

How to build Gen AI models in Python and through AWS

Saousan KADDAMI

Research scientist at Amazon

Agenda

da

Introduction to Generative AI

Generative AI applications and benefits

Building Gen AI models: Theoretical

Frameworks Building Gen AI models:

Practical Tools

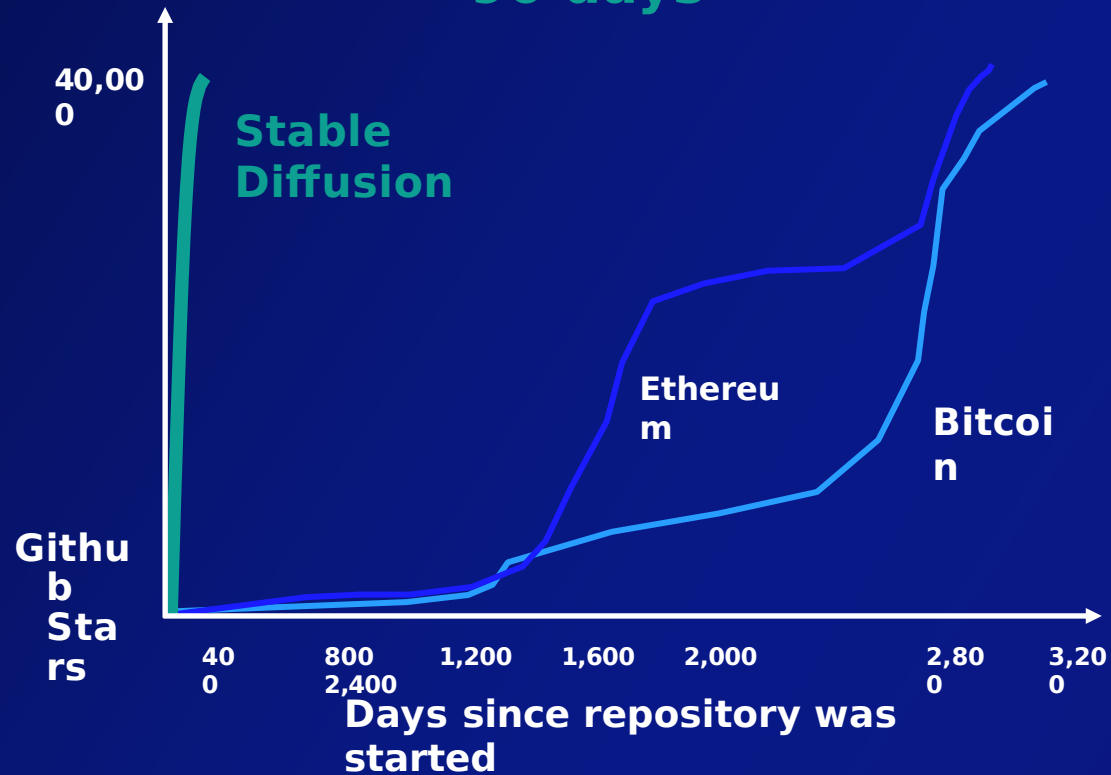
Tips and Tricks for Customized Gen AI models

Introduction to Generative AI

Generative AI is the fastest growing trend in AI

Developer adoption

Stable Diffusion accumulated 40k stars on GitHub in its first 90 days



Consumer adoption

ChatGPT reached the 1 million users mark in just 5 days



What is Generative AI?



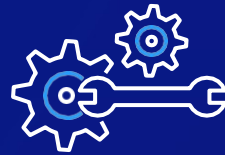
AI that can **generate content** close enough to human created content for real-world tasks



Powered by **foundation models** pre-trained on large sets of data with several hundred billion parameters



Applicable to **many use cases** like text summarization, question answering, digital art creation, code generation, etc.



Tasks can be **customized for specific domains** with minimal fine-tuning



New Volvo car concept design by midjourney
Credit: @sugar_design_1 Instagram

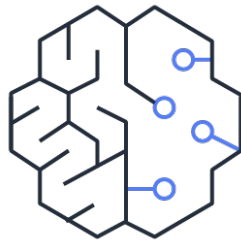
What is Generative AI?

Artificial Intelligence (AI)

Machine Learning (ML)

Deep Learning (DL)

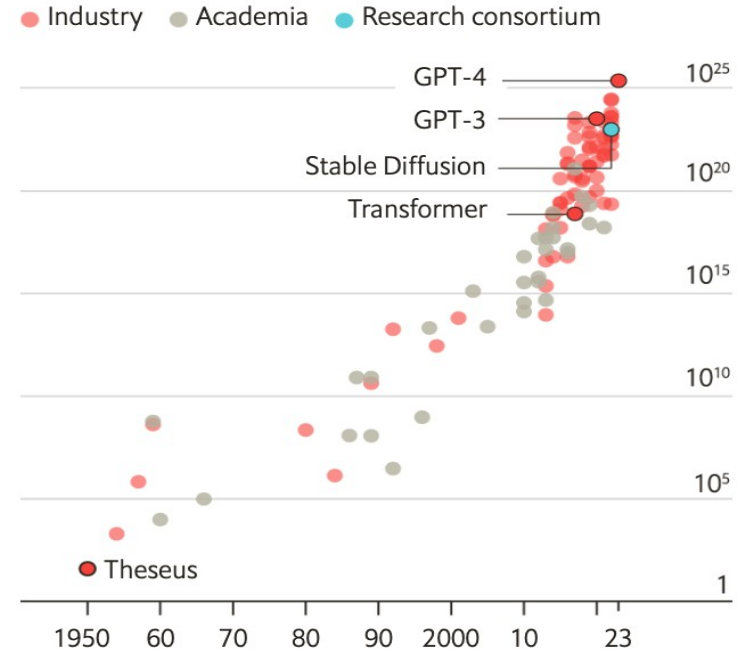
Generative AI



Faster, higher, more calculations

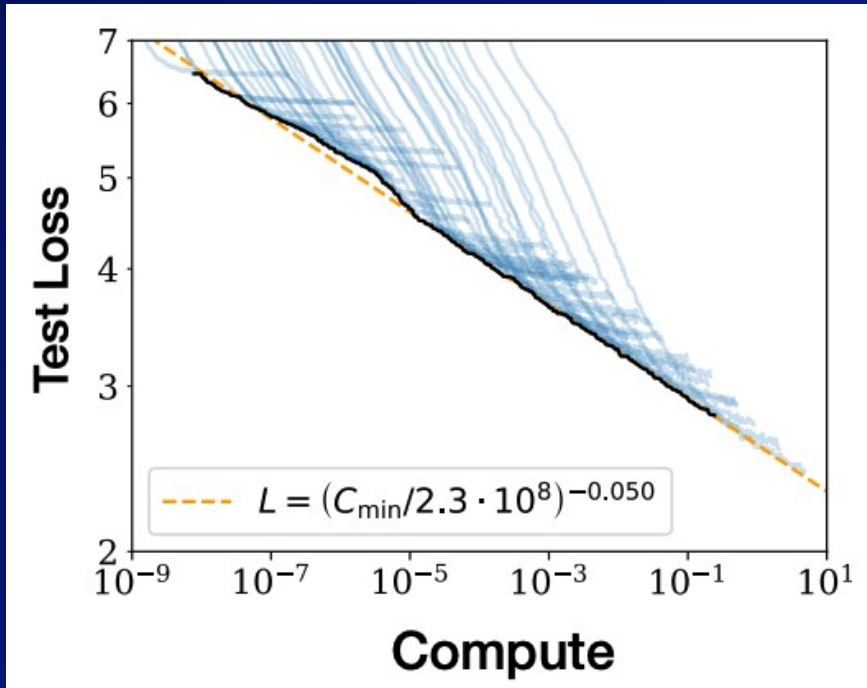
Computing power used in training AI systems

