JOURNAL INTERNATIONAL DE TECHNOLOGIE, DE L'INNOVATION,
DE LA PHYSIQUE, DE L'ENERGIE ET DE L'ENVIRONNEMENT

Spécialisation d'un modèle de langage pour la génération de gammes opératoires industrielles multi-étapes : injection d'un référentiel d'équipements par fine-tuning sur données structurées

L. E. Brunet

ISSN : 2428-8500

DOI : 10.52497/jitipee.v10i1.416

Journal International de Technologie, de l'Innovation, de la Physique, de l'Energie et de l'Environnement
ISSN : 2428-8500 vol. 10, n°1, 2, 2026
DOI : http://dx.doi.org/10.52497/jitipee.v10i1.416

Spécialisation d'un modèle de langage pour la génération de gammes opératoires industrielles multi-étapes : injection d'un référentiel d'équipements par fine-tuning sur données structurées

Luc E. Brunet[(1)]

[(1)] R&D Mediation

luc.brunet@insead.edu

Résumé *– La synthèse automatique de procédés industriels par modèles génératifs a jusqu'ici porté sur la topologie des flowsheets (enchaînement d'opérations unitaires), sans sélection d'équipements dans un référentiel nommé ni spécification des conditions opératoires. Nous posons la question suivante : un modèle de langage ouvert de petite taille (8 milliards de paramètres) peut-il internaliser un référentiel encyclopédique de 444 familles d'équipements de procédés et générer, à partir d'une simple spécification entrée/sortie, une gamme opératoire conceptuelle — enchaînement ordonné de familles d'équipements identifiées par leur référence, avec ordres de grandeur des conditions opératoires ? Nous comparons empiriquement trois stratégies d'injection de connaissances : la distillation de contexte dans les poids (prompt baking), le fine-tuning supervisé en texte libre et le fine-tuning supervisé sur données JSON structurées, par adaptation de rang faible (LoRA) — ainsi qu'un baseline sans fine-tuning avec le référentiel complet en contexte. Les résultats montrent (i) que le prompt baking échoue au-delà de quelques milliers de tokens (effondrement génératif sur un prompt de 59 203 tokens), (ii) qu'à modèle de base constant, le fine-tuning en texte libre ne restitue aucune référence du référentiel (0/10) tandis que le fine-tuning JSON produit des sorties valides avec références exactes (5/5) — contraste attribuable au format sous réserve d'une ablation à contenu strictement constant, les deux datasets différant aussi par leur composition —, (iii) qu'un dataset de 166 exemples suffit, en 40 minutes d'entraînement sur une station de travail personnelle, pour obtenir un modèle générant des gammes multi-étapes conformes aux pratiques industrielles documentées sur les cas vus en entraînement, (iv) que les baselines sans fine-tuning atteignent l'exactitude référentielle par recopie du catalogue (31/32 avec le catalogue seul, 33/36 en ajoutant des exemples few-shot) mais peinent sur la validité industrielle des enchaînements (0/5 puis ~1/5), et (v) qu'hors du domaine du corpus (15 procédés puis 3 cas extrêmes) le modèle fine-tuné présente la dissociation inverse — enchaînements plausibles mais ancrage référentiel effondré (2/91 puis 0/19). Cette double dissociation — exactitude sans grammaire, grammaire sans exactitude — montre que exactitude référentielle et validité des enchaînements sont deux compétences séparables, acquises par des mécanismes distincts. Nous discutons les limites de l'étude — conformité industrielle appréciée qualitativement, réentraînement sans fuite à mener, variables non contrôlées — et positionnons l'approche par rapport aux générateurs de flowsheets à base de transformers et aux agents LLM en génie des procédés.*

Mots clés *: modèle de langage, fine-tuning, LoRA, synthèse de procédés, génie des procédés, génération structurée, injection de connaissances*

DOI : *10.52497/jitipee.v10i1.416*

ISSN : 2428-8500 vol. 10, n°1, 2, 2026
DOI : http://dx.doi.org/10.52497/jitipee.v10i1.416

Introduction

La conception d'un procédé industriel consiste à sélectionner et enchaîner des opérations unitaires (broyage, séparation, séchage, réaction chimique, etc.) de manière à transformer une matière première en un produit fini répondant à des spécifications données [1, 2]. Cette tâche de synthèse de procédés (*process synthesis*) est traditionnellement abordée soit par des procédures heuristiques hiérarchiques [3], soit par optimisation sur superstructure [4], deux familles d'approches qui requièrent respectivement une expertise humaine approfondie et une formulation mathématique explicite de l'espace des solutions.

Les modèles de langage de grande taille (LLM) ont ouvert une troisième voie, fondée sur les données : l'apprentissage de la structure des procédés à partir de corpus [5]. La communauté du génie des procédés assistée par ordinateur (*process systems engineering*, PSE) identifie l'adaptation de ces modèles de fondation au domaine comme une frontière de recherche ouverte [6], et les premiers générateurs de flowsheets par transformers ont démontré la faisabilité de l'auto-complétion de topologies de procédés [7]. Cependant, ces approches génèrent des enchaînements de *types* d'opérations unitaires : elles ne sélectionnent pas d'équipements dans un référentiel réel, ne produisent pas de conditions opératoires, et ne sont pas interrogeables par une spécification fonctionnelle (« transformer X en Y »).

Ce travail étudie la question de recherche suivante : un LLM ouvert de taille modeste (8 Md de paramètres) peut-il internaliser, par fine-tuning, un référentiel encyclopédique d'équipements de procédés, au point de générer des gammes opératoires multi-étapes — familles d'équipements référencées, conditions opératoires indicatives — à partir d'une spécification entrée/sortie ? Elle se décline en trois sous-questions :

- Q1 (stratégie d'injection) : parmi la distillation de contexte dans les poids (*prompt baking* [8]), le fine-tuning supervisé (SFT) en texte libre et le SFT sur données structurées, laquelle permet le transfert effectif des connaissances du référentiel ?
- Q2 (rôle du format) : à modèle de base constant, le format de représentation des données d'entraînement (texte libre vs JSON structuré) conditionne-t-il la capacité du modèle à citer les références exactes du catalogue ?
- Q3 (frugalité) : quel volume de données et de calcul est nécessaire — l'approche est-elle praticable sur une station de travail personnelle ?

Les contributions sont les suivantes : (i) une méthodologie de dérivation d'un dataset d'instruction structuré à partir d'un référentiel encyclopédique d'équipements (444 machines, 24 catégories) ; (ii) une comparaison empirique de trois stratégies d'injection de connaissances, dont un résultat négatif documenté sur le prompt baking appliqué à un prompt de 59 203 tokens ; (iii) la mise en évidence du rôle déterminant du format structuré dans le transfert de références exactes ; (iv) un pipeline complet de spécialisation par LoRA [9] exécutable en 40 minutes sur matériel grand public. L'ensemble du protocole, des hyperparamètres et des données est décrit pour permettre sa reproduction.

ISSN : 2428-8500 vol. 10, n°1, 2, 2026
DOI : http://dx.doi.org/10.52497/jitipee.v10i1.416

1. État de l'art

1.1. Synthèse de procédés : approches systématiques

La synthèse de procédés dispose d'un corpus méthodologique établi depuis les années 1980 : la procédure de décision hiérarchique de Douglas [3], codifiée dans son ouvrage de référence [1], décompose la conception en niveaux successifs (mode batch/continu, structure entrée-sortie, réseau de recyclage, séparations, intégration énergétique). Les approches par optimisation, systématisées par Biegler, Grossmann et Westerberg [2], formulent la synthèse comme un problème d'optimisation mixte (MINLP) sur une superstructure énumérant les alternatives [4]. Ces deux familles présentent des limites duales : les heuristiques dépendent de l'expert, les superstructures exigent une énumération explicite préalable de l'espace des solutions. Les approches génératives fondées sur les données, dont ce travail relève, visent à apprendre implicitement cet espace à partir de corpus de procédés existants.

