Guide encyclopédique des modèles d'IA exécutables localement

Panorama exhaustif des modèles open-source et auto-hébergeables

Thierry Gayet

Édition 2026 — Mise à jour mai 2026

Contents

Chapter 1

Avant-propos

Auteur : Thierry Gayet
linuxmaine.org
https://linuxmaine.org

Ce document constitue une tentative encyclopédique de recenser, décrire et comparer les modèles d'intelligence artificielle dits *locaux* — c'est-à-dire conçus pour être téléchargés, exécutés et utilisés directement sur la machine de l'utilisateur, sans dépendance permanente à un service cloud propriétaire.

L'écosystème de l'IA locale a connu une explosion sans précédent depuis la libération en 2023 du modèle LLaMA de Meta, suivie en 2024-2025 par une vague d'ouverture massive des grands laboratoires (Mistral, Alibaba, DeepSeek, Google, Microsoft, IBM, Cohere, OpenAI). En 2026, il est désormais possible d'exécuter, sur un simple ordinateur portable récent, des modèles dont les capacités rivalisent avec les services propriétaires d'il y a deux ans.

Ce guide couvre :

- les **grands modèles de langage (LLM)** généralistes ;
- les **modèles spécialisés en code** ;
- les **modèles multimodaux** (vision-langage) ;
- les **modèles de génération d'images** ;
- les **modèles de génération vidéo** ;
- les **modèles audio** (TTS, STT, musique) ;
- les **modèles d'embedding** et de **reranking** ;
- les **modèles spécialisés** (segmentation, détection, OCR, scientifique) ;
- les **frameworks et outils** d'exécution locale.

Pour chaque modèle, l'auteur a tâché de fournir : nom, éditeur, pays d'origine, date de sortie, architecture, paramètres, tailles disponibles, contexte, licence, site officiel, dépôt source, cas d'usage et notes comparatives.

Les chiffres présentés (tailles en MB/GB, nombre de paramètres, scores) reflètent l'état des connaissances de l'auteur au moment de la rédaction. L'écosystème évoluant à une cadence soutenue, le lecteur est invité à consulter les liens fournis pour vérifier la dernière version disponible.

Bonne lecture.

Chapter 2

Concepts fondamentaux et grille de lecture

2.1 Qu'est-ce qu'un modèle local ?

Un modèle d'IA *local* est un modèle dont les **poids (weights)** — c'est-à-dire les paramètres numériques appris pendant l'entraînement — sont :

1. **Téléchargeables** par l'utilisateur, le plus souvent depuis Hugging Face, GitHub ou un site officiel ;
2. **Exécutables hors-ligne** sur la machine de l'utilisateur, avec un moteur d'inférence approprié ;
3. **Distribués sous une licence permettant cet usage** (open-source, *open-weights*, ou recherche).

Ne pas confondre :

- **Open-source** : code source ET poids ET données d'entraînement ouverts (rare ; ex. : OLMo, Pythia).
- **Open-weights** : poids librement diffusés, mais code/données partiellement fermés (LLaMA, Mistral, Qwen, DeepSeek…).
- **Source-available** : code visible mais usage commercial restreint.

2.2 Anatomie d'un modèle de langage

Élément	Description
Paramètres (B)	Nombre de poids du réseau, exprimé en milliards (Billions). 7B = 7 milliards.
Architecture	Transformer, Mixture-of-Experts (MoE), Mamba, hybride…
Contexte (tokens)	Quantité de texte que le modèle peut considérer simultanément.
Tokens d'entraînement	Volume total de données vues durant le pré-entraînement (en T = trillions).
Quantization	Réduction de la précision des poids (FP16 → INT8 → Q4 → Q2) pour gagner en taille.
Format	GGUF, GPTQ, AWQ, EXL2, safetensors, etc.

2.3 Tailles indicatives selon la quantization

Pour un modèle de 7 milliards de paramètres :

Précision	Bits / param.	Taille approx.	Qualité	Usage typique
FP32	32	~28 GB	Référence	Entraînement
FP16 / BF16	16	~14 GB	Quasi-parfaite	GPU haut de gamme
Q8_0	8	~7,2 GB	Excellente	GPU 12 GB+

Précision	Bits / param.	Taille approx.	Qualité	Usage typique
Q6_K	6	~5,5 GB	Très bonne	GPU 8 GB
Q5_K_M	5	~4,8 GB	Bonne	GPU 6 GB
Q4_K_M	4	~4,1 GB	Acceptable	CPU, GPU 4-6 GB
Q3_K_M	3	~3,3 GB	Dégradée	CPU léger
Q2_K	2	~2,7 GB	Très dégradée	Test uniquement

Règle empirique : **paramètres (×) bits / 8 = taille en GB** (approximation).

2.4 Formats de fichiers

- **GGUF** (GPT-Generated Unified Format) — format binaire de llama.cpp, supporte la quantization. *De facto* standard pour l'inférence CPU/Apple Silicon.
- **safetensors** — format sécurisé de Hugging Face, succède à `.bin` (pickle).
- **GPTQ** — quantization GPU 4-bit.
- **AWQ** — Activation-aware Weight Quantization.
- **EXL2** — format mixed-precision d'ExLlamaV2.
- **MLX** — format optimisé pour Apple Silicon.
- **ONNX** — format universel Microsoft.

2.5 Métriques de comparaison usuelles

Benchmark	Mesure
MMLU	Connaissances générales (57 disciplines).
GPQA	Raisonnement scientifique de niveau doctorat.
HumanEval	Génération de code Python.
MBPP	Problèmes de programmation basique.
GSM8K	Mathématiques niveau collège.
MATH	Mathématiques compétition.
HellaSwag	Sens commun.
ARC	Raisonnement scientifique.
TruthfulQA	Fiabilité factuelle.
IFEval	Suivi d'instructions.
MT-Bench	Conversation multi-tours.
AlpacaEval	Évaluation conversationnelle automatique.
LMSys Arena	Classement par préférence humaine (Elo).

Chapter 3

Partie I — Grands modèles de langage (LLM)

3.1 Famille Meta LLaMA

3.1.1 LLaMA 1

- **Éditeur** : Meta AI (Facebook AI Research)
- **Pays** : États-Unis
- **Sortie** : 24 février 2023
- **Description** : premier modèle « ouvert » d'envergure, fuité puis officiellement diffusé pour la recherche. A déclenché la vague open-weights.
- **Architecture** : Transformer décodeur, RMSNorm, SwiGLU, RoPE.
- **Tailles** : 7B (~13 GB FP16), 13B (~26 GB), 33B (~65 GB), 65B (~130 GB).
- **Contexte** : 2 048 tokens.
- **Tokens d'entraînement** : 1,0 à 1,4 T.
- **Licence** : recherche non-commerciale.
- **Cas d'usage** : recherche, fine-tuning de Vicuna, Alpaca, Guanaco.
- **Site / Code** : `https://github.com/meta-llama/llama`
- **Note** : aujourd'hui obsolète, mais base historique de l'écosystème.

3.1.2 LLaMA 2

- **Sortie** : 18 juillet 2023
- **Tailles** : 7B, 13B, 70B (versions Base et Chat).
- **Contexte** : 4 096 tokens.
- **Tokens d'entraînement** : 2 T.
- **Licence** : Llama 2 Community License (commerciale autorisée, sauf très grands acteurs >700 M MAU).
- **Innovations** : RLHF, GQA (Grouped-Query Attention) sur le 70B, sécurité renforcée.
- **Téléchargement** : `https://huggingface.co/meta-llama/Llama-2-7b-hf` (et variantes).
- **Cas d'usage** : socle de centaines de fine-tunings (CodeLlama, Llama-Guard, Vicuna v1.5…).

3.1.3 LLaMA 3

- **Sortie** : 18 avril 2024
- **Tailles** : 8B (~16 GB), 70B (~140 GB).
- **Contexte** : 8 192 tokens (étendu à 128 K en 3.1).
- **Tokens d'entraînement** : 15 T (7,5 (×) Llama 2).
- **Tokenizer** : 128 K vocabulaire (vs 32 K en Llama 2), bien plus efficace pour le code et le multilingue.
- **Performances** : MMLU 8B ~66 ; 70B ~82.
- **Licence** : Llama 3 Community License.
- **Cas d'usage** : assistant généraliste, chat, RAG, base de fine-tuning.

3.1.4 LLaMA 3.1

- **Sortie** : 23 juillet 2024
- **Tailles** : 8B, 70B, **405B** (premier *frontier model* open-weights).
- **Contexte** : **128 K tokens**.
- **Spécificités** : multilingue (8 langues), tool-use natif, distillation du 405B vers 8B/70B.
- **Téléchargement** : `https://huggingface.co/meta-llama/Meta-Llama-3.1-405B-Instruct`.
- **Note** : le 405B nécessite ~810 GB FP16, ~230 GB en Q4 ; usage typique multi-GPU H100.

3.1.5 LLaMA 3.2

- **Sortie** : 25 septembre 2024
- **Tailles** : **1B**, **3B** (texte), **11B**, **90B** (vision-langage).
- **Spécificités** : versions edge/mobile (1B-3B), premières variantes multimodales officielles Meta.
- **Cas d'usage** : on-device (smartphone), assistants embarqués, vision.