Selected systems, floating-point operations, log scale

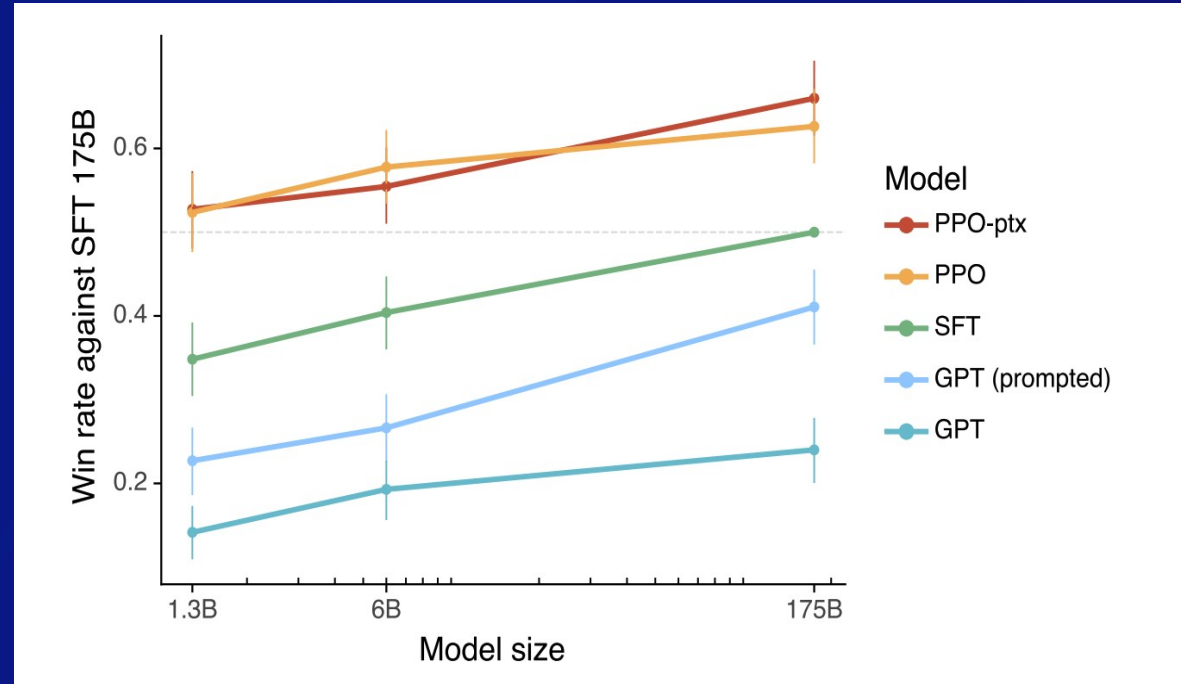


Sources: Sevilla et al., 2023; Our World in Data

Accuracy revolution



Scaling Laws for Neural Language Models
Kaplan et al, 2020



Training language models to follow instructions with human feedback (InstructGPT)

GPT-3 showed you can 3x accuracy with 10x model size increase

Accuracy revolution

“a picture of a very clean living room”



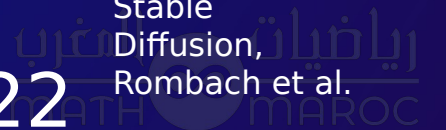
StackGAN,
Zhang et al.

2016



Stable
Diffusion,
Rombach et al.

2022



Gen AI applications and benefits

Generative AI is emerging across a range of domains ...



Productivity
Text generation



Chat
Virtual assistant



Summarization
Text extraction



Search



Code generation



Image generation



Image classification



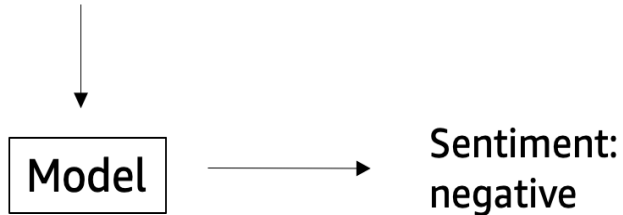
Create music



Generate videos

Many ML tasks can now be re-cast as generative

Text: I am not into this house; it's way too expensive and too far from the train line!



Traditional classification

Text: I am not into this house; it's way too expensive and too far from the train line!

Classify this sentence into positive or negative sentiment:

Agent: Negative sentiment

Using generation to classify text

Generative AI creates significant business value



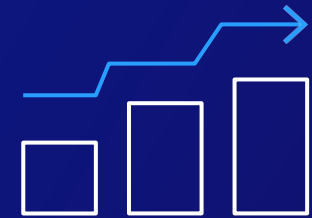
CREATIVITY

Create new content and ideas, including conversations, stories, images, videos, and music



PRODUCTIVITY

Radically improve productivity across all lines of business, for example 57% faster task completion with [Amazon CodeWhisperer](#)



ECONOMIC GROWTH

\$7T increase in global GDP over the next 10 years

Let's say I asked you to learn **everything** on the internet. How would you do it?