1.2. Modèles génératifs pour les flowsheets

Le groupe de Schweidtmann (TU Delft) a établi la faisabilité de la génération de flowsheets par modèles de langage. La notation SFILES 2.0 [10] sérialise un flowsheet en chaîne de caractères (à la manière des SMILES en chimie), rendant les topologies de procédés traitables par des modèles de séquence. Sur cette représentation, Vogel et al. [7] ont entraîné un transformer causal de type GPT à l'auto-complétion de flowsheets (pré-entraînement sur topologies synthétiques puis fine-tuning sur flowsheets réels), la rareté des données réelles étant partiellement compensée par augmentation des chaînes SFILES [11]. La même représentation a permis de formuler la génération de structures de contrôle (passage PFD → P&ID) comme une traduction séquence-à-séquence, atteignant 74,8 à 89,2 % de précision top-5 [12], puis l'autocorrection de flowsheets erronés (80 % top-1) [13] et la prédiction de structures de contrôle à partir de graphes [14]. En parallèle, des agents d'apprentissage par renforcement construisant des flowsheets séquentiellement ont été étudiés [15], au prix d'un simulateur fournissant la récompense.

Ces travaux partagent une limite structurelle au regard de notre objectif : ils manipulent des *types* d'opérations unitaires (colonnes, échangeurs, réacteurs génériques) et leur connectivité, mais ne sélectionnent pas d'équipements dans un référentiel nommé, ne produisent pas de conditions opératoires, et prennent en entrée un flowsheet partiel plutôt qu'une spécification fonctionnelle. Leur représentation capture en revanche la topologie (branchements, recyclages), que notre format séquentiel ne représente pas (§ 4.5).

1.3. LLM généralistes et agents en chimie et génie des procédés

Une seconde ligne de travaux mobilise des LLM généralistes de frontière, augmentés d'outils. Coscientist [16] (GPT-4) planifie et exécute des synthèses chimiques ; ChemCrow [17] couple GPT-4 à 18 outils de chimie computationnelle ; la revue de Ramos et al. [18] dresse le panorama de ces agents et souligne l'absence de benchmarks établis. Dans le domaine du génie des procédés, l'évaluation empirique de Schulze Balhorn et al. [19] montre que les réponses des LLM généralistes se dégradent avec la complexité des questions d'ingénierie, et la perspective de Decardi-Nelson et al. [6] identifie la spécialisation de modèles au domaine PSE comme un

ISSN : 2428-8500 vol. 10, n°1, 2, 2026
DOI : http://dx.doi.org/10.52497/jitipee.v10i1.416

verrou. Ces approches agentiques reposent sur des modèles fermés interrogés par API, avec injection des connaissances au moment de l'inférence (outils, contexte) ; notre travail explore la voie complémentaire d'un modèle ouvert, de petite taille, dont les poids internalisent le référentiel métier.

1.4. Spécialisation de modèles : fine-tuning, RAG et modèles de domaine

La spécialisation d'un LLM peut passer par l'apprentissage en contexte [5], la génération augmentée par récupération (RAG) [23] ou le fine-tuning. Le fine-tuning d'instruction [21] apprend au modèle à suivre un format de tâche, et LIMA [22] a montré qu'un millier d'exemples soigneusement choisis suffit à transférer format et style — résultat qui fonde notre hypothèse de frugalité (Q3). Les méthodes à paramètres efficients, LoRA [9] et sa variante quantifiée QLoRA [20], rendent cette spécialisation praticable sur matériel modeste en n'entraînant qu'une fraction des paramètres, ce qui limite en outre l'oubli catastrophique des capacités générales [25]. La comparaison systématique de Ovadia et al. [24] nuance toutefois l'intérêt du fine-tuning pour l'injection de connaissances *factuelles*, où le RAG est souvent supérieur ; nous revenons sur cette tension en discussion (§ 4.4), notre tâche relevant de la génération structurée de conceptions plutôt que de la restitution factuelle.
En chimie et sciences des matériaux, plusieurs modèles de domaine valident la voie du fine-tuning de petits modèles : ChemLLM [26] (7 Md) construit ses données d'instruction en convertissant des bases de données chimiques structurées en dialogues — stratégie analogue à notre dérivation encyclopédie → instructions ; LlaSMol[27] (Mistral-7B + LoRA, 0,58 % des paramètres) dépasse GPT-4 sur 14 tâches de chimie ; MechGPT [28] distille un corpus de mécanique des matériaux en paires question-réponse pour le fine-tuning. Aucun de ces modèles ne traite la synthèse de procédés industriels.

1.5. Génération structurée

La production de sorties structurées (JSON) par un LLM peut être obtenue au décodage — génération guidée par automate ou grammaire [30, 31], qui garantit la validité syntaxique sans entraînement — ou apprise par fine-tuning, à l'instar de Gorilla [32] où un LLaMA fine-tuné dépasse GPT-4 en exactitude d'appels d'API structurés, références comprises. Tam et al.[29] observent par ailleurs que les contraintes de format imposées au décodage peuvent dégrader les capacités de raisonnement d'un modèle généraliste. Notre approche apprend conjointement le format et le contenu dans les poids : la contrainte structurelle est internalisée à l'entraînement plutôt qu'imposée à l'inférence, et nous observons qu'elle peut alors jouer un rôle positif de tuteur du transfert de connaissances (Q2).

1.6. Distillation de contexte et prompt baking

Une alternative au SFT pour internaliser des connaissances est la distillation de contexte [33, 34] : entraîner le modèle sans prompt à imiter ses propres distributions avec prompt. Le prompt baking [8] en est la formulation récente, qui convertit un prompt en mise à jour de poids par minimisation de divergence KL ; il a été validé sur des prompts d'instruction et de personnalité de quelques milliers de tokens. Son comportement sur des corpus factuels volumineux (plusieurs dizaines de milliers de tokens) n'était pas documenté ; nous apportons sur ce point un résultat négatif (§ 3.2).

ISSN : 2428-8500 vol. 10, n°1, 2, 2026
DOI : http://dx.doi.org/10.52497/jitipee.v10i1.416

1.7. Distillation de contexte et prompt baking

Au croisement de ces lignes, le verrou que nous adressons est le suivant : les générateurs de flowsheets [7, 10-14] produisent des topologies sans équipements ni conditions ; les agents LLM [16-18] dépendent de modèles fermés et d'outils à l'inférence ; les modèles de domaine fine-tunés [26-28] n'ont pas abordé la synthèse de procédés. À notre connaissance, la génération de gammes opératoires référencées — sélection dans un référentiel clos de familles d'équipements, avec identifiants et conditions indicatives, à partir d'une spécification fonctionnelle — par un LLM ouvert spécialisé n'a pas été étudiée. C'est l'objet de ce travail.

2. Matériel et méthodes

2.1. Référentiel des machines de procédés

Le référentiel est une encyclopédie structurée en JSON de 444 machines et équipements de procédés industriels, organisée en 24 catégories (Tableau 1). Chaque machine est décrite par un identifiant unique (ex : 1.07), un nom bilingue (français/anglais), une sous-catégorie, ses entrées, ses sorties, ses industries d'application et ses principaux fournisseurs. Les identifiants sont propres au référentiel — il s'agit de familles génériques d'équipements, non de références de modèles constructeurs — et l'ontologie mêle des niveaux de granularité différents (opérations unitaires, familles d'équipements, architectures de réacteurs), hétérogénéité discutée en § 4.5. Les sources incluent le *Perry's Chemical Engineers' Handbook* [35], le *Coulson & Richardson's Chemical Engineering* [36], la documentation technique de constructeurs et des publications spécialisées.

#	Catégorie	Machines	Exemples
1	Réduction de taille	28	Concasseur à mâchoires, broyeur à boulets, microniseur
2	Augmentation de taille	15	Granulateur, presse à comprimer, tour de prilling
3	Mélange / Agitation	20	Mélangeur à rubans, agitateur Rushton, mélangeur statique
4	Séparation mécanique	45	Filtre-presse, centrifugeuse, hydrocyclone, flottation
5	Séparation thermique	38	Distillation, évaporation, extraction, chromatographie
6	Transfert de chaleur	34	Échangeur à plaques, four rotatif, condenseur
7	Séchage	19	Spray dryer, lit fluidisé, lyophilisateur
8	Réaction chimique	37	Réacteur batch, CSTR, lit fixe, bioréacteur
9–24	Autres	208	Formage, enrobage, membrane, électrochimie, plasma, etc.
	Total	**444**	

Tableau 1 : Catégories principales du référentiel et nombre de machines par catégorie.

