Actes de CORIA-TALN 2026
Nantes, France, 29 juin au 3 juillet 2026

Q-COMP : un jeu de données pour l'évaluation du traitement compositionnel de la quantification dans les grands modèles de langue

Shuyang Sun[1] Antoine Venant[1]
(1) Observatoire de linguistique Sens-Texte (OLST), Université de Montréal, Montréal, Canada
shuyang.sun@umontreal.ca, antoine.venant@umontreal.ca

RÉSUMÉ

Nous cherchons à déterminer si la manière dont les grands modèles de langue traitent les expressions quantifiées est cohérente avec les modèles de la sémantique formelle. Nous générons automatiquement des descriptions comportant un enchâssement récursif de quantifieurs, accompagnés de leur dénotation dans différents contextes. Nous constituons un jeu de données de 14 160 triplets description/contexte/interprétation, avec une profondeur d'enchâssement de 1 à 4, et évaluons les modèles sur trois tâches : (1) identifier tous les objets correspondant à la description, (2) en identifier un, et (3) vérifier la vérité d'une description. Les résultats montrent une forte baisse de performance lorsque le niveau d'enchâssement augmente, suggérant un traitement non compositionnel, ainsi que des difficultés propres aux quantificateurs *tous*, *aucun* et *plusieurs*.

ABSTRACT

Q-COMP : A Dataset for Evaluating the Compositional Treatment of Quantification in Large Language Models

We investigate whether large language models deal with quantified expressions in a way consistent with formal semantics. We automatically generate descriptions containing recursive nesting of quantifiers along with their denotation in different contexts. We construct a dataset of 14,160 description/context/interpretation triplets, with nesting depths ranging from 1 to 4, and evaluate the models on three tasks : (1) identifying all objects corresponding to the description, (2) identifying one such object, and (3) verifying the truth of a description relative to the context. The results show a strong decrease in performance as the level of nesting increases, suggesting non-compositional processing, as well as specific difficulties tied to the quantifiers *tous*, *aucun* and *plusieurs*.

MOTS-CLÉS : grands modèles de langue, quantification, sémantique, compositionnalité.

KEYWORDS: large language models, quantification, semantics, compositionality.

1 Introduction

Les grands modèles de langue (GML) ont produit des avancées considérables dans un large éventail de tâches linguistiques et cognitives (Zhao *et al.*, 2023). Malgré leurs performances impressionnantes sur des jeux d'évaluation généralistes (comme ceux de Wang *et al.*, 2018, 2019, 2022), leur degré de « compréhension » du langage – par opposition à celui auquel ils reproduiraient des motifs statistiques

Actes des 33ème Conférence sur le Traitement Automatique des Langues Naturelles.
Volume 1 : articles scientifiques originaux, pages 439–453.
Cette œuvre est mise à disposition sous licence Attribution 4.0 International.

contingents – reste débattu (Poibeau, 2025).

Bender & Koller (2020) argumentent, par des expériences de pensée, qu'un modèle n'ayant accès qu'aux « formes » devrait nécessairement afficher une compréhension et une capacité de généralisation limitées, parce qu'il ne dispose d'aucun moyen d'apprendre la correspondance entre ces formes et leurs dénotations. Cet argument informel est étayé, dans un cadre mathématique idéalisé, par la preuve de Merrill *et al.* (2021) que la relation d'équivalence sémantique d'une langue ne peut, en général, être simulée par une machine n'ayant observé qu'un nombre fini d'assertions portant sur cette relation. À l'inverse, des travaux comme y Arcas (2022); Piantadosi & Hill (2022); Pavlick (2023); Gubelmann (2024) contestent l'idée selon laquelle la « compréhension » doive nécessairement s'ancrer dans une perspective dénotationnelle (adoptant plutôt une perspective inférentialiste comme celle de Brandom, 1994), et soulignent la capacité remarquable des GMLs à capturer adéquatement de nombreuses *relations* de sens, et à *utiliser* le langage de manière appropriée.

Les tenants des perspectives dénotationnelle et inférentialiste de la compétence sémantique sont toutefois d'accord sur un certain nombre de ses manifestations observables (par exemple, que l'on puisse inférer que *Alice connaît Bob* à partir de la prémisse *Alice connaît chaque étudiant et Bob est étudiant*). Il apparaît donc intéressant de tester empiriquement la capacité des GMLs à reproduire de tels phénomènes, et à le faire de manière systématique. Le présent article s'inscrit dans une telle démarche, en se concentrant sur le phénomène de la quantification en français. La quantification est l'un des phénomènes les plus étudiés en sémantique formelle, elle intervient également dans l'analyse des constructions adverbiales (*toujours, jamais, en général, possiblement,* ...), des verbes modaux ou d'attitude (*pouvoir, croire*, ...), ainsi que de certains connecteurs discursifs (*donc, parce que,* ...).

Notre objectif principal est de tester si le traitement des quantifieurs par les GML est cohérent avec leur modélisation en sémantique formelle. Nous souhaitons enrichir les travaux existants sur le sujet 1) en isolant autant que possible les compétences grammaticales (logiques) des connaissances du monde et 2) en testant la capacité des modèles à généraliser compositionnellement leur traitement des syntagmes quantifiés simples, à des expressions complexes enchâssant ces syntagmes. Nous construisons pour ce faire un nouveau jeu de données synthétique composé de triplets $\langle \text{description}, \text{contexte}, \text{interprétation} \rangle$, où l'interprétation est calculée automatiquement à partir de l'expression et du contexte au moyen d'un modèle compositionnel symbolique [1]. Cette approche synthétique nous permet de séparer compétence logique et connaissance du monde en évaluant les modèles sur leur capacité à raisonner à partir de contextes arbitraires générés procéduralement.

Par exemple, le contexte présenté sur la figure 1 peut être associé à la description *triangle connecté à tous les losanges connectés à au moins un rond*, et à l'interprétation $\{6\}$. Plus précisément, dans ce contexte, le seul rond est l'objet 2. Les *losanges connectés à au moins un rond* sont donc les losanges connectés à l'objet 2. Les objets connectés à l'objet 2 sont l'objet 4 (qui est un losange), et l'objet 5 (qui est un triangle) : le seul losange pertinent est donc l'objet 4. La description *triangle connecté à tous les losanges connectés à au moins un rond* revient alors à chercher les triangles connectés à l'objet 4. Parmi les objets connectés à l'objet 4, on trouve l'objet 2 (qui est un rond), et l'objet 6 (qui est un triangle). L'unique triangle satisfaisant la description est donc l'objet 6, d'où l'interprétation $\{6\}$. Nous pouvons alors décrire le contexte à un modèle comme une succession de faits élémentaires (ou une image pour les modèles vision-langage, voir § 4.1 et § 4.3) et lui demander d'identifier tous les objets du contexte correspondant à la description (tâche 1), d'en identifier un (tâche 2), ou de

1. Bien que nous décrivions nos données en termes de sémantique dénotationnelle en § 3, les tâches présentées en § 4.3 peuvent être considérées comme inférentielles (à l'exception peut-être de celles impliquant des modèles vision-langage), et pourraient être définies de manière équivalente dans le cadre de la théorie de la preuve (*e.g.* Luo, 2012).

déterminer si un objet donné satisfait la description (tâche 3)[2].