3.1.6 LLaMA 3.3

- **Sortie** : 6 décembre 2024
- **Taille** : 70B uniquement.
- **Particularité** : performances équivalentes au 405B avec 6 (×) moins de paramètres ; améliorations math/code/instruction-following.

3.1.7 LLaMA 4

- **Sortie** : avril 2025
- **Variantes** : **Scout** (17B actifs / 109B total, MoE 16 experts, contexte 10 M), **Maverick** (17B actifs / 400B total, MoE 128 experts), **Behemoth** (annoncé, ~2 T total).
- **Architecture** : Mixture-of-Experts native, multimodalité (texte+image+vidéo) intégrée, *early fusion*.
- **Contexte** : jusqu'à 10 millions de tokens (Scout).
- **Téléchargement** : `https://huggingface.co/meta-llama/Llama-4-Scout-17B-16E-Instruct`.

3.2 Famille Mistral AI

3.2.1 Mistral 7B

- **Éditeur** : Mistral AI (Paris)
- **Pays** : France
- **Sortie** : 27 septembre 2023
- **Tailles** : 7,3B (~14 GB FP16, ~4,1 GB Q4).
- **Contexte** : 8 192 tokens (sliding window de 4 096).
- **Architecture** : Transformer + GQA + Sliding Window Attention (SWA).
- **Licence** : Apache 2.0 (totalement libre, commercial inclus).
- **Performances** : surpasse Llama 2 13B sur la majorité des benchmarks.
- **Site** : `https://mistral.ai`
- **Téléchargement** : `https://huggingface.co/mistralai/Mistral-7B-v0.1`
- **Cas d'usage** : référence open-weights, base de centaines de fine-tunings (Zephyr, OpenHermes, Dolphin).

3.2.2 Mixtral 8x7B

- **Sortie** : 11 décembre 2023
- **Architecture** : **Sparse Mixture-of-Experts** : 8 experts de 7B, 2 activés par token.
- **Paramètres** : 46,7B total / 12,9B actifs.
- **Taille** : ~87 GB FP16, ~26 GB Q4.
- **Contexte** : 32 K tokens.
- **Licence** : Apache 2.0.
- **Performances** : équivalent ou supérieur à GPT-3.5 et Llama 2 70B en utilisant moins de calcul à l'inférence.

3.2.3 Mixtral 8x22B

- **Sortie** : avril 2024
- **Paramètres** : 141B total / 39B actifs.
- **Contexte** : 64 K.
- **Taille** : ~262 GB FP16.
- **Licence** : Apache 2.0.

3.2.4 Mistral Nemo

- **Sortie** : juillet 2024 (en partenariat avec NVIDIA)
- **Taille** : 12B
- **Contexte** : 128 K
- **Tokenizer** : Tekken (efficace multilingue, 30% mieux que Mistral 7B sur du code).
- **Licence** : Apache 2.0.

3.2.5 Mistral Small / Medium / Large

- Versions commerciales (poids non publiés, sauf Small 3 et Small 3.1 publiés en 2025).
- **Mistral Small 3** (24B, jan 2025) — Apache 2.0, ~150 tokens/s sur RTX 4090.
- **Mistral Small 3.1** (mars 2025) — multimodal, 128 K contexte.

3.2.6 Codestral

- **Sortie** : mai 2024
- **Taille** : 22B
- **Spécialité** : code, 80 + langages.
- **Licence** : Mistral AI Non-Production License (MNPL).

3.2.7 Codestral Mamba

- **Architecture** : Mamba2 (state-space model, complexité linéaire).
- **Contexte** : théorique illimité, pratique 256 K+.
- **Licence** : Apache 2.0.

3.2.8 Mathstral

- **Sortie** : juillet 2024
- **Taille** : 7B
- **Spécialité** : mathématiques (basé sur Mistral 7B, fine-tuné STEM).

3.2.9 Pixtral

- **Sortie** : septembre 2024
- **Tailles** : 12B, 124B (Pixtral Large).
- **Modalité** : texte + image.

3.3 Famille Qwen (Alibaba)

3.3.1 Qwen 1 / 1.5

- **Éditeur** : Alibaba Cloud (équipe Tongyi Qianwen)
- **Pays** : Chine
- **Sortie** : août 2023 (Qwen 1), février 2024 (Qwen 1.5)
- **Tailles** : 0,5B / 1,8B / 4B / 7B / 14B / 32B / 72B / 110B.
- **Licence** : Tongyi Qianwen License (permissive, restrictions >100 M MAU).

3.3.2 Qwen2

- **Sortie** : juin 2024
- **Tailles** : 0,5B / 1,5B / 7B / 57B-A14B (MoE) / 72B.
- **Contexte** : jusqu'à 128 K (sur 7B et 72B).
- **Multilingue** : 27 + langues, fort sur chinois et anglais.

3.3.3 Qwen2.5

- **Sortie** : septembre 2024
- **Tailles** : 0,5B / 1,5B / 3B / 7B / 14B / 32B / 72B.
- **Contexte** : 128 K (génération 8 K).
- **Variantes spécialisées** : **Qwen2.5-Coder** (code, jusqu'à 32B), **Qwen2.5-Math** (mathématiques).
- **Téléchargement** : `https://huggingface.co/Qwen/Qwen2.5-72B-Instruct`.
- **Performances** : Qwen2.5-72B se hisse au niveau de GPT-4o sur de nombreux benchmarks.

3.3.4 Qwen3

- **Sortie** : avril 2025
- **Tailles** : dense de 0,6B à 32B ; MoE 30B-A3B et 235B-A22B.
- **Spécificité** : *thinking mode* commutable (raisonnement long-form vs réponse rapide).
- **Contexte** : 128 K à 1 M.
- **Licence** : Apache 2.0 (rupture vs licence Tongyi antérieure).

3.3.5 QwQ-32B (Qwen with Questions)

- **Sortie** : novembre 2024
- **Spécialité** : modèle de raisonnement (concurrent de o1-preview).
- **Contexte** : 32 K.

3.3.6 Qwen-VL / Qwen2-VL / Qwen2.5-VL

- Modèles multimodaux vision-langage.
- **Qwen2-VL** (août 2024) : 2B, 7B, 72B ; performances exceptionnelles sur OCR et compréhension de vidéo.

3.4 Famille DeepSeek

3.4.1 DeepSeek LLM

- **Éditeur** : DeepSeek AI (Hangzhou Huan Fang Quantitative Investment)
- **Pays** : Chine
- **Sortie** : novembre 2023
- **Tailles** : 7B, 67B.
- **Licence** : DeepSeek License (similaire LLaMA 2, commercial autorisé).

3.4.2 DeepSeek-Coder

- **Sortie** : novembre 2023
- **Tailles** : 1,3B / 6,7B / 33B.
- **Spécialité** : code (87 langages, infill, repository-level).
- **Performance** : meilleur modèle open-source de code en 2023.

3.4.3 DeepSeek-V2

- **Sortie** : mai 2024
- **Architecture** : MoE 236B / 21B actifs, **Multi-Head Latent Attention (MLA)** réduisant la consommation KV-cache.
- **Contexte** : 128 K.

3.4.4 DeepSeek-V3

- **Sortie** : 26 décembre 2024
- **Architecture** : MoE 671B / 37B actifs.
- **Tokens entraînement** : 14,8 T.
- **Coût d'entraînement** : 5,576 M USD (sensation médiatique).
- **Performances** : niveau GPT-4o et Claude 3.5 Sonnet.
- **Licence** : DeepSeek License + commercial.

3.4.5 DeepSeek-R1

- **Sortie** : 20 janvier 2025
- **Spécialité** : modèle de raisonnement à la o1, *open-weights*.
- **Tailles** : R1 (671B MoE), R1-Zero (sans SFT), distillations en Llama-8B/70B et Qwen-1.5B/7B/14B/32B.
- **Licence** : MIT.
- **Téléchargement** : `https://huggingface.co/deepseek-ai/DeepSeek-R1`.
- **Impact** : choc industriel — performances o1 à coût très inférieur, déclenche le débat sur la souveraineté IA.

3.4.6 DeepSeek-V3.1, V3.2, R2

- Itérations 2025 améliorant raisonnement multimodal et efficacité.

3.5 Famille Google Gemma

3.5.1 Gemma 1

- **Éditeur** : Google DeepMind
- **Pays** : États-Unis / Royaume-Uni
- **Sortie** : 21 février 2024
- **Tailles** : 2B, 7B (Base et Instruct).
- **Architecture** : dérivée de Gemini, Transformer décodeur.
- **Licence** : Gemma Terms of Use (commercial autorisé).
- **Contexte** : 8 192.

3.5.2 Gemma 2

- **Sortie** : juin 2024
- **Tailles** : 2B, 9B, 27B.
- **Innovations** : *logit soft-capping*, alternance d'attention locale/globale, distillation depuis modèle plus grand.
- **Contexte** : 8 192.

3.5.3 Gemma 3

- **Sortie** : mars 2025
- **Tailles** : 1B, 4B, 12B, 27B.
- **Multimodal** : vision intégrée (sauf 1B), 140 + langues.
- **Contexte** : 128 K (sauf 1B : 32 K).