2.2. Construction du dataset d'entraînement

Le dataset est construit selon un format JSON structuré entrée/sortie.

ISSN : 2428-8500 vol. 10, n°1, 2, 2026
DOI : http://dx.doi.org/10.52497/jitipee.v10i1.416

Format d'entrée (message utilisateur) :

```
{"entree": "Betteraves sucrières entières",
 "sortie": "Sucre cristallisé blanc (saccharose > 99.7%)"}
```

Format de sortie (réponse du modèle) :

```
{
  "procede": "Production de sucre cristallisé de betterave",
  "nb_etapes": 8,
  "etapes": [
    {"n": 1, "machine_id": "1.11", "machine": "Broyeur à couteaux",
     "operation": "Découpe des betteraves en cossettes",
     "entree": "Betteraves lavées",
     "sortie": "Cossettes (lanières 2-3 mm)",
     "conditions": {}},
    {"n": 2, "machine_id": "5.16",
     "machine": "Extracteur solide-liquide",
     "operation": "Diffusion : extraction du saccharose",
     "entree": "Cossettes + eau chaude",
     "sortie": "Jus de diffusion (15 Brix)",
     "conditions": {"temperature": "70-75°C",
                    "duree": "60-90 min"}},
    ...
  ]
}
```

Le dataset comprend 166 exemples répartis en deux types (Figure 1) :

- **Procédés multi-étapes** (72 exemples) : 24 procédés industriels issus de la littérature technique [35, 36], chacun décliné en 3 variantes de formulation d'entrée (formulation de base ; ajout d'un champ de contraintes précisant le contexte industriel ; formulation abrégée). Les procédés couvrent 10 secteurs industriels : mines (concentration de minerai de cuivre, de fer), chimie (acide sulfurique, ammoniac, biodiesel), agroalimentaire (sucre, lait en poudre, chocolat, frites surgelées), pharma (comprimés, anticorps monoclonaux), traitement des eaux (STEP, ZLD), polymères (PET, recyclage), énergie (hydrogène, biométhane) et métallurgie (aluminium).
- **Catalogues par opération unitaire** (94 exemples) : pour chaque sous-catégorie comptant au moins 2 machines, un catalogue structuré listant les options disponibles avec leurs entrées/sorties et industries.

Le dataset est scindé en 149 exemples d'entraînement et 17 de validation (90/10, graine aléatoire fixée).

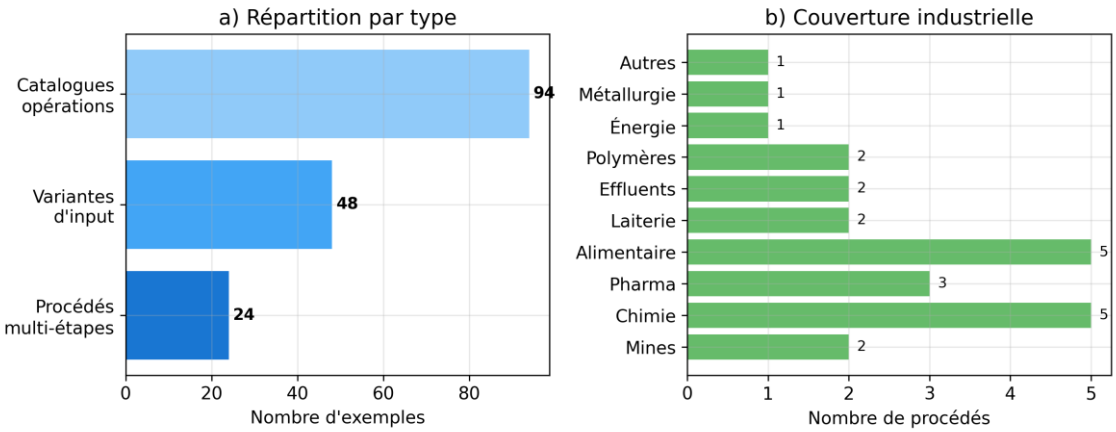

Figure 1 : Composition du dataset d'entraînement.

2.3. Modèle de base et fine-tuning

Le modèle de base est Qwen3-8B [37] dans une variante « ablitérée » publiquement disponible (`Josiefied-Qwen3-8B-abliterated-v1`), dont la direction de refus a été supprimée dans l'espace résiduel [38]. Ce choix est d'ordre pratique : il élimine les refus parasites sur des requêtes techniques (certains procédés chimiques déclenchant les garde-fous du modèle aligné). La tâche elle-même ne présentant pas de caractère sensible, l'incidence de cette variante sur les résultats est discutée comme variable non contrôlée en § 4.5.

Le fine-tuning est réalisé par LoRA [9] sur le framework MLX [39] optimisé pour Apple Silicon, avec masquage du prompt (la perte n'est calculée que sur la réponse du modèle). Les hyperparamètres sont présentés dans le Tableau 2.

Paramètre	Valeur
Méthode	LoRA
Paramètres entraînables	10,8 M (0,12 % du modèle)
Couches fine-tunées	16 / 36
Learning rate	1×10^{-5} (constant)
Itérations	2000
Batch size	1
Gradient accumulation	4
Max sequence length	1536 tokens
Gradient checkpointing	Oui
Mask prompt	Oui

Tableau 2 : Hyperparamètres de fine-tuning.

ISSN : 2428-8500 vol. 10, n°1, 2, 2026
DOI : http://dx.doi.org/10.52497/jitipee.v10i1.416

2.4. Protocole expérimental

Dans ce paragraphe, le protocole expérimental est décrit.

Stratégies comparées (Q1, Q2).

Trois stratégies d'injection de connaissances ont été évaluées successivement :

- Prompt baking[8] : injection du référentiel complet (59 203 tokens, tokenizer Qwen3.5) dans les poids par distillation KL entre un modèle *teacher* (avec prompt) et un modèle *student* (sans prompt). Testé sur Qwen3.5-text-9B, 2000 itérations.
- SFT texte libre : fine-tuning LoRA sur 633 exemples en texte libre (444 fiches machines + 94 chaînes de procédés + 95 exemples de raisonnement). Testé sur Qwen3.5-text-9B.
- SFT JSON structuré : fine-tuning LoRA sur les 166 exemples du § 2.2. Testé d'abord sur Qwen3.5-text-9B (même modèle de base que les stratégies 1 et 2), puis sur Qwen3-8B ablitéré.

La comparaison des stratégies 2 et 3 sur le même modèle de base (Qwen3.5-text-9B) fournit le contraste principal pour Q2 (rôle du format à modèle constant) ; le passage à Qwen3-8B évalue la sensibilité au modèle de base et l'efficacité computationnelle. On note que les datasets des stratégies 2 et 3 diffèrent en taille et en composition, le format conditionnant la nature des exemples constructibles ; cette limite du protocole est discutée en § 4.5.

Baselines sans fine-tuning.

Pour mesurer ce que le fine-tuning apporte réellement, le modèle de base Qwen3-8B (version alignée d'origine, bf16, décodage glouton) est évalué sur les mêmes cas de test dans trois conditions : (A) *zero-shot*, le schéma JSON cible étant décrit dans le prompt système mais sans accès au référentiel ; (B) *catalogue en contexte*, le référentiel complet des 444 machines étant sérialisé sous forme compacte (identifiant, nom, sous-catégorie, entrées, sorties — 19 243 tokens, tokenizer Qwen3-8B) et fourni dans le prompt système ; (C) *catalogue* + *few-shot*, identique à (B) mais avec deux démonstrations complètes de gammes issues du corpus (chocolat, biodiesel — hors des cinq cas de test) placées avant la requête, afin que le modèle dispose non seulement du vocabulaire mais aussi d'exemples de la logique d'enchaînement. Les conditions (B) et (C) relèvent d'un prompting à contexte long (fourniture intégrale, sans étape de récupération), non d'un RAG au sens strict ; (C) répond à l'objection selon laquelle (B) reçoit le vocabulaire sans jamais voir de gamme complète. Pour ces baselines, l'exactitude des références est mesurée par une métrique automatique stricte : un identifiant est compté correct si, et seulement si, il existe dans le référentiel *et* que le nom de machine généré correspond au nom référencé pour cet identifiant.

ISSN : 2428-8500 vol. 10, n°1, 2, 2026
DOI : http://dx.doi.org/10.52497/jitipee.v10i1.416

Métriques.