Structure



Storage



Time

5.74 B pages x 52
seconds = ~82 M hours

=> **40,000 human**

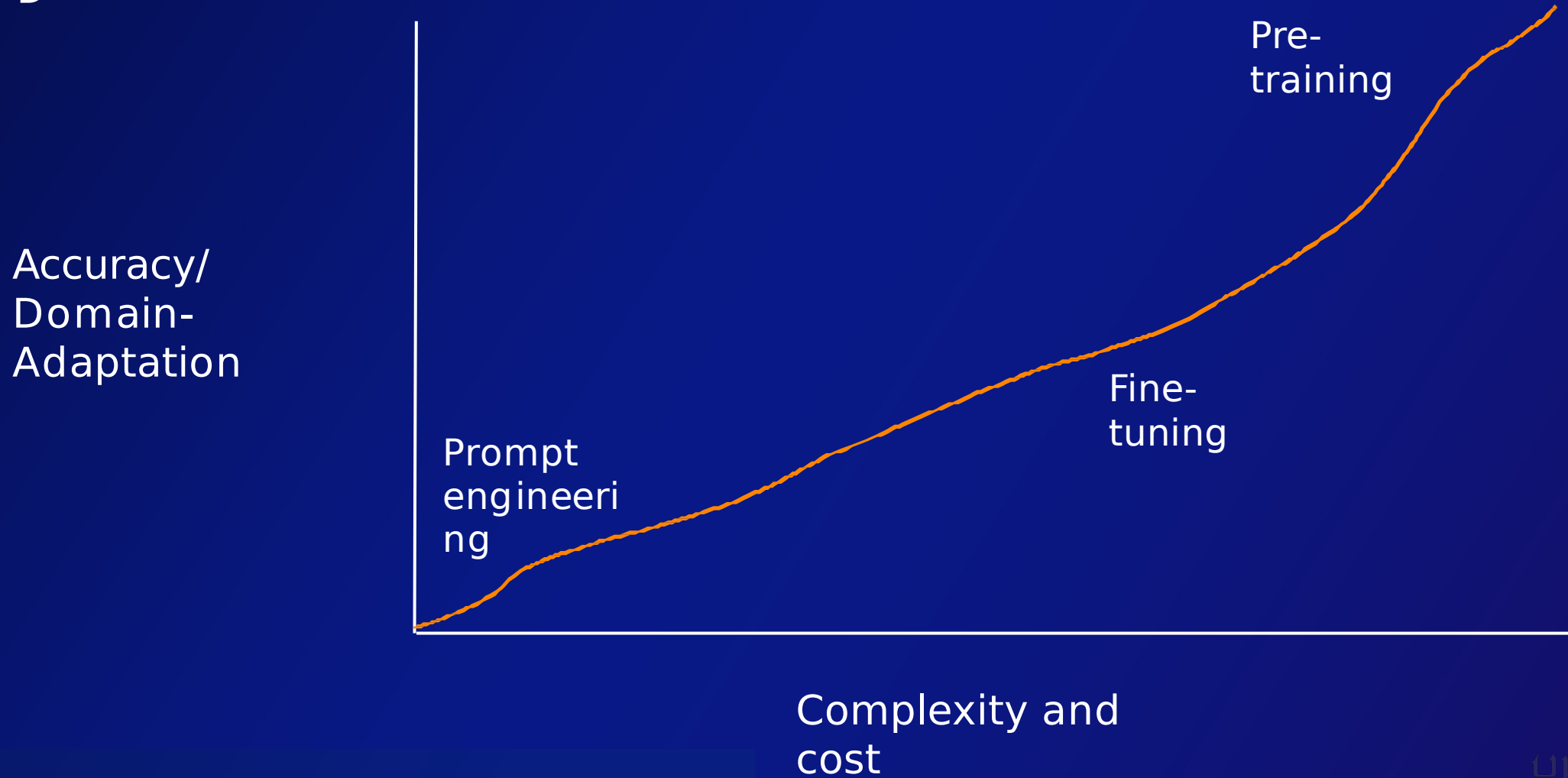
A foundation model
can do this in a few
minutes.

Building Gen AI : Theoretical Frameworks

Building Gen AI models:

	Prompt engineering on existing models	Fine tuning	Pretraining
Training duration (and cost)	Not required	Minutes to hours	Days to weeks to months
Customization	<ul style="list-style-type: none">• No customization on model• Customizing the prompt	Some <ul style="list-style-type: none">• Specific task tuning• Added domain-specific training data	FULL <ul style="list-style-type: none">• NN architecture and size• Vocabulary size• Context length• Training data
Expertize needed	Low	Medium	High

Customizing a foundation model to your domain

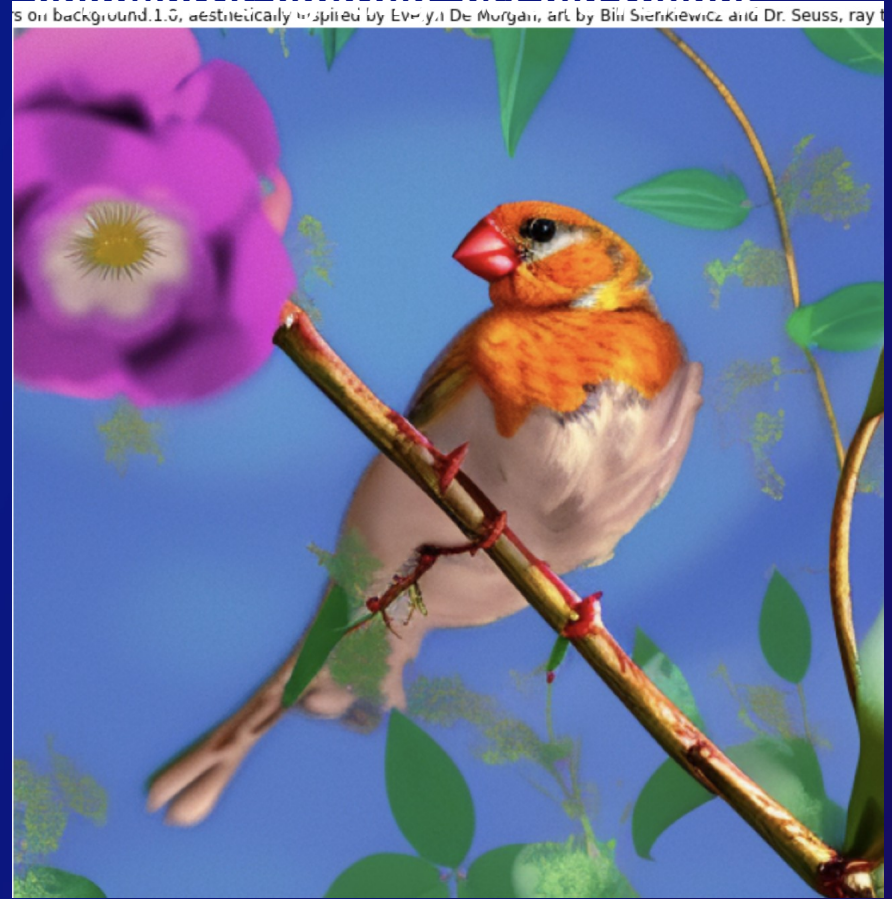


One-Shot prompting:

Finch



A tiny finch on a branch with spring flowers, aesthetically inspired by Evelyn De Morgan, art by Bill Sienkiewicz and Dr. Seuss, ray tracing, volumetric lighting, octane render



Few-shot prompting:

Review:

I like the film, but the actor James was not performing as usual

What are the entities and their associated sentiments? Film : positive, Actor James: Negative

Review:

I loved the food in the restaurant but the place was not clean

What are the entities and their associated sentiments? Food : positive, Place: negative

Review:

I enjoyed the hotel room but the reception guy was unpleasant

What are the entities and their associated sentiments? Hotel room: positive, Reception guy: negative

Review:

I liked the boat trip, but the weather was not good

What are the entities and their associated sentiments?