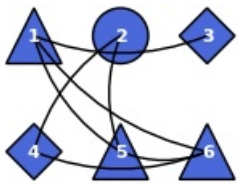

FIGURE 1 – Exemple de contexte

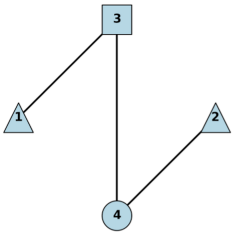

FIGURE 2 – Contexte jouet

Le reste de l'article est structuré comme suit. Le § 2 compare notre approche aux travaux connexes. Le § 3 présente les descriptions utilisées dans nos tâches et leur sémantique. Le § 4 décrit la méthodologie (génération des données, sous-ensembles et protocole expérimental). Le § 5 rapporte les résultats expérimentaux et leur analyse. Le § 6 présente les expériences de peaufinage. Le § 7 conclut.

2 Travaux connexes

Asher *et al.* (2023) argumentent d'un point de vue théorique une limitation intrinsèque des modèles de langue à apprendre de manière non supervisée la relation de conséquence logique d'un langage logique comprenant la quantification universelle. L'argument repose (non sans rappeler celui de Merrill *et al.*, 2021) sur la possibilité d'un domaine de quantification infini (et la consistance logique de la négation de la quantification universelle avec toute conjonction finie de prédications élémentaires). Nous nous concentrons sur une approche empirique restreinte à des cas de quantification sur des domaines finis.

Kalouli *et al.* (2022) montrent que plusieurs modèles de langue de type encodeur-décodeur font des prédictions similaires dans des phrases comportant des mots grammaticaux inconsistants (comprenant les quantifieurs anglais *all* et *no*), suggérant une tendance de ces modèles à reléguer la contribution sémantique de ces mots au second plan. Michaelov & Bergen (2023); Gupta (2023) montrent que les GMLs rencontrent globalement des difficultés avec des quantifieurs anglais tels que *few* et *most*, allant jusqu'à présenter une dégradation des performances lorsque la taille du modèle augmente dans le cas de *most*. Contrairement à notre approche, ces études ne cherchent pas à fixer le contexte d'évaluation mais reposent sur des connaissances générales (comme la relation entre *postmen* et *mail*).

Zhifei Wang & Steinert-Threlkeld (2023) proposent un jeu de données synthétique dans lequel des énoncés quantifiés sont associés à un contexte textuel. Cette méthode est très proche conceptuellement de la nôtre, mais se concentre davantage sur la diversité des quantifieurs testés et moins sur les capacités compositionnelles des modèles (15 quantifieurs différents sont testés dans des phrases ne comportant qu'un seul quantifieur portant sur un prédicat monadique).

Enfin, Kim & Linzen (2020); Li *et al.* (2023) développent un jeu de données explicitement conçu pour évaluer la généralisation compositionnelle. Toutefois, ils ne traitent pas de la quantification. En outre, une traduction vers un langage formel présenterait deux inconvénients vis-à-vis de notre objectif : elle introduirait dans l'évaluation les idiosyncrasies du langage formel cible, et elle dissocierait la forme logique de sa signification[3].

2. Notre code et données sont disponibles à https://github.com/EstelleYun/DatasetQCOMP
3. Par exemple, constater qu'un modèle est capable de produire la forme $\forall x\, P(x) \rightarrow Q(y)$ nous renseignerait peu sur sa

3 Descriptions et leur sémantique

Notre jeu de données comporte des descriptions telles que *triangle connecté à tous les carrés* ou *rond connecté à plusieurs losanges connectés à aucun carré*. Une **description** de niveau 0 est un prédicat simple (comme *triangle* ou *triangles*) tandis qu'une description de niveau $n+1$ est constituée d'un tel prédicat restreint par une relation quantifiée à un ensemble d'objets, récursivement spécifié par une description de niveau n. *carrés* constitue ainsi un exemple de description de niveau 0, et *triangle connecté à tous les carrés* une description de niveau 1. n correspond au nombre de quantifieurs enchâssés dans une description de niveau n.

Afin de préciser les notions de contexte et d'interprétation associées à ces descriptions, nous adoptons une approche formelle bien établie de la sémantique des syntagmes nominaux et de la modification adjectivale (Montague, 1970; Chierchia & McConnell-Ginet, 2000; Montague, 1973; Barwise & Cooper, 1981). Un **contexte** est un couple $c = \langle \mathcal{D}, [\![\cdot]\!]^c \rangle$, où $\mathcal{D}$ est un ensemble d'entités et $[\![\cdot]\!]^c$ associe aux prédicats unaires (dans notre cas *triangle*, *carré*, *rond* et *losange*) et binaires (*connecté*) respectivement des ensembles d'éléments et des relations sur $\mathcal{D}$. Par exemple, le contexte représenté graphiquement à la figure 2 est formellement décrit comme suit : $\mathcal{D} = \{o_1, o_2, o_3, o_4\}$, $[\![\text{triangle}]\!]^c = \{o_1, o_2\}$, $[\![\text{carré}]\!]^c = \{o_3\}$, $[\![\text{rond}]\!]^c = \{o_4\}$, $[\![\text{losange}]\!]^c = \emptyset$, $[\![\text{connecté}]\!]^c = \{\langle o_1, o_3\rangle, \langle o_3, o_1\rangle, \langle o_3, o_4\rangle, \langle o_4, o_3\rangle, \langle o_4, o_2\rangle, \langle o_2, o_4\rangle\}$.

Toutes les descriptions, quel que soit leur niveau d'enchâssement n, dénotent des ensembles d'entités. Par exemple, avec c comme ci-dessus, la description de niveau 0 *triangle* dénote l'ensemble $[\![\text{triangle}]\!]^c = \{o_1, o_2\}$, et l'intuition dicte que $[\![\text{triangle connecté à un carré}]\!]^c = \{o_1\}$.

L'interprétation des descriptions de niveau supérieur est modélisée de manière compositionnelle. Nous supposons qu'une description de niveau $n+1$ telle que *triangle connecté à* Q d_n (où Q est un quantifieur comme *un* ou *tous* et d_n est une description de niveau n, par exemple *carré* dans le cas $n = 0$) a une forme logique équivalente à $\lambda x\, \text{triangle}(x) \wedge Q(d_n)(\lambda y\, \text{connecté}(x, y))$ [4]. Suivant Barwise & Cooper (1981), les quantifieurs dénotent des relations entre ensembles d'entités. Par exemple, $[\![\text{un}]\!]^c(X)(Y)$ est vrai si $X \cap Y \neq \emptyset$, et $[\![\text{tous}]\!]^c(X)(Y)$ est vrai si $X \subseteq Y$ (quel que soit le contexte c). La description *triangle connecté à tous les* d_n dénote donc l'ensemble des objets qui sont des triangles et dont les voisins forment un sur-ensemble de la dénotation de d_n.

