DOCUMENT 5 : DOMAIN MAP

AIM²P — Cartographie du domaine et choix d'outils figés pour P2–P3

Phase : Phase 1 (cartographie du domaine) · Projet : AIM²P Initiative · Version : 0.1

Synthèse : ce que la phase P1 fige

Chaîne formelle : Lean v4.35.0, avec Mathlib et Physlib au tag v4.35.0 (commits épinglés dans `bench/lean/lake-manifest.json`) et `autoImplicit = false`. Rocq et Isabelle/HOL restent hors de la V0.1. Calcul formel : SymPy 1.14.0 avec mpmath 1.3.0, en local, comme critique faillible et mesuré ; SageMath ne sert qu'à des tests différentiels ponctuels. Vérification : dans la boucle, le REPL Lean (tag v4.35.0) piloté par LeanInteract, avec Lean gardé en mémoire ; pour la certification finale, `lake comparator` dans son bac à sable Linux (donc sous WSL2), avec la liste blanche `propext`, `Quot.sound`, `Classical.choice`. Modèles, sous contrainte de gratuité : `nvidia/nemotron-3-ultra-550b-a55b` (via OpenRouter, voir l'encadré « Jour 1 ») comme modèle généraliste pré-sélectionné, `moonshotai/kimi-k3.1-pro` en repli, et Goedel-Prover-V2-8B (Apache-2.0) sur notebooks Kaggle gratuits comme prouveur ouvert. Ces modèles ne seront gelés qu'après la porte du jour 1 (§8) et le gel de S8. Aucun résultat de performance ne figure ici. Les durées sont des mesures locales ponctuelles ou des estimations présentées comme telles. Les noms de modèles et les limites sont ceux observés le 18 novembre 2026 et peuvent changer.

Jour 1 (18 novembre 2026) : la route NVIDIA est remplacée par OpenRouter

La vérification téléphonique de build.nvidia.com a échoué pour le numéro du fondateur : le code SMS n'est jamais arrivé. Le même modèle, `nvidia/nemotron-3-ultra-550b-a55b`, est accessible gratuitement via OpenRouter (`nvidia/nemotron-3-ultra-550b-a55b:free`, servi par Nvidia), avec une simple connexion par compte Google ou GitHub. La limite est de 50 requêtes par jour sans achat de crédits.

Pilote mesuré le jour même sous WSL2, sur un problème jouet public : réponse correcte à un calcul en 3,9 s ; preuve Lean produite en 53 s (562 tokens) et **acceptée par** `lake comparator` (noyau Lean, axiomes standard, énoncé identique), pour un coût nul. Le modèle a écrit `import Mathlib`, ce qui a porté la certification à 161 s, contre 13 à 20 s avec des imports ciblés : le protocole imposera donc un en-tête d'imports fixe.

Conséquences : (i) le volume est limité à 50 requêtes par jour, d'où un pilote puis une décision sur le volume (§9) ; (ii) selon une source secondaire, les points d'accès gratuits d'OpenRouter peuvent servir à l'entraînement : seuls des items publics y passent tant que la décision du §9 n'est pas prise ; (iii) Kaggle exige aussi une vérification téléphonique : le prouveur sera d'abord testé sur Google Colab.

1. OBJET ET PÉRIMÈTRE

La phase P1 n'ajoute pas de revue de littérature : l'état de l'art est arrêté dans les documents 02 et 03, et le protocole dans le document 04. Son rôle est de **décider** : choisir et figer les outils dont dépendent la validité et la reproductibilité des mesures, avant la construction d'AMP-Bench-mini (P2, S3–S5) et les exécutions des baselines B0–B3 et des ablations A1–A2 (P3, S8–S10). Le document 04 (§5.6) exige de geler les versions de SymPy, de Math-Verify, de la chaîne Lean, de Mathlib et du REPL, ainsi que les identifiants de modèles. Ce document fixe les versions et pré-sélectionne les modèles ; le gel définitif des modèles reste prévu en S8, dans `results/baselines/config.yaml`.

Il s'appuie sur trois sources : les notes de recherche P1 du 18 novembre 2026 (offres gratuites de modèles, prouveurs ouverts, outillage de vérification Lean), les notes de P0, et des constats faits le même jour sur la machine du fondateur. Celle-ci n'a pas de GPU dédié (Intel Iris Xe intégrée, Intel i7-1355U, 16 Go de RAM, Windows 11). **L'inférence locale se limite donc à des tests de fumée avec de petits modèles quantifiés**, et toute génération en volume se fait ailleurs.