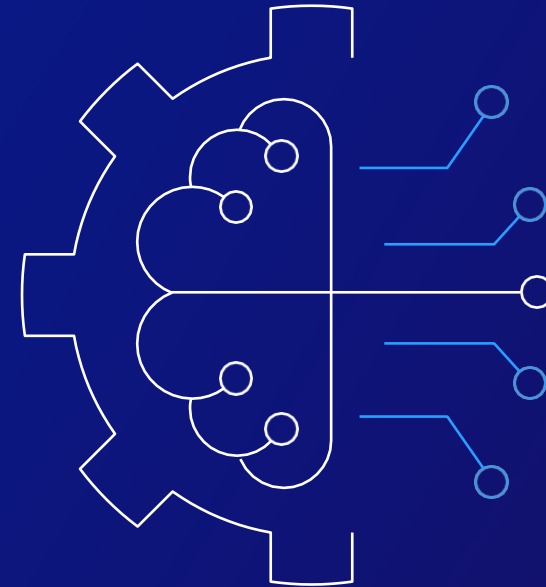
The reasoning result is: '['negative, Boat trip: positive, Weather: negative']'

Building Gen AI models: Practical Tools

NEW






Amazon Bedrock

The easiest way to
build and scale
generative AI
applications with FMs



How it works

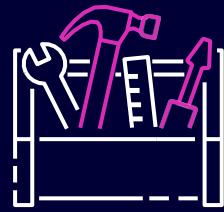
- Finding the right model from a list of FMs from leading AI startups and Amazon
- Privately Customizing FMs with specific data
- Integration and deployment into applications using AWS

Amazon Titan  <small>Text summarization, generation, classification, open-ended Q&A, information extraction, embeddings, and search.</small>	Jurassic-2  <small>Instruction-following LLMs for any language task including question answering, summarization, text generation, and more.</small>	Claude 2  <small>LLM for thoughtful dialogue, content creation, complex reasoning, creativity, and coding, based on Constitutional AI and harmlessness training.</small>
Command and Embed  <small>Text generation model for business applications and embeddings model for search, clustering, or classification in 100+ languages.</small>	Stable Diffusion  <small>Generation of unique, realistic, and high-quality images, art, logos, and designs.</small>	

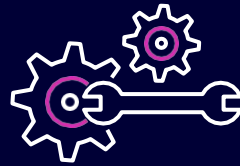
Amazon Bedrock key benefits



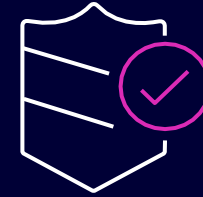
Accelerate development of generative AI applications using FMs through an API, without managing infrastructure



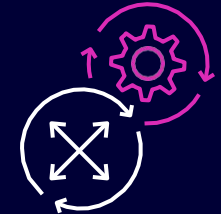
Choose FMs from AI21 Labs, Anthropic, Stability AI, and Amazon to find the right FM for your use case



Privately customize FMs using your organization's data

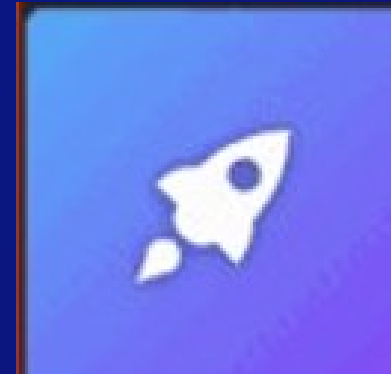


Enhance your data protection using comprehensive AWS security capabilities



Use AWS tools and capabilities that you are familiar with to deploy scalable, reliable, and secure generative AI applications

**Amazon
Jumpstart**
Provides pretrained
open-source models for
direct deployment or
Notebook
experimentations




Jumpstart Foundation Models

Home


Customize layout

▼ Quick actions




Open Launcher


Create notebooks and other resources




Import & prepare data visually



Open the Getting Started notebook




Read documentation



View guided tutorials


▼ Prebuilt and automated solutions

Deploy built-in algorithms, pre-built solutions, example notebooks, and build models from visual interface.



JumpStart

Pretrained models, notebooks, and prebuilt solutions



AutoML

Automatically build, train, and tune the best ML models

▼ Workflows and tasks

Kick off a new step in the machine learning workflow.

Prepare data	Build, train, tune model	Deploy model
<ul style="list-style-type: none">Connect to data sources	<ul style="list-style-type: none">View all experiments	<ul style="list-style-type: none">Get endpoint recommendation

Playground experience

Amazon SageMaker

Getting started
Studio
Studio Lab
Canvas
Workbench
Tutorialboard

Domains

SageMaker dashboard
Images
Lifecycle configurations
Search

Jumpstart
Foundation models
Computer vision models
Natural language processing models

Governance
Ground Truth

Notebook
Processing
Training
Inference

Compilation jobs
Marketplace model packages
Models
Endpoint configurations
Endpoints
Batch transform jobs
Shadow tests

Edge Manager

Augmented AI

AWS Marketplace


Tutorials
Documentation

Amazon SageMaker > Foundation models

Foundation models


Foundation models are pre-trained on large amounts of data so you can perform a wide range of tasks such as article summarization and text, image, or video generation. Choose from a variety of foundation models below to accelerate your application development.

Foundation models




AI21 Summarize
By AI21 Labs | Nov 1, 2022
THE INPUT TEXT SHOULD CONTAIN AT LEAST 40 "WORDS" AND NO MORE THAN 50,000 "CHARACTERS". THIS TRANSLATES TO ROUGHLY 10,000 WORDS, OR AN IMPRESSIVE 40 PAGES!
Summarize texts with our world-class summarization engine. Quick integration with high quality for all kinds of text.

View model




AI21 Jurassic-2 Jumbo
By AI21 Labs | Nov 1, 2022
PRE-TRAINED LANGUAGE MODEL TRAINED BY AI21 LABS ON A CORPUS OF WEB TEXT INCLUDING NATURAL LANGUAGE AND COMPUTER PROGRAMS WITH RECENT DATA - UPDATED TO MID 2022. THIS MODEL HAS A 8192 TOKEN CONTEXT WINDOW (I.E. THE LENGTH OF THE PROMPT + COMPLETION SHOULD BE AT MOST 8192 TOKENS).
Best-in-class large language model designed for maximum quality. Ideal for generating text using plain English.

View model




AI21 Jurassic-2 Jumbo Instruct
By AI21 Labs | Nov 1, 2022
PRE-TRAINED LANGUAGE MODEL TRAINED BY AI21 LABS ON A CORPUS OF WEB TEXT INCLUDING NATURAL LANGUAGE AND COMPUTER PROGRAMS WITH RECENT DATA - UPDATED TO MID 2022. THIS MODEL HAS A 8192 TOKEN CONTEXT WINDOW (I.E. THE LENGTH OF THE PROMPT + COMPLETION SHOULD BE AT MOST 8192 TOKENS).
Best-in-class instruction following model designed for maximum quality. Ideal for generating text using plain instructions.