Cette sémantique compositionnelle modélise ainsi la dénotation de toute description, à partir de postulats sur la dénotation des quantifieurs (déterminants) utilisés. Dans ce travail, nous expérimentons les quantifieurs français *tous, au moins un, au moins deux, plusieurs, aucun*, en adoptant les postulats suivants (on suppose que *au moins deux* et *plusieurs* sont synonymes) :

$[\![\text{tous}]\!]^c(X)(Y) = 1$ ssi $X \subseteq Y$ $\quad$ $[\![\text{au moins } n]\!]^c(X)(Y) = 1$ ssi $|X \cap Y| \geq n$

$[\![\text{plusieurs}]\!]^c(X)(Y) = 1$ ssi $|X \cap Y| \geq 2$ $\quad$ $[\![\text{aucun}]\!]^c(X)(Y) = 1$ ssi $X \cap Y = \emptyset$

Une limite de ce cadre théorique est que la sémantique développée ci-avant décrit parfois seulement **l'une** des différentes interprétations possibles d'une description. Deux sources d'ambiguïté peuvent ainsi affecter nos descriptions :

1. Une ambiguïté syntaxique liée à l'attachement du syntagme participial. Pour obtenir l'interprétation attendue, l'attachement doit se faire sur le substantif précédant immédiatement le syntagme participial : *triangle connecté à au moins un* **[rond [connecté à deux carrés]]**. Mais un attachement au substantif de la principale est également possible : **[triangle** *[connecté à au*

compréhension de la signification du symbole $\forall$.

4. Nous référons le lecteur à Barendregt (1985) pour une introduction au lambda-calcul.

moins un rond] **[connecté à deux carrés]]**. Cette ambiguïté est toutefois limitée par l'accord (*triangle connecté à plusieurs ronds connectés à deux carrés* n'a pas cette ambiguïté).

2. Une ambiguïté liée aux lectures cumulative et distributive des quantifieurs. L'interprétation attendue suppose une lecture distributive, alors qu'une lecture cumulative est parfois possible : dans l'exemple de la figure 1, la description *triangle connecté à deux triangles connectés à au moins un rond* a une dénotation vide selon la lecture distributive attendue, mais pourrait être interprétée comme dénotant o1 dans une lecture cumulative (o1 est un triangle connecté aux triangles o5 et o6 qui **à eux deux** cumulent au moins une connexion vers le rond o2).

Que ce soit en raison de leur caractère plus marginal ou des exemples fournis en k-shot, nos expériences préliminaires n'ont pas montré de tendance des modèles à privilégier ces interprétations alternatives par rapport à l'interprétation attendue. Nous avons donc choisi, pour éviter d'alourdir la syntaxe des descriptions, d'expérimenter avec ce format. Nos analyses qualitatives d'erreur (voir section 5.2) n'ont pas non plus dégagé de tendance à privilégier ces interprétations alternatives.

4 Méthodologie

4.1 Génération des données

Notre jeu de données est généré de manière procédurale. Chaque instance se compose de trois éléments : (i) une description (avec son arbre syntaxique), (ii) un contexte comportant six objets, et (iii) une interprétation sémantique indiquant quels objets du contexte satisfont la description. Nous avons d'abord généré un grand nombre d'instances, puis filtré les données afin de respecter les contraintes d'équilibrage décrites au § 4.2.

Les descriptions, ainsi que leur arbre syntaxique, sont générées à l'aide d'une grammaire hors-contexte (CFG) écrite manuellement. Cette grammaire est conçue pour produire des descriptions de niveaux variant de 1 à 4. Les contextes sont des graphes non-orientés à six sommets étiquetés générés aléatoirement : pour chacun des 6 sommets, on choisit aléatoirement une étiquette dans l'ensemble {triangle, rond, losange, carré} et on le relie aléatoirement à 1 à 3 autres sommets (les boucles sont interdites). Pour adapter le jeu de données aux modèles multimodaux langage–vision, chaque contexte est converti et stocké à la fois sous forme textuelle et sous forme d'image. Pour chaque paire ⟨description, contexte⟩, l'ensemble des objets dénotés par la description est calculé automatiquement au moyen d'une implémentation de la sémantique compositionnelle décrite au § 3.

Le choix de fixer le nombre d'objets à six est un compromis méthodologique : un nombre d'objets trop faible, combiné à une génération aléatoire des descriptions, conduit trop fréquemment à des dénotations vides (c'est-à-dire qu'aucun objet du contexte ne satisfait la description). Dans certains sous-ensembles de données, nous devions obtenir, pour chaque contexte, neuf descriptions distinctes dont l'interprétation était non vide. Avec cinq objets dans chaque contexte, nous ne trouvions pas toujours un nombre suffisant de descriptions satisfaisant cette contrainte aux niveaux d'enchâssement élevés. À l'inverse, un nombre d'objets trop élevé alourdit considérablement la description textuelle des contextes, au risque de mettre davantage l'accent sur la capacité de mémoire ou d'attention des modèles que sur leur compétence sémantique.

Les instances sont enfin converties en différents (fragments de) prompts correspondant aux trois tâches : identifier tous les objets satisfaisant la description (tâche 1), identifier un objet satisfaisant la

description (tâche 2), vérifier si un objet donné satisfait la description (tâche 3). Le tableau 1 présente la description d'un contexte, ainsi que deux exemples.

Contexte	Il y a 6 objets : o1, o2, o3, o4, o5, o6. o1 est un rond. o2 est un rond. o3 est un triangle. o4 est un carré. o5 est un carré. o6 est un carré. o1 est connecté à o4, o5, o6. o2 est connecté à o3, o4. o3 est connecté à o2, o5, o6. o4 est connecté à o1, o2, o5, o6. o5 est connecté à o1, o3, o4, o6. o6 est connecté à o1, o3, o4, o5.
Description (tâche 1/2)	Triangle connecté à aucun rond.
Dénotation	$\emptyset$
Question (tâche 3)	Est-ce que o5 est un carré connecté à au moins un rond connecté à plusieurs carrés ?
Réponse attendue	Oui

TABLE 1 – Deux exemples partageant un contexte, respectivement utilisés pour les tâches 1/2 et 3

4.2 Sous-ensembles et statistiques

Jeu de données	**Nombre d'exemples**
Enchâssement (standard)	480
Enchâssement (difficile)	480
Quantificateurs (standard)	600
Quantificateurs (difficile)	600
Peaufinage (standard)	6000
Peaufinage (difficile)	6000
Total	**14 160**

TABLE 2 – Répartition des six sous-ensembles de données

Jeux par profondeur, quantifieur et fine-tuning : Nous avons généré six sous-ensembles de données, présentés dans le tableau 2. Les sous-ensembles *Enchâssement* contiennent des descriptions de niveau 1 à 4, avec 120 exemples par niveau. Les sous-ensembles *Quantificateurs*, contiennent 120 exemples de profondeur 1 pour chaque quantifieur. Dans les sous-ensembles *peaufinage*, l'ensemble d'entraînement contient des descriptions de niveau 1 à 4, avec 1200 exemples par niveau (soit 4800 au total), tandis que l'ensemble de test consiste en un ensemble distinct de contextes et de descriptions, avec 300 exemples par profondeur (soit 1200 au total).

