GOUVERNEMENT
Liberté
Égalité
Fraternité

Pôle d'Expertise de la Régulation Numérique – PEReN

RAPPORT TECHNIQUE

Etude IA & Environnement

Réalisée par le PEReN au titre de son programme de travail 2025, cette étude a été produite dans le cadre de son partenariat avec l'Arcep qui a souhaité la rendre publique. Les travaux réalisés ont bénéficié d'un accès aux moyens de calcul de l'IDRIS au travers de l'allocation de ressources 2025-AD011016936 attribuée par GENCI.

Introduction

L'intelligence artificielle générative (IAG) est devenue partie intégrante de la vie des français, dont un quart des plus de 18 ans s'en sert quotidiennement, et les trois quarts au moins une fois par semaine[1]. L'impact environnemental du numérique des Français, évalué chaque année par l'Arcep, doit donc prendre en compte ces nouveaux usages.

Or la consommation énergétique de modèles d'IA varie selon de nombreux critères. Le premier à prendre en compte est le type de tâche demandée, qui peut faire varier la consommation du simple au centuple : classification de texte ou d'image, détection d'objet ou encore description d'image. La génération d'images, par exemple, consomme dix fois plus que la génération de texte selon Luccioni et. al. (Figure 1[2]).

1 IPSOS, "L'USAGE DE L'INTELLIGENCE ARTIFICIELLE PAR LES FRANÇAIS", février 2025

2 Les valeurs présentes dans ce graphique illustrent l'état des connaissances en mai 2024. Certains travaux plus récents et certaines annonces industrielles revoient néanmoins les émissions de CO_2 à la baisse, d'un ou plusieurs ordres de grandeur, sans remettre en question les écarts les différents types de génération. Voir par exemple Carbon in Motion: Characterizing Open-Sora on the Sustainability of Generative AI for Video Generation, 04/2025.

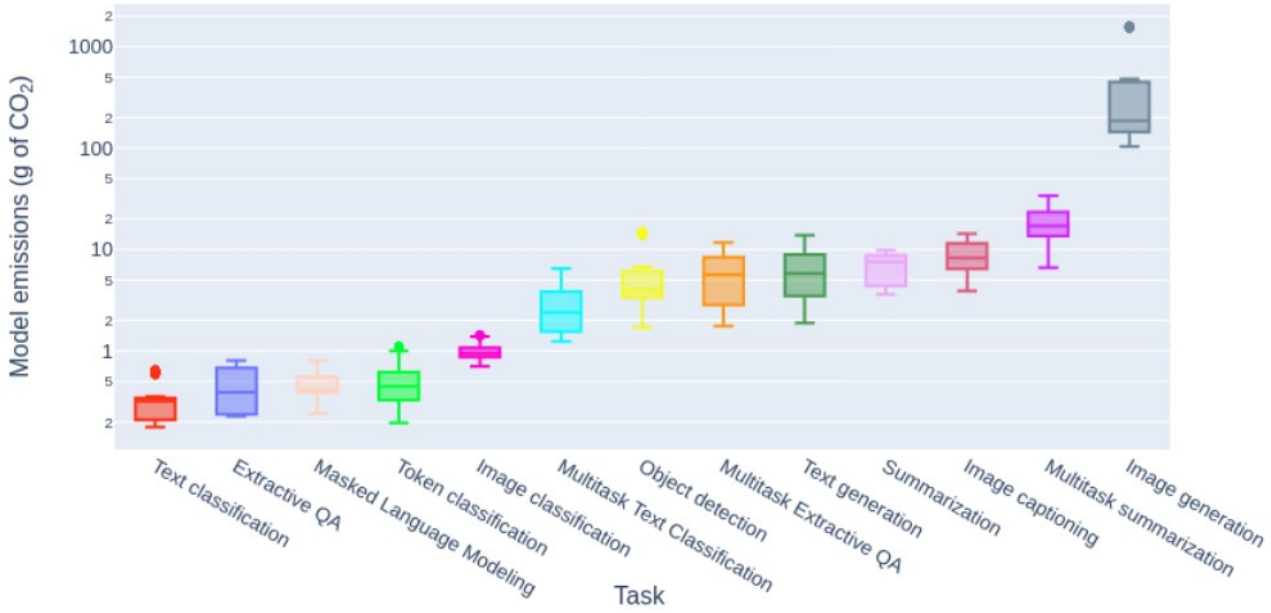

Figure 1: Luccioni et. al., Power Hungry Processing: Watts Driving the Cost of AI Deployment?, 05/2024

La présente étude se consacrera **uniquement à l'évaluation de la consommation énergétique de la génération de texte à l'inférence**, en s'intéressant aux variations de consommation qui peuvent exister entre ces types de tâches (raisonnement de bon sens, questions de connaissances, raisonnement mathématique[3], etc.).

Elle cherchera à évaluer, si l'on connaît à l'avance le type de génération de texte à effectuer, quels modèles optimiseraient le rapport performance / coût énergétique de ce type de génération.

Pour répondre à cette question, nous avons :

1. Lancé des modèles sur des jeux d'évaluation correspondant à différents types de tâches ;
2. Calculé des métriques de qualité et de consommation énergétique issues des processus lancés en 1. ;
3. Analysé ces métriques pour comprendre si certains modèles ont effectivement un rapport qualité / consommation énergétique meilleur que d'autres, et si oui essayer de comprendre pourquoi.

La première partie de cette étude expliquera la méthodologie utilisée pour réaliser ces trois étapes. La deuxième partie présentera les résultats extraits de notre analyse.

3 Voir les catégories de tâches proposées dans Meta AI, LLaMA: Open and Efficient Foundation Language Models, 02/2023

Table des matières

1. Méthodologie

1.1. Choix des jeux d'évaluation

1.1.1. L'évaluation en lien avec des usages réels

Nous nous sommes basés sur deux publications scientifiques (que l'on nommera Handa[4] et Realm[5]) ainsi que sur un sondage professionnel (Amperly[6]) pour dégager plusieurs thématiques d'usages fréquents de l'IA générative par le grand public. Nous avons ensuite confronté ces thématiques aux catégories de questions posées par les usagers du comparateur de modèles Compar:IA[7,8] afin de confirmer nos hypothèses.

Tableau 1: Thématiques d'utilisation des LLMs en ligne dans la littérature

Thème identifié	Usages Handa	Usages Realm	Usages Amperly	Usages Compar :IA
Code	Develop & maintain software / Program & debug IT / Perform IT system administration & maintenance	(Comp & Math) Content creation / Decision making / Assistance / Automation	Code or scripts	API & Python Development
Inventivité, créativité	Produce film, TV, theatre, music / Manage public relations & strategic communications / Develop marketing strategies	(Art & Media) Content creation	Creative writing or media	Image & Design Tools / Creative Poetry & Lyrics / Title & Name Suggestions
Usage générique quotidien, éducatif	Teach & instruct diverse subjects / Provide personal financial advice	(Edu & Library) Content creation, Assistance	Research and info / Educational Purposes	Information Queries / Recipe Instructions & Requests / Travel Planning / Entertainment Recommendations
Prise de décision,	Record, analyze,	(Bus & Finance,	Customer support	Marketing

4 K. Handa et al., Which Economic Tasks are Performed with AI? Evidence from Millions of Claude Conversations. 2025.
5 J. Cheng, K. Ghate, W. Hua, W. Y. Wang, H. Shen, and F. Fang, "REALM: A Dataset of Real-World LLM Use Cases," in Findings of the Association for Computational Linguistics: ACL 2025, Jul. 2025, pp. 8331–8341. doi: 10.18653/v1/2025.findings-acl.437
6 https://amperly.com/llm-survey-generative-ai-adoption-statistics/
7 https://huggingface.co/datasets/ministere-culture/comparia-conversations
8 Nous nous appuyons en particulier sur une analyse des thématiques présentes dans le jeu de données, conduite par Bunka

analyse spécialisée, professionnel	report data / Develop marketing strategies / Analyze financial data & develop budget / Provide customer service / Conduct chemical analyses	Management) Content creation / Decision Making / Assistance		Strategy & Innovation / Financial Operations / Property & Tax Law / Legal Procedures & Rights / Matrerial Properties & Measurement
Production de documents, résumé, aide à la rédaction	Manage book & document publishing /	Content synthesis	Emails and communication / Reports and Documents	Writing Assistance / Contract & Payment Letters

Nous dégageons ainsi cinq grandes thématiques d'utilisation des LLMs en ligne par le grand public dans la première colonne du Tableau 1.

1.1.2. Liste des jeux d'évaluation utilisés

Nous avons cherché différents benchmarks pour évaluer ces thématiques. Un benchmark est un jeu de données annoté qui permet d'évaluer un modèle. Bien que l'évaluation d'un modèle via un benchmark présente des limites identifiées dans la littérature scientifique[9], celle-ci reste la méthode d'évaluation de référence dans l'industrie. Par ailleurs, un benchmark est conçu pour évaluer le modèle sur un aspect précis. Dans notre cas, ces benchmarks sont des questions ou problèmes visant uniquement à évaluer la pertinence des réponses fournies par les modèles sur les thématiques sus-citées. Nous n'évaluons donc pas d'autres aspects pouvant intéresser des chercheurs ou des industriels, comme l'évitement de contenu inapproprié.

Il existe deux types d'évaluation :

- par comparaison de bonnes réponses, dans le cas où la formulation des réponses est unique.
- en utilisant un modèle tiers, ou modèle-juge (*llm-as-a-judge*), dans le cas où la réponse nécessite une évaluation plus complexe. Par exemple, pour HealthBench, l'évaluation est basée sur différents critères que le LLM-juge utilise pour évaluer la réponse générée[10].

Nous choisissons finalement six jeux d'évaluation (voir Tableau 2) pour leur adéquation avec les thématiques de l'étude, leur facilité à être implémentés avec la bibliothèque python Inspect[11] et leur popularité dans la littérature scientifique.

- Pour la thématique « usage générique quotidien, éducatif », trois benchmarks complémentaires ont été sélectionnés pour couvrir l'ensemble de cet usage :
    - des questions de culture générale, avec du texte libre (SimpleQA) ;

9 A. M. Bean et al., Measuring what Matters: Construct Validity in Large Language Model Benchmarks. 2025
10 Pour plus de détails, voir l'exemple donné sur le blog d'OpenAI : https://openai.com/fr-FR/index/healthbench/
11 Voir détails dans la section 1.3.1. Paramétrages

- des questions de « bon sens » portant sur les intéractions physiques (PIQA) ;
- des questions d'analyse d'image, puisque beaucoup de LLMs en ligne sont désormais multimodaux ;
- il aurait été utile de tester la recherche internet (*agentic web browsing*), cependant cela était impossible en raison de notre environnement de calcul sans connexion à internet directe[12].

- Le cas d'usage « Production de documents, résumé, aide à la rédaction », invitant notamment au RAG (*Retrieval Augmented Generation*), n'a pas pu être testé par manque de benchmark de ce type dans Inspect.