Chaque procédé généré est évalué selon : (i) *validité JSON* — la sortie est-elle un objet JSON syntaxiquement analysable ; (ii) *exactitude des références* — proportion des identifiants machines cités qui existent dans le référentiel et correspondent à l'opération décrite ; (iii) *présence des conditions opératoires* — proportion d'étapes portant des conditions non vides ; (iv) *conformité du procédé* — appréciation qualitative de l'enchaînement au regard des pratiques industrielles documentées [35, 36].

Jeux de test.

Cinq spécifications entrée/sortie pour la comparaison principale : trois vues à l'entraînement sous d'autres formulations (*in-distribution* : minerai de cuivre → concentré ; betteraves → sucre ; lait → poudre) et deux absentes du dataset (*hors-distribution proche* : eaux usées → eau traitée ; graines de tournesol → huile). S'y ajoutent deux jeux hors-distribution plus larges : un jeu *OOD étendu* de 15 procédés sans parenté avec les 24 du corpus, couvrant sept secteurs (agroalimentaire, boissons, hydrométallurgie, énergie, chimie, eau, matériaux — p. ex. raisin → vin, eau de mer → eau potable, pétrole brut → essence, spodumène → hydroxyde de lithium, coques de coco → charbon actif) ; et un jeu d'*extrapolation extrême* de 3 procédés particulièrement éloignés (batteries lithium-ion usagées → carbonate de lithium ; paille de blé → bioéthanol anhydre ; cartes électroniques broyées → or affiné). Tous sont générés dans les conditions d'inférence standard et évalués avec la métrique stricte.

3. Résultats

3.1. Convergence de l'entraînement

La Figure 2 présente les courbes de convergence du fine-tuning LoRA sur Qwen3-8B. La perte d'entraînement converge rapidement, passant de 1,29 à 0,03 en 600 itérations. La perte de validation atteint son minimum à l'itération 600 (0,351) puis remonte progressivement (0,501 à l'itération 2000), signature d'un surapprentissage. Le modèle final (itération 2000) restitue toutefois plus fidèlement les procédés vus en entraînement que le point de validation optimal, comportement attendu d'un régime de mémorisation d'un corpus expert restreint (cf. § 4.3).

L'entraînement complet nécessite 40 minutes sur un Mac Studio M4 Ultra (256 Go de mémoire unifiée), avec un pic de consommation mémoire de 18,6 Go — soit 7 % de la mémoire disponible (Q3).

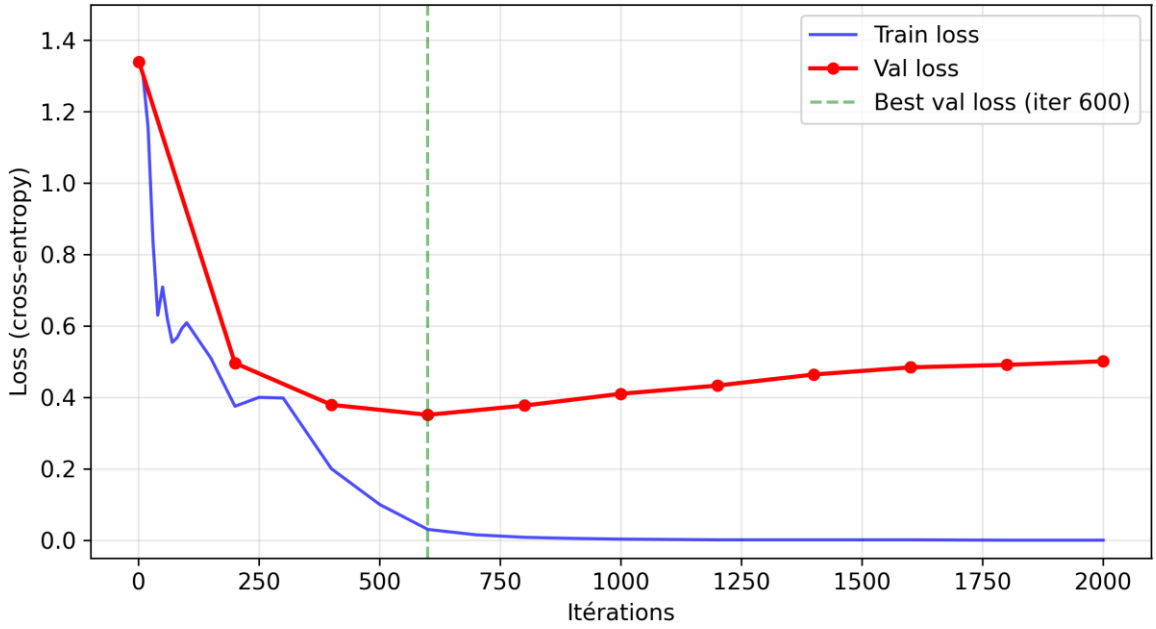

Figure 2 : Convergence du fine-tuning LoRA (Qwen3-8B).

3.2. Comparaison des stratégies d'injection

La Figure 3 et le Tableau 3 résument la comparaison entre les quatre configurations testées.

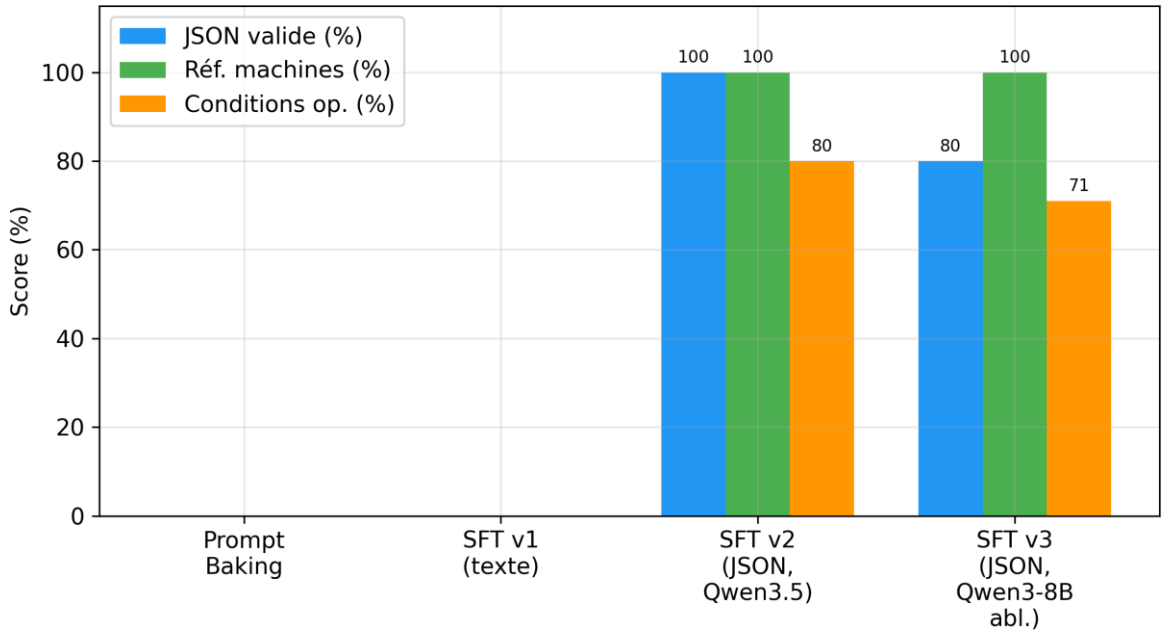

Figure 3 : Comparaison des approches de spécialisation.

Approche	Modèle	JSON valide	Réf. machines	Conditions	Durée
Prompt baking	Qwen3.5-9B	0/5	—	—	9 h
SFT texte libre	Qwen3.5-9B	—	0/10	Non	3 h
SFT JSON (Qwen3.5)	Qwen3.5-9B	5/5	5/5	Oui	3 h
SFT JSON (Qwen3-8B abl.)	**Qwen3-8B**	**4/5**	**4/4**	**Oui**	**0,7 h**

Tableau 3 : Évaluation comparative sur 5 procédés de test (3 in-distribution, 2 hors-distribution).