3.5.4 CodeGemma

- **Sortie** : avril 2024
- **Tailles** : 2B (Base), 7B (Base/Instruct).
- **Spécialité** : code, infilling.

3.5.5 PaliGemma / PaliGemma 2

- Modèles vision-langage compacts (3B), performants sur OCR, tâches spécialisées après fine-tuning.

3.5.6 RecurrentGemma

- **Architecture** : Griffin (mélange attention + récurrence linéaire).
- **Tailles** : 2B, 9B.

3.5.7 ShieldGemma

- Modèles de modération de contenu (2B/9B/27B).

3.5.8 DataGemma

- Versions ancrées sur les statistiques publiques (Data Commons).

3.6 Famille Microsoft Phi (« Small Language Models »)

3.6.1 Phi-1 / Phi-1.5

- **Sortie** : juin 2023 / septembre 2023
- **Tailles** : 1,3B
- **Approche** : entraîné sur des données « textbook quality », démontrant que la qualité des données prime sur la quantité.

3.6.2 Phi-2

- **Sortie** : décembre 2023
- **Taille** : 2,7B
- **Performances** : équivalent à Llama 2 7B sur de nombreux benchmarks.
- **Licence** : MIT.

3.6.3 Phi-3

- **Sortie** : avril 2024
- **Variantes** : Mini (3,8B), Small (7B), Medium (14B), Vision (4,2B).
- **Contexte** : 4 K et 128 K (variantes).
- **Cas d'usage** : on-device, mobile.

3.6.4 Phi-3.5

- **Sortie** : août 2024
- **Variantes** : Mini Instruct (3,8B), MoE (16x3.8B / 6,6B actifs), Vision Instruct (4,2B).

3.6.5 Phi-4

- **Sortie** : décembre 2024
- **Taille** : 14B (dense)
- **Performances** : surpasse Llama 3.3 70B sur GPQA et MATH.
- **Variantes** : Phi-4-mini (3,8B), Phi-4-multimodal (5,6B, audio + vision + texte).

3.6.6 Phi-4 Reasoning / Reasoning-Plus

- **Sortie** : avril/mai 2025
- **Spécialité** : raisonnement chain-of-thought entraîné par RL.

3.7 Famille Anthropic — *non-locale*

Anthropic (Claude) **ne publie pas** de poids ouverts. Les modèles Claude (Haiku, Sonnet, Opus) ne sont accessibles que par API. Ce guide n'en traite donc que par contraste.

3.8 Famille OpenAI

3.8.1 GPT-OSS

- **Éditeur** : OpenAI
- **Sortie** : 5 août 2025
- **Tailles** : **gpt-oss-20b** (MoE, 21B total / 3,6B actifs), **gpt-oss-120b** (117B total / 5,1B actifs).
- **Architecture** : Mixture-of-Experts.
- **Licence** : **Apache 2.0** (rupture historique pour OpenAI).
- **Contexte** : 128 K.
- **Particularité** : premier modèle aux poids ouverts d'OpenAI depuis GPT-2.
- **Téléchargement** : `https://huggingface.co/openai/gpt-oss-20b`.

3.8.2 GPT-2

- **Sortie** : 2019
- **Tailles** : 124M, 355M, 774M, 1,5B.
- **Licence** : MIT.
- **Intérêt actuel** : pédagogique, embarqué très contraint.

3.9 Famille Cohere

3.9.1 Command R

- **Éditeur** : Cohere
- **Pays** : Canada
- **Sortie** : mars 2024
- **Taille** : 35B
- **Spécialité** : RAG, tool-use, multilingue (10 langues).
- **Contexte** : 128 K.
- **Licence** : CC-BY-NC 4.0 (recherche non-commerciale).

3.9.2 Command R+

- **Sortie** : avril 2024
- **Taille** : 104B
- **Performances** : top des modèles open-weights sur RAG et function-calling à sa sortie.

3.9.3 Command R7B / R+ 08-2024 / Command A

- Itérations 2024-2025 (compactes ou améliorées).

3.9.4 Aya / Aya-23 / Aya Expanse

- **Spécialité** : multilingue à très grande échelle (101 langues sur Aya, 23 sur Aya-23, 23 sur Aya Expanse).
- **Tailles** : 8B, 32B, 35B, 101B.

3.10 Famille IBM Granite

- **Éditeur** : IBM Research
- **Pays** : États-Unis
- **Licence** : Apache 2.0
- **Cible** : entreprise, gouvernance, traçabilité des données.

3.10.1 Granite 3.0 (octobre 2024)

- Tailles : 1B / 3B (MoE), 2B / 8B (dense), variantes Code et Time-Series.

3.10.2 Granite 3.1 / 3.2 / 3.3

- Améliorations contexte (128 K), Vision (Granite Vision 3.1 / 3.2), embeddings.

3.10.3 Granite-Code

- Tailles : 3B / 8B / 20B / 34B
- Langages : 116
- Cible : génération, explication, traduction de code.

3.11 Famille Falcon (TII Abu Dhabi)

- **Éditeur** : Technology Innovation Institute
- **Pays** : Émirats Arabes Unis

3.11.1 Falcon 7B / 40B (mai/juin 2023)

- Licence : Apache 2.0 (initialement TII, puis Apache).
- Premier grand modèle ouvert non-américain.

3.11.2 Falcon 180B (septembre 2023)

- 180 milliards de paramètres ; entraîné sur 3,5 T tokens.

3.11.3 Falcon Mamba 7B (août 2024)

- Premier grand modèle Mamba (state-space) compétitif.

3.11.4 Falcon 3 (décembre 2024)

- Tailles : 1B / 3B / 7B / 10B (Mamba 7B inclus), Apache 2.0.

3.12 Famille 01.AI / Yi

- **Éditeur** : 01.AI (Kai-Fu Lee)
- **Pays** : Chine

3.12.1 Yi-6B / 34B (novembre 2023)

- Licence : Yi License (commercial après inscription).

3.12.2 Yi-1.5 (mai 2024)

- Tailles : 6B, 9B, 34B.

3.12.3 Yi-Coder (septembre 2024)

- Tailles : 1,5B, 9B.
- Excellent rapport perf/taille en code.

3.13 Famille Apple

3.13.1 OpenELM

- **Sortie** : avril 2024
- **Tailles** : 270M / 450M / 1,1B / 3B.
- **Particularité** : modèle entièrement open (poids + code + recettes).

3.13.2 Apple Intelligence Foundation (AFM)

- Modèles on-device et serveur d'Apple Intelligence (~3B on-device).
- Poids non publiés ; documenté par Apple.

3.14 Famille NVIDIA

3.14.1 Nemotron-4

- **Tailles** : 15B (340B annoncé en 2024).
- **Licence** : NVIDIA Open License.

3.14.2 Llama-3.1-Nemotron-70B-Instruct

- Fine-tuning de Llama 3.1 par NVIDIA, top du leaderboard Arena à sa sortie (oct 2024).

3.14.3 Mistral-NeMo-Minitron-8B-Base

- Compression de Mistral Nemo 12B → 8B par *pruning* + distillation.

3.14.4 Cosmos

- Famille de world-models pour la robotique (poids ouverts en partie, 2025).

3.15 Modèles communautaires et fine-tunes notables

3.15.1 Vicuna (mars 2023)

- LMSYS, fine-tuning Llama sur conversations ShareGPT.
- Tailles : 7B, 13B, 33B.

3.15.2 Alpaca (mars 2023)

- Stanford, fine-tuning Llama sur instructions GPT-3.5.

3.15.3 Guanaco / QLoRA (mai 2023)

- Démonstration de la quantization 4-bit + LoRA.

3.15.4 WizardLM / WizardCoder / WizardMath

- Microsoft, fine-tuning par évolution d'instructions (Evol-Instruct).

3.15.5 Nous Research — Hermes

- **Hermes 2 Mistral 7B**, **Hermes 2 Pro Llama-3 8B**, **Hermes 3 Llama-3.1 70B/405B**.
- Spécialité : tool-use, agentique, JSON structuré.

3.15.6 OpenHermes (Teknium)

- Fine-tuning Mistral très populaire.

3.15.7 Dolphin (Eric Hartford / Cognitive Computations)

- *Uncensored* fine-tunings (Mistral, Mixtral, Llama, Qwen) ; mention « 8x7B Dolphin », « Dolphin 2.9 Llama 3 8B ».

3.15.8 Zephyr (HuggingFace H4)

- **Zephyr 7B Beta** (Mistral 7B + DPO) — preuve de concept du DPO grand-public.

3.15.9 OpenChat

- **OpenChat 3.5** : Mistral 7B fine-tuné, top open-source en novembre 2023.

3.15.10 Solar (Upstage)

- **Solar 10.7B** : *depth up-scaling* à partir de Llama 2.

3.15.11 Snowflake Arctic

- MoE 480B / 17B actifs, Apache 2.0 (avril 2024).

3.15.12 Reka

- Reka Flash / Core / Edge (modèles multimodaux).

3.15.13 Inflection / Pi

- Pas de poids ouverts.

3.15.14 xAI / Grok

- **Grok-1** : 314B MoE, Apache 2.0 (mars 2024). Poids bruts publiés.
- **Grok-2** : poids partiellement publiés en 2025.