Tableau 2: Résumé des spécificités des benchmarks utilisés

	Tâche visée	Nombre d'exemples	Méthode d'évaluation
SimpleQA[13] (verified)	Questions courtes portant sur des faits connus	1 000	LLM-as-a-judge
ZeroBench[14]	Capacités de raisonnement sur des images	434 (200 + 134)	Comparaison
PIQA[15]	Raisonnement sur le monde physique	1 838	Comparaison
Healthbench[16]	Questions de santé	5 000	LLM-as-a-judge
HumanEval[17]	Génération de code	164	Comparaison
Writingbench[18]	Capacité rédactionnelle	128 (sous-catégories *Creative Writing* et *Advertising & Marketing)*	LLM-as-a-judge

Les jeux d'évaluation ayant un nombre de questions total très différent, dans la section 2. Résultats. **Par la suite, les consommations énergétiques seront données par question du jeu et non sur l'ensemble du jeu.**

12 Ibid. 11
13 L. Haas et. al., SimpleQA Verified: A Reliable Factuality Benchmark to Measure Parametric Knowledge, 2025.
14 J. Roberts et al., ZeroBench: An Impossible Visual Benchmark for Contemporary Large Multimodal Models. 2025.
15 Y. Bisk, R. Zellers, R. L. Bras, J. Gao, and Y. Choi, PIQA: Reasoning about Physical Commonsense in Natural Language. 2019.
16 R. K. Arora et al., HealthBench: Evaluating Large Language Models Towards Improved Human Health. 2025.
17 M. Chen et al., Evaluating Large Language Models Trained on Code. 2021.
18 Y. Wu et al., WritingBench: A Comprehensive Benchmark for Generative Writing. 2025.

1.2. Choix des modèles

Afin de pouvoir mesurer de manière fiable la consommation énergétique des modèles, nous devions réaliser nous-même les inférences sur une infrastructure connue. Nous avons donc été contraints de choisir des **modèles à poids ouverts** (*open-weights*) pour cette étude. Nous avons également rejeté les modèles avec des licences trop contraignantes, comme certains modèles de Deepseek. Enfin, nous avons choisi de concentrer cette étude sur l'IA générative. Pour des cas d'usages plus simples, des modèles d'IA non génératifs, voire des scripts simples, pourraient suffire à donner des résultats corrects, avec une consommation énergétique largement inférieure.

Nous avons ensuite cherché à maximiser la diversité dans le type de modèles choisis :

- venant d'entreprises différentes (Alibaba, Google, Meta, Mistral, Z.ai, ...) ;
- de plusieurs tailles différentes ;
- pouvant traiter uniquement du texte, ou des types de données variées (multimodaux) ;
- méthode d'entraînement : entraînement non supervisé classique, modèle spécialisé[19] sur un type de données, distillation, modèles de raisonnement ;
- avec des astuces diverses dans la construction du modèle afin de maximiser le rapport performance / vitesse : précision numérique (quantification), ou encore architecture (mélange d'experts) ;

Définition : Le **mélange d'experts** (MoE, ou *Mixture of Experts*) est une approche qui consiste à diviser un modèle en sous-réseaux distincts (les experts), chacun spécialisé dans un sous-ensemble de données d'entrée, afin d'effectuer conjointement une tâche[20]. Les modèles n'utilisant pas cette approche sont à distinguo appelés « denses ». Lors de l'inférence, un MoE n'utilise que la partie du réseau dont il a besoin, le nombre de ces paramètres dits « activés » est ainsi inférieur au nombre de paramètres totaux du modèle.

En pratique, nous avons choisi des modèles publiés entre juin 2024 et septembre 2025 car :

- Les modèles précédant cette période nous paraissaient trop vieux et risquaient de ne pas être compétitifs avec des modèles plus récents ;
- Cette période d'un an et demi a été riche en innovations technologiques dans les modèles, nous permettant de comparer l'impact de certaines innovations sur le rapport performance / consommation. Par exemple, en 2024, peu de modèles denses (non-MoE) ont été publiés. Au contraire, les modèles MoE ont pris leur essor à partir de septembre 2024, comme les modèles de raisonnement.

19 Voir Figure 9 pour plus de détails sur les modèles spécialisés.
20 https://www.ibm.com/fr-fr/think/topics/mixture-of-experts

Nous avons choisi des modèles entre 3 milliards et 123 milliards de paramètres. Nous avons divisé ceux-ci en trois catégories de taille, selon la classification du AI Energy Score :

- petit : moins de 20 milliards de paramètres totaux
- moyen : entre 20 et 66 milliards de paramètres totaux
- gros : plus de 66 milliards de paramètres totaux
- nous n'avons pas testé de très gros modèles (de plus de 125 milliards de paramètres) car nous n'avions à disposition que quatre GPUs maximum pour nos tests.

La méthodologie suivie pour sélectionner les modèles ainsi que la liste finale des modèles testés est détaillée en Annexe 1 : Modèles utilisés.

1.3. Implémentation

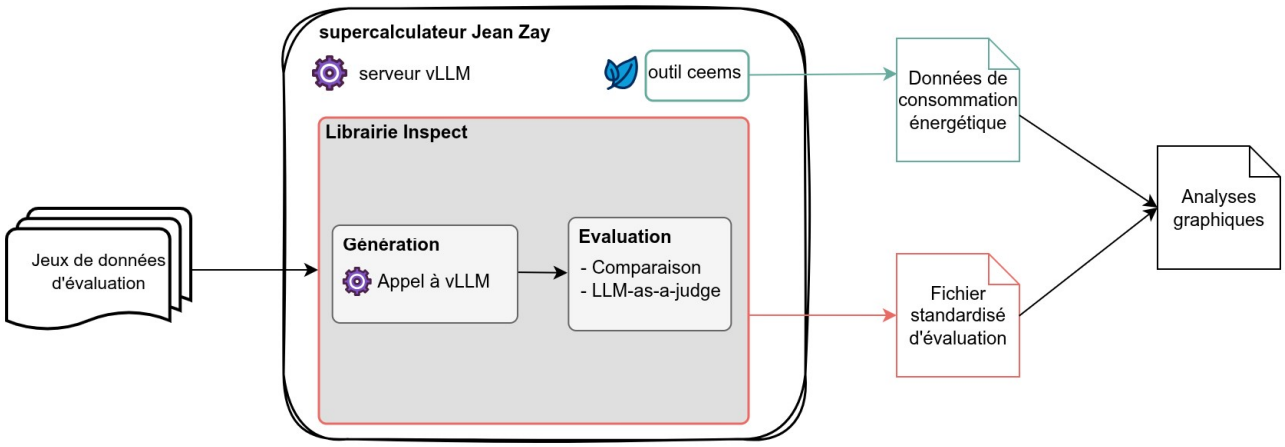

Figure 2: Schéma de l'implémentation technique pour notre étude

1.3.1. Paramétrages

Outils et ressources utilisés

Inspect AI[21] est une librairie open-source d'évaluation de grands modèles de langage développée par l'institut britannique pour la sécurité de l'intelligence artificielle (UK AI Security Institute). Elle contient des méthodes permettant de lancer l'inférence d'un LLM sur un jeu d'évaluation et de générer un fichier standardisé contenant les générations faites à l'inférence ainsi que l'ensemble des métriques d'évaluation. Elle permet également de paralléliser les requêtes sous différentes modalités.

Pour l'inférence, nous avons utilisé **vLLM**[22], qui est l'un des moteurs d'inférence de LLMs les plus populaires et le projet open-source comptant le plus de contributeurs sur Github en 2025[23]. Il a été

21 https://inspect.aisi.org.uk/

22 En version 0.11.0. Kwon et. Al., Efficient Memory Management for Large Language Model Serving with PagedAttention, 2023

23 https://github.blog/news-insights/octoverse/octoverse-a-new-developer-joins-github-every-second-as-ai-leads-typescript-to-1/

développé avec l'objectif d'optimiser l'utilisation de mémoire lors de l'inférence.

Toutes les évaluations ont été effectuées sur le **supercalculateur Jean Zay**[24] du Grand équipement national de calcul intensif (GENCI). Chaque évaluation a mobilisé entre **1 et 4 GPUs** NVIDIA H100[25] en fonction de la taille du modèle, toutes sur un même nœud de calcul. Les CPUs utilisés sont quant à eux des Intel Xeon Gold 6248.

Paramètres d'inférence

La qualité des réponses et la consommation énergétique des LLMs dépendent de nombreux paramètres d'inférence. Les paramètres d'échantillonnage et de limite de la longueur de génération sont souvent choisis de manière empirique pour maximiser la qualité des réponses selon les modèles et les tâches sur lesquels ils sont appelés.

Les valeurs choisies pour ces paramètres sont détaillées en Annexe 2 : Valeurs des paramètres.

1.3.2. Mesure d'énergie

La mesure de la consommation d'énergie dans le contexte de l'IA est un sujet extrêmement complexe, pouvant mener à des résultats très éloignés suivant les éléments effectivement mesurés et la méthodologie utilisée. Nos mesures s'appuient sur CEEMS[26], une bibliothèque open-source permettant de réaliser des calculs de consommation énergétique intégrée au sein du supercalculateur Jean Zay[27]. Pour cette étude, nous présenterons uniquement la **consommation énergétique des GPUs lors de la phase d'inférence** du modèle. En effet, nous avons remarqué que la consommation énergétique du CPU relevée par CEEMS variait beaucoup lorsque l'on reproduisait plusieurs fois la même expérience[28]. Par mesure de précaution (bien que cela réduise jusqu'à un tiers de l'énergie totale consommée par le modèle lors de l'inférence), nous préférons donc ne pas l'inclure dans les résultats.

Notre but n'est en effet pas de donner une valeur absolue de la consommation des modèles, mais de les comparer sur la base de leurs différentes propriétés. Ainsi, le seul risque lié à la non prise en compte de la consommation du CPU serait une pénalisation des modèles ayant une consommation CPU plus élevée que les autres, à consommation GPU équivalente. Cette hypothèse nous semble peu probable, et nos observations tendent à montrer que la consommation CPU est quasiment proportionnelle à la consommation GPU. Nous détaillons en Annexe 3 : Détail des mesures de consommation énergétique tous les facteurs ayant un impact sur l'empreinte carbone de l'IA qui ont été ignorés.

2. Résultats

Cette section présente les conclusions principales que nous avons pu tirer des expérimentations décrites ci-dessus.

#:~:text=The%20top%20open%20source%20projects%20by%20contributors

24 http://www.idris.fr/jean-zay/jean-zay-presentation.html

25 Plus précisément, pour un modèle donné, nous avons exploité le nombre minimal de GPU sur lequel il pouvait loger en mémoire. Dans le cas où 3 GPUs étaient nécessaires, nous avons arrondi à 4 car le quatrième n'était pas exploitable parallèlement.

26 https://github.com/ceems-dev/ceems

27 http://www.idris.fr/jean-zay/jean-zay-doc-energie.html

28 Voir 2.1.1. Variance des résultats

2.1. Étude préliminaire

Afin de choisir les paramétrages utilisés pour les analyses portant sur l'efficacité énergétique dans différents scénarios, nous avons réalisé les études ci-dessous, portant sur la fiabilité et les configurations de nos outils et modèles.

2.1.1. Variance des résultats

Pour des raisons pratiques et logistiques, pour la très grande majorité des modèles, nous n'avons effectué qu'**une seule mesure de la consommation et de la qualité des réponses sur un benchmark précis**.

Pour vérifier qu'une mesure unique reste valable, nous avons effectué cinq fois une même inférence sur le benchmark WritingBench *(écriture créative)* pour cinq modèles, afin d'observer s'il existait de la variance entre les résultats.

