

# GenAI Fundamentals For Beginners

Unlocking the Power of Generative AI



What is Generative AI?



How does it work?  
(Models & Data)



Applications & Future Trends







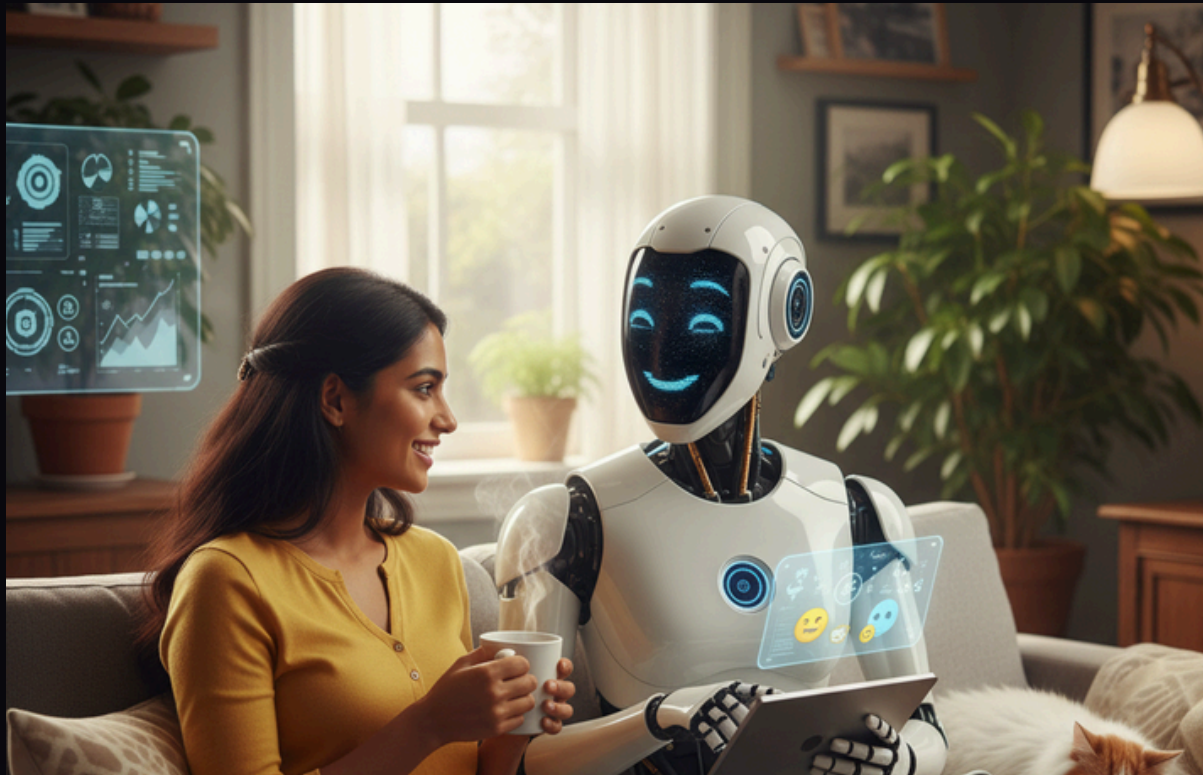
# We Will Be Covering Today:

- What is GenAI?
- Traditional AI (ML) vs GenAI
- What are Foundational Models?
- What is pre-training?
- What is fine-tuning?
- What are LLM?
- Token/Tokenization & Vectors
- Encoding & Decoding
- Basics of Transformers
- No-code AI Platforms/Apps

# The Big Question: 🤔



# What Is GenAI?



Imagine you're talking to a friend. You ask:

- "Can you write me a poem about coffee?"
- "Can you design me a logo idea?"
- "Can you explain quantum physics like I'm 10?"

Generative AI (GenAI) is like that friend — but powered by machines.

# What Is GenAI?

## Definition (simplified):

- GenAI = Artificial Intelligence that can create new things (text, images, code, music, videos) instead of just analyzing old data.
- Think of it as a creative assistant trained on huge amounts of information.