3.16 Modèles compacts et embarqués

3.16.1 TinyLlama 1.1B

- Réplique de Llama 2 sur 1,1B paramètres, entraîné sur 3 T tokens.
- Apache 2.0.

3.16.2 SmolLM / SmolLM2 (HuggingFace)

- Tailles : 135M, 360M, 1,7B.
- Excellents pour edge.

3.16.3 MobileLLM (Meta)

- 125M, 350M, 600M, 1B, 1,5B — optimisé smartphone.

3.16.4 MiniCPM (OpenBMB)

- MiniCPM 2B/3B/4B et MiniCPM-V (vision compacte).

3.16.5 Stable LM Zephyr 3B / Stable LM 2

- Stability AI, 1,6B et 12B.

3.16.6 RWKV (Bo Peng)

- Architecture RNN-Transformer hybride.
- Tailles : RWKV-4 (jusqu'à 14B), RWKV-5 « Eagle », RWKV-6 « Finch », RWKV-7 « Goose ».
- Apache 2.0, contexte théorique illimité.

3.16.7 Mamba / Mamba-2 (Tri Dao, Albert Gu)

- State-space models.
- Mamba-2.8B, Mamba-2-2.7B.

3.16.8 Jamba (AI21 Labs)

- Hybride Transformer + Mamba + MoE.
- Jamba 1.5 Mini (52B/12B actifs) et Large (398B/94B actifs).

3.17 Modèles européens et autres

3.17.1 BLOOM (BigScience, 2022)

- 176B, multilingue (46 langues), licence RAIL.

3.17.2 OPT (Meta, 2022)

- 125M à 175B.

3.17.3 GPT-NeoX-20B (EleutherAI, 2022)

- 20B, Apache 2.0.

3.17.4 Pythia (EleutherAI)

- Suite 70M → 12B, totalement reproductible.

3.17.5 OLMo / OLMo 2 (Allen AI)

- Tailles : 1B / 7B / 13B / 32B (OLMo 2).
- **Totalement open** : poids + données + code + checkpoints intermédiaires.

3.17.6 MAP-Neo

- 7B, totalement open, équipe sino-américaine.

3.17.7 Cerebras-GPT

- 111M → 13B, Apache 2.0.

3.17.8 Croissant (LightOn / Inria)

- Modèles français.

3.17.9 CroissantLLM / Lucie / Mistral / TheBloke fine-tunes

- Écosystème francophone.

3.17.10 EuroLLM

- 1,7B, 9B — projet européen multilingue (24 langues UE).

3.17.11 Salamandra (Barcelona Supercomputing Center)

- 2B, 7B, 40B — multilingue (35 langues européennes).

3.17.12 Teuken (OpenGPT-X, Allemagne)

- 7B, 24 langues UE.

3.17.13 Mistral-7B et dérivés français

- Voir section Mistral.

Chapter 4

Partie II — Modèles de génération de code

4.1 CodeLlama (Meta)

- **Sortie** : août 2023
- **Base** : Llama 2
- **Tailles** : 7B / 13B / 34B / 70B
- **Variantes** : Base, Instruct, Python.
- **Contexte** : 16 K (entraîné), 100 K (extrapolé).
- **Licence** : Llama 2.

4.2 DeepSeek-Coder / Coder-V2

- Voir section DeepSeek.
- **DeepSeek-Coder-V2** : MoE 236B / 21B actifs ou Lite 16B / 2,4B.
- 338 langages supportés.

4.3 StarCoder / StarCoder2 (BigCode)

- **Éditeur** : BigCode (Hugging Face + ServiceNow)
- **StarCoder** (mai 2023) : 15,5B, 80+ langages.
- **StarCoder2** (février 2024) : 3B / 7B / 15B, jusqu'à 600+ langages, contexte 16 K.
- **Licence** : BigCode OpenRAIL-M.

4.4 CodeGemma

- Voir section Google.

4.5 Codestral / Codestral Mamba

- Voir section Mistral.

4.6 Qwen2.5-Coder

- Tailles : 0,5B / 1,5B / 3B / 7B / 14B / 32B.
- Top open-source en code à fin 2024.

4.7 WizardCoder

- Évolution d'instructions sur StarCoder/Llama.

4.8 Replit Code

- Replit-code-v1-3b, optimisé éditeurs.

4.9 SantaCoder

- Précurseur de StarCoder.

4.10 OpenCoder (Infly AI, novembre 2024)

- 1,5B, 8B, totalement open (poids + données).

4.11 Yi-Coder

- Voir section 01.AI.

4.12 Granite-Code

- Voir section IBM.

4.13 Aider polyglot leaders

- Modèles évalués spécifiquement pour assistance code agentique.

Chapter 5

Partie III — Modèles multimodaux (vision-langage)

5.1 LLaVA (Large Language and Vision Assistant)

- **Éditeur** : University of Wisconsin-Madison + Microsoft
- **Sortie** : avril 2023 (LLaVA), 2024 (LLaVA-1.5, LLaVA-NeXT/1.6).
- **Architecture** : encodeur visuel CLIP (ou SigLIP) + projection + LLM (Llama / Mistral / Vicuna).
- **Tailles** : 7B / 13B / 34B (LLaVA-NeXT 34B) / 72B / 110B (NeXT-Stronger).
- **Licence** : Apache 2.0 (poids), code parfois MIT.
- **Site** : `https://llava-vl.github.io`

5.2 BakLLaVA (Skunkworks AI)

- Mistral 7B + LLaVA.

5.3 MiniGPT-4 / MiniGPT-v2

- Université King Abdullah (KAUST).

5.4 CogVLM / CogAgent (Tsinghua / Zhipu)

- 17B, fortes capacités OCR et UI.

5.5 InternVL (Shanghai AI Lab + SenseTime)

- InternVL 1.5 / 2.5 / 3.
- Tailles : 1B → 78B.
- État de l'art open-source en multimodal en 2024-2025.

5.6 Qwen-VL / Qwen2-VL / Qwen2.5-VL

- Voir Qwen.

5.7 Idefics / Idefics2 / Idefics3 (HuggingFace)

- 8B / 80B / 9B / 80B.
- Multimodal entrelacé (images dans une conversation).

5.8 Florence-2 (Microsoft)

- 230M / 770M.
- Vision multitâche (caption, détection, segmentation, OCR).
- Licence MIT.

5.9 Phi-3-Vision / Phi-3.5-Vision / Phi-4-multimodal

- Voir section Microsoft.

5.10 PaliGemma / PaliGemma 2

- Voir section Google.

5.11 Moondream

- 1,8B, ultra-compact, conçu pour edge.
- Licence Apache 2.0.
- Site : `https://moondream.ai`

5.12 NVILA / VILA

- NVIDIA, vision-langage performant.

5.13 Pixtral

- Voir Mistral.

5.14 Llama 3.2 Vision (11B / 90B)

- Voir Meta.

5.15 Molmo (Allen AI)

- Tailles : 1B (MoE), 7B, 72B.
- Données entièrement open ; performances proches de Claude 3 Sonnet.

5.16 Aria (Rhymes AI)

- MoE multimodal natif (3,9B actifs / 25,3B total).

5.17 Eagle 2 (NVIDIA)

- Vision-langage multimodal performant.

5.18 Ovis / Ovis2 (Alibaba)

- Approche d'alignement des embeddings visuels.

Chapter 6

Partie IV — Modèles de génération d'images

6.1 Stable Diffusion (Stability AI)

6.1.1 Stable Diffusion 1.4 / 1.5

- **Éditeur** : Stability AI + CompVis (LMU Munich) + Runway
- **Pays** : Royaume-Uni / États-Unis / Allemagne
- **Sortie** : août/octobre 2022
- **Architecture** : Latent Diffusion Model (LDM) — VAE + UNet + CLIP text-encoder.
- **Résolution native** : 512 (×) 512.
- **Taille** : ~4 GB FP32, ~2 GB FP16.
- **Licence** : CreativeML Open RAIL-M.
- **Téléchargement** : `https://huggingface.co/runwayml/stable-diffusion-v1-5`.
- **Cas d'usage** : socle historique de l'écosystème (LoRA, ControlNet, fine-tunings : Realistic Vision, DreamShaper…).

6.1.2 Stable Diffusion 2.0 / 2.1

- **Sortie** : novembre 2022 / décembre 2022
- **Résolution** : 768 (×) 768.
- **Encodeur** : OpenCLIP.

6.1.3 SDXL 1.0

- **Sortie** : juillet 2023
- **Architecture** : 2 UNets (base + refiner), 3,5B + 6,6B paramètres.
- **Résolution** : 1024 (×) 1024.
- **Taille** : ~13 GB total.

6.1.4 SDXL Turbo

- **Sortie** : novembre 2023
- **Spécificité** : 1-4 steps grâce à *Adversarial Diffusion Distillation* (ADD).

6.1.5 SDXL Lightning (ByteDance)

- 1, 2, 4, 8 steps.

6.1.6 Stable Diffusion 3 / 3.5

- **Sortie** : juin 2024 (3 Medium), octobre 2024 (3.5 Large/Medium/Turbo).
- **Architecture** : Multimodal Diffusion Transformer (MMDiT).
- **Tailles** : 2B (Medium), 8B (Large).
- **Licence** : Stability Community License (gratuit <1 M USD revenu).