Versions standard et difficile : Chaque type de sous-ensemble se décline en deux versions "standard" et "difficile". La version difficile exclut les cas où la dénotation du prédicat principal est exactement identique à l'ensemble des réponses attendues. Par exemple, lorsque la description est *rond connecté à tous les carrés connectés à aucun losange*, et que la restriction est triviale dans le contexte considéré, de sorte que tous les ronds satisfont la description. Cette mesure vise à empêcher qu'une heuristique simple consistant à ne considérer que le prédicat principal permette d'obtenir de bonnes performances.

Distribution des réponses : Nous avons imposé un ratio de 3/1 entre réponses non vides et réponses vides dans les six sous-ensembles. Ce choix découle de la logique de génération des questions dans la tâche de vérification, où les modèles doivent déterminer si un objet spécifique o satisfait une description d (voir le second exemple dans le tableau 1). On distingue deux types d'instances négatives : (a) lorsque d a une dénotation vide, et (b) lorsque d a une dénotation non vide mais que o ne satisfait pas d. L'objectif était d'assurer un équilibre entre instances positives et négatives, et que les instances négatives soient équilibrées entre les cas (a) et (b). On obtient ainsi 50 % d'instances positives (donc non vides), 25 % de réponses négatives vides et 25 % de réponses négatives non vides.

4.3 Expériences

Nous avons conçu deux catégories d'expériences : la première vise à mesurer l'impact de la profondeur d'enchâssement des quantifieurs à l'aide des sous-ensembles *Enchâssement*, la seconde à évaluer l'effet du type de quantifieur à l'aide des sous-ensembles *Quantificateurs*. Chaque catégorie comporte une expérience pour chacune des trois tâches (*Tous*, *Un* et *Vérification*) et pour chaque version du jeu de données (*standard*, *difficile*). Nous avons également contrôlé des variantes de prompts et comparé les configurations suivantes :

`k=1|2 :` le prompt inclut un exemple (non vide) pour la tâche | inclut deux exemples (un vide et un non vide) pour la tâche.
`+|-reasoning :` le prompt demande un raisonnement détaillé étape par étape | demande une réponse directe.

Par exemple, la configuration `k=1 +reasoning` correspond au type de prompt suivant pour la tâche 1 (*Tous*) : *Votre tâche est de résoudre le problème suivant. Je vous donnerai une description et un contexte. Vous devez renvoyer la liste des identifiants d'objets du contexte qui satisfont la description. Si aucun objet ne la satisfait, répondez* ***none****. La réponse finale doit apparaître exactement une fois entre ** et **, par exemple* ***[o1,o5]*** *ou* ***none****. Voici un exemple : <description, contexte et réponse>. IMPORTANT : avant de donner la réponse finale, effectuez un raisonnement logique en expliquant les étapes de votre analyse.*

Modèles testés : Nous avons évalué quatre LLM textuels et un modèle langage–vision. Les quatre modèles textuels sont Qwen2.5-14B-Instruct et Meta-LLaMA-3-8B-Instruct, choisis comme représentants populaires de modèles à poids ouverts de taille (relativement) modérée, DeepSeek-R1-Distill-Qwen-7B comme représentant d'un modèle de raisonnement obtenu par distillation à partir d'un modèle de très grande taille, et GPT-4o un modèle industriel de très grande taille (non reproductible). Le modèle langage–vision testé est Qwen2.5-VL-7B-Instruct. Par souci de concision, nous désignerons désormais Qwen2.5-14B-Instruct par Qwen2.5-14B, Meta-LLaMA-3-8B-Instruct par LLaMA3-8B, DeepSeek-R1-Distill-Qwen-7B par DeepSeek-R1-Distill-Qwen et Qwen2.5-VL-7B-Instruct par Qwen-VL.

Détails d'implémentation : Toutes les expériences ont été réalisées sur un seul ordinateur équipé de deux GPU RTX 4090 (avec une quantification sur 8 bits pour Qwen2.5-14B-Instruct), à l'aide de la bibliothèque Hugging Face, à l'exception de GPT-4o pour lequel nous avons utilisé l'API OpenAI. Chaque modèle a été exécuté avec quatre initialisations aléatoires (seeds) différentes, et nous rapportons l'exactitude moyenne sur ces quatre exécutions.

5 Résultats expérimentaux et analyse

5.1 Synthèse des résultats expérimentaux

Sauf mention explicite, nos observations valent pour l'ensemble des configurations de prompting, mais pour des raisons d'espace, nous rapportons principalement l'exactitude des modèles, évalués dans la configuration `k=2 +reasoning`.

La profondeur d'enchâssement réduit les performances : La plupart des modèles présentent une baisse significative d'exactitude lorsque la profondeur d'enchâssement augmente, ce qui suggère

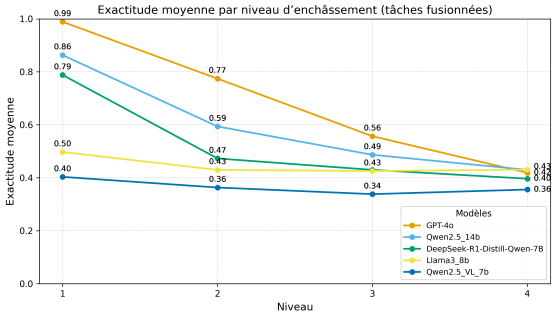

FIGURE 3 – Exactitude moyenne par profondeur d'enchâssement, ensemble standard, moyenne sur les trois tâches, `k=2 +reasoning`.

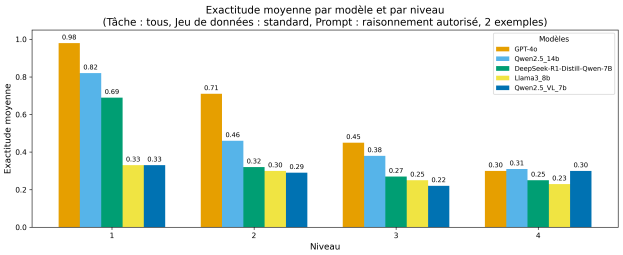

FIGURE 4 – Exactitude moyenne par profondeur d'enchâssement pour la tâche 1, ensemble standard, `k=2 +reasoning`.

que l'interprétation des GMLs s'écarte des modèles symboliques compositionnels aux niveaux d'enchâssement élevés (voir figure 3). La figure 4 propose un examen plus détaillé de la tâche 1. Même le modèle le plus performant, GPT-4o, passe de 0,98 à la profondeur 1 à 0,30 à la profondeur 4. À chaque niveau supplémentaire, l'exactitude diminue de 0,15 à 0,27.