## Analogy:

- Traditional AI is like a calculator (it gives you answers to problems you type in).
- GenAI is like a storyteller (it can create stories, images, songs on its own).

# Traditional-AI VS GenAI



# Traditional-AI VS GenAI

## Chef ML (Traditional AI/ML)

- Needs a specific recipe (training for one narrow task, e.g., predicting house prices).
- Works best on numbers, structured tasks.
- If you ask him to make something outside his recipe book → he fails.

## Chef GenAI

- Has read millions of recipes from around the world.
- Can invent new dishes by combining knowledge creatively.
- If you ask him for a “fusion sushi pizza,” he’ll attempt it!

# Traditional-AI **VS** GenAI

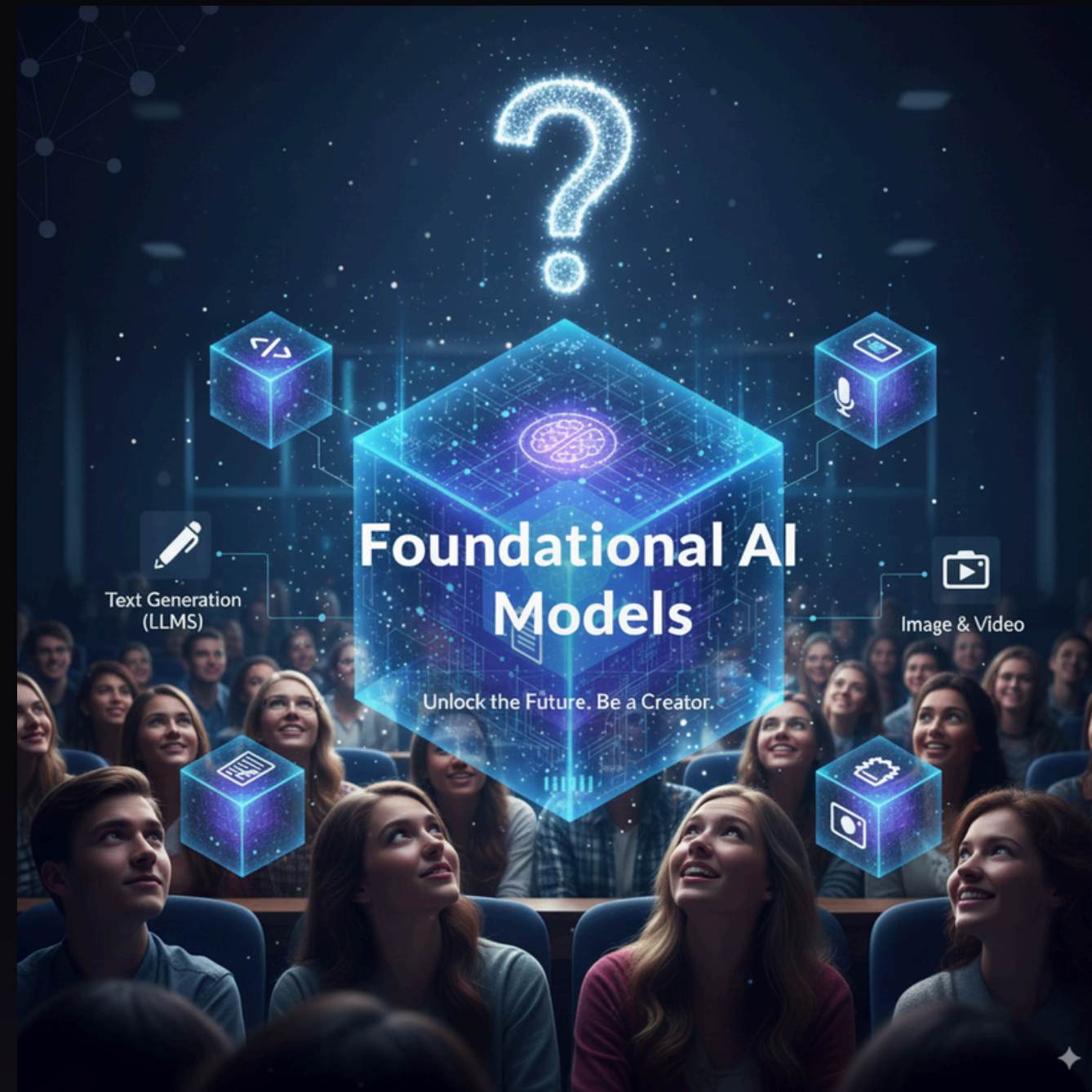
## Traditional AI (ML):