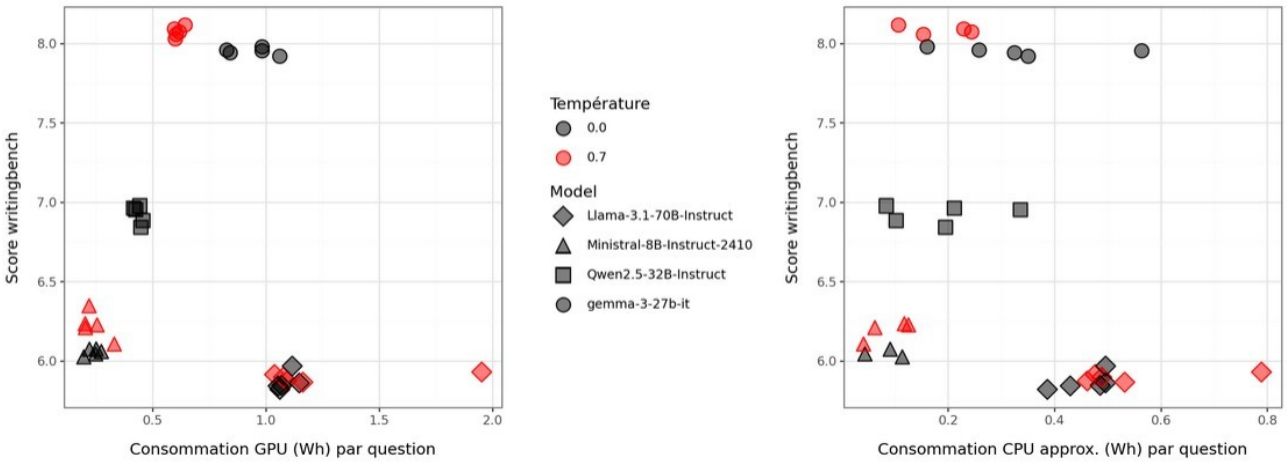

Figure 3: Consommations d'énergie GPU (à gauche) / approximation CPU (à droite) et score sur le benchmark Writingbench, dupliquées pour cinq modèles afin d'observer la variance des résultats (max_connections=10).

Stabilité des scores obtenus

On observe sur la Figure 3 que **pour une température et type de processeur (CPU vs GPU) fixés, le score des réponses du modèle varie peu**, ce qui est d'autant plus remarquable qu'il existe une première variation potentielle dans la réponse du modèle évalué, puis une seconde lors de l'évaluation de la réponse par *LLM-as-a-judge* dans le cas de Writingbench.

Choix de la mesure d'énergie

Concernant la mesure de consommation énergétique, sur GPU, pour la température haute comme basse, la consommation reste plutôt stable. **Pour la consommation GPU, nous pouvons donc considérer qu'une unique mesure faite pour chaque modèle est suffisante pour ne pas altérer la qualité des résultats.**

En revanche, on observe une très grande variation dans la consommation CPU relevée par l'outil CEEMS. Nous présenterons donc **la suite des résultats uniquement avec la consommation GPU.**

Choix de la température

La température d'un modèle contrôle le niveau d'aléatoire introduit dans les réponses. La réponse fournie pouvant grandement varier selon la température choisie, les résultats d'expérimentation pouvaient être très différents selon ceux effectués avec la température de 0 et celle de 0.7.

Utiliser une température haute de 0.7 impliquait que nos résultats étaient moins reproductibles par de futures études, car les réponses données par le LLM seraient beaucoup plus diverses en reproduisant plusieurs fois exactement le même paramétrage qu'avec la température de 0. Cependant, cela pouvait permettre d'améliorer la qualité des réponses des modèles dans certains cas testés, selon les recommandations des auteurs de certains benchmarks ou modèles utilisés.

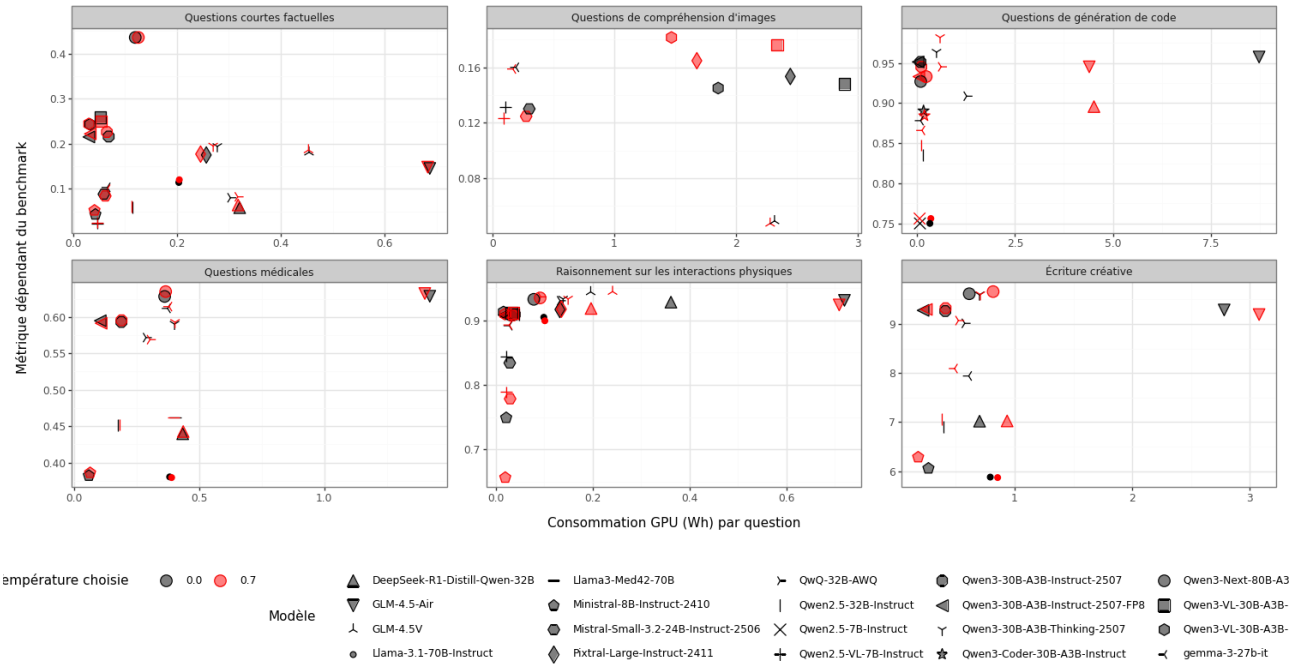

Figure 4: Variance des résultats en fonction de la température pour plusieurs modèles sur tous les jeux d'évaluation (max_connections=10)

En pratique, il est difficile de tirer une conclusion définitive sur l'influence de la température au vu des résultats illustrés sur la Figure 4. Pour les benchmarks SimpleQA (questions courtes) et Healthbench (questions médicales), qui consistent tous les deux en réponses ouvertes nécessitant l'évaluation d'un *LLM-as-a-judge*, les points correspondant aux températures 0 et 0.7 sont superposés. Certains modèles montrent des valeurs disjointes en fonction de la température sur d'autres jeux d'évaluation, mais la tendance peut être inversée entre différents benchmarks (par exemple pour GLM-4.5-Air, Deepseek-R1-Distill-Qwen ou Ministral-8B). Au global, aucun benchmark ou modèle ne montre de cas où une température aurait une performance systématiquement meilleure qu'une autre, que ce soit sur le score ou la consommation énergétique.

Nous choisissons donc, pour la suite, de ne présenter les résultats qu'avec une température égale à zéro afin de simplifier la lecture des graphiques et favoriser la reproductibilité des expériences.

2.2. Présentation des résultats

2.2.1. Le nombre de paramètres du modèle n'est pas équivalent à sa consommation énergétique

Nous avons constaté que les modèles très consommateurs sont toujours ceux avec le plus grand nombre de paramètres, mais l'inverse n'est pas toujours vrai (voir Figure 5). Certains modèles consomment autant ou moins d'énergie que des modèles beaucoup plus « petits ».

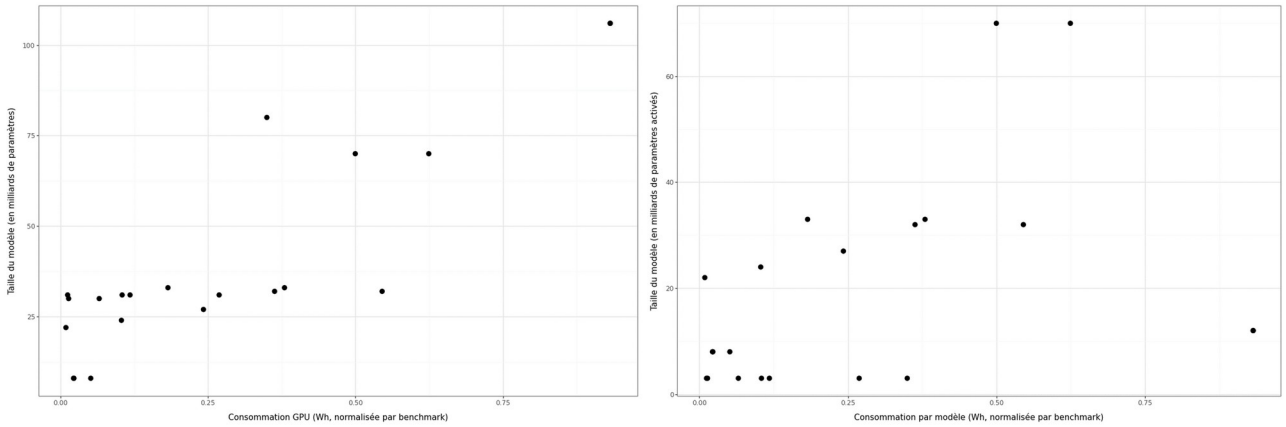

Figure 5: Consommation moyenne des modèles sur l'ensemble des jeux d'évaluation en fonction (a) à gauche, du nombre de paramètres du modèle ; (b) à droite, du nombre de paramètres activés du modèle. Pour réduire la variance due aux différences entre les jeux, la consommation affichée est entre 0 et 1 après une normalisation min-max sur chaque benchmark puis un calcul de la moyenne des consommations pour chaque modèle.

De fait, nos tests ont confirmé que la consommation énergétique est fortement corrélée à la durée d'inférence à utilisation du matériel constante (voir Figure 6)[29].