View model




AI21 Jurassic-2 Grande
By AI21 Labs | Nov 1, 2022
PRE-TRAINED LANGUAGE MODEL TRAINED BY AI21 LABS ON A CORPUS OF WEB TEXT INCLUDING NATURAL LANGUAGE AND COMPUTER PROGRAMS WITH RECENT DATA - UPDATED TO MID 2022. THIS MODEL HAS A 8192 TOKEN CONTEXT WINDOW (I.E. THE LENGTH OF THE PROMPT + COMPLETION SHOULD BE AT MOST 8192 TOKENS).
Best-in-class large language model with optimal quality-latency trade-off. Ideal for generating text using plain English.

View model




AI21 Jurassic-2 Grande Instruct
By AI21 Labs | Nov 1, 2022
PRE-TRAINED LANGUAGE MODEL TRAINED BY AI21 LABS ON A CORPUS OF WEB TEXT INCLUDING NATURAL LANGUAGE AND COMPUTER PROGRAMS WITH RECENT DATA - UPDATED TO MID 2022. THIS MODEL HAS A 8192 TOKEN CONTEXT WINDOW (I.E. THE LENGTH OF THE PROMPT + COMPLETION SHOULD BE AT MOST 8192 TOKENS).
Best-in-class instruction following model with optimal quality-latency trade-off. Ideal for generating text using plain instructions.

View model




AI21 Jurassic-2 Large
By AI21 Labs | Nov 1, 2022
PRE-TRAINED LANGUAGE MODEL TRAINED BY AI21 LABS ON A CORPUS OF WEB TEXT INCLUDING NATURAL LANGUAGE AND COMPUTER PROGRAMS WITH RECENT DATA - UPDATED TO MID 2022. THIS MODEL HAS A 8192 TOKEN CONTEXT WINDOW (I.E. THE LENGTH OF THE PROMPT + COMPLETION SHOULD BE AT MOST 8192 TOKENS).
Designed for speed, Jurassic-2 Large is the ideal choice for single language tasks that require maximum affordability and minimal latency.

View model




Jurassic-1 Grande (17B)
By AI21 Labs | Nov 2022/1/20
TEXT GENERATION, LONG-FORM GENERATION, SUMMARIZATION, PARAPHRASING, CHAT, INFORMATION EXTRACTION, QUESTION ANSWERING, CLASSIFICATION
A Large Language Model (LLM) that you can use for any language comprehension or generation task.

View model




Cohere Generate Model - Command-Light
By Cohere | Nov 2022/1/20
TEXT GENERATION, GENERATIVE AI, CONTENT GENERATION, AI TEXT WRITER, COPY WRITING, SUMMARIZATION, SUMMARY GENERATOR, ENTITY EXTRACTION
Powered by a large language model use Cohere Generate for tasks like copywriting, named entity recognition, paraphrasing or summarization.

View model




Lyra-Fr 10B
By Lyra | Nov 2022/1/20
TEXT GENERATION, KEYWORD EXTRACTION, INFORMATION EXTRACTION, QUESTION ANSWERING, SUMMARIZATION, SENTIMENT ANALYSIS, CLASSIFICATION
State-of-the-art French text generation AI with a REST API.

View model




Flan UL2
By Hugging Face | Nov 1, 2022
TEXT2TEXT GENERATION
Flan UL2 is an instruction-tuned model trained on top of UL2 20B model and thus capable of performing various zero-shot natural language processing tasks, as well as the few-shot in-context learning tasks. In comparison to previous UL2 20B model, the Flan UL2 uses a receptive field of 2048 which makes it more suitable for few-shot in-context learning. With appropriate prompt, it can perform zero-shot NLP tasks such as text summarization, common sense reasoning, natural language inference, question and answering, sentence/sentiment classification, translation, and pronoun resolution.

View model




BloomZ 7B1
By Hugging Face | Nov 1, 2022
TEXT2TEXT GENERATION
BloomZ 7B1 is an instruction-tuned model based on Bloom 7B1 and thus capable of performing various zero-shot natural language processing tasks, as well as the few-shot in-context learning tasks. With appropriate prompt, it can perform zero-shot NLP tasks such as text summarization, common sense reasoning, natural language inference, question and answering, sentence/sentiment classification, translation, and pronoun resolution.

View model




GPT-J 6B
By Hugging Face | Nov 1, 2022
TEXT GENERATION
GPT-J 6B is a text generation model with 6 billion of parameters released by Eleuther AI. The GPT-J 6B has been trained on a large corpus of text data - the Pile dataset - and thus is capable of performing various few-shot natural language processing tasks such as text generation, text classification, text summarization, question and answering, grammar and spelling correction, paraphrasing, and intent classification.

View model




Flan T5 XL
By Hugging Face | Nov 2022/1/20
TEXT2TEXT GENERATION
FLAN T5 is an instruction-tuned model and thus capable of performing various zero-shot natural language processing tasks, as well as the few-shot in-context learning tasks. With appropriate prompt, it can perform zero-shot NLP tasks such as text summarization, common sense reasoning, natural language inference, question and answering, sentence/sentiment classification, translation, and pronoun resolution.

View model



Stable Diffusion 1.4
By Stability AI | Nov 1, 2022
IMAGE GENERATION (FROM TEXT)
Stable Diffusion v1.4 is a latent text-to-image diffusion model, capable of generating photo-realistic images given any text input.

View model



Stable Diffusion 2.1 Base
By Stability AI | Nov 2, 2022
IMAGE GENERATION (FROM TEXT)
Stable Diffusion v2.1 greatly improves the quality of the generated images compared to earlier v1 releases.

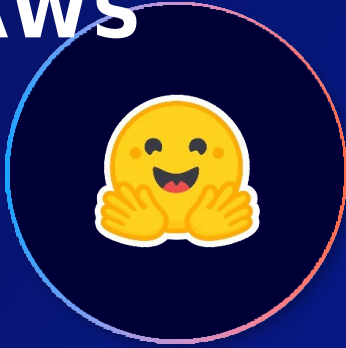
View model

Foundation models available on SageMaker JumpStart for self-managed access

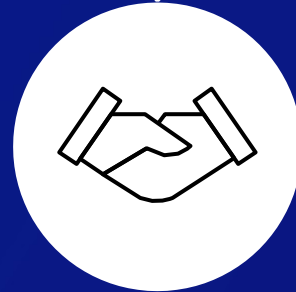
Publicly available			Proprietary models		
stability.ai	alexai	🤖	co:here	Lighten	AI21labs
Models Stable Diffusion Upscaling	Models AlexaTM 20B	Models More than 100 models! Including: FLAN-T5, FLAN-UL2, GPT2, GPTJ, GPT-Neo, BLOOM, Openjourney	Models Cohere generate-med	Models Lyra-10B	Models Jurassic-2 (multiple variants)
Tasks Generate photo-realistic images from text input Improve quality of generated images	Tasks Machine translation Question answering Summarization Annotation Data generation	Tasks Machine translation Question answering Summarization Annotation Data generation	Tasks Text generation Information extraction Question answering Summarization	Tasks Text generation Keyword extraction Information extraction Question answering Summarization Sentiment	Tasks Text generation Long-form generation Summarization Paraphrasing Chat Information extraction Question answering
Features Fine-tuning on SD 2.1 model					

