% / 26.9% |
| <b>Gemma 2B IT</b> | <b>56.5%</b>          |
| 95% Conf. Interval | [54.4%, 58.6%]        |
| Win / Tie / Loss   | 44.8% / 22.9% / 32.3% |

Source: Gemma paper



# The Responsible Generative AI Toolkit



# Responsible Generative AI Toolkit

## safety classifiers

A hate speech classifier.

Methodology to build any classifier with limited data points.

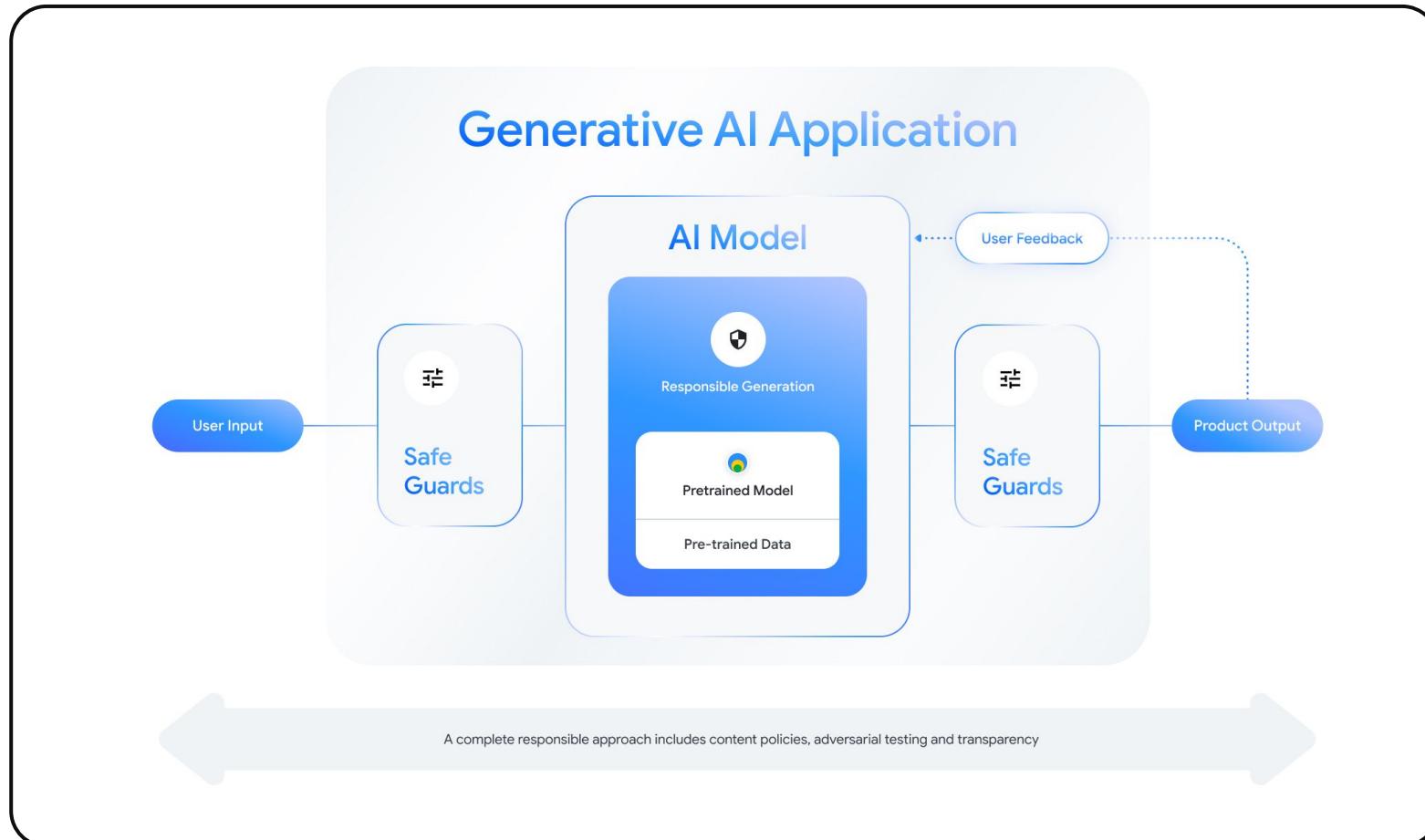
## Model debugging

The first LLM prompt-debugger, based on saliency methods.