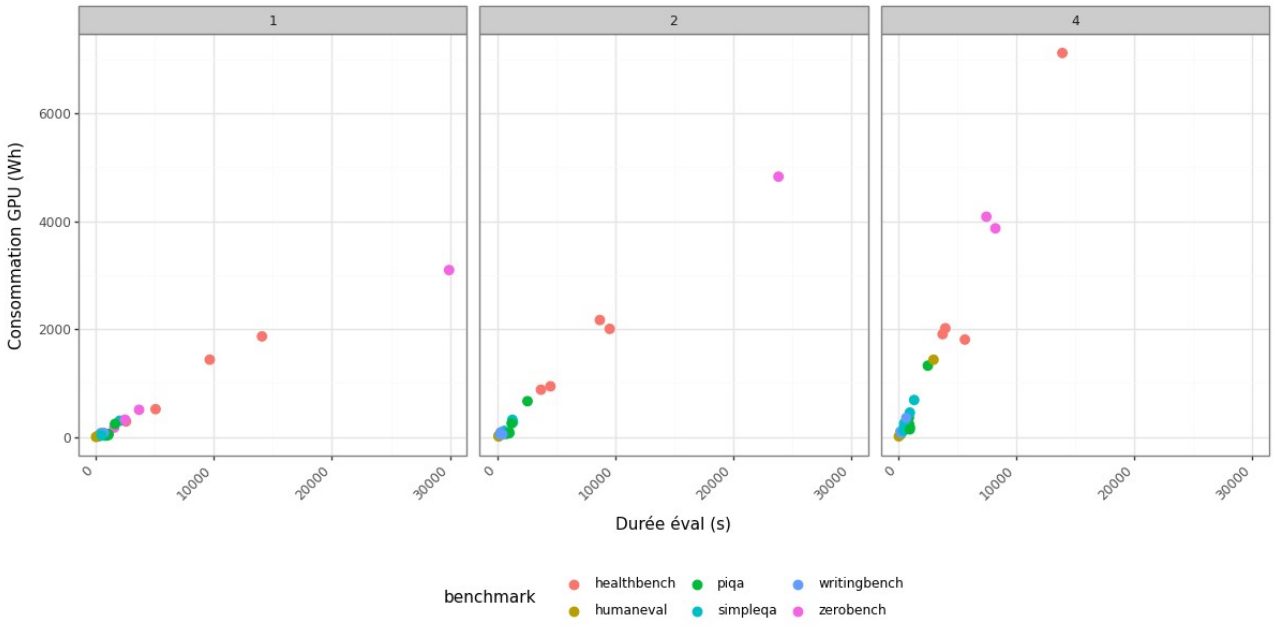

Figure 6: Consommation énergétique (en Wh) en fonction de la durée d'évaluation (temp=0) pour les modèles tournant sur 1, 2, ou 4 GPUs (max_connections=10).

29 Pour connaître le nombre de GPUs sur lequel tourne un modèle en particulier, voir l'Annexe 2 : Valeurs des paramètres

Or la durée d'inférence dépend à la fois du nombre de paramètres activés lors de l'inférence, d'éventuelles compressions du modèle (quantification) et du nombre de tokens générés. Ce nombre de tokens générés (voir Figure 7) dépend lui de nombreux paramètres, comme la température d'échantillonnage[30], l'architecture du modèle (dense, MoE), mais aussi sa procédure d'entraînement.

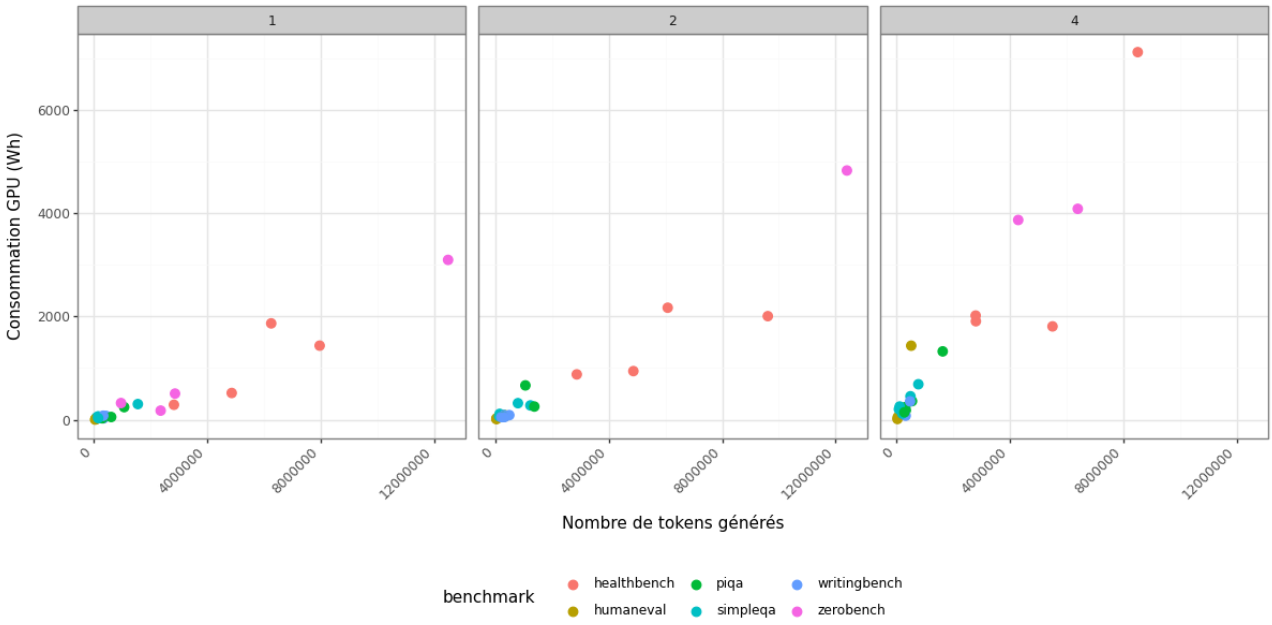

Figure 7: Consommation énergétique (en Wh) en fonction du nombre de tokens générés pour 1, 2, 4 GPUs. On voit une corrélation, mais moins évidente que pour la durée de génération (max_connections=10).

Afin de mieux comprendre la dépendance entre la consommation énergétique et les différentes caractéristiques d'un modèle, nous avons effectué une régression linéaire généralisée à effets mixtes sur la consommation énergétique, dont les détails se trouvent en Annexe 4 : Détail de la modélisation statistique de la consommation énergétique. Les principaux constats sont les suivants :

- Le premier facteur explicatif de la consommation énergétique est le nombre de paramètres.
- Le niveau de quantification, l'activation ou non de raisonnement dans la réponse et l'architecture de modèle (dense/MoE) ont également un impact qui, bien qu'étant moins important que le nombre de paramètres, reste significatif.
- L'effet de la quantification 8-bit sur la consommation énergétique est statistiquement significatif, avec un gain moyen de 39±18% de consommation.
- Les modèles type « mélange d'experts » consomment moins que des modèles denses ayant le même nombre de paramètres mais davantage que des modèles denses ayant le même nombre de paramètres activés. En moyenne, ils consomment 45±12 % de moins que les modèles ayant un nombre de paramètres équivalent.
- Le raisonnement a lui un effet très variable à la fois selon les modèles et selon les jeux de données. En moyenne il augmente significativement la consommation énergétique (+92 %),

30 Pour connaître les valeurs choisies pour les paramètres comme la limite maximale de tokens générés, la température d'échantillonnage, etc. consulter l'Annexe 2 : Valeurs des paramètres

mais l'effet réel semble dépendre du type de questions posées. Par exemple sur HumanEval l'effet moyen observé est de +849 % tandis que sur simpleqa de +41 %.

- Une partie de la variance reste non expliquée par le modèle statistique. Plusieurs facteurs pourraient être ajoutés : l'architecture du modèle qui est susceptible d'améliorer son efficacité, le nombre moyen de tokens générés pour une même requête, etc.

Estimer la consommation d'un modèle *a priori* est donc plus complexe que compter le nombre de paramètres.

2.2.2. Les modèles spécialisés étudiés ne sont pas gage de meilleure performance sur les tâches spécialisées

Nous avons testé trois modèles spécialisés dans deux domaines : la programmation informatique et les connaissances médicales.

Pour les modèles de code, la version *Coder* de Qwen3-30B-A3B produit un moins bon score que sa version *Instruct*. Codestral-v0.1, le modèle spécialiste de code de Mistral AI datant de juin 2024 (une mise à jour a été réalisée, mais non ouverte), performe quant à lui moins bien sur le jeu humaneval que presque l'intégralité des autres modèles plus récents, sans amélioration notable concernant la consommation électrique.

Concernant le domaine professionnel médical, la version spécialisée médicale de Llama-3-70B réussit effectivement mieux que la version de base[31]. Cependant, de nombreux autres modèles généralistes testés fournissent un meilleur rapport performance / consommation énergétique.

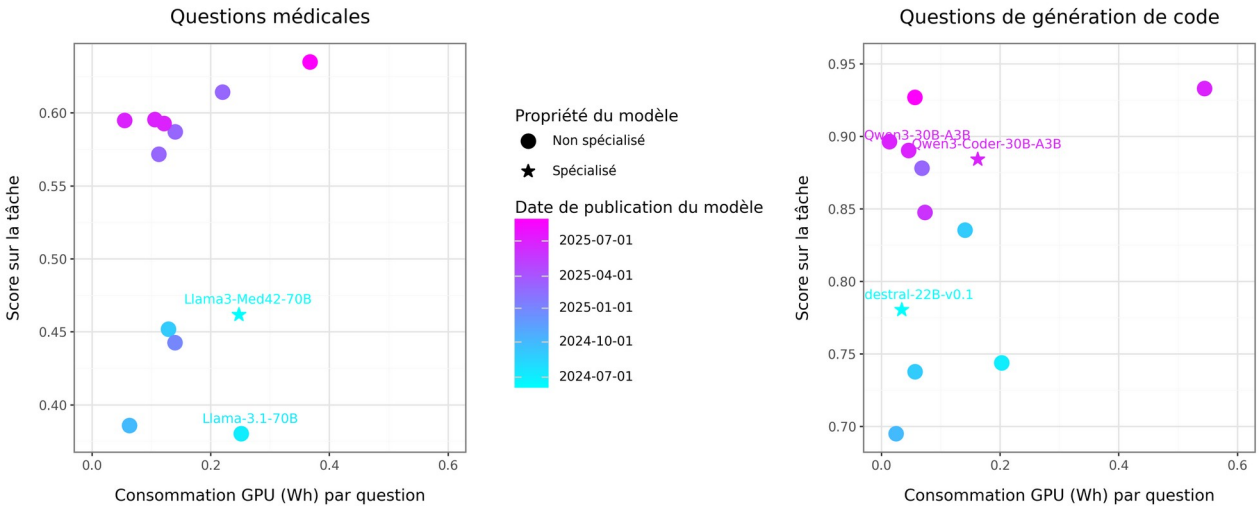

Figure 8: Score des modèles spécialisés ou non en fonction de leur consommation énergétique, avec un focus sur l'impact de leur date de publication

Comment expliquer ces résultats alors qu'un modèle spécialisé peut en théorie fournir de meilleures performances à consommation maîtrisée que la version généraliste dont il découle ? Nous voyons deux explications :

31 Plus précisément, nous comparons à Llama-3 .1-70B-Instruct, sorti trois mois plus tard.

- un modèle spécialisé est parfois entraîné sur une tâche trop précise pour assurer une bonne généralisation sur d'autres tâches dans le même domaine de spécialité. Par exemple, la documentation technique de Qwen3-Code-30B-A3B-Instruct le présente comme un modèle d'*agentic coding*[32], servant d'assistant dans un logiciel de développement. D'où sa performance plus faible sur le benchmark humaneval, contenant des questions plus ouvertes telles que « développe une fonction réalisant ceci… ». **Un modèle spécialisé ne peut donc correctement fonctionner que si on lui soumet des tâches très proches de ce à quoi il a été entraîné**, et non des tâches relevant seulement d'un domaine similaire.
- Il existe très peu de modèles spécialisés ouverts et les publications de ces derniers sont par conséquent largement moins fréquentes que celles des modèles généralistes. Or, comme l'illustre la Figure 8, les performances des LLM progressent avec leur date de publication. Cette évolution s'explique notamment par le fait que les nouveaux modèles intègrent les avancées techniques les plus récentes afin de dépasser l'état de l'art antérieur. Ainsi, si les modèles spécialisés ont généralement des performances plus élevées que les modèles généralistes de même génération, ils sont en revanche dépassés par les modèles plus récents.