A strong collaboration to make NLP easy and accessible for all

**Hugging Face
AWS**



Hugging Face is the most popular open source company providing state-of-the-art NLP technology



Amazon SageMaker offers high performance resources to train and use NLP models

Development in a Jupyter Notebook or Direct Deployment

```
Terminal 2  model-txt2img-stabilityai-... Home  Demo.ipynb  model-txt2img-stabilityai-...
Cluster  Data Science 2.0  Python 3  2 vCPU + 4 GiB  Share

Introduction to JumpStart - Text to Image

Welcome to Amazon SageMaker JumpStart! You can use JumpStart to solve many Machine Learning tasks through one-click in SageMaker Studio, or through SageMaker JumpStart API. In this demo notebook, we demonstrate how to use the JumpStart API to generate images from text using state-of-the-art Stable Diffusion models. Furthermore, we show how to fine-tune the model to your dataset.

Stable Diffusion is a text-to-image model that enables you to create photorealistic images from just a text prompt. A diffusion model trains by learning to remove noise that was added to a real image. This de-noising process generates a realistic image. These models can also generate images from text alone by conditioning the generation process on the text. For instance, Stable Diffusion is a latent diffusion where the model learns to recognize shapes in a pure noise image and gradually brings these shapes into focus if the shapes match the words in the input text.

Training and deploying large models and running inference on models such as Stable Diffusion is often challenging and include issues such as cuda out of memory, payload size limit exceeded and so on. JumpStart simplifies this process by providing ready-to-use scripts that have been robustly tested. Furthermore, it provides guidance on each step of the process including the recommended instance types, how to select parameters to guide image generation process, prompt engineering etc. Moreover, you can deploy and run inference on any of the 80+ Diffusion models from JumpStart without having to write any piece of your own code.

In the first part of this notebook, you will learn how to use JumpStart to generate highly realistic and artistic images of any subject, object, environment or scene. This may be as simple as an image of a cute dog or as detailed as a hyper-realistic image of a beautifully decorated cozy kitchen by pixer in the style of Greg Rutkowski with dramatic sunset lighting and long shadows with cinematic atmosphere. This can be used to design products and build catalogs for ecommerce business needs or to generate realistic art pieces or stock images.

In the second part of this notebook, you will learn how to use JumpStart to fine-tune the Stable Diffusion model to your dataset. This can be useful when creating art, logos, custom designs, NFTs, and so on, or fun stuff such as generating custom AI images of your pets or avatars of yourself.

Model license: By using this model, you agree to the CreativeML Open RAIL-M++ license.

[43]: from ipywidgets import Dropdown
      from sagemaker.jumpstart.notebook_utils import list_jumpstart_models

      # Retrieves all Text-to-Image generation models.
      filter_value = "task == txt2img"
      txt2img_models = list_jumpstart_models(filter=filter_value)


      # display the model-ids in a dropdown to select a model for inference.
      model_dropdown = Dropdown(
          options=txt2img_models,
          value="model-txt2img-stabilityai-stable-diffusion-v2-1-base",
          description="Select a model",
          style={"description_width": "initial"},
          layout={"width": "max-content"},
      )
      display(model_dropdown)

      Select a model  model-txt2img-stabilityai-stable-diffusion-v2-1-base  v

[44]: # model_version="*" fetches the latest version of the model
      model_id, model_version = model_dropdown.value, "*"

[34]: text = "cottage in impressionist style"
      query_response = query(model_predictor, text)
      img, prmt = parse_response(query_response)
      display_img_and_prompt(img, prmt)

      cottage in impressionist style


```


Development in a Jupyter Notebook or Direct Deployment

Prompt


portrait photo of a african old warrior chief, tribal panther make up, gold on white, side profile, looking away, serious eyes, 50mm portrait photography, hard rim lighting photography-beta -ar 2:3 -beta

Negative prompt (optional)

Use negative prompt to avoid certain objects, colors, styles, attributes, and more

Generate image

Output



Example

Choose an option

General info

Width

512

Height

512

Number of inference steps

50

Guidance scale

7

Seed

1

Stable diffusion

width

Type: 1

Name: width

Label: Width

Default: 512

MinValue: 512

MaxValue: 512

Description: Height of the generated image. If specified, it must be a positive integer divisible by 8.

height

Type: 1

Name: height

Label: Height

Jupyter Notebook

Using Python Jupyter
Notebook to retrieve
weights of a pre-trained
model and prompt/fine-
tune it



Pretrained Model Retrieval

```
model_checkpoint='google/flan-t5-base'
```

```
from transformers import AutoTokenizer  
tokenizer = AutoTokenizer.from_pretrained(model_checkpoint, use_fast=True)
```

```
from transformers import AutoModelForSeq2SeqLM  
model = AutoModelForSeq2SeqLM.from_pretrained(model_checkpoint)
```

```
inputs = tokenizer(few_shot_prompt, return_tensors='pt')  
output = tokenizer.decode(  
    model.generate(  
        inputs["input_ids"],  
        max_new_tokens=50,  
    )[0],  
    skip_special_tokens=True  
)  
print(f'FEW SHOT RESPONSE: {output}')summary = dataset['test'][example_indices[0]]['summary']  
print(f'EXPECTED RESPONSE: {summary}')
```

```
FEW SHOT RESPONSE: Tom is late for the train. He has to catch it at 9:30.
```

Prompting/ Fine-tuning on Jupyter Notebook

```
start_prompt = 'Summarize the following conversation.\n'
end_prompt = '\n\nSummary: '
dialogue = dataset['test'][example_indices[1]]['dialogue']
summary = dataset['test'][example_indices[1]]['summary']
prompt = f'{start_prompt}{dialogue}{end_prompt}'
```

```
inputs = tokenizer(prompt, return_tensors='pt')
output = tokenizer.decode(
    model.generate(
        inputs["input_ids"],
        max_new_tokens=50,
    )[0],
    skip_special_tokens=True
```

```
)
print(f'INPUT PROMPT:\n{prompt}\n')
print(f'MODEL GENERATION:\n{output}\n')
print(f'BASELINE SUMMARY:\n{summary}')
```

INPUT PROMPT:
Summarize the following conversation.
#Person1#: May, do you mind helping me prepare for the picnic?
#Person2#: Sure. Have you checked the weather report?
#Person1#: Yes. It says it will be sunny all day. No sign of rain at all. This is your father' Daniel.
#Person2#: No, thanks Mom. I'd like some toast and chicken wings.
#Person1#: Okay. Please take some fruit salad and crackers for me.
#Person2#: Done. Oh, don't forget to take napkins disposable plates, cups and picnic blanket.
#Person1#: All set. May, can you help me take all these things to the living room?
#Person2#: Yes, madam.
#Person1#: Ask Daniel to give you a hand?
#Person2#: No, mom, I can manage it by myself. His help just causes more trouble.