## RAI guidance

Guidance on developing Responsible models

# Build your safety classifier



# Build your safety classifier

## Custom classifier

1. Collect 80-500 training data examples
2. Parameter efficient tune using LoRA
3. Get model scores or predictions and evaluate

Hate speech classifier

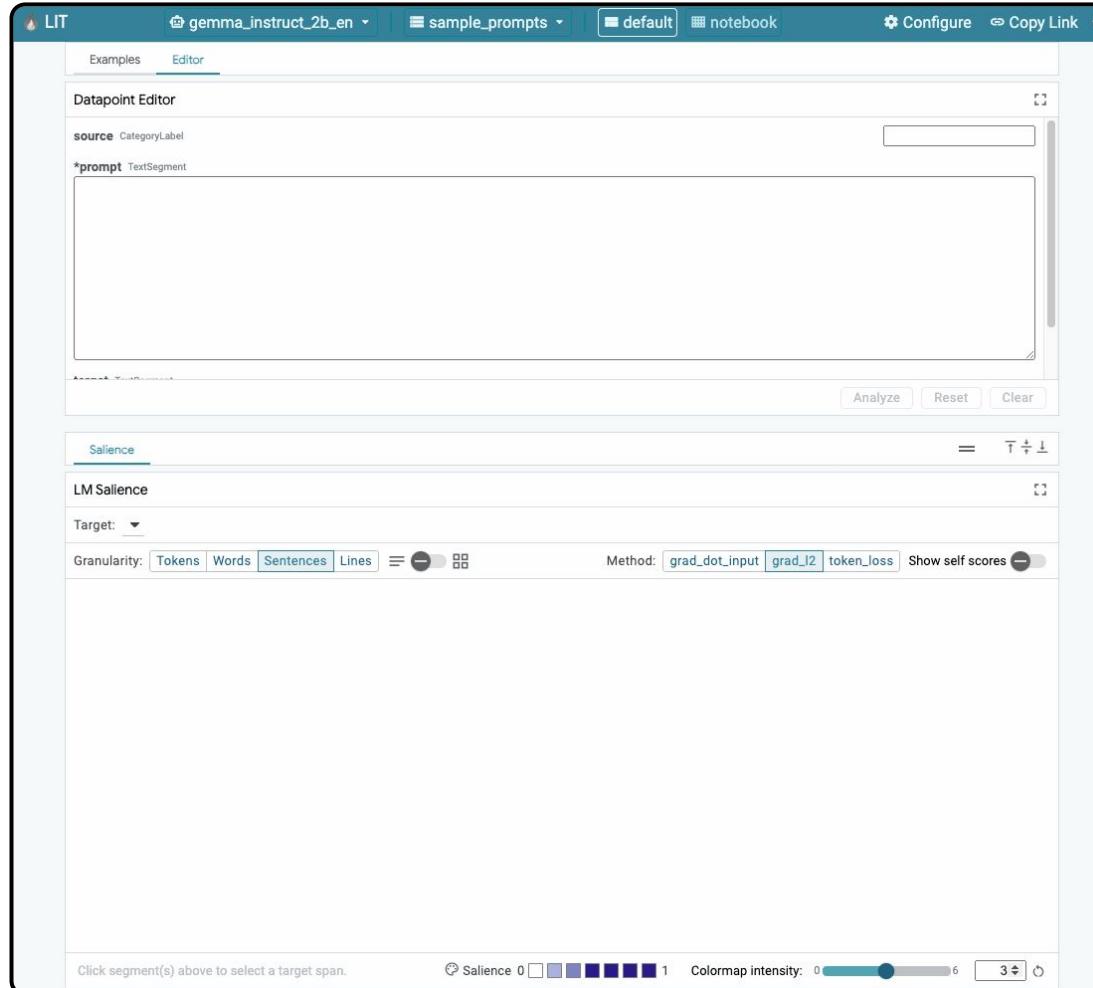
200 data points  
SOTA on ETHOS  
leaderboard