Ainsi, l'absence d'une offre suffisante de modèles spécialisés ouverts rend ce levier inopérant : pour un utilisateur souhaitant un modèle ouvert, il peut être préférable d'utiliser un modèle généraliste plus récent.

2.2.3. Les modèles multimodaux ont des performances et une consommation similaires aux modèles textuels

Sur des tâches purement textuelles, nous avons constaté que les modèles multimodaux ont une consommation énergétique et une performance comparables à des modèles uniquement textuels. Si l'on regarde en particulier le modèle Qwen3-30B-A3B et sa version quantisée FP8, pour lesquels il existe une version multimodale, la version multimodale consomme autant pour une performance équivalente. La Figure 9 montre que les autres modèles multimodaux, comme ceux de Mistral ou Gemma, n'ont pas une consommation énergétique particulièrement supérieure à d'autres modèles évalués.

32 https://huggingface.co/Qwen/Qwen3-Coder-30B-A3B-Instruct

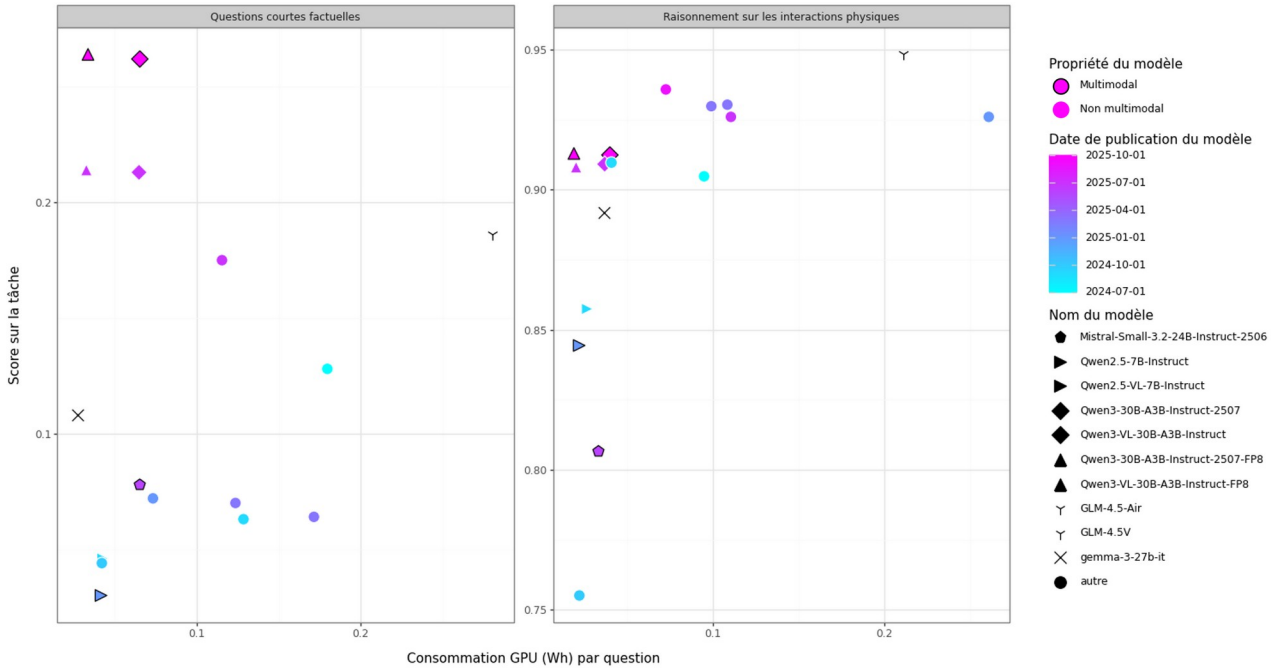

Figure 9: Score des modèles en fonction de leur consommation énergétique, avec un focus sur les modèles multimodaux.

En revanche, ces modèles présentent une consommation systématiquement plus élevée sur des tâches impliquant de l'analyse d'image. La Figure 10 montre par exemple que la consommation des modèles Mistral-Small-3.2-24B-Instruct-2506 et GLM-4.5V est de l'ordre du double sur de la compréhension d'image par rapport à des tâches purement textuelles.

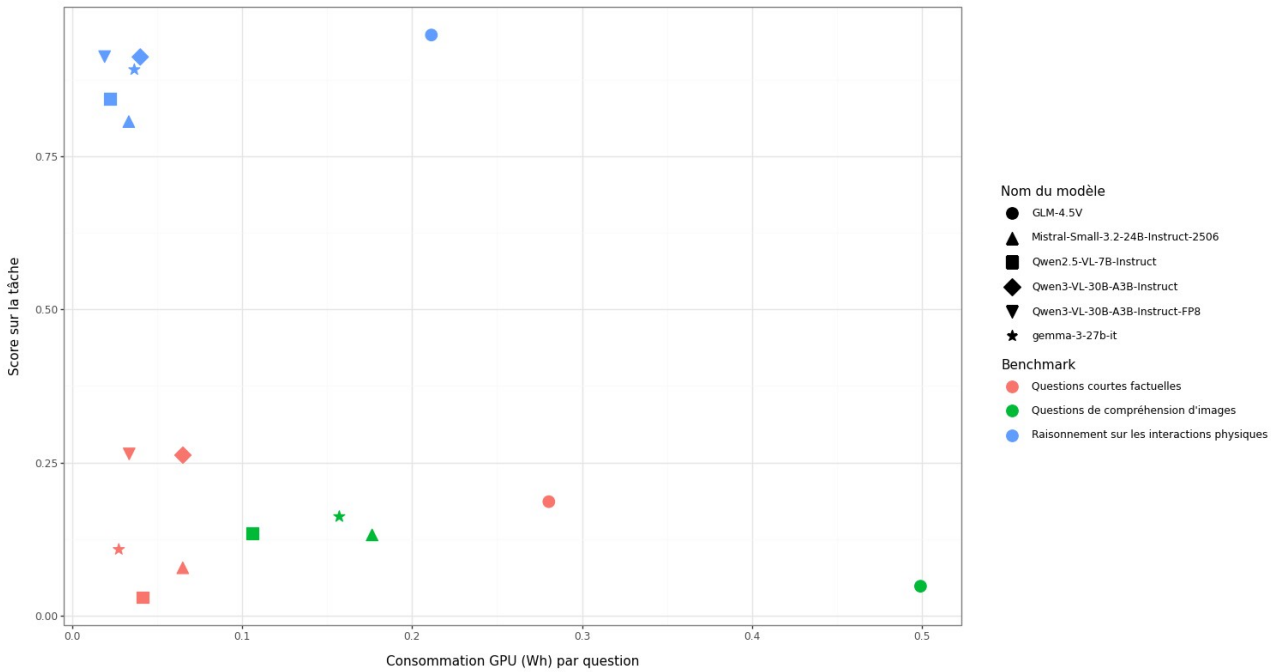

Figure 10: Score des modèles multimodaux en fonction de leur consommation énergétique sur des tâches purement textuelles et de compréhension d'images

2.2.4. Les modèles consomment davantage lorsqu'ils raisonnent, pour un gain de qualité inégal selon les tâches

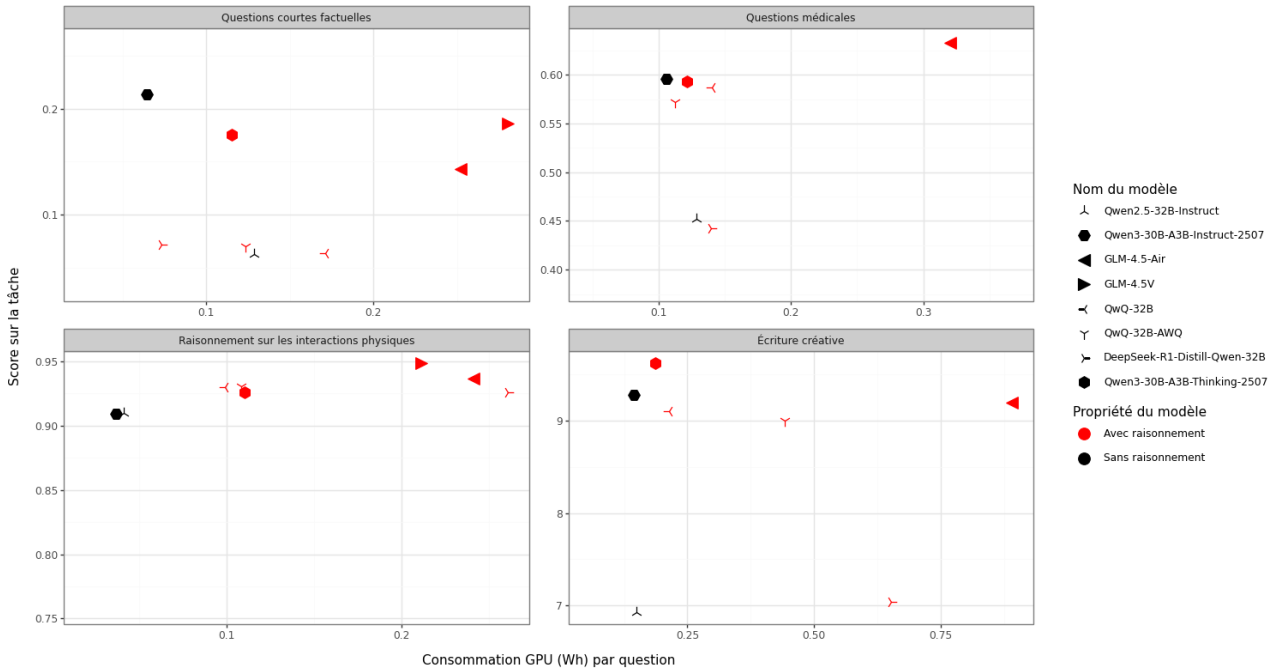

Figure 11: Score des modèles en fonction de leur consommation énergétique, avec un focus sur les modèles de raisonnement ainsi que les modèles généralistes dont ils sont issus.

Dans nos résultats, les modèles qui ont été entraînés spécifiquement pour résoudre des tâches complexes nécessitant plusieurs étapes de raisonnement consomment globalement plus d'énergie que des modèles classiques de taille équivalente, car ils génèrent davantage de tokens.

L'introduction de la propriété de raisonnement permet parfois une augmentation nette de la performance (par exemple, QwQ-32B sur l'écriture créative ou le code), mais une amélioration plutôt négligeable dans d'autres cas. D'autre part, la sur-consommation engendrée est parfois faible (de l'ordre de quelques dizaines de pourcent) et parfois très élevée (plusieurs centaines de pourcent) selon les questions posées.

Enfin, les modèles GLM-4.5 n'ont pas de version sans propriété de raisonnement auxquelles les comparer. Toutefois, nous observons que ces deux modèles sont les plus consommateurs parmi les modèles de cette étude, et ce sur tous les jeux sur lesquels ils ont été évalués (voir Figure 4).

Un utilisateur de LLM devrait donc s'assurer au préalable qu'un tel modèle est adapté à son cas d'usage, ou que le mode « raisonnement » est désactivé avant de l'utiliser.

2.2.5. Plus performant ne veut pas dire plus consommateur, et inversement

Sur les graphiques précédents, il apparaît qu'indépendamment de la spécialisation, de la multimodalité ou de l'activation d'un mode raisonnement, la performance n'est pas une fonction croissante de la consommation énergétique. Par conséquent, limiter l'empreinte énergétique n'implique pas toujours des compromis sur la performance, ce qui rend d'autant plus pertinente et intéressante la recherche d'efficacité énergétique.