6.2 FLUX (Black Forest Labs)

- **Éditeur** : Black Forest Labs (ex-Stability AI Allemagne)
- **Pays** : Allemagne
- **Sortie** : août 2024
- **Architecture** : MMDiT hybride + flow matching.
- **Variantes** :
  - **FLUX.1 [pro]** : API uniquement.
  - **FLUX.1 [dev]** : 12B, non-commercial, ~24 GB.
  - **FLUX.1 [schnell]** : 12B, Apache 2.0, 1-4 steps, ~24 GB.
- **Performances** : top open-source 2024-2025, suit le prompt et le texte de manière exceptionnelle.
- **Site** : `https://blackforestlabs.ai`.
- **Téléchargement** : `https://huggingface.co/black-forest-labs/FLUX.1-dev`.

6.3 Kandinsky (Sber AI / AI Forever)

- **Pays** : Russie
- **Versions** : Kandinsky 2.0 → 3.1.
- Architecture spécifique (priors latents).

6.4 DeepFloyd IF (Stability AI / DeepFloyd)

- **Sortie** : avril 2023
- Pixel-space cascadé, fort sur le texte dans l'image.

6.5 Würstchen / Stable Cascade

- Architecture compressive très efficace.

6.6 PixArt-Alpha / PixArt-Sigma (Huawei + universités)

- DiT compacts, efficaces.

6.7 Playground v2 / v2.5 / v3 (Playground AI)

- Affinés esthétique.

6.8 DALL-E mini / Craiyon

- Modèles modestes, historiques.

6.9 HiDream

- Modèles propriétaires partiellement publiés en 2025.

6.10 Lumina-Next / Lumina-T2X

- DiT, performants.

6.11 OmniGen (Vector Institute / 2024)

- Génération unifiée (texte+image vers image).

6.12 Cogview / CogView2 / CogView3 / CogView4 (Tsinghua / Zhipu)

- Pré-CogVideoX, Chinois fort.

6.13 Mitsua

- Modèles entraînés exclusivement sur des données opt-in (éthique).

Chapter 7

Partie V — Modèles de génération vidéo

7.1 Stable Video Diffusion (Stability AI)

- **Sortie** : novembre 2023
- 14 / 25 frames, 576 (×) 1024.

7.2 AnimateDiff

- Module ajoutable à SD 1.5 / SDXL.

7.3 CogVideoX (Tsinghua / Zhipu, août 2024)

- Tailles : 2B, 5B, **CogVideoX-1.5** 5B (octobre 2024).
- 6-10 secondes 720p/1080p.
- Apache 2.0.
- Site : `https://github.com/THUDM/CogVideo`.

7.4 HunyuanVideo (Tencent, décembre 2024)

- 13B paramètres, top open-source à sa sortie.
- Licence : Tencent Hunyuan Community License.

7.5 Mochi 1 (Genmo, octobre 2024)

- 10B, Apache 2.0, 480p.

7.6 LTX-Video (Lightricks, novembre 2024)

- 2B, génération temps-réel sur GPU consommateur.
- Apache 2.0.

7.7 Open-Sora / Open-Sora-Plan (PKU-YuanGroup, HPC-AI Tech)

- Réplication open de Sora.

7.8 Wan / Wan 2.1 (Alibaba, 2025)

- 14B, multi-tâches (text-to-video, image-to-video).

7.9 Allegro (RhymesAI)

- 3B, 6 secondes 720p.

7.10 SkyReels (2025)

- Vidéo réaliste centrée humain.

7.11 VideoCrafter / DynamiCrafter

Chapter 8

Partie VI — Audio, parole et musique

8.1 Reconnaissance vocale (Speech-to-Text)

8.1.1 Whisper / Whisper Large v3 / Turbo (OpenAI)

- **Sortie** : septembre 2022 (v1), 2023 (v2/v3), octobre 2024 (Turbo).
- **Tailles** : tiny (39M), base (74M), small (244M), medium (769M), large (1,55B), large-v3, large-v3-turbo (809M).
- **Langues** : 99.
- **Licence** : MIT.
- **Téléchargement** : `https://huggingface.co/openai/whisper-large-v3`.

8.1.2 Distil-Whisper

- 6 (×) plus rapide, ~50 % plus petit.

8.1.3 Faster-Whisper (CTranslate2)

- Réimplémentation 4 (×) plus rapide en C++.

8.1.4 Whisper.cpp (Georgi Gerganov)

- Portage C/C++ pour CPU et embarqué.

8.1.5 Wav2Vec 2.0 / HuBERT / WavLM (Meta / Microsoft)

- Encodeurs audio, base de nombreux pipelines.

8.1.6 MMS (Meta)

- 1 100 + langues reconnues / 4 000 + en TTS.

8.1.7 NeMo Canary (NVIDIA)

- Multitâche STT/Traduction.

8.1.8 Moonshine

- STT compact pour edge (octobre 2024).

8.1.9 Parakeet (NVIDIA)

- 0,6B / 1,1B, FastConformer.

8.1.10 SeamlessM4T / SeamlessM4T v2 (Meta)

- Traduction parole-parole 100 (×) 100 langues.

8.2 Synthèse vocale (Text-to-Speech)

8.2.1 Bark (Suno, avril 2023)

- Génération expressive (rires, intonations).
- MIT.

8.2.2 XTTS / XTTS-v2 (Coqui)

- Voice cloning multilingue (16 langues).
- CPML (recherche / commercial restreint).

8.2.3 Coqui TTS

- Suite open de TTS, modèles Glow-TTS, VITS, Tacotron, FastSpeech.

8.2.4 Piper (Rhasspy)

- TTS local rapide et léger, multi-voix, multi-langues.
- MIT.
- `https://github.com/rhasspy/piper`

8.2.5 StyleTTS 2

- TTS expressif diffusion-based.

8.2.6 Parler-TTS (HuggingFace)

- TTS contrôlable par description.

8.2.7 F5-TTS (octobre 2024)

- *Flow matching* TTS, voice cloning de qualité.

8.2.8 Fish-Speech / Fish-Audio

- Voice cloning multilingue.

8.2.9 MeloTTS (MyShell)

- Multilingue temps réel.

8.2.10 OpenVoice / OpenVoice-V2 (MyShell)

- Voice cloning + style.

8.2.11 MetaVoice-1B

- 1,2B, voice cloning.

8.2.12 Kokoro (octobre 2024)

- 82M paramètres, qualité excellente, Apache 2.0.

8.2.13 CosyVoice / CosyVoice 2 (Alibaba)

- Voice cloning multilingue.

8.2.14 Chatterbox (Resemble AI, 2025)

- Modèle TTS open source de référence.

8.2.15 MaskGCT

- TTS non-autorégressif performant.

8.2.16 IndexTTS (Bilibili, 2025)

- TTS expressif chinois/anglais.

8.2.17 Spark-TTS

- TTS basé LLM, contrôle de style.

8.2.18 Dia / Sesame CSM-1B / Orpheus-TTS

- Modèles TTS conversationnels 2025.

8.3 Génération musicale

8.3.1 MusicGen (Meta)

- **Sortie** : juin 2023
- Tailles : 300M / 1,5B / 3,3B.
- Licence : MIT (code), CC-BY-NC (poids).

8.3.2 AudioGen (Meta)

- Sons et bruitages.

8.3.3 MusicLM

- Pas de poids ouverts.

8.3.4 Stable Audio Open (Stability AI, juin 2024)

- 1,1B, 47 s, génération audio depuis texte.
- Licence Stability Open.

8.3.5 Riffusion

- SD fine-tuné sur spectrogrammes.

8.3.6 YuE (M-A-P, janvier 2025)

- Génération chanson complète (paroles + voix + musique), 7B.

8.3.7 DiffRhythm

- Musique avec voix chantée.

8.3.8 ACE-Step

- 3,5B, Apache 2.0, musique avec paroles.

8.3.9 Suno Bark

- Voir TTS.

8.4 Séparation et traitement audio

8.4.1 Demucs (Meta)

- Séparation de sources musicales (voix, batterie, basse, autres).
- v4 (Hybrid Transformer Demucs).

8.4.2 Spleeter (Deezer)

- Séparation sources, antérieur à Demucs.

8.4.3 NoiseTorch / RNNoise

- Suppression de bruit en temps réel.

8.4.4 Whisperer / Vad-silero / pyannote-audio

- VAD, diarisation.

Chapter 9

Partie VII — Modèles d'embedding et de reranking

9.1 Embeddings textuels

9.1.1 sentence-transformers / all-MiniLM-L6-v2

- **Éditeur** : UKP Lab + HuggingFace
- 22M paramètres, dimension 384.
- Référence légère universelle.

9.1.2 BGE (Beijing Academy of AI / FlagEmbedding)

- **bge-small / base / large** (en, zh, multilingual).
- **bge-m3** (multi-fonction, multi-langue, multi-granularité).
- **bge-en-icl** (in-context learning).

9.1.3 E5 (Microsoft)

- e5-small / base / large / mistral-7B-instruct.
- multilingual-e5-large.

9.1.4 GTE (Alibaba)

- gte-small / base / large / Qwen2-7B-instruct.