On peut noter que le prompt baking échoue complètement (Q1). En effet, le modèle produit des séquences dégénérées (caractères répétés). Le préfixe de 59 203 tokens excède d'un ordre de grandeur les prompts de quelques milliers de tokens sur lesquels l'algorithme a été validé [8] ; la convergence de la perte KL masque l'effondrement de la capacité générative du *student* en régime autorégressif.

À modèle de base constant (Qwen3.5-text-9B), le contraste entre les deux datasets est massif (Q2). On observe que le SFT en texte libre produit des réponses fluides et thématiquement pertinentes mais ne cite aucune référence du catalogue (0/10), le déséquilibre du dataset (444 fiches simples pour 94 chaînes de procédés) biaisant le modèle vers la restitution de fiches individuelles ; le SFT sur JSON structuré produit systématiquement du JSON valide portant les identifiants exacts du référentiel (5/5) et les conditions opératoires.

Le transfert de la même recette JSON sur Qwen3-8B ablitéré préserve la qualité (4/5 en validité JSON, le cas d'échec étant une troncature à la limite de tokens ; 4/4 en exactitude des références sur les sorties complètes) pour un coût d'entraînement divisé par 4 (§ 3.4).

3.3. Baseline sans fine-tuning

Le Tableau 4 présente les résultats du modèle de base (non fine-tuné) sur les mêmes 5 cas de test.

Conditions	JSON valide	IDs existants	IDs corrects	Étapes avec cond.	Procédés conformes
A — zero-shot	5/5	27/27	**0/27**	27/27	0/5
B — catalogue en contexte (19,2 K tokens)	5/5	32/32	**31/32**	32/32	**0/5**
C — catalogue + 2 exemples (few-shot)	5/5	36/36	**33/36**	31/36	**~1/5**
SFT JSON (§ 3.2, pour référence)	4/5	4/4	4/4	Oui	3/3 in-dist.

Tableau 4 : Baseline sans fine-tuning (Qwen3-8B aligné, décodage glouton) sur les 5 procédés de test. « IDs corrects » : identifiant existant dans le référentiel et nom de machine correspondant

On observe en zero-shot que le modèle de base produit du JSON valide au schéma demandé, mais ses références sont des hallucinations plausibles : les 27 identifiants cités existent par coïncidence dans la numérotation du référentiel, mais aucun (0/27) ne désigne la machine nommée. Ce résultat illustre la nécessité d'une métrique de correspondance stricte : la simple existence de l'identifiant surestime massivement l'exactitude référentielle.

ISSN : 2428-8500 vol. 10, n°1, 2, 2026
DOI : http://dx.doi.org/10.52497/jitipee.v10i1.416

Avec le catalogue en contexte, l'exactitude référentielle devient quasi parfaite (31/32) — le modèle recopie des entrées du catalogue — mais les procédés générés sont industriellement incohérents (0/5 conformes) : la chaîne du cuivre omet la flottation, étape essentielle du traitement des minerais sulfurés, au profit d'une centrifugation-décantation ; la chaîne du sucre broie les betteraves au broyeur à boulets puis enchaîne cinq broyages successifs du sucre, sans diffusion, épuration ni cristallisation ; l'écrémage du lait est confié à une « presse à briqueter » et la pasteurisation est absente ; le traitement d'eaux usées à forte charge organique (DBO 8 000 mg/L) ne comporte aucun traitement biologique. Les conditions opératoires, systématiquement présentes, sont des remplissages génériques (« ambiante, atmosphérique, 10-15 min ») là où le modèle fine-tuné produit des paramètres métier (P80, pH de flottation, degrés Brix).

L'ajout de deux exemples de gammes complètes (few-shot, condition C) modifie ce constat de façon instructive et corrige l'asymétrie du protocole : montré comment enchaîner, le modèle généraliste conserve l'exactitude référentielle (33/36) et, surtout, produit une chaîne du cuivre désormais correcte :

— concassage → broyage → classification → flottation à pH 9-10 → décantation —

La démonstration transfère donc une part de la grammaire, pas seulement le format. Mais la conformité industrielle reste faible (~1/5 sur notre appréciation) : le sucre est toujours broyé au broyeur à boulets puis « extrait » par une colonne de rectification ; le lait écrémé est broyé au broyeur à boulets ; le traitement d'eaux à DBO 8 000 mg/L reste dépourvu d'étape biologique et se termine par une « dépyrogénation » incongrue ; l'huile de tournesol est « raffinée » par osmose inverse. L'exactitude des identifiants (33/36) et la validité des enchaînements (~1/5) se dissocient nettement : le few-shot résout le vocabulaire et une partie de la grammaire, mais la grammaire industrielle robuste sur les procédés in-distribution demeure l'apport propre du fine-tuning (3/3).

Ces baselines suggèrent la nature de l'apport du fine-tuning : le catalogue en contexte fournit le *vocabulaire* (les équipements et leurs entrées/sorties), les exemples few-shot y ajoutent le *format* et une grammaire partielle, mais la *grammaire* industrielle complète (quelles séquences sont valides, avec quelles conditions métier) reste ce que le SFT sur chaînes complètes injecte dans les poids. L'avantage s'accompagne d'un bénéfice pratique : le modèle fine-tuné opère sans le prompt de 19,2 K tokens requis à chaque requête par les baselines B et C.

3.4. Analyse qualitative des procédés générés

La Figure 4 illustre le procédé généré pour la concentration du minerai de cuivre, détaillé au Tableau 5.

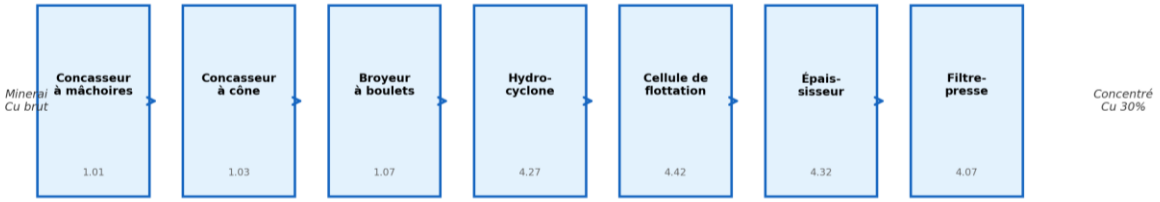

Figure 4 : Exemple de procédé généré : concentration du minerai de cuivre.

Étape	Machine (ID)	Opération	Conditions
1	Concasseur à mâchoires (1.01)	Concassage primaire	ouverture 150-300 mm
2	Concasseur à cône (1.03)	Concassage secondaire	—
3	Broyeur à boulets (1.07)	Broyage fin	P80 75-150 µm, milieu humide
4	Hydrocyclone (4.27)	Classification	d50 75 µm
5	Cellule de flottation (4.42)	Flottation sélective	pH 9-11, xanthate
6	Épaississeur (4.32)	Épaississement du concentré	—
7	Filtre-presse (4.07)	Déshydratation	pression 10-15 bar

Tableau 5 : Détail du procédé généré pour la concentration du minerai de cuivre (in-distribution).

La séquence concassage → broyage → classification → flottation → épaississement → filtration correspond au schéma standard de traitement du minerai sulfuré de cuivre décrit dans la littérature [35,36], et les conditions opératoires générées (P80 de 75-150 µm, pH de flottation 9-11, pression de filtration 10-15 bar) sont cohérentes avec les plages documentées.

Procédé	Dist.	Étapes	IDs machines	Cond.	Évaluation
Cuivre → concentré	In	7	1.01, 1.03, 1.07, 4.27, 4.42, 4.32, 4.07	5/7	Chaîne identique au schéma standard, conditions réalistes
Betteraves → sucre	In	8	1.11, 5.16, 4.30, 5.10, 5.12, 13.03, 4.25, 7.04	6/8	8 étapes identiques au corpus, avec conditions
Lait → poudre	In	6	12.03, 4.24, 11.01, 5.10, 7.03, 2.02	6/6	72°C/15s ; séchage : air d'entrée 180-200°C — conforme au corpus
Eaux usées → eau traitée	Hors	10	4.01, 4.33, 3.08, 8.14, …	6/10	Plausible : DAF + anaérobie + MBR
Tournesol → huile	Hors	—	—	—	JSON tronqué (max tokens)

Tableau 6 : Résultats détaillés sur les 5 procédés de test.