F1: 0.8



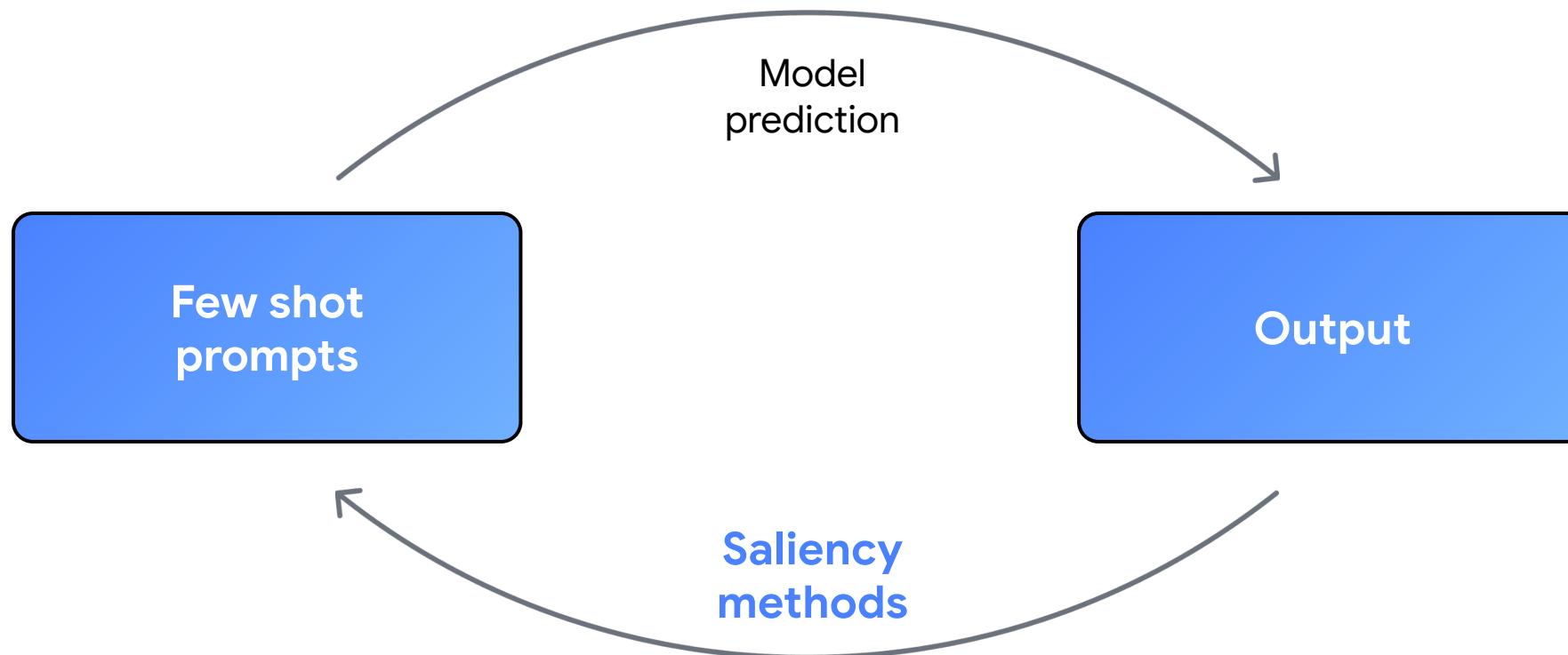
Start Codelab

# Investigate your prompts



<https://pair-code.github.io/lit/>

# Investigate your prompts



# Apply Responsible AI best practices



## Align with GenAI content policies

Standardized content safety policies.



## Model Mitigations

Generate content that is safe and supportive of diverse voices & cultures.



## Input and output filtering

Detect and avoid showing harmful content to users.



## Adversarial testing & eval

Assess model performance to measure risks of failures with high-quality data.



## Simple, Helpful Explanations

Transparency, feedback and user control.

For more information on Responsible AI

<https://ai.google.dev/responsible>



# Safety as a Shared Responsibility

- Gemma is an open model – accessible for further development.
- Those adapting Gemma must uphold safety and quality standards.
- Responsibility extends to creators and end-users.





[goo.gle/gemma](https://goo.gl/gemma)

[goo.gle/gemma-discord](https://goo.gle/gemma-discord)

# Thank you!



Gus Martins  
he/him  
AI Developer Relations Engineer



Ju-yeong Ji  
he/him  
AI Developer Relations Engineer