9.1.5 Nomic Embed (Nomic AI)

- nomic-embed-text-v1, v1.5, v2.
- Apache 2.0, totalement reproductible.

9.1.6 Jina Embeddings v2 / v3 / v4

- 8 K contexte, multilingue, multi-tâches.

9.1.7 mxbai-embed-large-v1 (mixedbread.ai)

- 335M, performant et compact.

9.1.8 Snowflake Arctic Embed

- Famille S/M/L/XL.

9.1.9 Stella

- stella_en_1.5B, stella_en_400M.

9.1.10 Voyage AI (modèles partiellement ouverts)

9.1.11 NV-Embed (NVIDIA)

- État de l'art MTEB en 2024.

9.1.12 Linq-Embed-Mistral

9.1.13 Qwen3-Embedding (2025)

9.2 Embeddings multimodaux

9.2.1 CLIP (OpenAI, janvier 2021)

- ViT-B/32, ViT-B/16, ViT-L/14, ViT-L/14@336, ViT-H, ViT-G.
- MIT.

9.2.2 OpenCLIP / LAION

- Réimplémentations entraînées sur LAION.
- ViT-bigG-14, EVA-CLIP.

9.2.3 SigLIP / SigLIP 2 (Google)

- Sigmoid loss au lieu de softmax, plus efficace.

9.2.4 DINO / DINOv2 / DINOv3 (Meta)

- Self-supervised vision (pas de texte), excellents features universels.

9.2.5 MetaCLIP (Meta)

- Réplication transparente de CLIP.

9.2.6 Jina-CLIP

- Texte + image, dimension partagée.

9.3 Reranking

9.3.1 bge-reranker-base / large / v2-m3 / v2-gemma

9.3.2 Jina-reranker-v1 / v2

9.3.3 Cohere Rerank (API + modèle Aya à venir)

9.3.4 MS MARCO MiniLM cross-encoders

Chapter 10

Partie VIII — Modèles spécialisés

10.1 Détection / Segmentation / Classification

10.1.1 YOLO — *You Only Look Once*

- **YOLOv5** (Ultralytics, 2020) — référence pratique.
- **YOLOv8** (2023) — segmentation et classification incluses.
- **YOLOv9** (2024) — Programmable Gradient Information.
- **YOLOv10** (2024) — sans NMS.
- **YOLOv11** / **YOLO12** (2024-2025).
- Licences : AGPL-3 / commerciale.

10.1.2 Grounding DINO

- Détection en *open-vocabulary* à partir de texte.

10.1.3 SAM / SAM 2 (Meta)

- **Segment Anything Model** (avril 2023).
- **SAM 2** (juillet 2024) : images + vidéos.
- Apache 2.0.

10.1.4 Grounded-SAM, Florence-2, OWLv2

10.1.5 MM-Detection (OpenMMLab)

- Suite complète de détecteurs.

10.1.6 Detectron2 (Meta)

- Cadre de détection / segmentation.

10.2 OCR

10.2.1 Tesseract (Google, historique)

- Open-source, lourd mais éprouvé.

10.2.2 EasyOCR (Jaided AI)

10.2.3 PaddleOCR (Baidu)

- Multilingue, performant.

10.2.4 TrOCR (Microsoft)

- OCR Transformer.

10.2.5 Surya (VikParuchuri, 2024)

- OCR multilingue moderne, layout-aware.

10.2.6 MinerU (OpenDataLab)

- Extraction PDF de qualité publication.

10.2.7 docTR

10.2.8 Marker (PDF → Markdown)

10.2.9 Got-OCR2

10.2.10 Florence-2 (peut faire OCR)

10.2.11 Qwen2.5-VL (très bon OCR)

10.3 Modèles scientifiques / spécialisés

10.3.1 AlphaFold 2 / 3 (Google DeepMind)

- Repliement de protéines.
- AlphaFold 2 : poids ouverts.
- AlphaFold 3 : code ouvert (académique), poids restreints.

10.3.2 ESMFold / ESM-2 / ESM-3 (Meta)

- Modèles de langage protéiques.
- ESM-2 : 8M → 15B.

10.3.3 RoseTTAFold (Baker Lab)

10.3.4 OpenFold

- Réimplémentation open d'AlphaFold.

10.3.5 ChemBERTa / Galactica

10.3.6 MolMIM / DiffDock

10.3.7 Boltz-1 / Boltz-2 (MIT, 2024-2025)

- Repliement et docking, MIT license.

10.3.8 Chai-1 (Chai Discovery)

10.3.9 Climate / Earth modeling

- ClimaX, GraphCast, Pangu-Weather, Aurora.

10.3.10 MedicalLLMs

- Med-PaLM (non-ouvert), Meditron, BioGPT, ClinicalBERT, OpenBioLLM.

10.4 Modèles 3D

10.4.1 TRELLIS (Microsoft, 2024)

- Texte/image vers 3D.

10.4.2 Hunyuan3D / Hunyuan3D 2.0 (Tencent)

10.4.3 Trellis / SF3D / Stable Fast 3D / Stable Zero123

10.4.4 LGM, LRM, InstantMesh

10.4.5 Shap-E / Point-E (OpenAI)

Chapter 11

Partie IX — Frameworks et outils d'exécution locale

11.1 Inférence LLM

11.1.1 llama.cpp (Georgi Gerganov)

- **Langage** : C/C++.
- **Cible** : CPU, GPU (CUDA, ROCm, Metal, Vulkan, SYCL), Apple Silicon.
- **Format** : GGUF.
- **Site** : `https://github.com/ggerganov/llama.cpp`.
- **Importance** : moteur fondamental ; alimente Ollama, LM Studio, Jan, Text Generation WebUI…

11.1.2 Ollama

- **Site** : `https://ollama.com`.
- **Description** : gestionnaire de modèles façon Docker (`ollama run llama3.1`).
- **Repose sur** : llama.cpp.
- Multi-plateforme.

11.1.3 LM Studio

- Application desktop avec UI graphique, recherche Hugging Face, serveur OpenAI-compatible.
- `https://lmstudio.ai`

11.1.4 Jan

- Alternative open-source (AGPL) à LM Studio.
- `https://jan.ai`

11.1.5 GPT4All (Nomic)

- Application desktop, modèles préconfigurés.

11.1.6 text-generation-webui (oobabooga)

- WebUI puissante pour expérimentation, multi-backends.

11.1.7 KoboldCpp

- Spin-off de llama.cpp orienté écriture créative et roleplay.

11.1.8 vLLM

- Serveur d'inférence haute performance, *PagedAttention*.
- Cible serveur GPU.

- `https://github.com/vllm-project/vllm`

11.1.9 SGLang (LMSYS)

- Concurrent moderne de vLLM, RadixAttention.

11.1.10 TensorRT-LLM (NVIDIA)

- Optimisation extrême sur GPU NVIDIA.

11.1.11 MLC-LLM

- Compilation pour multiples backends (WebGPU, mobile).

11.1.12 ExLlamaV2 / ExLlamaV3

- Inférence GPU 4-bit ultra-rapide (format EXL2).

11.1.13 Transformers (HuggingFace)

- Bibliothèque Python de référence.

11.1.14 llamafile (Mozilla)

- Distribution mono-binaire d'un modèle + moteur (Cosmopolitan Libc).

11.1.15 GPT4All / PrivateGPT / Anything LLM / Continue / Open WebUI

- Surcouches utilisateur.

11.1.16 LocalAI

- Serveur OpenAI-compatible multi-modèles.

11.1.17 Cortex (Jan)

- Couche serveur de Jan.

11.1.18 Llamafile, llama-cpp-python, ctransformers

11.2 Inférence images

11.2.1 ComfyUI

- Interface noeud-graphique pour Stable Diffusion / FLUX / vidéo.
- `https://github.com/comfyanonymous/ComfyUI`

11.2.2 Automatic1111 (stable-diffusion-webui)

- WebUI historique pour SD.

11.2.3 Forge / reForge / SD.Next

- Forks optimisés.

11.2.4 InvokeAI

- WebUI commerciale-amicale (Apache).

11.2.5 Fooocus (lllyasviel)

- Approche simplifiée.

11.2.6 Diffusers (HuggingFace)

- Bibliothèque Python de diffusion.

11.3 Inférence audio / vidéo

11.3.1 whisper.cpp / faster-whisper / WhisperX / Speech Note

11.3.2 Coqui-TTS / Piper / Tortoise / Bark

11.3.3 CogVideoX-Diffusers / Open-Sora-CLI / ComfyUI-VideoHelper

11.4 Fine-tuning

11.4.1 Unsloth

- 2-5 (×) plus rapide, 50-70 % moins de VRAM.
- `https://github.com/unslothai/unsloth`

11.4.2 Axolotl

- Configuration YAML simple, multi-méthodes.

11.4.3 LLaMA-Factory (HiYouGa)

- WebUI + CLI, nombreuses méthodes.

11.4.4 LoRA / QLoRA (PEFT)

- Low-Rank Adaptation : entraîner peu de paramètres adaptateurs.

11.4.5 TRL (HuggingFace)

- DPO, PPO, RLHF.

11.4.6 DeepSpeed / FSDP / Accelerate

- Parallélisme distribué.