Summary:

MODEL GENERATION:
The weather report says it will be sunny all day.

```
trainer = Trainer(
    model=model,
    args=training_args,
    train_dataset=sample_tokenized_dataset['train'],
    eval_dataset=sample_tokenized_dataset['validation']
)
```

```
trainer.train()
```

```
/opt/conda/lib/python3.7/site-packages/transformers/optimization.py:395: FutureWarning: This i
ill be removed in a future version. Use the PyTorch implementation torch.optim.AdamW instead,
sable this warning
FutureWarning,
```

[33/33 06:54, Epoch 1/1]

Epoch	Training Loss	Validation Loss
-------	---------------	-----------------

1	No log	35.455990
---	--------	-----------

```
supervised_fine_tuned_model_path = "./flan-dialogue-summary-checkpoint"
# supervised_fine_tuned_model_path = f"./{output_dir}/<put-your-checkpoint-dir-here>"
```

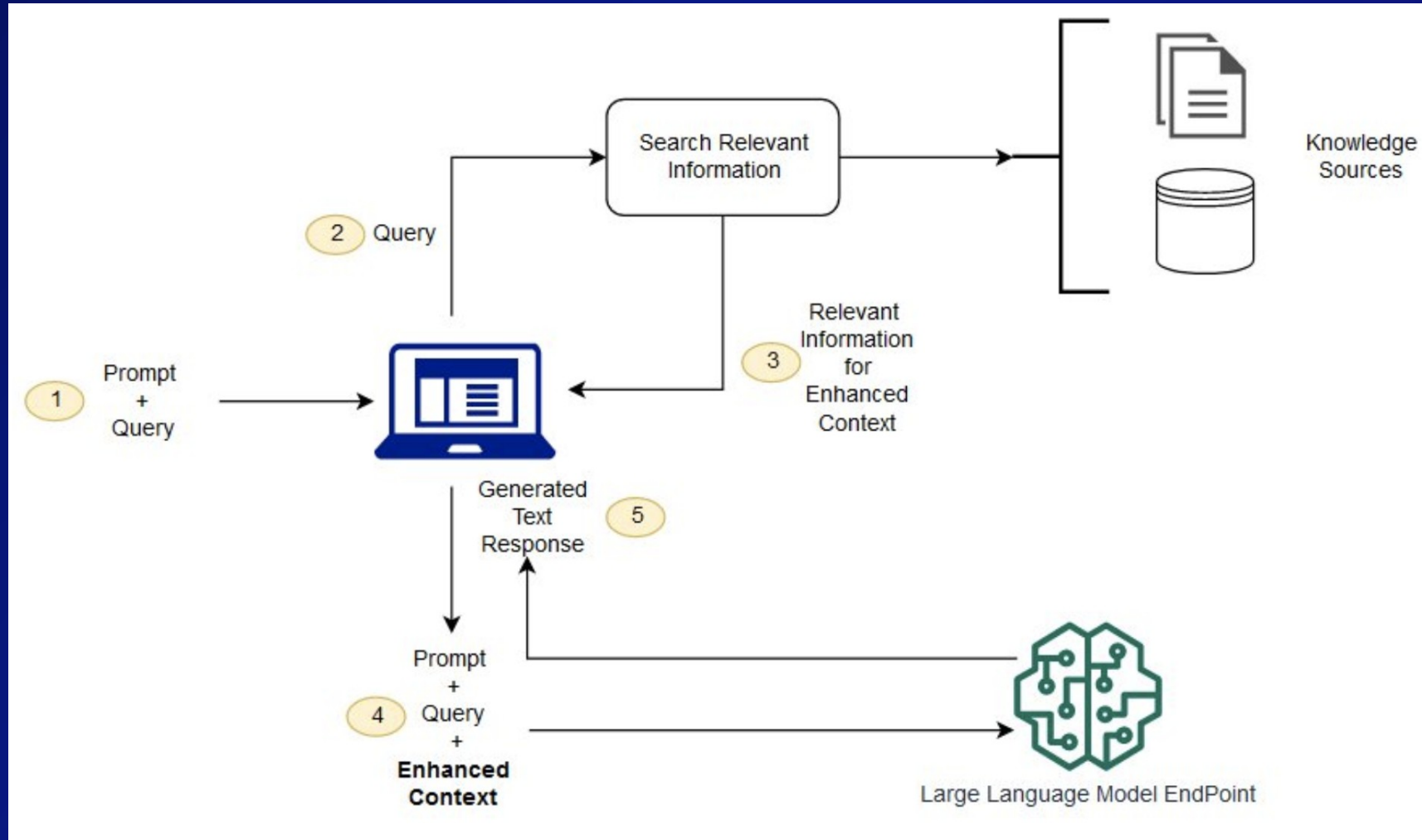
```
tuned_model = AutoModelForSeq2SeqLM.from_pretrained(supervised_fine_tuned_model_path)
model = AutoModelForSeq2SeqLM.from_pretrained(model_checkpoint)
```

```
outputs = tuned_model.to('cpu').generate(
    model_input,
    GenerationConfig(max_new_tokens=200, num_beams=1),
)
text_output = tokenizer.decode(outputs[0], skip_special_tokens=True)
```