Les quantifieurs ne sont pas équivalents : *tous*, *aucun* et *plusieurs* semblent systématiquement plus difficiles que *au moins un* ou *au moins deux*. En examinant de plus près la tâche 1 (sous-ensemble Quantifieurs, version standard, figure 5), GPT-4o obtient des performances presque parfaites, tandis que le deuxième meilleur modèle, Qwen2.5-14B, n'atteint que 66 % d'exactitude sur *tous*, soit environ 30 % de moins que pour les autres quantifieurs. DeepSeek-R1-Distill-Qwen tombe sous les 60 % pour *tous*, *aucun* et *plusieurs*. Les schémas d'erreurs récurrents sont analysés au § 5.2.

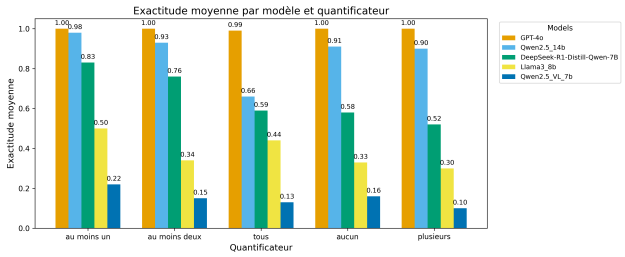

FIGURE 5 – Exactitude moyenne par modèle et quantificateur (tâche 1, ensemble standard, `k=2 +reasoning`).

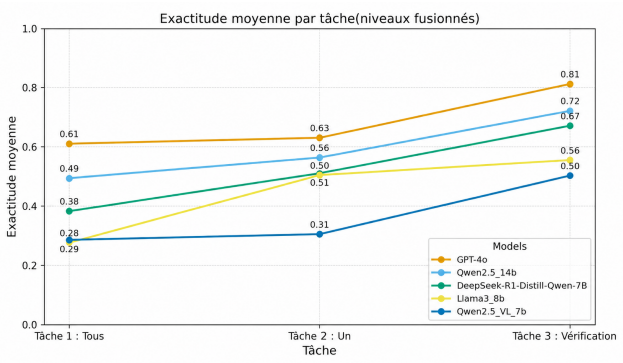

FIGURE 6 – Exactitude moyenne sur les trois tâches (enchâssements fusionnés, `k=2 +reasoning`).

Les tâches diffèrent en difficulté : Comme le montre la figure 6, la tâche 1 (trouver tous) est la plus difficile, suivie de la tâche 2 (trouver un), tandis que la tâche 3 (vérification) est la plus facile, conformément à l'intuition.

Performances globales des modèles : En termes d'exactitude globale : GPT-4o > Qwen2.5-14B > DeepSeek-R1-Distill-Qwen > LLaMA3-8B > Qwen-VL. À taille de modèle comparable, DeepSeek-R1-Distill-Qwen surpasse LLaMA3-8B et Qwen-VL (figure 6). De plus, Qwen-VL (multimodal) ne montre aucun avantage décisif par rapport aux modèles textuels de taille similaire. Son exactitude est souvent inférieure à celle de LLaMA3-8B et DeepSeek-R1-Distill-Qwen, et sa variance est plus

élevée. La présentation des contextes sous forme d'images semble donc poser un défi supplémentaire.

Standard vs difficile : L'ensemble "difficile", qui ne comporte pas de cas de restrictions triviales, où se concentrer uniquement sur le prédicat principal (*carré*, *triangle*, ...) et ignorer la restriction conduit à la bonne réponse, ne semble pas mériter son nom. (Pour une explication plus détaillée des versions standard et difficile du jeu de données, voir la section 4.2.) Les performances sont très proches entre les ensembles standard et difficile, et souvent meilleures sur le second (table 3). Cela suggère que la plupart des modèles n'ont pas une tendance marquée à ignorer la restriction, à l'exception de LLaMA3-8B qui accuse la plus grande baisse de performance entre les versions standard et difficile.

Modèle	Standard	Difficile
DeepSeek-R1 Distill Qwen (R)	0.51	0.53
GPT-4o (R)	0.68	0.75
LLaMA3-8B	0.44	0.30
LLaMA3-8B (R)	0.45	0.38
Qwen2.5-14B	0.43	0.44
Qwen2.5-14B (R)	0.59	0.62
Qwen-VL (R)	0.37	0.34

TABLE 3 – Exactitude moyenne des modèles sur les jeux de données standard et difficile dans l'expérience d'enchâssement. Les modèles (R) correspondent à la version `+reasoning`.

Impact de la stratégie de prompt : L'ajout d'un exemple négatif dans le prompt n'apporte qu'un faible gain d'exactitude (1–3 points de pourcentage). En revanche, l'exigence d'un raisonnement étape par étape a un effet nettement plus marqué : l'exactitude de Qwen2.5-14B augmente d'environ 16 points, et celle de LLaMA3-8B d'environ 4 points.

5.2 Analyse

Pour approfondir l'analyse des erreurs, nous examinons les réponses de GPT-4o, le modèle globalement le plus performant. Malgré une exactitude quasi parfaite au niveau 1, son score en tâche 1 chute nettement au niveau 2 (98 $\rightarrow$ 71%), ce qui suggère qu'il semble avoir acquis le sens des quantifieurs évalués, sans pour autant avoir développé un traitement compositionnel.

5.2.1 Erreurs systématiques associées à « aucun + $\emptyset$ » La majorité des 41 erreurs du modèle concernent des combinaisons contenant *aucun* (voir table 4)[5]. Les erreurs se concentrent dans la

Combinaison	Total	Erreurs	Part des erreurs
aucun | *au moins deux*	43	24	58.5%
aucun | *tous*	10	7	17.1%
aucun | *aucun*	10	8	19.5%
tous | *au moins un*	3	1	2.4%
plusieurs | *au moins deux*	5	1	2.4%
aucun | *au moins un*	10	0	0.0%

TABLE 4 – Distribution des erreurs de GPT-4o au niveau 2 (tâche 1, `k=2 +reasoning`) pour certaines combinaisons de quantificateurs.

Combinaison (« aucun + $\emptyset$ »)	Cas	Erreurs	Taux d'erreur
aucun | *au moins deux*	40	23	57.5%
aucun | *tous*	8	5	62.5%
aucun | *aucun*	10	8	80.0%
Taux d'erreur global	58	36	62.1%

TABLE 5 – Performance de GPT-4o sur les configurations « aucun + $\emptyset$ » (niveau 2, tâche 1, `k=2 +reasoning`).