- Learns patterns to predict outcomes.
- Works best on structured data (Excel tables, numbers, labels).
- Example: Predicting credit card fraud.

## Generative AI:

- Learns from massive unstructured data (text, images, audio).
- Can generate new data that looks human-made.
- Example: Writing a movie script, generating realistic images.

# What Are Foundational Models?



# What Are Foundational Models?

Think of LEGO blocks. Instead of building a new set from scratch every time, you get a big box of pre-built LEGO pieces. You can then rearrange them for cars, houses, spaceships.

- Foundational Models = **very large, general-purpose** AI models trained on a **vast range of data**.
- They are called “foundational” because you can use them as a **foundation to build many applications**.

Examples:

- GPT (text generation)
- Stable Diffusion (image generation)
- Whisper (speech-to-text)

# What Is Pre-training?



# What Is Pre-training?

Storytelling approach – Think of learning a language.

- When you were a child, you didn't start with Shakespeare.
- You first absorbed lots of words, grammar, and conversations around you.

That's pre-training for a model:

- Feeding it huge amounts of data (books, articles, code, images).
- It learns patterns of language and meaning but not specific tasks.

Analogy:

- Pre-training = “going to school” (learning general knowledge).
- Later, you specialize in a subject (medicine, law, art) → that's fine-tuning.

# What Is Fine-tuning?



# What Is Fine-tuning?

**Storytelling approach:** Imagine you learned English in school (pre-training).

- Now you want to become a lawyer → you learn legal English.
- Or a doctor → you learn medical terms.

That **specialization = fine-tuning**.

For AI models:

- Pre-trained model knows general language.
- Fine-tuning = training it further on specific domain data (like medical records, legal contracts, customer support chats).

Example:

- ChatGPT (general knowledge).
- A hospital fine-tunes it → becomes a medical chatbot.
- A bank fine-tunes it → becomes a finance assistant.

# What are LLM?



# What are LLM?

**Storytelling approach:** Think of a super librarian who has read almost every book, article, and website in existence. If you ask them a question, they don't just look up an answer — they can write a new book or article for you.

That's an LLM (**Large Language Model**).

- LLM (Large Language Model):
  - A special kind of AI that **understands and generates human-like text**.
  - Trained on **massive amounts of written data** (books, articles, websites).
  - Examples: ChatGPT, Claude, Gemini.

# What are LLM?

Now, what if this librarian doesn't just read books, but can also look at pictures, listen to music, and watch videos?  
That's multimodal AI.

- **Multimodal Models:**
  - Can work with more than one type of data (text, image, audio, video).
  - Example: You upload a photo + ask, "Describe this image" → AI replies with text.