11.4.7 MLX-LM / mlx-examples (Apple)

- Fine-tuning sur Apple Silicon.

11.5 Quantization

11.5.1 GPTQ / AutoGPTQ / GPTQModel

11.5.2 AWQ / autoawq

11.5.3 bitsandbytes

11.5.4 llama.cpp `quantize`

11.5.5 ExLlamaV2 conversion

11.5.6 HQQ (Half-Quadratic Quantization)

Chapter 12

Partie X — Tableaux comparatifs

12.1 Comparatif LLM généralistes (synthèse 2025)

Modèle	Éditeur	Param. (B)	Contexte	MMLU	Licence
Llama 3.1 8B	Meta	8	128 K	73	Llama 3
Llama 3.1 70B	Meta	70	128 K	86	Llama 3
Llama 3.1 405B	Meta	405	128 K	88	Llama 3
Llama 3.3 70B	Meta	70	128 K	86	Llama 3
Llama 4 Scout	Meta	17/109 (MoE)	10 M	79	Llama 4
Llama 4 Maverick	Meta	17/400 (MoE)	1 M	86	Llama 4
Mistral 7B v0.3	Mistral	7	32 K	62	Apache 2.0
Mixtral 8x7B	Mistral	13/47 (MoE)	32 K	71	Apache 2.0
Mixtral 8x22B	Mistral	39/141 (MoE)	64 K	78	Apache 2.0
Mistral Nemo 12B	Mistral/NV	12	128 K	68	Apache 2.0
Mistral Small 3	Mistral	24	32 K	81	Apache 2.0
Qwen2.5 7B	Alibaba	7	128 K	74	Qwen
Qwen2.5 72B	Alibaba	72	128 K	86	Qwen
Qwen3 235B	Alibaba	22/235 (MoE)	128 K	87	Apache 2.0
DeepSeek-V3	DeepSeek	37/671 (MoE)	128 K	89	DeepSeek
DeepSeek-R1	DeepSeek	37/671 (MoE)	128 K	90	MIT
Gemma 2 9B	Google	9	8 K	71	Gemma
Gemma 2 27B	Google	27	8 K	75	Gemma
Gemma 3 27B	Google	27	128 K	78	Gemma
Phi-4 14B	Microsoft	14	16 K	85	MIT
Command R+	Cohere	104	128 K	76	CC-BY-NC
Granite 3.2 8B	IBM	8	128 K	67	Apache 2.0
Yi-1.5 34B	01.AI	34	32 K	77	Apache 2.0
GPT-OSS 120B	OpenAI	5/117 (MoE)	128 K	84	Apache 2.0

Note : les scores MMLU sont indicatifs et varient selon les protocoles d'évaluation.

12.2 Comparatif modèles de code

Modèle	Param. (B)	Contexte	HumanEval (%)	Licence
Qwen2.5-Coder 32B	32	128 K	92	Apache 2.0
DeepSeek-Coder-V2 236B	21/236	128 K	90	DeepSeek
DeepSeek-Coder-V2 Lite	2,4/16	128 K	81	DeepSeek
Codestral 22B	22	32 K	81	MNPL
CodeLlama 70B	70	16 K	67	Llama 2
StarCoder2 15B	15	16 K	47	OpenRAIL-M
Granite-Code 34B	34	8 K	60	Apache 2.0
Yi-Coder 9B	9	128 K	85	Apache 2.0

Modèle	Param. (B)	Contexte	HumanEval (%)	Licence
OpenCoder 8B	8	8 K	68	Apache 2.0

12.3 Comparatif modèles d'images

Modèle	Éditeur	Param.	Résolution	Licence
SD 1.5	RunwayML/CompVis	0,9 B	512	RAIL-M
SDXL 1.0	Stability	3,5 + 6,6 B	1024	RAIL-M
SDXL Turbo	Stability	3,5 B	512	SAI-NC
SD 3.5 Large	Stability	8 B	1024	SCL
FLUX.1 [dev]	BFL	12 B	1024	FLUX-NC
FLUX.1 [schnell]	BFL	12 B	1024	Apache 2.0
PixArt-Sigma	Huawei	0,6 B	4 K	OpenRAIL
Kandinsky 3.1	Sber	3,3 B	1024	Apache 2.0
Lumina-Next	Lumina	1,7 B	1024	Apache 2.0

Chapter 13

Partie XI — Méthodologie pratique

13.1 Choisir un modèle selon son matériel

VRAM disponible	LLM recommandé	Quant.
4 GB	Phi-3.5-Mini, Llama 3.2 3B, Gemma 2 2B	Q4
6 GB	Llama 3.1 8B, Mistral 7B	Q5
8 GB	Qwen2.5 7B, Llama 3.1 8B	Q6
12 GB	Mistral Nemo 12B, Phi-4 14B	Q5
16 GB	Qwen2.5 14B, Phi-4 14B	Q8
24 GB	Qwen2.5 32B, Codestral 22B	Q5
48 GB	Llama 3.1 70B, Qwen2.5 72B	Q4
80 GB+	Llama 3.1 70B, Mixtral 8x22B	Q8
2× 80 GB	Llama 3.1 405B	Q3

Pour CPU pur (sans GPU), comptez :

- 8 GB RAM : 1B-3B Q4
- 16 GB RAM : 7B Q4-Q5
- 32 GB RAM : 13B Q5, 7B Q8
- 64 GB RAM : 32B Q4-Q5
- 128 GB RAM : 70B Q4

13.2 Apple Silicon — cas particulier

La mémoire unifiée des Mac M1/M2/M3/M4 sert de VRAM. Un Mac Studio M2 Ultra 192 GB peut faire tourner Llama 3.1 405B en Q3, ce qu'aucun GPU consommateur n'autorise. Privilégier MLX ou llama.cpp avec backend Metal.

13.3 Pipeline RAG local recommandé

1. **Embedding** : `bge-m3` ou `nomic-embed-text-v1.5`.
2. **Vector store** : Qdrant, Chroma, LanceDB, Milvus, pgvector.
3. **LLM** : Qwen2.5 / Llama 3.1 selon ressources.
4. **Reranker** (optionnel) : `bge-reranker-v2-m3`.
5. **Orchestration** : LangChain, LlamaIndex, Haystack, ou code direct.

Chapter 14

Partie XII — Liens utiles

14.1 Plateformes et catalogues

- **Hugging Face** : `https://huggingface.co` — référentiel central des poids et datasets.
- **Hugging Face Spaces** : `https://huggingface.co/spaces` — démos en ligne.
- **Open LLM Leaderboard** : `https://huggingface.co/spaces/open-llm-leaderboard/open_llm_leaderboard`.
- **MTEB Leaderboard** (embeddings) : `https://huggingface.co/spaces/mteb/leaderboard`.
- **LMSys Chatbot Arena** : `https://lmarena.ai`.
- **Artificial Analysis** : `https://artificialanalysis.ai`.
- **Papers With Code** : `https://paperswithcode.com`.
- **GitHub Trending AI** : `https://github.com/trending`.
- **Replicate** : `https://replicate.com` (cloud, mais référence des modèles).
- **Modelscope** (Alibaba) : `https://modelscope.cn`.
- **Civitai** (modèles SD) : `https://civitai.com`.
- **TheBloke (archives)** : `https://huggingface.co/TheBloke`.
- **bartowski (GGUF)** : `https://huggingface.co/bartowski`.
- **mradermacher (GGUF)** : `https://huggingface.co/mradermacher`.

14.2 Outils

- **Ollama** : `https://ollama.com`
- **LM Studio** : `https://lmstudio.ai`
- **Jan** : `https://jan.ai`
- **GPT4All** : `https://gpt4all.io`
- **llama.cpp** : `https://github.com/ggerganov/llama.cpp`
- **vLLM** : `https://github.com/vllm-project/vllm`
- **SGLang** : `https://github.com/sgl-project/sglang`
- **ComfyUI** : `https://github.com/comfyanonymous/ComfyUI`
- **AUTOMATIC1111** : `https://github.com/AUTOMATIC1111/stable-diffusion-webui`
- **Open WebUI** : `https://openwebui.com`
- **Continue** : `https://continue.dev`
- **Aider** : `https://aider.chat`
- **AnythingLLM** : `https://anythingllm.com`
- **PrivateGPT** : `https://github.com/zylon-ai/private-gpt`
- **LocalAI** : `https://localai.io`
- **Unsloth** : `https://unsloth.ai`
- **LLaMA-Factory** : `https://github.com/hiyouga/LLaMA-Factory`
- **Axolotl** : `https://github.com/axolotl-ai-cloud/axolotl`

14.3 Ressources éditoriales et communautés

- **r/LocalLLaMA** : `https://reddit.com/r/LocalLLaMA` — communauté de référence.
- **r/StableDiffusion** : `https://reddit.com/r/StableDiffusion`.
- **HuggingFace Blog** : `https://huggingface.co/blog`.

- **Simon Willison's blog** : `https://simonwillison.net`.
- **Sebastian Raschka — Ahead of AI** : `https://magazine.sebastianraschka.com`.
- **Lilian Weng's blog** : `https://lilianweng.github.io`.
- **AK (@_akhaliq)** sur X — veille quotidienne.
- **Discord HuggingFace, llama.cpp, Mistral, ComfyUI** — supports techniques.

