LLaMA – Open and Efficient Foundation Language Models

Analyse d'article – Cours Validation Numérique

Encadrée par Mme. Mouakher Amira et M. Revy Guillaume

Salma BENSMAIL

Université de Perpignan Via Domitia (UPVD)
Master 2 Calcul Haute Performance & Simulation (CHPS)
salma.bensmail@etudiant.univ-perp.fr

Janvier 2026

Plan de la soutenance

Efficacité LLaMA : combinaison de données de qualité, architecture épurée et optimisations HPC.

1. Contexte et Motivation

Contexte : le paradoxe des LLMs

Le défi

Entraîner un LLM coûte **extrêmement cher**. Exemple : GPT-3 (175B) $\rightarrow$ milliers de GPUs, plusieurs mois.

Le paradoxe

Plus gros $\neq$ meilleur. Hoffmann et al. (2022) montrent qu'on peut atteindre les mêmes performances avec un modèle **plus petit entraîné plus longtemps**.

Approche LLaMA

- Modèles plus petits : **7B à 65B**
- Entraînement long : **1–1.4T tokens**
- Données **100% publiques**

Idée clé

L'efficacité globale d'un LLM dépend de :

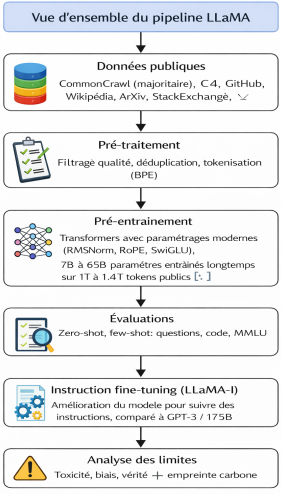
- **L'architecture**
- **Les données**
- **L'implémentation HPC**

Message clé : LLaMA optimise ces trois aspects simultanément.

2. Pipeline LLaMA : Étapes

Pipeline LLaMA : de la donnée au modèle

Vue end-to-end : données → entraînement → évaluation



3. Architecture Transformer Moderne

Architecture Transformer Moderne (LLaMA)

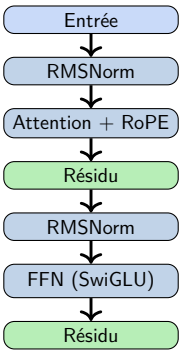
3 innovations clés

RMSNorm : normalisation stable en pré-norm.
RoPE : positionnement rotatoire (longues séquences).
SwiGLU : activation gated, plus performante.

Idée centrale

Pas une “révolution”, mais un assemblage **cohérent** + une exécution HPC efficace.

Flux d’un bloc Transformer LLaMA



4. Données & Tailles : Chiffres Clés

Données de pré-entraînement (mix)

Composition des données : la qualité/variété explique une grande partie des performances.

Source	Proportion (%)
CommonCrawl (filtré)	67.0
C4	15.0
GitHub	4.5
Wikipédia	4.5
Livres (Gutenberg + Books3)	4.5
ArXiv	2.5
StackExchange	2.0

À retenir : CommonCrawl domine $\Rightarrow$ filtrage/déduplication critiques.

Tailles des modèles & architecture

Hyperparamètres principaux (architecture Transformer).

Modèle	Dim.	Têtes	Couches	Tokens
LLaMA-7B	4096	32	32	1.0T
LLaMA-13B	5120	40	40	1.0T
LLaMA-33B	6656	52	60	1.4T
LLaMA-65B	8192	64	80	1.4T

Message : plus petit modèle + plus de tokens = bon compromis inference/training.

5. Résultats & Impact HPC

Résultats essentiels : comparaison

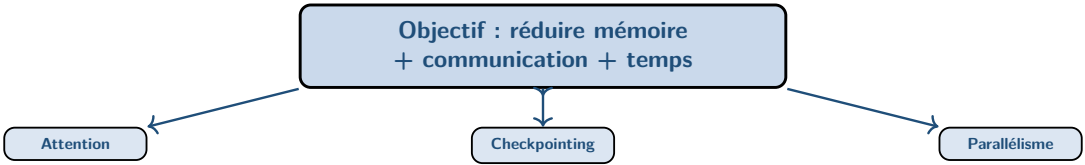
Récapitulatif des performances clés sur plusieurs benchmarks.

Modèle	Code (pass@1)		Common Sense (0-shot)		MMLU
	HumanEval	MBPP	HellaSwag	PIQA	5-shot Avg
GPT-3 175B	–	–	78.9	81.0	43.9
PaLM 62B	15.9	21.4	79.7	80.5	53.7
Chinchilla 70B	–	–	80.8	81.8	67.5
LLaMA-13B	**15.8**	**22.0**	79.2	80.1	**46.9**
LLaMA-65B	**23.7**	**37.7**	**84.2**	**82.8**	**63.4**

LLaMA-13B rivalise avec des modèles bien plus grands, et LLaMA-65B est compétitif face aux meilleurs.

Implémentation efficace : l'angle HPC

Stratégies d'optimisation pour réduire mémoire et temps



380 tokens/s/GPU A100 $\Rightarrow$ LLaMA-65B en $\sim$21 jours

Coût énergétique (ordre de grandeur)

Consommation énergétique et équivalent carbone.

Modèle	Énergie (MWh)	Carbone (tCO_2eq)
LLaMA-7B	36	14
LLaMA-13B	59	23
LLaMA-33B	233	90
LLaMA-65B	449	173

L'optimisation HPC aide, mais la taille du modèle reste le facteur dominant.

6. Conclusion et Perspectives

Conclusion : les 3 piliers

Architecture

RMSNorm, RoPE, SwiGLU : simples, mais efficaces.

Données

Données publiques + pré-traitement sérieux $\Rightarrow$ performance solide.

HPC

Attention efficiente + checkpointing + parallélisme rendent l'entraînement faisable.

Message clé

LLaMA est une **recette complète** : pipeline + optimisation + exécution.

Perspectives

Quantification (FP8), kernels CUDA optimisés, distillation, LoRA.

Merci !

Des questions ?

Remerciements
Mme. Mouakher Amira et Revy Guillaume

salma.bensmail@etudiant.univ-perp.fr