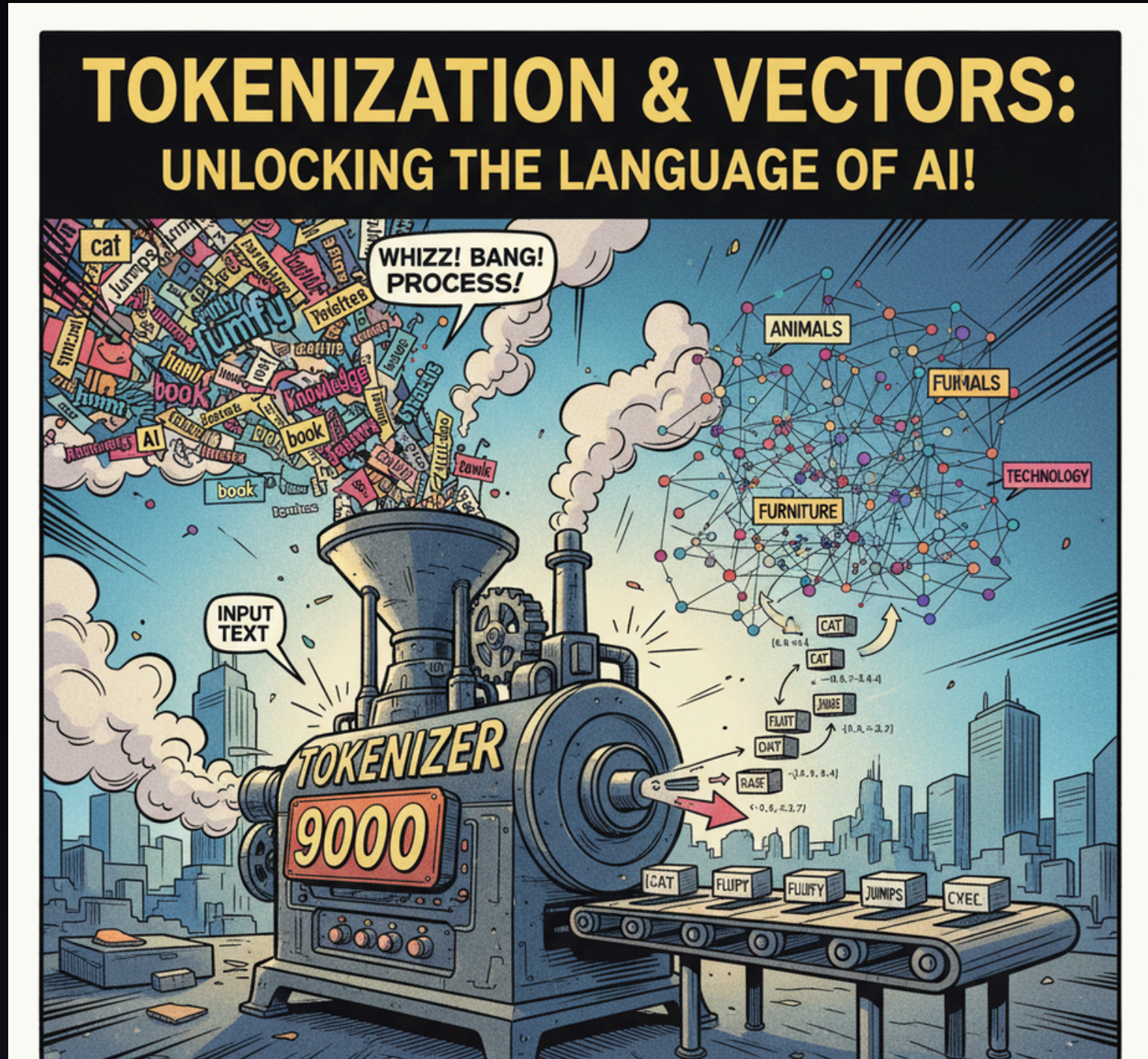
## Analogy:

- LLM = A master of words.
- Multimodal = A master of words, pictures, and sounds together.

# Some Popular LLMs:

- GPT (OpenAI): Great at conversation, coding, creative writing.
- Claude (Anthropic): Known for safe, friendly, detailed responses.
- Gemini (Google): Multimodal, integrates with search and Google tools.
- LLaMA (Meta): Open-source, widely used for research & custom apps.
- Mistral: Lightweight, efficient, often used in open-source setups.