5. Gpt-4o est toutefois toujours correct pour *aucun* | *au moins un*, ce qui indique que la difficulté ne tient pas uniquement au quantificateur *aucun* lui-même.

configuration ***aucun* + ∅** (table 5), dans les contextes où le restricteur de *aucun* a une dénotation vide (par exemple, *rond connecté à aucun triangle* dans un contexte sans triangles). Au lieu de considérer que tous les candidats satisfont la condition, GPT-4o répond souvent « aucun », interprétant une absence de « contre-exemple » comme une absence de « solution ». En moyenne, ce schéma entraîne un taux d'erreur supérieur à 60 %, atteignant 80 % pour *aucun* | *aucun*. Dans les autres modèles également, les configurations *« aucun + ∅ »* concentrent la majorité des erreurs (voir tableau 7 en annexe). En tâche *vérification* au niveau 2, Qwen2.5-14B commet 27 erreurs, dont 19 (70 %) relèvent de ce type, et DeepSeek-R1-Distill-Qwen 40 erreurs, dont 33 (82 %) de ce type.
Exemple 1 (GPT-4o, « aucun + ∅ »)
Description : *losange connecté à aucun carré connecté à au moins deux losanges*
Contexte : aucun carré dans la scène.
Interprétation attendue : [o2, o3] (tous les losanges, puisque aucun ne peut être disqualifié, tous satisfont la condition).
Réponse (extrait) : . . . Dans le contexte, il n'y a aucun carré parmi les objets (seulement des losanges, des triangles et un rond). Par conséquent, aucun losange ne peut être connecté à un carré . . . Ainsi, puisqu'aucun losange ne satisfait la condition décrite dans l'énoncé, la réponse est **aucun**.

5.2.2 Autres types d'erreurs

Comme GPT-4o atteint une exactitude parfaite sur le jeu "Quantificateurs", nous analysons les deuxième et troisième meilleurs modèles, Qwen2.5-14B et DeepSeek-R1-Distill-Qwen. Des exemples de chacune des catégories d'erreurs discutées ci-après sont présentés en annexe C.

(1) Pour *tous*, l'exactitude de Qwen2.5-14B atteint 66 % en tâche 1 et 67 % en tâche 2, tandis que DeepSeek-R1-Distill-Qwen atteint respectivement 59 % et 65 %. Pour Qwen2.5-14B, 44 % des erreurs en tâche 1 découlent d'une lecture « sur-pragmatisée » de *tous* présupposant l'existence de deux objets dans le restricteur. 27 % des erreurs de DeepSeek-R1-Distill-Qwen en tâche 1 sont du même type, et 31 % proviennent d'hallucinations dans le raisonnement.
Exemple 2 (Qwen2.5-14B sur-pragmatisation tous → plusieurs)
Description : *losange connecté à tous les triangles*
Contexte : un seul triangle (o6) ; un losange (o3) y est connecté.
Interprétation attendue : [o3].
Réponse (extrait) : . . . Les losanges sont o2, o3, o4 et o5. Aucun n'est connecté à tous les triangles (seuls o3 et o4 sont connectés à o6). Conclusion : aucun.
(2) Pour *plusieurs*, l'exactitude de DeepSeek-R1-Distill-Qwen en tâche 1 tombe à 52 %. Parmi ses erreurs, 43 % résultent d'une inversion des rôles (retour des objets auxquels la cible est connectée) et 23 % d'une interprétation de *plusieurs* comme « au moins un ».
(3) Pour *aucun* en tâche 1, Qwen2.5-14B atteint 91 % d'exactitude, contre 58 % pour DeepSeek-R1-Distill-Qwen ; 54 % de ses erreurs coïncident avec un raisonnement confondant *aucun* et *un*, notamment lorsque l'ensemble cible est vide (« aucun + ∅ »).

6 Peaufinage

Nous avons utilisé les ensembles « peaufinage » pour peaufiner les modèles Qwen2.5-14B et LLaMA-3-8B *via* QLoRA (quantification 4 bits + LoRA, configuration `-reasoning`), en les entraînant sur les profondeurs 1 et 2 (2400 exemples) et en évaluant, hors domaine, sur les niveaux 3 et 4 des ensembles de test correspondants (600 exemples). L'objectif était d'évaluer si un tel peaufinage

renforce le traitement compositionnel.

Modèle	Niveau 3 gain(1–2)	Niveau 4 gain(1–2)
Qwen2.5-14B	0.28→0.41 (+0.13)	0.23→0.40 (+0.17)
LLaMA-3-8B	0.02→0.18 (+0.16)	0.01→0.34 (+0.33)

TABLE 6 – Gains d'exactitude en tâche 1 aux niveaux profonds avant/après peaufinage (entraînement 1–2, test 3–4, ensemble difficile, `k=2 -reasoning`).

Le peaufinage améliore l'exactitude des modèles aux niveaux 3 et 4 bien que ceux-ci n'aient pas servi au peaufinage. La table 6 examplifie ce point à travers le gain d'exactitude pour la tâche 1 (version "difficile") (les résultats détaillés sont dans l'annexe B). Ces résultats vont dans le sens d'une meilleure généralisation compositionnelle, mais sont à prendre avec précaution. Le peaufinage pourrait avoir orienté les modèles vers des biais présents également aux niveaux profonds.

7 Conclusion et limites

Nos expériences montrent que, si plusieurs des GMLs testés semblent maîtriser le sens des quantificateurs employés dans des phrases simples, aucun ne semble capable de généralisation compositionnelle robuste sur des phrases à quantificateurs enchâssés : l'exactitude diminue systématiquement avec la profondeur d'enchâssement, en particulier pour la tâche 1 (trouver tous les référents). Les erreurs fréquentes incluent l'interprétation de *aucun* + $\emptyset$ comme contradictoire plutôt que tautologique, ainsi qu'une surgénéralisation d'inférences pragmatiques dans un contexte où celles-ci ne paraissent pas nécessairement fondées (la lecture de *tous* comme « ≥ 2 »), ainsi que des hallucinations.

Notre étude présente toutefois certaines limites. Notre évaluation ne prend pas quantitativement en compte les ambiguïtés discutées en section 3. Pour améliorer ce point et confirmer nos conclusions, nous avons préparé une version du jeu de données désambiguïsée à l'aide des marqueurs *lui-même* et *chacun* (par exemple, *carré connecté à plusieurs ronds qui sont chacun connectés à au moins un triangle qui est lui-même connecté à deux losanges*). Nous sommes présentement en train d'obtenir les résultats expérimentaux pour cette version du jeu de données. Notre étude se limite par ailleurs à un ensemble plutôt restreint de quantifieurs ; l'inclusion de quantifieurs non exprimables au premier ordre, constituerait une extension intéressante.