14.4 Dépôts d'évaluations et benchmarks

- **lm-evaluation-harness** : `https://github.com/EleutherAI/lm-evaluation-harness`.
- **lighteval** (HuggingFace) : `https://github.com/huggingface/lighteval`.
- **EvalPlus** (code) : `https://evalplus.github.io`.
- **BigCode Evaluation Harness**.
- **AI2 Reasoning Challenge**.

14.5 Bibliothèques de référence

- **Transformers** : `https://github.com/huggingface/transformers`.
- **Diffusers** : `https://github.com/huggingface/diffusers`.
- **PEFT** : `https://github.com/huggingface/peft`.
- **TRL** : `https://github.com/huggingface/trl`.
- **Accelerate** : `https://github.com/huggingface/accelerate`.
- **Datasets** : `https://github.com/huggingface/datasets`.
- **PyTorch** : `https://pytorch.org`.
- **JAX / Flax** : `https://github.com/google/jax`.
- **MLX** (Apple) : `https://github.com/ml-explore/mlx`.
- **LangChain** : `https://www.langchain.com`.
- **LlamaIndex** : `https://www.llamaindex.ai`.
- **Haystack** : `https://haystack.deepset.ai`.

14.6 Ressources francophones

- **linuxmaine.org** : `https://linuxmaine.org` (site de l'auteur).
- **Mistral AI** : `https://mistral.ai`.
- **CroissantLLM, Lucie, OpenLLM-France** : initiatives francophones.
- **France IA** : actualité institutionnelle.
- **LinuxFr.org** : suivi communautaire.

Chapter 15

Partie XIII — Lexique

Activation Sortie d'une couche du réseau neuronal après passage de la fonction non-linéaire.

Adapter (LoRA, QLoRA, DoRA) Petit ensemble de paramètres ajoutés à un modèle figé pour le spécialiser sans tout réentraîner.

Agent Système qui combine un LLM avec des outils (web, code, fichiers) pour accomplir des tâches en plusieurs étapes.

Attention Mécanisme du Transformer permettant à chaque token de pondérer son influence sur les autres.

AWQ (Activation-aware Weight Quantization) Méthode de quantization 4-bit prenant en compte l'importance des activations.

Backbone Réseau de base, généralement vision, sur lequel on greffe une tête de tâche.

Beam search Stratégie de décodage maintenant plusieurs hypothèses en parallèle.

BERT Bidirectional Encoder Representations from Transformers (Google, 2018), encodeur fondateur.

BF16 (BFloat16) Format flottant 16 bits avec dynamique exponentielle équivalente à FP32.

BPE (Byte-Pair Encoding) Algorithme de tokenisation utilisé par GPT, Llama, Mistral.

Calibration set Petit ensemble de données utilisé pour calibrer une quantization (GPTQ, AWQ).

Checkpoint État sauvegardé d'un modèle à un instant de son entraînement.

Chinchilla scaling Loi (DeepMind, 2022) suggérant un ratio paramètres/tokens d'entraînement optimal d' 1:20.

CLIP Contrastive Language-Image Pre-training (OpenAI, 2021), pose les bases du multimodal.

Context window / Contexte Quantité maximale de tokens traités simultanément en entrée.

ControlNet Module ajouté à un modèle de diffusion pour le conditionner sur une carte (canny, profondeur, pose...).

CUDA API GPGPU de NVIDIA. La plupart des frameworks IA en dépendent.

DDPM / DDIM Algorithmes de débruitage des modèles de diffusion.

Decoder-only Architecture Transformer ne comportant que la partie décodeur (Llama, GPT, Mistral).

Diffusion Famille de modèles génératifs reposant sur un processus de débruitage progressif.

Distillation Entraîner un petit modèle ('élève') à imiter un grand ('professeur').

DPO (Direct Preference Optimization) Alternative au RLHF, plus simple et stable.

Embedding Vecteur dense représentant un texte, une image ou un autre objet dans un espace continu.

Emergence Apparition de nouvelles capacités au-delà d'une certaine échelle de paramètres / données.

Encoder-decoder Architecture comportant un encodeur (BERT-like) et un décodeur (T5, BART, mBART).

Fine-tuning Entraînement supplémentaire d'un modèle pré-entraîné sur des données spécialisées.

Flash Attention Implémentation efficace du mécanisme d'attention (Tri Dao, 2022, 2023, 2024).

FP16 / FP32 Précision flottante 16 / 32 bits.

Function calling / Tool use Capacité d'un LLM à appeler des outils externes via JSON structuré.

GGUF Format binaire de stockage / quantization de llama.cpp.

GPTQ Méthode de quantization 4-bit pour GPU (Frantar et al., 2022).

GQA (Grouped-Query Attention) Variante d'attention partageant les têtes K/V (Llama 2 70B+).

Hallucination Génération d'une information fausse ou inventée par un LLM.

HellaSwag, ARC, MMLU Benchmarks usuels.

HQQ Half-Quadratic Quantization, sans calibration.

Inference Phase d'utilisation d'un modèle déjà entraîné.

Instruct / Chat / Base Variantes d'un modèle (suivi d'instructions, dialogue, brut).

KV-cache Cache des clés/valeurs d'attention accélérant la génération token-par-token.

LAION Large-scale Artificial Intelligence Open Network — datasets d'images-texte.

Latent space Espace compressé où opèrent les modèles de diffusion latente (SD).

Llama.cpp Moteur d'inférence C++ pour LLM, fondamental.

LLM (Large Language Model) Modèle de langage de grande taille.

LoRA (Low-Rank Adaptation) Méthode efficace de fine-tuning par adaptateurs de bas rang.

Mamba / SSM (State Space Model) Architecture à complexité linéaire (Albert Gu, Tri Dao).

Mixture of Experts (MoE) Architecture où seuls quelques sous-réseaux ('experts') sont activés par token.

MMLU Massive Multitask Language Understanding, benchmark généraliste.

Open-weights Poids librement accessibles, mais éventuellement sans le code/données d'entraînement.

Open-source Au sens strict : poids + code + données + recettes ouverts.

Overfitting Sur-apprentissage : le modèle mémorise au lieu de généraliser.

Parameter (paramètre) Poids appris du réseau.

PEFT (Parameter-Efficient Fine-Tuning) Fine-tuning de peu de paramètres (LoRA, prefix-tuning, etc.).

Perplexity (PPL) Mesure de qualité d'un modèle de langage (plus bas = mieux).

Pre-training Entraînement initial massif sur données non-étiquetées.

Prompt engineering Art de formuler les requêtes pour optimiser les réponses.

Quantization Réduction de la précision des poids (FP16 → INT8 / INT4 / etc.).

RAG (Retrieval-Augmented Generation) Architecture où le modèle est augmenté par une récupération de documents.

Reranker Modèle classant des documents par pertinence après une recherche initiale.

RLHF Reinforcement Learning from Human Feedback.

RoPE (Rotary Position Embedding) Encodage positionnel par rotation, standard moderne.

SafeTensors Format sécurisé de Hugging Face (sans pickle).

SDPA (Scaled Dot-Product Attention) Attention de base, accélérée nativement dans PyTorch 2.

SFT (Supervised Fine-Tuning) Fine-tuning supervisé classique.

Sliding Window Attention (SWA) Attention sur une fenêtre locale (Mistral 7B).

SoTA (State of the Art) État de l'art.

Tokenizer Algorithme convertissant le texte en tokens.

Token Unité élémentaire de texte traitée par un LLM (4 caractères en moyenne).

Transformer Architecture introduite par 'Attention Is All You Need' (Vaswani et al., 2017).

ViT (Vision Transformer) Transformer appliqué aux images par patches.

VRAM Video RAM (mémoire GPU).

Whisper Modèle STT d'OpenAI.

XLA / JIT Compilation à la volée pour accélérer les graphes de calcul.

Zero-shot / Few-shot Capacité à résoudre une tâche sans / avec quelques exemples seulement.

Chapter 16

Conclusion

L'IA locale est passée en trois ans (2023-2026) de la curiosité technologique à un outil mature, capable de couvrir la grande majorité des cas d'usage courants : assistance code, RAG documentaire, génération créative, traitement audio-visuel, agents autonomes.

Trois tendances se dessinent pour 2026 :

1. **La compression** : modèles de plus en plus petits à performances égales (Phi-4, Gemma 3, Qwen3 dense).
2. **La spécialisation** : familles dédiées (raisonnement R1, code Coder-V2, multimodal Pixtral).
3. **L'agentique locale** : combinaison LLM + outils + mémoire entièrement sur l'appareil.

Pour l'utilisateur, le compromis confidentialité / coût / qualité penche aujourd'hui clairement en faveur du local pour les usages quotidiens. Un assistant Mistral Nemo ou Qwen2.5 14B sur un PC à 1 500 € rivalise avec un service cloud d'il y a 18 mois, et ce, sans transmettre une seule donnée à l'extérieur.

L'auteur encourage le lecteur à expérimenter, contribuer aux projets open-source, partager modèles et fine-tunings, et soutenir les acteurs européens et francophones de l'écosystème.

« Le meilleur modèle d'IA est celui qui s'exécute là où vit votre donnée. »

Thierry Gayet — linuxmaine.org
Édition 2026