# Tokenization & Vectors



# Tokenization & Vectors

**Storytelling approach:** Imagine you're translating English into a secret code. Instead of writing "I love pizza" you break it into pieces:

- "I" → 101
- "love" → 245
- "pizza" → 872

**Tokens:** Small chunks of text (words, sub-words, or even characters).

**Tokenization:** Breaking text into these small chunks so the computer can process them.

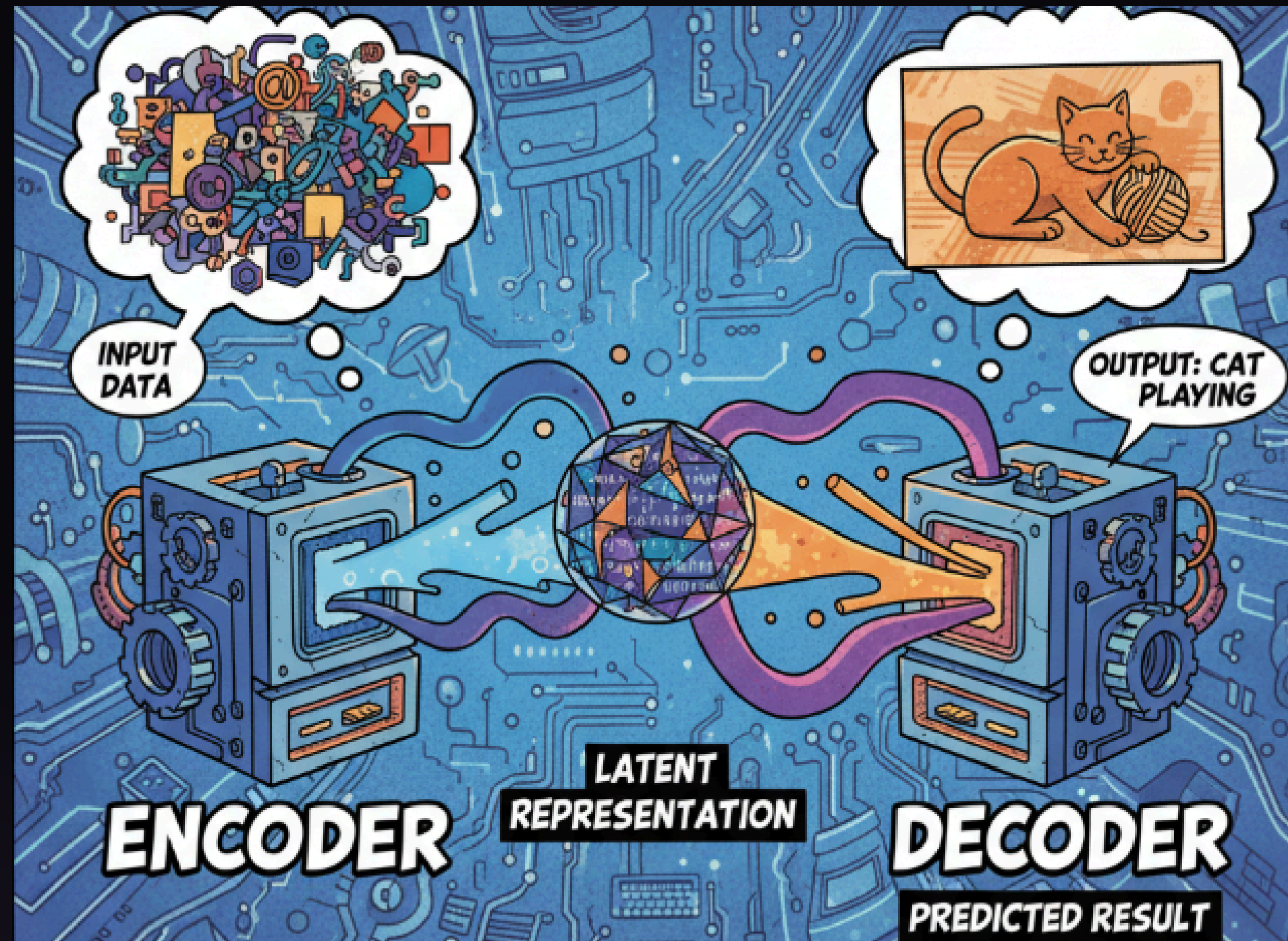
# Tokenization & Vectors

But computers don't understand words, they understand numbers. So each token is converted into a vector — basically, a list of numbers that captures its meaning.