Sur les cas hors-distribution proches, le modèle compose des enchaînements plausibles à partir des motifs appris (le procédé de traitement des eaux combine flottation à air dissous, digestion anaérobie et bioréacteur à membrane de manière cohérente), mais avec des identifiants machines parfois inexacts et une sensibilité à la longueur maximale de génération.

ISSN : 2428-8500 vol. 10, n°1, 2, 2026
DOI : http://dx.doi.org/10.52497/jitipee.v10i1.416

Généralisation hors-distribution (15 procédés).

Sur les 15 procédés OOD sans parenté avec le corpus, le modèle produit une sortie JSON complète dans 12 cas sur 15 (les 3 échecs sont des troncatures à la limite de tokens, non des JSON invalides). Sur les 91 étapes des sorties complètes, 87 (96 %) portent un identifiant existant, mais seulement 2 sur 91 (2 %) satisfont la correspondance stricte identifiant ↔ nom, et 69 % portent des conditions. La capacité à structurer une chaîne cohérente et à la garnir de conditions se maintient hors distribution, mais l'ancrage dans le référentiel — l'association apprise entre une opération et l'identifiant exact de sa machine — s'effondre presque complètement.

Extrapolation extrême.

Les trois spécifications les plus éloignées confirment et aggravent ce constat, avec 0/19 correspondances identifiant ↔ nom. La structure conceptuelle reste pourtant macroscopiquement plausible : la chaîne bioéthanol suit la route lignocellulosique réelle (broyage → hydrolyse → fermentation à 30 °C, 40-50 h, avec inhibition à 6-7 % d'éthanol → distillation sous vide → déshydratation), la chaîne or se conclut par fusion puis électrolyse (famille du procédé Wohlwill), la chaîne lithium s'ouvre sur un traitement thermique à 750-850 °C cohérent avec la pyrolyse des cellules. Le modèle génère des noms de machines adaptés à l'opération (« Fermenteur », « Fonderie »), parfois absents du référentiel, et leur attache des identifiants arbitraires quoique existants — les 19/19 « identifiants existants » masqueraient entièrement cet effondrement sans la métrique stricte. Le format se dégrade également (un champ `conditions` généré en chaîne de caractères, des conditions incohérentes telles qu'une électrolyse de « 1-3 mois »). Le Tableau 7 résume la symétrie observée.

Régime	Références exactes (id ↔ nom)	Grammaire des enchaînements
Baseline B, catalogue en contexte (§ 3.3)	31/32 (recopie)	0/5
Baseline C, catalogue + few-shot (§ 3.3)	33/36 (recopie)	~1/5 (partielle)
Fine-tuné, in-distribution	4/4	3/3
Fine-tuné, OOD étendu (15 procédés)	2/91 (2 %)	plausibilité macroscopique
Fine-tuné, extrapolation extrême	0/19	plausibilité macroscopique

Tableau 7 : Dissociation vocabulaire/grammaire selon le régime d'évaluation. « Grammaire » : validité industrielle des enchaînements (appréciation qualitative, cf. § 4.5).

La lecture d'ensemble du Tableau 7 est le résultat le plus net de l'étude. En effet, l'exactitude référentielle et la validité des enchaînements sont deux compétences dissociables, acquises par des mécanismes distincts. La fourniture en contexte (recopie) donne les références sans les enchaînements ; le few-shot y ajoute une grammaire partielle ; le fine-tuning donne les deux, mais seulement sur son domaine — hors domaine, l'ancrage référentiel disparaît tandis que la compétence structurale, héritée du pré-entraînement, subsiste.

ISSN : 2428-8500 vol. 10, n°1, 2, 2026
DOI : http://dx.doi.org/10.52497/jitipee.v10i1.416

3.5. Performance computationnelle

Le Tableau 8 donne les ressources computationnelles utilisées en RAM, en vitesse et en durée totale.

Approche	Modèle	RAM pic	Vitesse	Durée totale
Prompt baking	Qwen3.5-9B	51,7 Go	16 s/iter	535 min
SFT texte (Qwen3.5)	Qwen3.5-9B	60 Go	7 s/iter	180 min
SFT JSON (Qwen3-8B)	Qwen3-8B	18,6 Go	1,2 s/iter	40 min

Tableau 8 : Comparaison des ressources computationnelles.

L'architecture standard de Qwen3-8B est nettement plus efficace que l'architecture hybride DeltaNet de Qwen3.5-9B sur ce matériel (18,6 vs 60 Go de RAM pic, itérations 6 à 8 fois plus rapides). Combinée à LoRA (0,12 % de paramètres entraînables) et au gradient checkpointing, elle rend la spécialisation complète praticable en 40 minutes sur une station de travail personnelle (Q3), conformément aux ordres de grandeur rapportés pour le fine-tuning à paramètres efficients [9, 20].

4. Discussion

4.1. Format structuré comme tuteur du transfert de connaissances

Le contraste central de l'étude (Q2) — 0/10 références exactes en texte libre contre 5/5 en JSON, à modèle de base constant — suggère que le format structuré joue un rôle de tuteur dans l'apprentissage : les champs obligatoires (`machine_id`, `conditions`) créent un signal de supervision explicite qui force l'association entre opération et référence catalogue, là où le texte libre laisse le modèle paraphraser sans ancrage. Ce résultat prolonge les observations sur le fine-tuning d'instruction [21] et rejoint le constat de Gorilla [32] : un petit modèle fine-tuné sur appels structurés dépasse un modèle de frontière en exactitude référentielle. Il éclaire aussi, en creux, le débat ouvert par Tam et al [29] : la contrainte de format dégrade les performances quand elle est imposée au décodage d'un modèle généraliste, mais devient bénéfique quand elle est internalisée à l'entraînement. Enfin, la frugalité observée (166 exemples) est cohérente avec LIMA [22] — pour l'apprentissage du format et des motifs ; l'acquisition de connaissances généralisables demande, elle, davantage (cf. § 4.3).

4.2. Résultat négatif : le prompt baking à grande échelle factuelle

L'échec du prompt baking sur 59 203 tokens (Q1), alors que la méthode est validée sur des prompts d'instruction de 3-5 K tokens [8], documente une limite non rapportée par les auteurs : la distillation KL sur un préfixe factuel massif conduit à un optimum dégénéré où la perte converge tandis que la génération autorégressive s'effondre (*exposure bias*). Ce constat est cohérent avec la nature de la distillation de contexte [33, 34], conçue pour internaliser des *comportements* (instructions, style, raisonnement) plutôt que des corpus factuels volumineux. Il suggère que l'injection de connaissances encyclopédiques requiert une supervision par exemples (SFT) plutôt qu'une imitation de distributions conditionnées.

4.3. Mémorisation, généralisation et régime d'usage

Le régime observé (perte d'entraînement ≈ 0, perte de validation remontant à 0,5) est un régime de mémorisation assumé : sur les procédés vus en entraînement, le modèle restitue des chaînes exactes ; hors distribution, il compose des enchaînements plausibles mais avec des références dégradées. L'évaluation OOD étendue (15 procédés) et l'extrapolation extrême (§ 3.4)

quantifient cette limite : au-delà du domaine du corpus, ce n'est pas la capacité à structurer une chaîne qui disparaît — les motifs macroscopiques, hérités du pré-entraînement, subsistent — mais l'ancrage dans le référentiel, qui chute à 2 % (2/91) puis 0 % (0/19). La dissociation vocabulaire/grammaire est ainsi observée dans les deux directions (Tableau 7) : le contexte seul donne le vocabulaire sans la grammaire, le fine-tuning hors de son domaine donne une grammaire approximative sans le vocabulaire. L'association opération ↔ identifiant est acquise par mémorisation et ne s'étend pas au-delà du corpus, tandis que la compétence structurale se transfère. Ce comportement délimite le régime d'usage actuel : le modèle est un *compilateur de connaissances expertes interrogeable par spécification fonctionnelle*, fiable sur le domaine couvert par le corpus, exploratoire au-delà — toute sortie hors domaine nécessitant validation, ses conditions pouvant devenir incohérentes (électrolyse de « 1-3 mois »). L'élargissement du corpus de procédés (24 actuellement, contre par exemple les milliers de topologies utilisées par les générateurs SFILES [7, 11]) est la voie principale d'amélioration de la généralisation, avec un ordonnancement du taux d'apprentissage (décroissance cosinus) et un arrêt anticipé calibré sur des métriques de tâche plutôt que sur la perte de validation.