Conclusion

Dans cette étude, nous avons cherché à identifier, pour un panel de cinq thématiques d'usages de modèles de langage génératifs (LLMs) par le grand public, quels types de modèles permettent d'optimiser le rapport entre qualité des réponses du modèle et consommation énergétique du GPU à l'inférence. A cette fin, nous avons testé 23 modèles développés par plus de cinq fournisseurs différents, sur six benchmarks. Au total, les expériences menées dans ce projet ont généré une consommation énergétique de 0,229 tCO2e en ce qui concerne les coûts CPU et GPU.

Les principales conclusions de ces expériences sont les suivantes : (1) la consommation énergétique dépend de nombreuses caractéristiques du modèle et non uniquement du nombre de paramètres ; (2) la qualité des réponses n'est pas corrélée à la consommation énergétique ; (3) les trois modèles spécialisés étudiés n'ont pas permis d'obtenir systématiquement de meilleures performances sur des tâches spécialisées ; (4) les modèles multimodaux ont des performances et une consommation similaires aux modèles textuels pour des tâches textuelles ; et (5) la propriété de raisonnement augmente la consommation énergétique, mais pas toujours la qualité des réponses du modèle.

Cette étude présente néanmoins un certain nombre de limites. D'abord, le périmètre des modèles testés a été réduit aux modèles à poids ouverts. Par conséquent, seulement trois modèles spécialisés ont pu être inclus, ce qui n'a pas permis de tirer des conclusions claires quant à leur performance. De plus, afin de limiter le nombre d'expériences pour respecter les contraintes de ressources et de temps, la variance entre deux exécutions a été négligée et chaque expérience n'a été lancée qu'une fois. Enfin, l'utilisation de benchmarks a permis d'évaluer les LLMs sur plusieurs milliers de questions et dans un cadre standardisé, mais elle s'accompagne aussi des défauts inhérents aux benchmarks, qui ne reflètent pas toujours les interactions réelles avec un LLM.

Par ailleurs, il reste important de mettre ces résultats en perspective avec l'impact du choix de l'infrastructure de calcul. Nous avons constaté, à matériel constant, que les modèles les plus consommateurs ont une consommation GPU quatre à cinq fois plus élevée que les modèles les moins consommateurs. Néanmoins un écart comparable peut être observé lorsqu'un même modèle tourne sur deux infrastructures GPUs différentes[33]. Des innovations technologiques sur les puces informatiques, peu diffusées dans l'industrie car difficiles à mettre en œuvre, permettent une réduction allant jusqu'à un facteur mille de la consommation énergétique[34].

Nous invitons donc de futurs travaux à consolider les résultats obtenus. Ces résultats pourraient également être confrontés aux données de Compar:IA, qui utilisent des votes d'utilisateurs plutôt que des performances basées sur des jeux d'évaluation.

33 Chen et. al., An Empirical Study on the Power Consumption of LLMs with Different GPU Platforms, 2024
34 Karchris et. al., A Survey on Hardware Accelerators for Large Language Models, Appl. Sci.**2025**, *15*(2), 586

Annexe 1 : Modèles utilisés

Méthodologie suivie pour sélectionner les modèles à tester

- Nous avons cherché à minimiser le nombre de modèles à évaluer en raison des délais et du temps de calcul alloué à cette étude sur l'infrastructure utilisée. Ainsi, pour chaque distribution de propriétés présentées en début de section 1.2. Choix des modèles (par exemple dense + multimodal + quantisé), nous avons donc cherché à sélectionner à l'avance le meilleur modèle dans cette catégorie.
- Pour les modèles denses, nous avons cherché un exemple de modèle petit, un moyen et un gros. Nous avons utilisé le podium OpenLLM Leaderboard qui donne des croisements performance - relevés de consommation énergétiques pour sélectionner le modèle avec le meilleur rapport { score / coût CO2 } pour chaque type de benchmark. OpenLLM Leaderboard est le leaderboard le plus complet que nous ayons trouvé avec à la fois des performances énergétiques et des scores de qualité. Cependant, les modèles référencés datent au plus tard de 2024 et sont peu nombreux. Pour tous les modèles présentant des caractéristiques qui n'étaient pas référencées dans OpenLLM leaderboard (comme les MoE, les modèles quantisés, les modèles de raisonnement, les modèles spécialisés), nous avons utilisé LMArena. LMArena est un leaderboard populaire pour connaître la performance des modèles plus récents où il n'existe pas de données de consommation énergétique, de nouveau sur chaque type de benchmark.

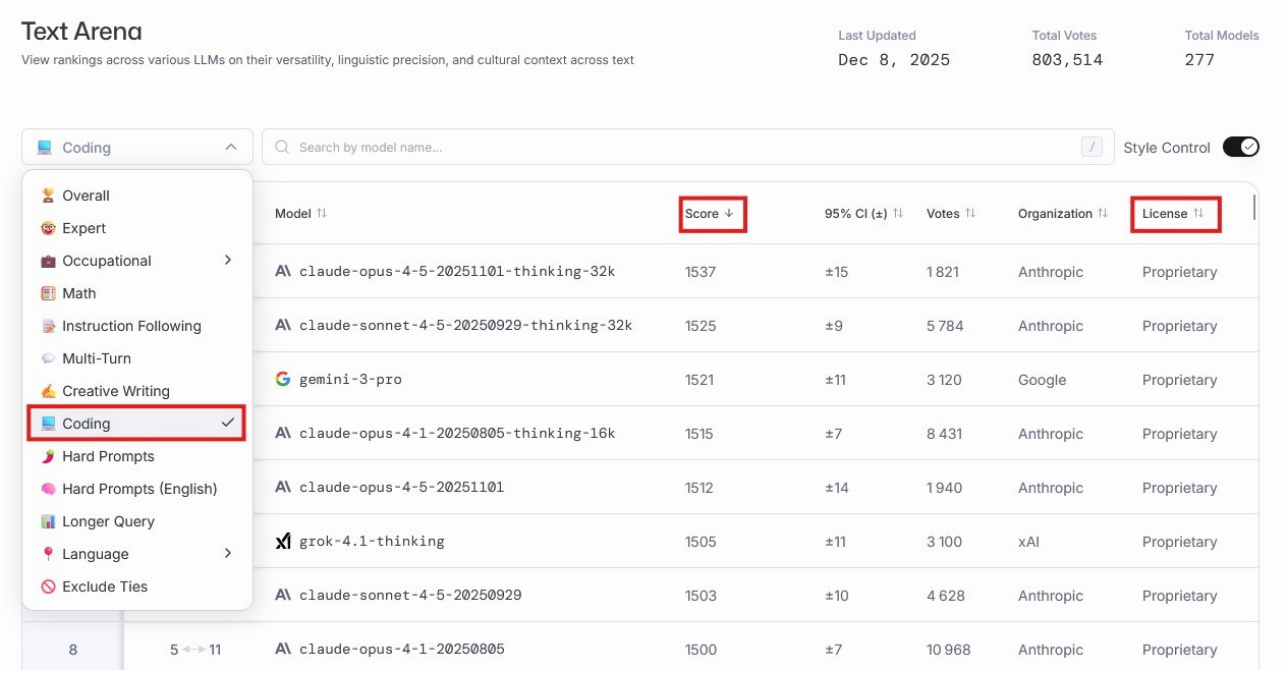

Figure 12: Capture de LMArena. Nous sélectionnons les meilleurs modèles pour un type de benchmark donné, avec une license ouverte.

- Pour les innovations technologiques liées à l'entraînement, la précision ou l'architecture, nous avons essayé de choisir les modèles en incrémentant à chaque fois la modification à

étudier. Par exemple, choisir un modèle M dense et sa version quantifiée, ou bien un modèle N et sa version spécialisée (voir *Figure 13*). Cependant, il n'existait pas toujours des modèles respectant toutes ces contraintes et permettant donc d'évaluer parfaitement l'impact de l'ajout d'une innovation technologique.

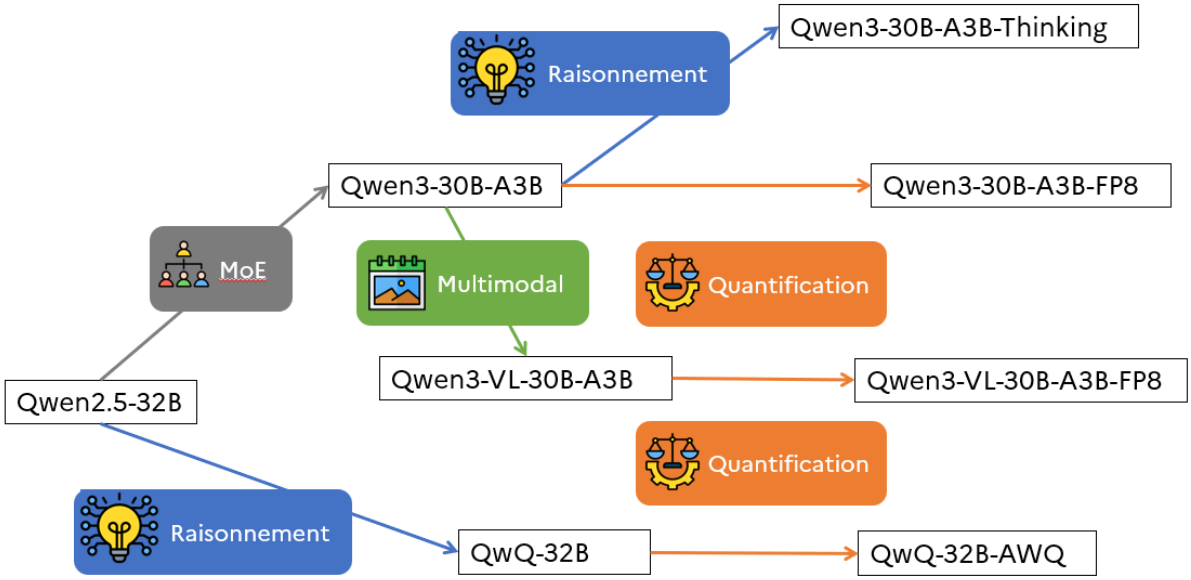

Figure 13: Choix de modèles en incrémentant une nouvelle spécificité pour les modèles Qwen de taille moyenne

- La très grande majorité des modèles étudiés sont de type *Instruct*, puisqu'il s'agit des modèles conversationnels mis à disposition du public.
- Pour certains benchmarks, le score de performance qualitative du modèle nécessitait l'intervention d'un LLM *as-a-judge* pour noter les réponses du modèle évalué. Nous avons utilisé en tant que juge le modèle Qwen3-Next-80B-A2B-Instruct, qui était le modèle open-source et de taille compatible avec notre infrastructure le mieux noté sur LMArena. Ce choix est conforté par nos résultats qui montrent que ce modèle figure parmi ceux ayant les meilleurs résultats qualitatifs. Le LLM *as-a-judge* était systématiquement lancé sur 4 GPUs NVIDIA A100.