## Analogy:

- **Tokens** = Lego bricks (pieces of a sentence).
- **Vectors** = The “coordinates” of each brick in 3D space.

# Language to Machine – Machine to Language (Encoding & Decoding)



# Language to Machine – Machine to Language (Encoding & Decoding)

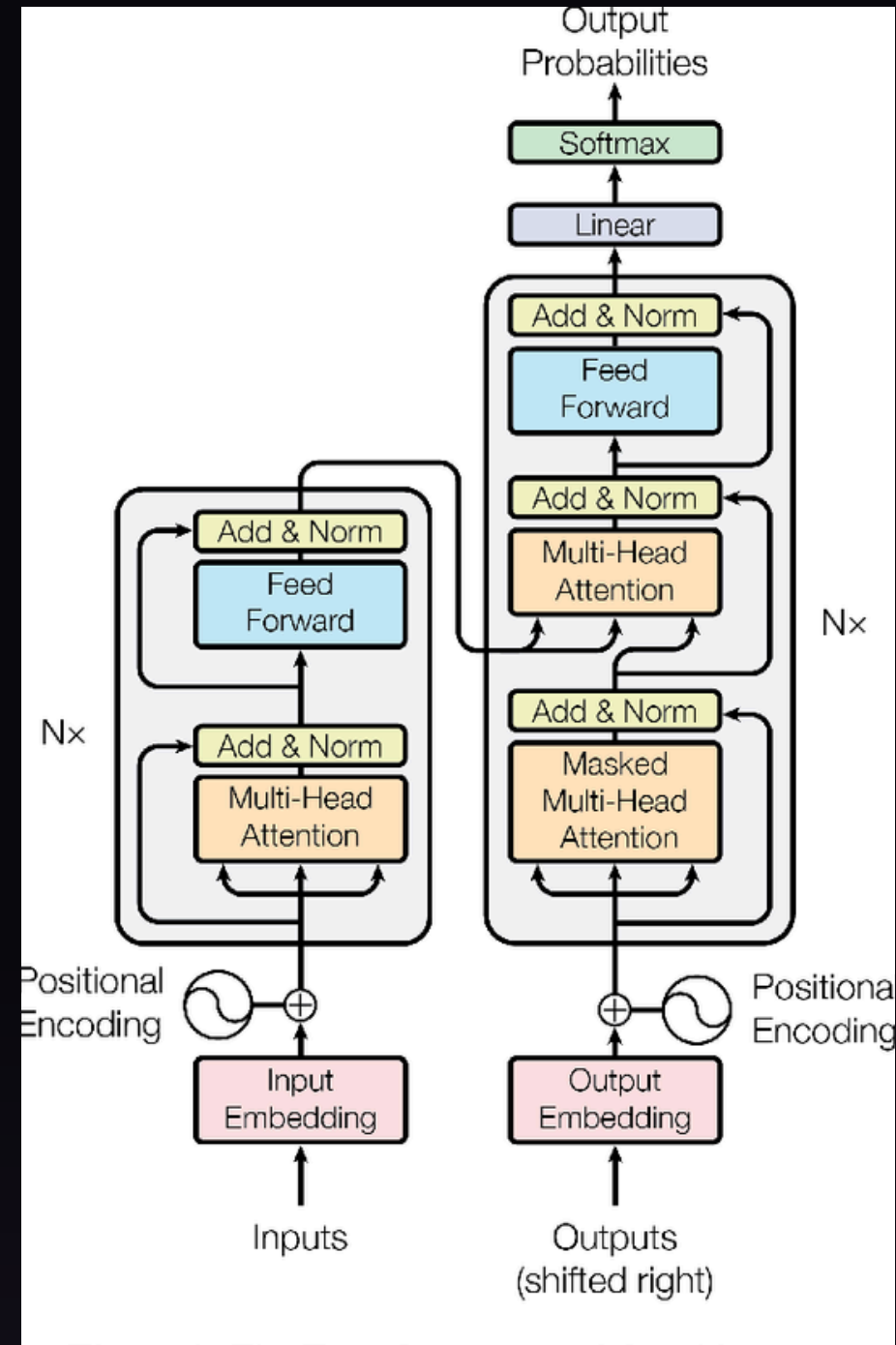
**Storytelling approach:** Imagine you're traveling to Japan but only know English. You need a translator.

