Categories of Generative AI Models

Category	Description	Examples
1. Language Models (LLMs)	Generate or understand human language	GPT-4, Claude 3, LLaMA 3, Mistral, Gemini, PaLM
2. Embedding Models	Convert text/images into vector representations (for search, clustering, etc.)	text-embedding-3, bge, e5, MiniLM, LaBSE
3. Code Generation Models	Write or explain code, generate scripts	Codex, StarCoder2, Code LLaMA, DeepSeek-Coder
4. Multimodal Models	Process multiple inputs (e.g., text + image)	GPT-4o, Gemini 1.5, Claude 3 Opus, Kosmos-2
5. Vision Models	Understand and generate images	CLIP, DINOv2, SAM, OpenCV-AI, BLIP-2
6. Text-to-Image Models	Create images from text prompts	DALL·E 3, Stable Diffusion, Midjourney, Kandinsky
7. Image-to-Text Models	Caption images or explain visual content	BLIP, Flamingo, GPT-4o (vision), Gemini
8. Text-to-Video Models	Generate videos from text descriptions	Sora (OpenAI), Runway Gen-2, Pika, AnimateDiff
9. Speech Models	Convert voice to text or vice versa	Whisper (STT), Bark (TTS), VALL-E, Deepgram
10. Audio Generation Models	Create music, sound effects, or voice clones	MusicGen, AudioCraft, Riffusion, ElevenLabs
11. Conversational Agents	Designed for back-and-forth dialogue	ChatGPT, Claude, Pi (Inflection), Google Gemini Chat

Category	**Description**	**Examples**
12. RAG Models / Frameworks	Combine retrieval with generation using LLMs	LangChain, LlamaIndex, Haystack (not models but frameworks), RAG-ready LLMs
13. Agentic Models	Execute multi-step tasks (agents)	AutoGPT, BabyAGI, AgentGPT, OpenAI GPTs (custom)
14. Diffusion Models	Probabilistic models to generate data	Stable Diffusion, Imagen, Glide (text-to-image/video/audio)
15. Fine-Tuning & Adapter Models	Lightweight task-specific tuning	LoRA, QLoRA, PEFT models (not standalone models but techniques)

Choosing the Right Model for Your Use Case

Use Case	Recommended Models	Notes
Text Generation / Chatbots	GPT-4, Claude 3 Opus, Gemini 1.5 Pro, LLaMA 3 70B	GPT-4 is best in general performance; Claude 3 has strong reasoning
Summarization	GPT-4, Gemini, Claude 3, Mistral Medium	Claude is good for summarizing long content
Coding / Code Generation	GPT-4, Claude 3 Opus, StarCoder2, CodeLLaMA, DeepSeek-Coder	GPT-4 and Claude 3 Opus best for coding and reasoning
Search / RAG (Retrieval-Augmented Generation)	GPT-4, Claude, LLaMA 3, Mistral	Use with vector DB like FAISS, Weaviate
Document Q&A / Enterprise Search	Claude 3, GPT-4, Gemini	Claude handles long context best
Multimodal (Text + Image)	GPT-4o, Gemini 1.5 Pro, Claude 3 Opus	GPT-4o is fastest and good at vision
Voice / Speech-to-Text	Whisper (OpenAI), Deepgram, Meta MMS	For real-time transcription
Data Analysis / Tables / Math	GPT-4o, Claude 3 Opus	GPT-4o supports charts; Claude is strong in math
Small Devices / On-Device	Mistral 7B, LLaMA 3 8B, Phi-3 Mini	Run on CPU/GPU with quantization
Multilingual Tasks	GPT-4, Mistral, XGLM, BLOOM	GPT-4 and Gemini support >30 languages well
Privacy/On-Prem	LLaMA 3, Mistral, Falcon, OpenHermes, Zephyr	Open-source options for private deployment

Model Comparison Criteria

When evaluating models, consider:

- Performance (accuracy, reasoning)
- Context Length (tokens the model can "remember")
- Cost (tokens per dollar if using APIs)
- Latency (speed)
- License (commercial use allowed?)
- Hardware Compatibility (can it run locally?)

Example Use Case and Model Match

Use Case: Internal enterprise document Q&A
Best Models:

- **Claude 3 Opus** – long context, handles documents well
- **GPT-4** – accurate and safe
- **LLaMA 3 70B** – strong open-source alternative (via RAG)

Key Open Embedding Models:

Model Name	**Vector Size**	**Open Source**	**Performance**
text-embedding-3-small (OpenAI)	1536	NO	Very High (state-of-the-art)
all-MiniLM-L6-v2 (Sentence Transformers)	384	YES	Fast, good for small tasks
bge-base-en-v1.5 (BAAI)	768	YES	Very strong for English
e5-base-v2 (Intel Labs)	768	YES	Good multilingual support
InstructorXL	768	YES	Can follow instructions during embedding

How to Choose the Right Embedding Model

Parameter	Why It Matters	Tips
Vector Quality	Better embeddings → more accurate similarity	Use newer models (text-embedding-3, bge, e5)
Language Support	Needed for multilingual data	Choose models like e5-multilingual, LaBSE
Vector Size	Larger = better accuracy, but slower search	Use 768 or 1536 unless on edge devices
Open Source vs API	Cost and privacy	Use open-source (e.g., BGE) if deploying locally
Inference Speed	Needed for real-time results	MiniLM and bge-small are fast on CPU
RAG Compatibility	Can you use with retrieval + generation?	Yes – embeddings are fed into LLMs like GPT-4 later