Liste des modèles testés

Tableau 3: Liste des modèles utilisés pour l'étude[35]. Entre parenthèse la catégorie de modèle est précisé. (G) pour gros, (M) pour moyen et (P) pour petit. La couleur fait référence au fournisseur de modèle Alibaba, Google, Meta, Mistral, Z.ai, Autre

Benchmark	Modèles denses	MoE	Quantisés	Autre spécificité
Tous hors ZeroBench (modèles textuels communs)	(G) Llama-3.1-70B-Instruct (M) Qwen2.5-32B-Instruct, gemma-3-27b-it (P) Qwen2.5-7B-Instruct, Ministral-8B-Instruct-2410	(G) GLM-4.5-Air, Qwen3-Next-80B-A3B-instruct (M) Qwen3-30B-A3B-Instruct-2507	Qwen3-30B-A3B-Instruct-2507-FP8 (Raisonnement + Quantisé) QwQ-32B-AWQ	(Distillé) DeepSeek-R1-Distill-Qwen-32B (Raisonnement) Qwen3-30B-A3B-Thinking-2507, QwQ-32B
ZeroBench + SimpleQA + PIQA (modèles multimodaux)	(G) Pixtral-Large-Instruct-2411 (M) Mistral-Small-3.2-24B-Instruct-2506 (P) Qwen2.5-VL-7B-Instruct	(G) GLM-4.5V (M) Qwen3-VL-30B-A3B-Instruct-2507	Qwen3-VL-30B-A3B-Instruct-2507-FP8	
HumanEval	(M) Mistral-Small-3.2-24B-Instruct-2506			(spécialisé) Qwen3-Coder-30B-A3B-Instruct, Codestral-22B-v0.1
HealthBench				(spécialisé) Llama3-Med42-70B

35 Les détails de chaque modèle peuvent être trouvés dans leur documentation HuggingFace sur le lien https://huggingface.co/fournisseur/nom-modèle, par exemple https://huggingface.co/mistralai/Mistral-Small-3.2-24B-Instruct-2506

Annexe 2 : Valeurs des paramètres

Pour définir les choix de paramètres d'échantillonnage des évaluations de cette étude, nous avons effectué une revue des publications associées aux jeux d'évaluation et de modèles pour identifier les paramètres recommandés. Le Tableau 3 en présente les principaux résultats.

Tableau 3: Revue des paramètres d'échantillonnage des publications académiques de jeux d'évaluation et de modèles

	temperature	top-p	top-k	max-gen-tokens
Paramètres pour les jeux d'évaluation				
ZeroBench[36] (basse température)	0	1	-	-
ZeroBench (haute température)	0,7	-	0,95	-
WritingBench[37]	0,7	20	0,8	16 000
Paramètres pour les modèles				
Mistral[38]	0	-	-	-
Meta[39]	0	-	-	-
Qwen3[40] (avec raisonnement)	0,6	20	0,95	32 768
Qwen3 (sans raisonnement)	0,7	20	0,8	32 768

Par conséquent, les évaluations de cette étude ont été effectuées avec les paramètres suivants :

- **température d'échantillonnage** : le modèle évalué est testé avec deux valeurs de `temperature` , 0 et 0.7 (min : 0, max : 2). Le modèle "juge" a quant à lui une température fixée à 0.2, la valeur par défaut suggérée par Inspect AI.

36 ZeroBench:An Impossible* Visual Benchmark for Contemporary Large Multimodal Models.
37 WritingBench: A Comprehensive Benchmark for Generative Writing.
38 https://github.com/mistralai/mistral-evals/tree/main
39 https://github.com/meta-llama/llama-cookbook/blob/b5f64c0b69d7ff85ec186d964c6c557d55025969/tools/benchmarks/llm_eval_harness/meta_eval_reproduce/eval_config.yaml
40 Qwen3 Technical Report

- **seuils d'échantillonnage** : les paramètres vLLM `top-k` et `top-p` limitent le nombre de tokens échantillonnés à chaque étape de la génération à la valeur égale à `top-k` et / ou aux tokens dont la probabilité cumulée est égale à `top-p`.
- *nombre de tokens générés* : le paramètre `max-gen-tokens` de vLLM est utilisé pour limiter la génération à 16 000 tokens pour les évaluations sur WritingBench, et non renseigné sinon.

D'autres paramètres concernent la configuration de l'infrastructure :

- **nombre de processeurs** : chaque modèle tourne sur le nombre minimum de GPUs sur lesquels il peut loger en mémoire[41]. Le processus n'a d'accès qu'à ce nombre minimum, et le paramètre vLLM `tensor-parallel-size`, pour charger les modèles sur plusieurs GPUs, est fixé en conséquence. Pour chaque GPU réservé, nous utilisons 24 CPUs[42]. Pour choisir ce nombre de GPUs, nous avons utilisé l'approximation conventionnelle de calcul de la VRAM utilisée par la communauté ingénieure[43] :

  $$VRAM = \left(\frac{P \times 4}{\frac{32}{Q}} \right) \times 1.3 \quad ,$$

  où *P* correspond au nombre total de paramètres du modèle et Q à la quantification utilisée (FP16, 8, 4). Après des tests empiriques, nous avons utilisé une marge de 1.3 pour la surcharge VRAM car de nombreux modèles ne rentraient pas en mémoire avec la marge habituelle de 1.2.
- **fraction de mémoire GPU** utilisée pour charger les poids, activation et KV cache du modèle : le paramètre vLLM `gpu-memory-utilization` a été fixé à 0.9.
- nombre maximum de **connexions concurrentes** : nous avons fait tourner chaque couple (modèle, benchmark) avec trois valeurs (10, 100, 1000) pour le paramètre Inspect AI `max-connections`[44], qui permet de paralléliser les requêtes. Afin de simuler au mieux les paramétrages d'un fournisseur professionnel, les résultats présentés dans la partie 2. Résultats utilisent, pour chaque modèle, la valeur de `max_connections` permettant de minimiser le temps d'exécution sur le benchmark.

Nous avons cherché à tester l'influence du nombre de requêtes maximal au serveur vLLM sur la consommation énergétique du modèle. En effet, les produits grand public recevant des requêtes d'utilisateurs à destination de LLMs en ligne, tels que ChatGPT ou « Le Chat », ne traitent pas ces requêtes les unes après les autres : les requêtes sont traitées parallèlement dans un « batch » par le LLM afin de pouvoir diminuer le temps de réponse (voir Figure 14).

41 Dans le cas où 3 GPUs étaient nécessaires, nous avons arrondi à 4 car vLLM nécessite un nombre pair de GPUs pour pourvoir paralléliser.
42 Conformément aux recommandations de la documentation Jean zay pour la partition H100
43 https://lipikaaggarwal.github.io/LLM-Memory-Estimator/, https://apxml.com/posts/how-to-calculate-vram-requirements-for-an-llm
44 https://inspect.aisi.org.uk/parallelism.html

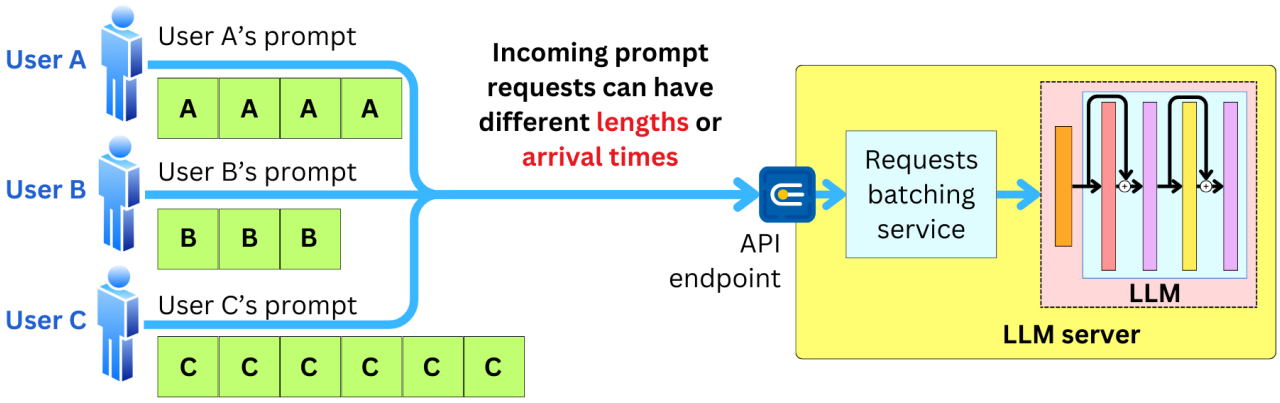

Figure 14: Pour trois utilisateurs différents, les requêtes peuvent avoir des tailles variées et arriver à différents moments. Le serveur LLM les regroupe dans un même batch pour les traiter. (Source : Damien Benveniste, The Ai Edge Newsletter)

Cependant, un trop grand nombre de connexions simultanées pourrait dépasser le seuil du nombre de requêtes possibles, ce qui entraînerait des délais supplémentaires et augmenterait ainsi la consommation d'énergie. En théorie, il existe donc un nombre de requêtes parallèles optimisant le temps de réponse et la consommation énergétique.

Annexe 3 : Détail des mesures de consommation énergétique

Consommations prises en compte

Notre travail se focalise sur la mesure de la consommation énergétique générée lors de l'utilisation des modèles d'IA par les particuliers. En particulier :

- les consommations non-énergétiques (d'eau, et de ressources minières en particulier) ne sont pas prises en compte ;
- les émissions de CO_2 ne sont pas mesurées, nous permettant de nous abstraire des conditions de production d'énergie locales[45] (utilisation d'énergie renouvelable, nucléaire ou fortement carbonée, compensation carbone, etc.).

Seule une partie réduite du cycle de vie de l'IA est évaluée : l'utilisation par les particuliers. Nous ne tenons pas compte, côté fournisseur, des coûts liés à la conception initiale des data centers, à leur entretien (remplacement des pièces usagées par exemple), à l'entraînement des modèles, et au chargement des modèles en mémoire. Côté utilisateur, nous ignorons les coûts de fabrication et d'utilisation des terminaux utilisateurs, ainsi que le coût énergétique lié à l'utilisation du réseau internet pour transmettre les requêtes et réponses aux serveurs d'IA. En effet, ces requêtes engendrent la même consommation énergétique que des usages numériques plus classiques, comme publier un texte sur Facebook, qui sont négligeables comparées au coût de génération. À noter que l'augmentation des usages de l'IA ainsi que les éventuels "effets rebonds"[46] (augmentation de l'usage compensant les gains d'efficacité liés à l'amélioration d'une technologie) sont également ignorés dans cette étude.

En résumé, notre travail étudie donc la consommation énergétique liée à la génération de réponses par les modèles d'IA au sein d'un data-center, à l'exclusion de tout autre facteur, comme illustré en Figure 15.