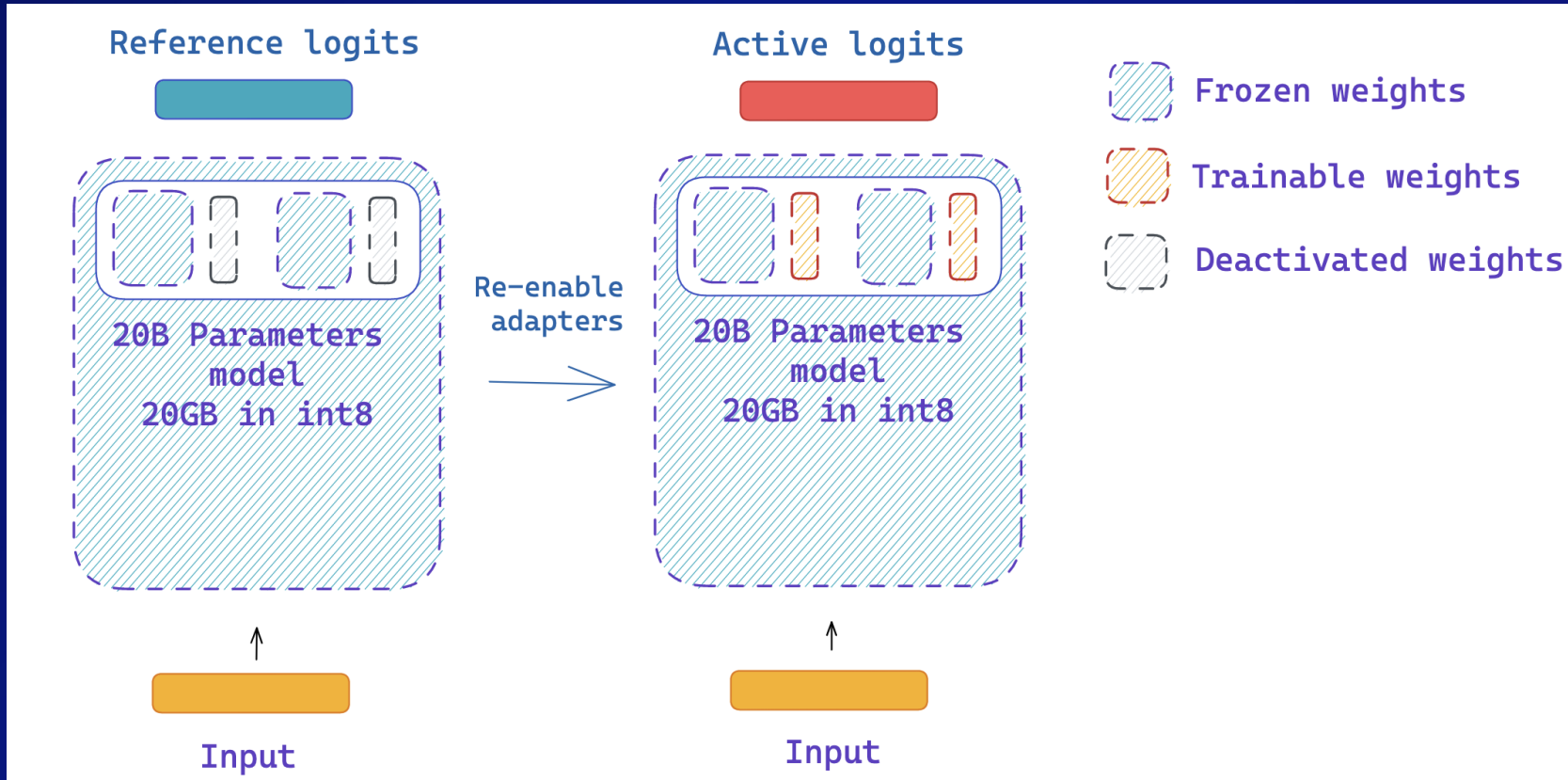
Tips and Tricks for Customized Gen AI

The background of the slide is a solid blue color. In the lower half, there are large, flowing, abstract shapes in shades of purple and orange, creating a modern and dynamic aesthetic.

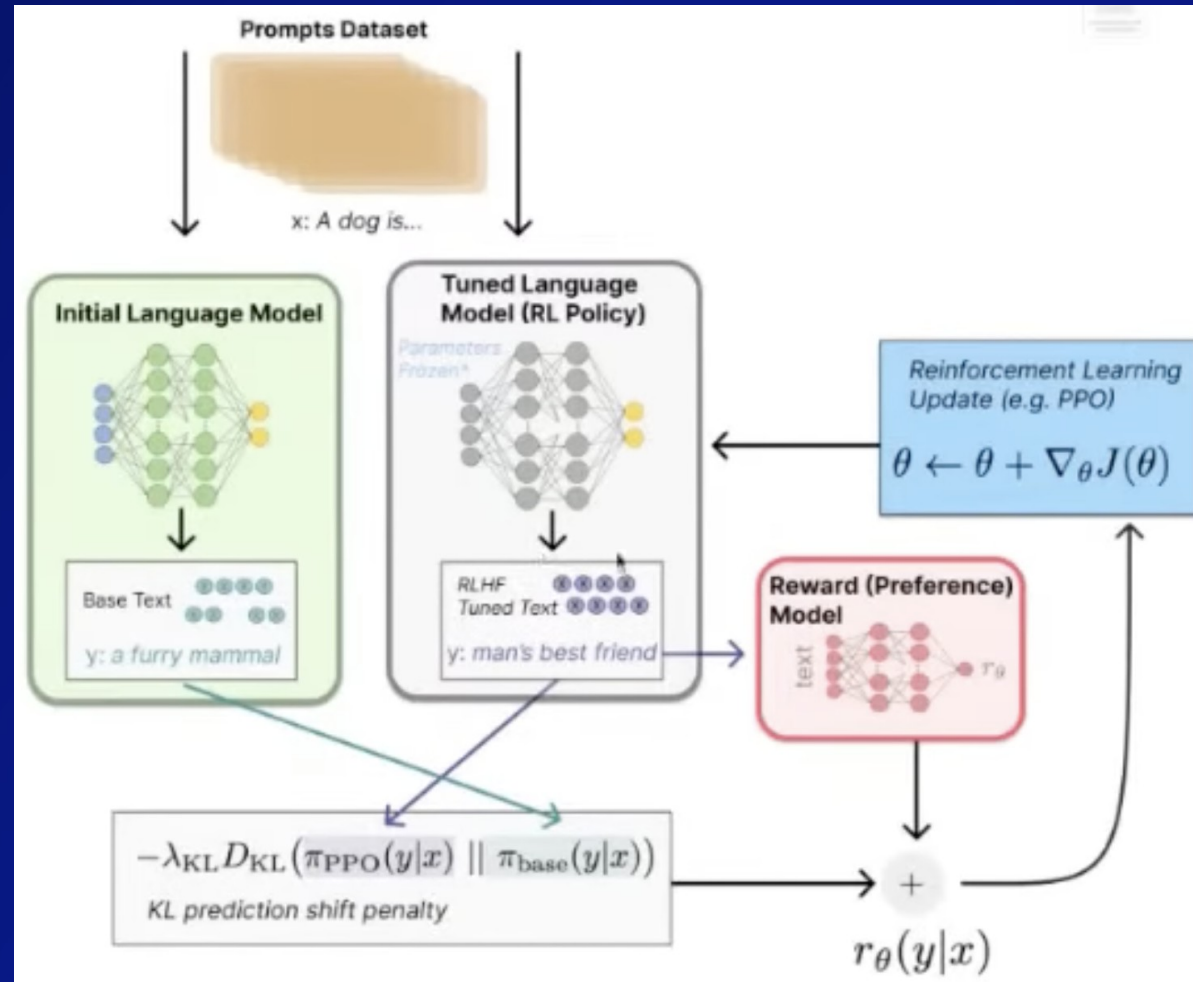
Retrieval Augmented Generation



PEFT - Partial Efficient Fine Tuning



RHLF - Reinforcement Learning with Human Feedback



Thank you!