8 Remerciements

Nous remercions les relecteurs anonymes de TALN pour leurs commentaires et suggestions. La présente recherche a été financée par le Conseil de Recherche en Sciences Naturelles et en Génie du Canada (RN001462).

Références

ASHER N., BHAR S., CHATURVEDI A., HUNTER J. & PAUL S. (2023). Limits for learning with language models. *arXiv preprint arXiv :2306.12213*.

BARENDREGT H. P. (1985). *The lambda calculus - its syntax and semantics*, volume 103 de *Studies in logic and the foundations of mathematics*. North-Holland.

BARWISE J. & COOPER R. (1981). Generalized quantifiers and natural language. In *Philosophy, language, and artificial intelligence : Resources for processing natural language*, p. 241–301. Springer.

BENDER E. M. & KOLLER A. (2020). Climbing towards nlu : On meaning, form, and understanding in the age of data. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, p. 5185–5198.

BRANDOM R. B. (1994). *Making It Explicit : Reasoning, Representing, and Discursive Commitment*. Harvard University Press.

CHIERCHIA G. & MCCONNELL-GINET S. (2000). *Meaning and grammar : An introduction to semantics*. MIT press Cambridge, MA.

GUBELMANN R. (2024). Pragmatic norms are all you need – why the symbol grounding problem does not apply to LLMs. In Y. AL-ONAIZAN, M. BANSAL & Y.-N. CHEN, Éds., *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, p. 11663–11678, Miami, Florida, USA : Association for Computational Linguistics. DOI : 10.18653/v1/2024.emnlp-main.651.

GUPTA A. (2023). Probing quantifier comprehension in large language models : Another example of inverse scaling. *arXiv preprint arXiv :2306.07384*.

KALOULI A.-L., SEVASTJANOVA R., BECK C. & ROMERO M. (2022). Negation, coordination, and quantifiers in contextualized language models. In N. CALZOLARI, C.-R. HUANG, H. KIM, J. PUSTEJOVSKY, L. WANNER, K.-S. CHOI, P.-M. RYU, H.-H. CHEN, L. DONATELLI, H. JI, S. KUROHASHI, P. PAGGIO, N. XUE, S. KIM, Y. HAHM, Z. HE, T. K. LEE, E. SANTUS, F. BOND & S.-H. NA, Éds., *Proceedings of the 29th International Conference on Computational Linguistics*, p. 3074–3085, Gyeongju, Republic of Korea : International Committee on Computational Linguistics.

KIM N. & LINZEN T. (2020). COGS : A compositional generalization challenge based on semantic interpretation. In B. WEBBER, T. COHN, Y. HE & Y. LIU, Éds., *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, p. 9087–9105, Online : Association for Computational Linguistics. DOI : 10.18653/v1/2020.emnlp-main.731.

LI B., DONATELLI L., KOLLER A., LINZEN T., YAO Y. & KIM N. (2023). SLOG : A structural generalization benchmark for semantic parsing. In H. BOUAMOR, J. PINO & K. BALI, Éds., *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, p. 3213–3232, Singapore : Association for Computational Linguistics. DOI : 10.18653/v1/2023.emnlp-main.194.

LUO Z. (2012). Formal semantics in modern type theories with coercive subtyping. *Linguistics and Philosophy*, **35**(6), 491–513. DOI : 10.1007/s10988-013-9126-4.

MERRILL W., GOLDBERG Y., SCHWARTZ R. & SMITH N. A. (2021). Provable limitations of acquiring meaning from ungrounded form : What will future language models understand ? *Transactions of the Association for Computational Linguistics*, **9**, 1047–1060. DOI : 10.1162/tacl_a_00412.

MICHAELOV J. A. & BERGEN B. K. (2023). Rarely a problem ? language models exhibit inverse scaling in their predictions following few-type quantifiers. In A. ROGERS, J. BOYD-GRABER & N. OKAZAKI, Éds., *Findings of the Association for Computational Linguistics :*

ACL 2023, p. 14162–14174, Toronto, Canada : Association for Computational Linguistics. DOI : 10.18653/v1/2023.findings-acl.891.

MONTAGUE R. (1970). Universal grammar, theoria 36 ; przedruk w thomason, r.(red.)(1974) formal philosophy, selected papers of richard montague.

MONTAGUE R. (1973). The proper treatment of quantification in ordinary English. In K. J. J. HINTIKKA, J. MORAVCSIC & P. SUPPES, Éds., *Approaches to Natural Language*, p. 221–242. Dordrecht : Reidel.

PAVLICK E. (2023). Symbols and grounding in large language models. *Philosophical Transactions of the Royal Society A : Mathematical, Physical and Engineering Sciences*, **381**(2251), 20220041. DOI : 10.1098/rsta.2022.0041.

PIANTADOSI S. & HILL F. (2022). Meaning without reference in large language models. In *NeurIPS 2022 Workshop on Neuro Causal and Symbolic AI (nCSI)*.

POIBEAU T. (2025). *Understanding Conversational AI*. London : Ubiquity Press. DOI : 10.5334/bde.

WANG A., PRUKSACHATKUN Y., NANGIA N., SINGH A., MICHAEL J., HILL F., LEVY O. & BOWMAN S. (2019). Superglue : A stickier benchmark for general-purpose language understanding systems. In H. WALLACH, H. LAROCHELLE, A. BEYGELZIMER, F. D'ALCHÉ-BUC, E. FOX & R. GARNETT, Éds., *Advances in Neural Information Processing Systems*, volume 32 : Curran Associates, Inc.

WANG A., SINGH A., MICHAEL J., HILL F., LEVY O. & BOWMAN S. (2018). GLUE : A multi-task benchmark and analysis platform for natural language understanding. In T. LINZEN, G. CHRUPAŁA & A. ALISHAHI, Éds., *Proceedings of the 2018 EMNLP Workshop BlackboxNLP : Analyzing and Interpreting Neural Networks for NLP*, p. 353–355, Brussels, Belgium : Association for Computational Linguistics. DOI : 10.18653/v1/W18-5446.

WANG Y., MISHRA S., ALIPOORMOLABASHI P., KORDI Y., MIRZAEI A., NAIK A., ASHOK A., DHANASEKARAN A. S., ARUNKUMAR A., STAP D., PATHAK E., KARAMANOLAKIS G., LAI H., PUROHIT I., MONDAL I., ANDERSON J., KUZNIA K., DOSHI K., PAL K. K., PATEL M., MORADSHAHI M., PARMAR M., PUROHIT M., VARSHNEY N., KAZA P. R., VERMA P., PURI R. S., KARIA R., DOSHI S., SAMPAT S. K., MISHRA S., REDDY A S., PATRO S., DIXIT T. & SHEN X. (2022). Super-NaturalInstructions : Generalization via declarative instructions on 1600+ NLP tasks. In Y. GOLDBERG, Z. KOZAREVA & Y. ZHANG, Éds., *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, p. 5085–5109, Abu Dhabi, United Arab Emirates : Association for Computational Linguistics. DOI : 10.18653/v1/2022.emnlp-main.340.