Un point méthodologique important sous-tend ces mesures. Le découpage train/validation d'origine était aléatoire sur les 166 exemples : les trois reformulations d'un même procédé pouvant tomber de part et d'autre, la perte de validation mesurait en partie la reconnaissance de reformulations plutôt que la généralisation à de nouveaux procédés. Nous avons reconstruit un découpage *par groupe de procédé* (quatre procédés — huile d'olive, ciment Portland, biodiesel, minerai de fer — réservés intégralement en validation, absents de l'entraînement). Sur ce découpage sans fuite, la perte de validation initiale sur les procédés jamais vus est de 1,29, du même ordre que celle du modèle non entraîné, ce qui confirme que le découpage aléatoire sous-estimait l'écart de généralisation. Le réentraînement complet sur ce découpage, qui préciserait le plancher de généralisation atteint, reste à mener (limite § 4.5).

4.4. Fine-tuning et fourniture en contexte : vocabulaire vs grammaire

Ovadia et al. [24] montrent que le RAG surpasse souvent le fine-tuning pour l'injection de connaissances *factuelles*. Notre baseline (§ 3.3) précise empiriquement les termes de cette comparaison pour une tâche de *conception* : la fourniture du référentiel en contexte suffit à l'exactitude référentielle — le modèle recopie le catalogue — mais échoue sur la validité des enchaînements (0/5 conformes) et sur les conditions opératoires, qui relèvent de connaissances procédurales absentes du catalogue lui-même. Le fine-tuning sur chaînes complètes semble injecter cette connaissance relationnelle — quelles machines s'enchaînent, dans quel ordre, sous quelles conditions — sous la réserve d'asymétrie du protocole notée en § 3.3. Les deux mécanismes sont d'ailleurs complémentaires plutôt que concurrents : un système opérationnel pourrait combiner poids spécialisés (grammaire des procédés), catalogue en contexte ou récupéré (vocabulaire à jour, machines ajoutées après l'entraînement), décodage contraint par grammaire [30, 31] pour la garantie syntaxique (éliminant le cas de troncature observé), et validation programmatique des `machine_id` contre le référentiel en post-traitement. Il reste à comparer avec un modèle de frontière (contexte long, capacités de raisonnement supérieures) muni du même catalogue, qui pourrait combler une partie du déficit de grammaire procédurale par ses connaissances générales en génie des procédés.

ISSN : 2428-8500 vol. 10, n°1, 2, 2026
DOI : http://dx.doi.org/10.52497/jitipee.v10i1.416

4.5. Menaces à la validité

Nous étudions dans ce paragraphes les menaces à la validité.

- **Comparaisons partiellement confondues** : les stratégies 2 et 3 partagent le modèle de base mais diffèrent par la taille et la composition du dataset (633 vs 166 exemples), le format conditionnant la nature des exemples constructibles ; l'attribution de l'écart au seul format reste donc une inférence, qu'une ablation à contenu strictement constant devra confirmer.
- **Conformité industrielle qualitative** : l'exactitude référentielle et le format sont mesurés automatiquement (métrique stricte identifiant ↔ nom, validité JSON) sur 5 + 15 + 3 procédés ; en revanche la *validité industrielle des enchaînements* — colonne « grammaire » du Tableau 7 — reste une appréciation qualitative de l'auteur, non un jugement d'experts indépendants ni une vérification par simulateur. C'est la principale faiblesse d'évaluation restante : une notation en aveugle par plusieurs ingénieurs procédés, ou l'export vers un simulateur séquentiel modulaire, est nécessaire pour objectiver cette dimension.
- **Réentraînement sans fuite non abouti** : le découpage d'origine comportait une fuite (variantes d'un même procédé réparties entre train et validation) ; le découpage corrigé par groupe de procédé a été construit (§ 4.3) et la perte de validation initiale mesurée (1,29), mais le réentraînement complet sur ce découpage, qui donnerait le plancher de généralisation, n'a pu être exécuté sur la station de travail utilisée (limite matérielle) et reste à mener.
- **Variante ablitérée** : le modèle de base modifié [39] introduit une variable non contrôlée ; la recette devra être répliquée sur le modèle aligné d'origine (que les baselines § 3.3 utilisent déjà).
- **Exécution unique** : les résultats proviennent d'un entraînement par configuration, sans répétition multi-graines.
- **Baseline de frontière manquant** : les baselines portent sur le même modèle de 8 Md de paramètres ; un modèle de frontière à long contexte muni du même catalogue et des mêmes exemples reste à mesurer (§ 4.4).
- **Représentation linéaire** : la sortie encode une séquence ordonnée d'étapes, non un flowsheet — recyclages, courants parallèles, purges et utilités ne sont pas représentés. Les objets générés sont des gammes opératoires conceptuelles, non des schémas de procédé exploitables en ingénierie de détail.
- **Ontologie hétérogène** : le référentiel mêle opérations unitaires, familles d'équipements et architectures de réacteurs, ce qui rend la notion d'« exactitude machine » partiellement conventionnelle.

ISSN : 2428-8500 vol. 10, n°1, 2, 2026
DOI : http://dx.doi.org/10.52497/jitipee.v10i1.416

Conclusion

Nous avons étudié l'injection d'un référentiel encyclopédique de 444 familles d'équipements de procédés dans un modèle de langage ouvert de 8 milliards de paramètres. Cinq résultats se dégagent : le prompt baking échoue à internaliser un corpus factuel de 59 K tokens (résultat négatif documenté) ; à modèle constant, le dataset JSON structuré a permis le transfert des références exactes du catalogue là où le dataset en texte libre a échoué (0/10 contre 5/5), contraste attribuable au format sous réserve d'une ablation à contenu strictement constant ; 166 exemples structurés suffisent, en 40 minutes sur une station de travail personnelle, à obtenir un modèle générant des gammes opératoires multi-étapes référencées avec conditions indicatives, conformes aux pratiques documentées sur le domaine couvert par le corpus ; les baselines sans fine-tuning atteignent l'exactitude référentielle par recopie du catalogue (31/32 avec le catalogue seul, 33/36 en ajoutant deux exemples few-shot) mais peinent sur la validité des enchaînements (0/5 puis ~1/5), les exemples few-shot transférant le format et une grammaire partielle sans la conformité industrielle complète ; enfin, hors du domaine du corpus (15 procédés OOD puis 3 cas extrêmes), le modèle fine-tuné présente la dissociation inverse — enchaînements macroscopiquement plausibles, ancrage référentiel effondré (2/91 puis 0/19). Cette double dissociation — exactitude sans grammaire d'un côté, grammaire sans exactitude de l'autre — est le résultat central : elle montre que les deux compétences sont séparables et acquises par des mécanismes distincts, et délimite le régime d'usage du modèle : fiable sur le domaine couvert par le corpus, exploratoire et nécessitant validation au-delà.

L'approche se distingue des générateurs de flowsheets à base de transformers, centrés sur la topologie des opérations unitaires, en produisant des sorties structurées programmatiquement validables (identifiants d'équipements, conditions indicatives) — au prix, en l'état, d'une représentation séquentielle sans topologie —, et des agents LLM de frontière en reposant sur un modèle ouvert, local et frugal. Les travaux futurs porteront, dans l'ordre des menaces identifiées : sur une évaluation quantitative à grande échelle avec vérification programmatique de la cohérence des enchaînements contre le référentiel et jugement d'experts indépendants ; sur un baseline de frontière (modèle à long contexte muni du catalogue) ; sur une ablation du format à contenu strictement constant ; sur l'élargissement du corpus de procédés (80-100 procédés) ; et sur l'hybridation poids spécialisés + catalogue en contexte + décodage contraint esquissée en § 4.4. Une passerelle vers la notation SFILES 2.0 [10] permettrait en outre la comparaison directe avec les générateurs de flowsheets existants sur des benchmarks communs.