- When you speak English, the translator encodes it into Japanese for the locals.
- When locals reply in Japanese, the translator decodes it back to English for you.

**AI works the same way:**

- Encoding: Human language → tokens → vectors (machine language).
- Decoding: Vectors → tokens → human language (text output).

# Basics of Transformers



# Basics of Transformers

**Storytelling approach:** Before transformers, AI struggled with long sentences. Imagine reading a novel but only remembering the last 5 words — that's how older models worked!

Then came Transformers (2017, by Google researchers). They introduced a **game-changing idea: "attention"**.

- Transformers can **look at all words** at once and **decide which ones are important**.
- Example: In the sentence "The cat sat on the mat because it was tired,"
  - "it" refers to cat, not mat → the model figures this out using attention.

# Basics of Transformers

## Key idea:

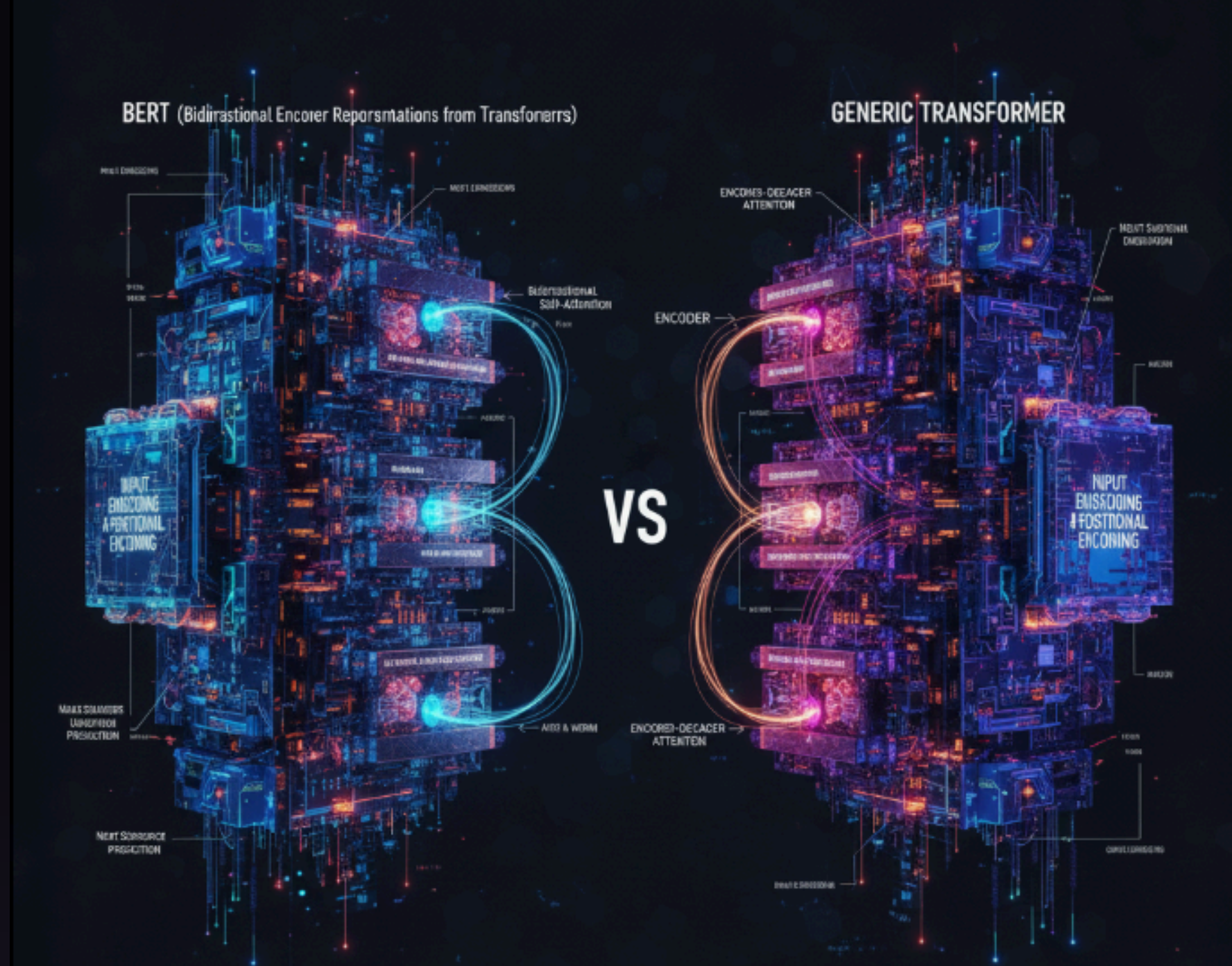
Transformers = The architecture that powers almost all modern GenAI (GPT, BERT, Claude, etc.).