Y ARCAS B. A. (2022). Do large language models understand us ? *Daedalus*, **151**(2), 183–197.

ZHAO W. X., ZHOU K., LI J., TANG T., WANG X., HOU Y., MIN Y., ZHANG B., ZHANG J., DONG Z., DU Y., YANG C., CHEN Y., CHEN Z., JIANG J., REN R., LI Y., TANG X., LIU Z., LIU P., NIE J.-Y. & WEN J.-R. (2023). A survey of large language models. *arXiv preprint arXiv :2303.18223*.

ZHIFEI WANG L. & STEINERT-THRELKELD S. (2023). GQG : Generalized quantifier generalization - a dataset for evaluating quantifier semantics understanding in language models. In D. HUPKES, V. DANKERS, K. BATSUREN, K. SINHA, A. KAZEMNEJAD, C. CHRISTODOULOPOULOS, R. COTTERELL & E. BRUNI, Éds., *Proceedings of the 1st GenBench Workshop on (Benchmarking) Generalisation in NLP*, p. 185–192, Singapore : Association for Computational Linguistics. DOI : 10.18653/v1/2023.genbench-1.15.

A Résultats supplémentaires sur les structures « aucun + ∅ »

Modèle	Erreurs	*aucun*+∅	Proportion
Qwen2.5-14B	27	19	70 %
DS-R1-Distill-Qwen	40	33	82 %

TABLE 7 – Erreurs sur les structures « aucun+∅ » en tâche *vérification* (niveau 2, `k=2 +reasoning`).

B Résultats détaillés du peaufinage

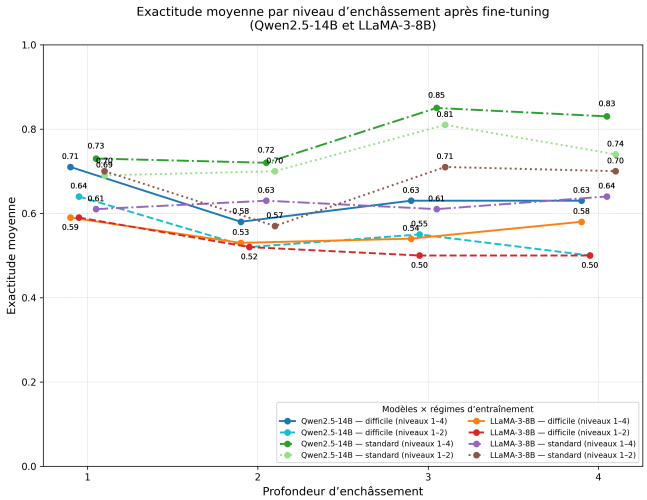

FIGURE 7 – Exactitude moyenne selon la profondeur d'enchâssement après peaufinage (toutes tâches confondues, Qwen2.5-14B et LLaMA-3-8B, données d'entraînement standard vs difficiles, peaufinage sur les niveaux 1–2 ou sur tous les niveaux (1–4).

Modèle	Niveau 3 gain(1–2)	Niveau 4 gain(1–2)
Qwen2.5-14B	0.28→0.41 (+0.13)	0.23→0.40 (+0.17)
LLaMA-3-8B	0.02→0.18 (+0.16)	0.01→0.34 (+0.33)

TABLE 8 – Gains d'exactitude en tâche 1 aux niveaux profonds avant/après peaufinage (entraînement 1–2, test 3–4, ensemble difficile, `k=2 -reasoning`).

Modèle	Niveau 3 gain(1–2)	Niveau 4 gain(1–2)
Qwen2.5-14B	0.37→0.56 (+0.19)	0.35→0.45 (+0.10)
LLaMA-3-8B	0.23→0.55 (+0.32)	0.25→0.51 (+0.26)

TABLE 9 – Gains d'exactitude en tâche 2 aux niveaux profonds avant/après peaufinage (entraînement 1–2, test 3–4, ensemble difficile, `k=2 -reasoning`).

Modèle	Niveau 3 gain(1–2)	Niveau 4 gain(1–2)
Qwen2.5-14B	0.57→0.68 (+0.11)	0.51→0.66 (+0.15)
LLaMA-3-8B	0.50→0.64 (+0.14)	0.50→0.64 (+0.14)

TABLE 10 – Gains d'exactitude en tâche 3 aux niveaux profonds avant/après peaufinage (entraînement 1–2, test 3–4, ensemble difficile, `k=2 -reasoning`).

Modèle	Niveau 3 gain(1–4)	Niveau 4 gain(1–4)
Qwen2.5-14B	0.28→0.51 (+0.23)	0.23→0.46 (+0.23)
LLaMA-3-8B	0.02→0.37 (+0.35)	0.01→0.43 (+0.42)

TABLE 11 – Gains d'exactitude en tâche 1 aux profondeurs élevées avant vs après peaufinage (entraînement 1–4, test 1–4, jeu de données difficile, `k=2 -reasoning`).

Modèle	Niveau 3 gain(1–4)	Niveau 4 gain(1–4)
Qwen2.5-14B	0.25→0.85 (+0.60)	0.23→0.74 (+0.51)
LLaMA-3-8B	0.20→0.57 (+0.37)	0.20→0.67 (+0.47)

TABLE 12 – Gains d'exactitude en tâche 1 aux profondeurs élevées avant vs après peaufinage (entraînement 1–4, test 1–4, jeu de données standard, `k=2 -reasoning`).

Modèle	Niveau 3 gain(1–2)	Niveau 4 gain(1–2)
Qwen2.5-14B	0.25→0.78 (+0.53)	0.23→0.74 (+0.51)
LLaMA-3-8B	0.20→0.62 (+0.42)	0.20→0.67 (+0.47)

TABLE 13 – Gains d'exactitude en tâche 1 aux niveaux profonds avant/après peaufinage (entraînement 1–2, test 1–4, jeu de données standard, `k=2 -reasoning`).

C Autres exemples d'erreurs

Exemple 3 (DeepSeek, inversion des rôles).
Description : *losange connecté à plusieurs triangles*
Contexte : o6 est un losange ; o2 et o3 sont des triangles.
Interprétation attendue : [o6].
Réponse (extrait) : . . . Il faut trouver les objets connectés au losange qui sont des triangles. . . Donc les objets sont o2 et o3. Conclusion : [o2, o3].

Exemple 4 (DeepSeek, confusion $\geq$1).
Description : *losange connecté à plusieurs triangles*
Contexte : un seul triangle (o6).
Interprétation attendue : $\emptyset$.
Réponse (extrait) : . . . o2 est connecté à o6 ; o3 est connecté à o6. . . Donc les losanges connectés à des triangles sont o2 et o3. Conclusion : [o2, o3].