Références

[1] Douglas J. M. (1988), "Conceptual Design of Chemical Processes", McGraw-Hill.
[2] Biegler L. T., Grossmann I. E., Westerberg A. W. (1997), "Systematic Methods of Chemical Process Design", Prentice Hall.
[3] Douglas J. M. (1985), "A hierarchical decision procedure for process synthesis." AIChE Journal, 31(3), 353-362. https://doi.org/10.1002/aic.690310302
[4] Mencarelli L., Chen Q., Pagot A., Grossmann I. E. (2020), "A review on superstructure optimization approaches in process system engineering." Computers & Chemical Engineering, 136, 106808. https://doi.org/10.1016/j.compchemeng.2020.106808
[5] Brown T. B. et al. (2020), "Language Models are Few-Shot Learners." Advances in Neural Information Processing Systems, 33. https://arxiv.org/abs/2005.14165
[6] Decardi-Nelson B., Alshehri A. S., Ajagekar A., You F. (2024), "Generative AI and process systems engineering: The next frontier." Computers & Chemical Engineering, 187, 108723. https://doi.org/10.1016/j.compchemeng.2024.108723
[7] Vogel G., Schulze Balhorn L., Schweidtmann A. M. (2023), "Learning from flowsheets: A generative transformer model for autocompletion of flowsheets." Computers & Chemical Engineering, 171, 108162. https://doi.org/10.1016/j.compchemeng.2023.108162
[8] Bhargava A., Witkowski C., Detkov A., Thomson M. (2024), "Prompt Baking". https://arxiv.org/abs/2409.13697
[9] Hu E. J. et al. (2022), "LoRA: Low-Rank Adaptation of Large Language Models." Proc. International Conference on Learning Representations. https://arxiv.org/abs/2106.09685
[10] Vogel G., Hirtreiter E., Schulze Balhorn L., Schweidtmann A. M. (2023), "SFILES 2.0: an extended text-based flowsheet representation." Optimization and Engineering, 24(4), 2911-2933. https://doi.org/10.1007/s11081-023-09798-9
[11] Schulze Balhorn L., Hirtreiter E., Luderer L., Schweidtmann A. M. (2023), "Data augmentation for machine learning of chemical process flowsheets." Computer Aided Chemical Engineering (ESCAPE-33), Elsevier. https://doi.org/10.1016/B978-0-443-15274-0.50320-6
[12] Hirtreiter E., Schulze Balhorn L., Schweidtmann A. M. (2024), "Toward automatic generation of control structures for process flow diagrams with large language models." AIChE Journal, 70(1), e18259. https://doi.org/10.1002/aic.18259
[13] Schulze Balhorn L., Caballero M., Schweidtmann A. M. (2024), "Toward autocorrection of chemical process flowsheets using large language models." Computer Aided Chemical Engineering, 53, 3109-3114. https://doi.org/10.1016/B978-0-443-28824-1.50519-6
[14] Schulze Balhorn L., Degens K., Schweidtmann A. M. (2025), "Graph-to-SFILES: Control structure prediction from process topologies using generative artificial intelligence." Computers & Chemical Engineering, 199, 109121. https://doi.org/10.1016/j.compchemeng.2025.109121

[15] Gao Q., Schweidtmann A. M. (2024), "Deep reinforcement learning for process design: Review and perspective." Current Opinion in Chemical Engineering, 44, 101012. https://doi.org/10.1016/j.coche.2024.101012

[16] Boiko D. A., MacKnight R., Kline B., Gomes G. (2023), "Autonomous chemical research with large language models." Nature, 624, 570-578. https://doi.org/10.1038/s41586-023-06792-0

[17] Bran A. M., Cox S., Schilter O., Baldassari C., White A. D., Schwaller P. (2024), "Augmenting large language models with chemistry tools." Nature Machine Intelligence, 6, 525-535. https://doi.org/10.1038/s42256-024-00832-8

[18] Ramos M. C., Collison C. J., White A. D. (2025), "A review of large language models and autonomous agents in chemistry." Chemical Science, 16, 2514-2572. https://doi.org/10.1039/D4SC03921A

[19] Schulze Balhorn L., Weber J. M., Buijsman S., Hildebrandt J. R., Ziefle M., Schweidtmann A. M. (2024), "Empirical assessment of ChatGPT's answering capabilities in natural science and engineering." Scientific Reports, 14, 4998. https://doi.org/10.1038/s41598-024-54936-7

[20] Dettmers T., Pagnoni A., Holtzman A., Zettlemoyer L. (2023), QLoRA: "Efficient Finetuning of Quantized LLMs." Advances in Neural Information Processing Systems, 36. https://arxiv.org/abs/2305.14314

[21] Wei J. et al. (2022), "Finetuned Language Models Are Zero-Shot Learners." Proc. International Conference on Learning Representations. https://arxiv.org/abs/2109.01652

[22] Zhou C. et al. (2023), "LIMA: Less Is More for Alignment." Advances in Neural Information Processing Systems, 36, 55006-55021. https://arxiv.org/abs/2305.11206

[23] Lewis P. et al. (2020), "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks." Advances in Neural Information Processing Systems, 33. https://arxiv.org/abs/2005.11401

[24] Ovadia O., Brief M., Mishaeli M., Elisha O. (2024), "Fine-Tuning or Retrieval? Comparing Knowledge Injection in LLMs." Proc. Conference on Empirical Methods in Natural Language Processing (EMNLP). https://arxiv.org/abs/2312.05934

[25] Luo Y., Yang Z., Meng F., Li Y., Zhou J., Zhang Y. (2023), "An Empirical Study of Catastrophic Forgetting in Large Language Models During Continual Fine-tuning." https://arxiv.org/abs/2308.08747

[26] Zhang D., Liu W., Tan Q. et al. (2024), "ChemLLM: A Chemical Large Language Model." https://arxiv.org/abs/2402.06852

[27] Yu B., Baker F. N., Chen Z., Ning X., Sun H. (2024), "LlaSMol: Advancing Large Language Models for Chemistry with a Large-Scale, Comprehensive, High-Quality Instruction Tuning Dataset". Proc. Conference on Language Modeling (COLM). https://arxiv.org/abs/2402.09391

[28] Buehler M. J. (2024), "MechGPT, a Language-Based Strategy for Mechanics and Materials Modeling That Connects Knowledge Across Scales, Disciplines and Modalities." Applied Mechanics Reviews, 76(2), 021001. https://doi.org/10.1115/1.4063843

[29] Tam Z. R., Wu C.-K., Tsai Y.-L., Lin C.-Y., Lee H.-y., Chen Y.-N. (2024), "Let Me Speak Freely? A Study on the Impact of Format Restrictions on Performance of Large Language Models." Proc. EMNLP Industry Track. https://arxiv.org/abs/2408.02442

[30] Willard B. T., Louf R. (2023), "Efficient Guided Generation for Large Language Models." https://arxiv.org/abs/2307.09702

[31] Geng S., Josifoski M., Peyrard M., West R. (2023), "Grammar-Constrained Decoding for Structured NLP Tasks without Finetuning." Proc. Conference on Empirical Methods in Natural Language Processing (EMNLP), 10932-10952. https://arxiv.org/abs/2305.13971

[32] Patil S. G., Zhang T., Wang X., Gonzalez J. E. (2024), "Gorilla: Large Language Model Connected with Massive APIs." Advances in Neural Information Processing Systems, 37. https://arxiv.org/abs/2305.15334

[33] Askell A. et al. (2021), "A General Language Assistant as a Laboratory for Alignment." https://arxiv.org/abs/2112.00861

[34] Snell C., Klein D., Zhong R. (2022), "Learning by Distilling Context." https://arxiv.org/abs/2209.15189

[35] Green D. W., Southard M. Z. (eds.) (2019), "Perry's Chemical Engineers' Handbook", 9e éd., McGraw-Hill Education.

[36] Coulson J. M., Richardson J. F., Backhurst J. R., Harker J. H. (2002), "Coulson & Richardson's Chemical Engineering, Volume 2: Particle Technology and Separation Processes", 5e éd., Butterworth-Heinemann.

[37] Yang A. et al. (2025), "Qwen3 Technical Report " https://arxiv.org/abs/2505.09388

[38] Arditi A. et al. (2024), "Refusal in Language Models Is Mediated by a Single Direction." Advances in Neural Information Processing Systems, 37. https://arxiv.org/abs/2406.11717

[39] Hannun A., Digani J., Katharopoulos A., Collobert R. (2023), "MLX: Efficient and flexible machine learning on Apple silicon." Logiciel, version 0.x. https://github.com/ml-explore/mlx