## Analogy:

- Old models = A person with short-term memory.
- Transformers = A person who can highlight and focus on important words in a whole paragraph.

# Transformers VS BERT

## BERT vs TRANSFORMERS ARCHITECTURE



# Transformers VS BERT

Once transformers came out, many new models were built. One famous one is **BERT (Bidirectional Encoder Representations from Transformers)**.

- BERT basics:
  - Reads sentences in both directions (left-to-right & right-to-left).
  - Great for understanding context.
  - Example: “Bank” → could mean river bank or money bank.
    - BERT uses surrounding words to guess the right meaning.

# Transformers VS BERT

- **Transformers** = The general architecture (like the invention of electricity).
- **BERT** = A specific model built using transformers (like the first light bulb).

## Analogy:

- Transformers = A powerful engine design.
- BERT = A specific car built using that engine.

# No-code AI Platforms/Apps – Free vs Paid

Imagine you want to drive a car 🚗. You don't need to know how to build the engine or code the GPS system. You just need a ready-made car with a steering wheel.

That's what No-code AI platforms are for GenAI:

- They give you a steering wheel interface (simple app or website).
- Behind the scenes, the complex engine (LLM, AI model) is running.

# No-code AI Platforms/Apps – Free vs Paid

Imagine you want to drive a car 🚗. You don't need to know how to build the engine or code the GPS system. You just need a ready-made car with a steering wheel.

That's what No-code AI platforms are for GenAI:

- They give you a steering wheel interface (simple app or website).
- Behind the scenes, the complex engine (LLM, AI model) is running.

# No-code AI Platforms/Apps – Free vs Paid

## Free Tools (Good for Starters):

- ChatGPT Free (GPT-3.5) – basic text generation.
- Perplexity AI – research + fact-checking.
- Canva AI – design + image generation.
- Grammarly AI – writing assistant.
- Poe (free tier) – access to multiple models in one app.

**Pros:** Easy access, no payment needed, great for experimentation.

**Cons:** Limited usage, slower responses, fewer advanced features.

# No-code AI Platforms/Apps – Free vs Paid

Paid Tools (For Power Users/Business Needs):

- ChatGPT Plus (GPT-4) – advanced reasoning, creativity, plugins.
- Claude Pro (Anthropic) – larger context window, safer conversations.
- Gemini Advanced (Google) – integrates with Gmail, Docs, YouTube.
- Notion AI / Jasper – specialized in writing, productivity, marketing.
- Runway AI / MidJourney – advanced image & video generation.

**Pros:** Access to the latest models, faster, more accurate, extra features.

**Cons:** Subscription fees, may still have limits.