45 Voir par exemple les différences obtenues dans les mesures de CO2 entre Bloom et GPT-3 : https://news.climate.columbia.edu/2023/06/09/ais-growing-carbon-footprint/#:~:text=According%20to%20Microsoft%2C%20all%20the,their%20energy%20from%20renewable%20sources

46 https://fr.wikipedia.org/wiki/Effet_rebond_(%C3%A9conomie)

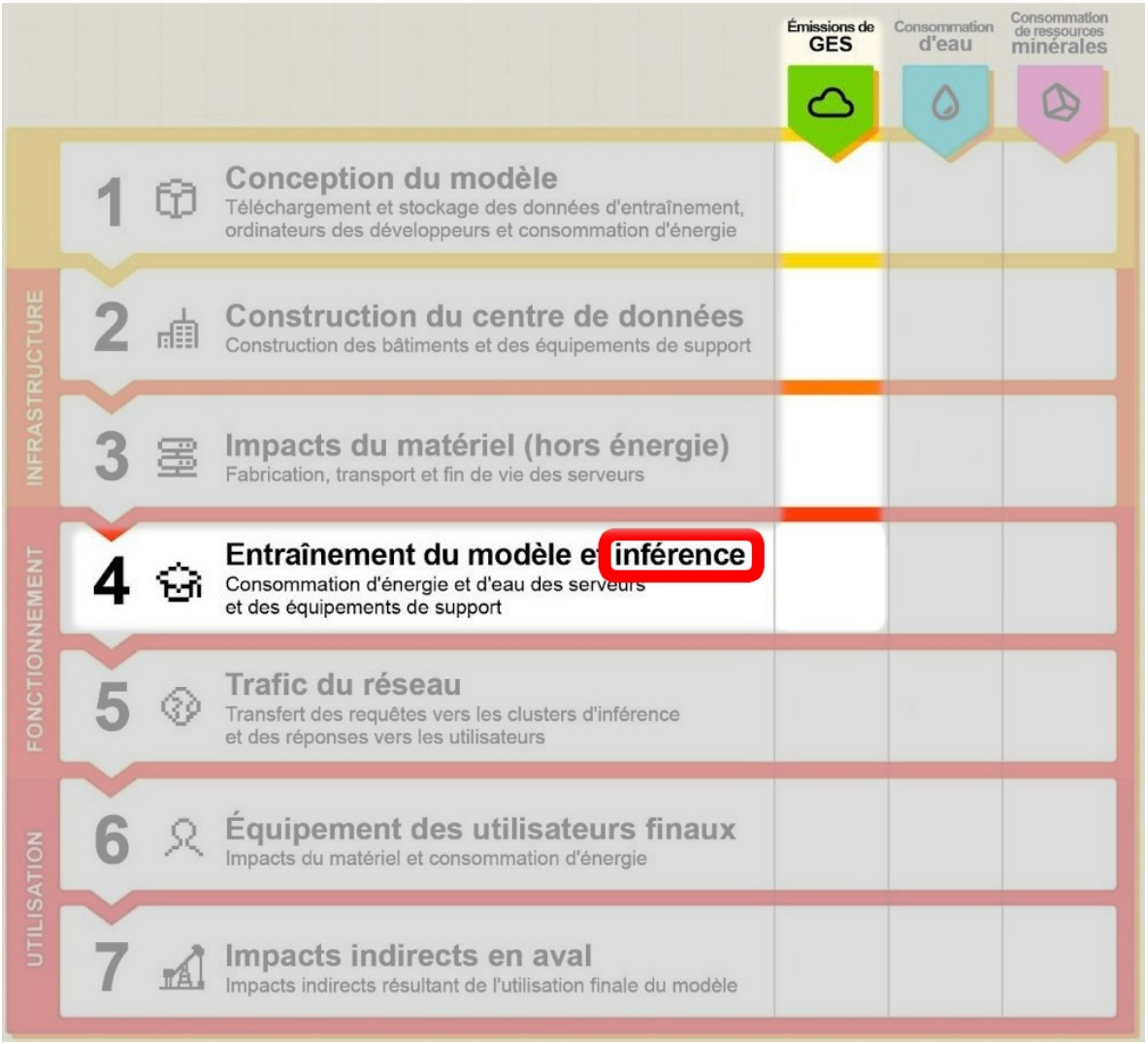

Figure 15: Tableau illustratif de l'objet de notre mesure, repris et adapté des travaux de Mistral en collaboration avec l'ADEME et Carbone 4. ***Nous mesurons néanmoins l'énergie consommée en Wh et non les émissions de gaz à effet de serre.*** *https://mistral.ai/fr/news/our-contribution-to-a-global-environmental-standard-for-ai*

Méthodologie de calcul

L'outil CEEMS mesure uniquement la consommation au niveau des nœuds. Les coûts du refroidissement ou des interconnexions ne sont pas pris en compte[47].

Lorsque l'on réalise une inférence de modèle sur un benchmark, le processus à partir duquel la consommation d'énergie est calculée contient la consommation liée à la génération des réponses et celle liée au chargement des modèles en mémoire. Pour contourner ce problème, nous avons effectué des mesures supplémentaires sur des tâches se limitant au chargement du modèle en mémoire. Ces consommations obtenues ont par la suite été soustraites de celles calculées lors de nos expérimentations principales, comme illustré en Figure 16[48].

47 En dernière analyse, ces coûts restent complexes à isoler au sein d'un data-center partagé entre plusieurs usages.

48 Une restriction sur Jean Zay empêche également la collecte des résultats pour les processus durant moins de 5 minutes : des périodes d'inactivité ("*sleep*") de cette durée ont donc ajoutées aux processus trop courts, et la consommation de cette inactivité a été mesurée

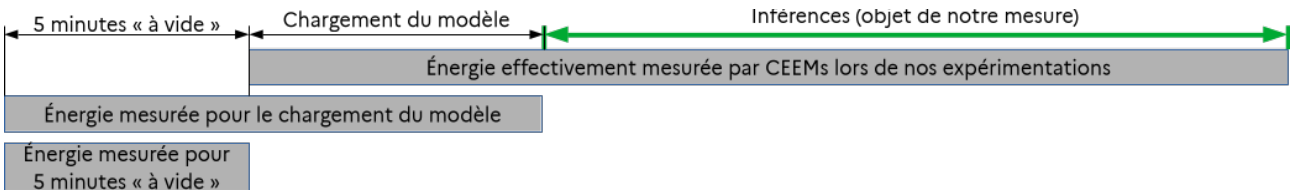

Figure 16: Illustration des différentes opérations effectuées pour obtenir la mesure des inférences, abstraction faite des autres effets

puis retranchée adéquatement.

Annexe 4 : Détail de la modélisation statistique de la consommation énergétique

Modèle utilisé

Les contraintes de la modélisation sont les suivantes :

- (C1) La consommation énergétique est une variable continue positive
- (C2) Tous les modèles n'ont pas été évalués sur tous les jeux de données
- (C3) L'effet principal observé semble être lié au nombre de paramètres (ou au nombre de paramètres activés) mais des effets « d'ordre 2 » sont également visibles
- (C4) Pour un même modèle, la consommation énergétique relative à la génération d'un sample d'un jeu de données peut varier selon la complexité du jeu de données

Ces différentes contraintes ont été prises en compte de la manière suivante :

- C1 : Modélisation de la distribution de la variable cible par une loi Gamma
- C3 : La variable cible utilisée n'est pas la consommation énergétique de l'inférence d'un modèle sur tout le jeu de données mais la consommation énergétique (en Wh) par paramètre (en milliards) et par tâche
- C2/C4 : Ajout d'effets mixtes dans le modèle permettant de modéliser l'effet d'un jeu de données (d'évaluation)

Le modèle choisi est un modèle linéaire généralisé à effet mixte (GLMM) avec distribution Gamma. Les effets mixtes permettent de capturer directement la spécificité des jeux de données via une constante multiplicative propre à chaque jeu d'évaluation.

Résultats

La formule correspondante à la modélisation finale est la suivante :

$$\log(\text{conso-by-parameter}) = (\alpha + \delta_{\text{dataset}}) + \beta_0 * \text{is-quantized} + (\beta_1 + \gamma_{\text{dataset}}) * \text{is-reasoning} + \beta_2 * \text{is-moe}$$

Le R^2 obtenu est de 0.65.

Les valeurs des paramètres sont indiquées dans le tableau ci-dessous. Nous indiquons dans le tableau également les valeurs brutes obtenus lors de la régression linéaire, pour la reproductibilité de l'expérience, ainsi que sa conversion en %, pour une interprétation plus aisée.

Lors de la régression, d'autres variables avaient été testées (influence de la multimodalité[49], effets mixtes du jeu d'évaluation (*dataset*, C2/C4) pour la quantification et l'architecture MoE) mais les résultats démontrant qu'elles ont une influence non significative, elles ne sont pas présentées. De même, les effets mixtes dépendent de chaque jeu d'évaluation, ce qui donne six valeurs pour chacun des facteurs δ et γ, que nous omettons pour simplifier le tableau.

49 Ce résultat est d'ailleurs cohérent avec l'observation du « 2.2.4. Les modèles consomment davantage lorsqu'ils raisonnent, pour un gain de qualité inégal selon les tâches »

Tableau 4: Résultat des valeurs obtenues par régression linéaire pour les paramètres principaux

	Moyenne (brut)	**Effet moyen (%)**	**Déviation standard**	**HDI @95% (brut)**	**HDI @95% (%)**
α	1.49	N/A	0.3	[0.88, 2.26]	N/A
β_0 (quantification)	-0.50	-39%	0.15	[-0.80, -0.20]	[-55%, -18%]
β_1 (raisonnement)	0.65	+92%	0.38	[-0.23, 1.5]	[-21%, 348%]
β_2 (moe)	-0.60	-45%	0.11	[-0.81, -0.37]	[-56%, -31%]

Lecture des paramètres :

En ignorant les effets mixtes des jeux d'évaluation, pour un modèle non quantifié, sans raisonnement ni MoE de 30B de paramètres, la consommation estimée se situera en moyenne autour de 0.13 Wh pour une requête :

$$conso_{30} = e^{1.49} \times 30$$

Pour un même modèle quantifié en 8 bits, la consommation estimée se situera en moyenne autour de 0.081 Wh :

$$conso_{30,FP8} = e^{1.49-0.5} \times 30 = e^{1.49} \times 30 \times e^{-0.5} = conso_{30} \times (-39\%)$$

Le Highest Density Interval (HDI) correspond à l'intervalle de confiance lorsque la distribution de probabilité n'est pas symétrique. Les HDI inqués ci-dessus correspondent à 95 % des résultats, ceux qui ont une probabilité de se trouver entre 2.5 % et 97.5 % des échantillons (voir Figure 17).

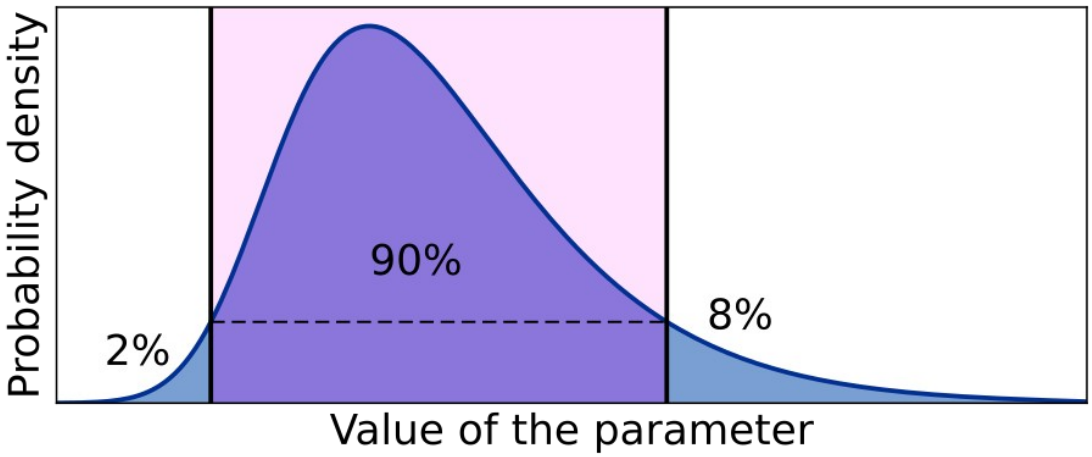

Figure 17: HDI à 90% d'une distribution de probabilité entre 2 et 8%. Source: https://en.wikipedia.org/wiki/Credible_interval