Contrainte de gratuité : un choix méthodologique

À ce stade, AIM²P n'utilise que des modèles gratuits, accessibles par une API gratuite ou sur du calcul gratuit, sans crédit payant. Ce choix rend le protocole rejouable sans budget, mais il a un coût : les conclusions porteront sur ces modèles et non sur les modèles fermés de pointe, et les offres gratuites changent sans préavis (§4). Si des crédits sont obtenus (document 04, §10), les modèles ajoutés feront l'objet d'une version datée et ne remplaceront jamais l'analyse principale gelée.

2. MATRICE COMPARATIVE

Composant	Licence	Coût	Exécution locale	Automatisation, API	Couverture physique	Maturité, communauté	Rôle dans AIM²P
LLM généralistes (catalogue NVIDIA)	Variable : `openmdw-1.1` (Nemotron 3 Ultra), licence propre (Kimi K3.1 Pro)	Gratuit en mode d'essai ; limites non publiées (§4)	Non : plus de 500 milliards de paramètres, aucun GPU local	API REST ; liste publique `/v1/models` (32 identifiants le 18 nov.), aucun prouveur Lean hébergé	Non mesurée pour ces modèles. Sur LeanPhysBench, un modèle généraliste (35 %) devance le meilleur prouveur spécialisé (16 %)	Force de Nemotron en mathématiques non vérifiée de façon indépendante ; Kimi K3.1 Pro est le premier modèle ouvert du classement global de MathArena (54,3 % ± 5,8 %)	Modèle des baselines B0–B3 et de la boucle
Prouveurs Lean ouverts (Goedel-Prover-V2-8B…)	Apache-2.0 (Goedel, Kimina) ; licence DeepSeek à restrictions d'usage (DeepSeek-Prover-V2)	Gratuit sur notebooks ; aucune API gratuite	Tests de fumée seulement (GGUF sur CPU, 3 à 10 tokens/s, estimation)	Génération par lots (vLLM), sans outils	Transfert faible : 16 % pour le meilleur prouveur mathématique sur LeanPhysBench	83,0 % sur miniF2F en pass@32 selon sa carte ; entraînés ou évalués sur Lean 4.9 à 4.15	Baseline à poids ouverts sur les items formels
Lean 4 + Mathlib + Physlib	Apache-2.0 (les trois)	Gratuit	Oui, mesuré : installation en 26 min, test de fumée élaboré en 171 s à froid	REPL JSON, LeanInteract (Python), `lake comparator`, `lake check`	Physlib : environ 451 000 lignes sur 20 domaines, dont unités typées, mécanique hamiltonienne et commutateurs ; ni Krylov ni Lanczos	Mathlib : 8 718 fichiers ; Physlib : 7 mainteneurs, 3 884 commits ; benchmarks retenus disponibles en Lean	Certification, et retour d'erreurs dans la boucle
SymPy (+ SageMath)	SymPy : BSD à 3 clauses ; SageMath : GPL-3.0 pour la distribution	Gratuit	Oui : Python pur, SymPy 1.14.0 installé	API Python ; pont Sage–Lean LeanSage (MIT, en développement)	`sympy.physics` (unités, quantique, mécanique) disponible ; aucun certificat hors algèbre polynomiale	Le traqueur garde une étiquette `simplify` de bogues de correction ouverts ; aucune étude systématique de son taux d'erreur	Critique faillible et mesuré ; calcul des clés N5 (avec mpmath)
Rocq (ex-Coq)	LGPL-2.1	Gratuit	Possible, non testé	Non évaluée en P1	Programmes quantiques (SQIR/VOQC, CoqQ) ; aucune preuve de Lanczos trouvée	412 formalisations dans PutnamBench	Hors V0.1
Isabelle/HOL	Licence de type BSD	Gratuit	Possible, non testé	Sledgehammer (oracles ATP/SMT avec reconstruction de preuve) ; non évaluée en P1	Grandeurs et unités (entrée AFP Physical_Quantities, ISQ/SI), circuits quantiques (Isabelle Marries Dirac)	640 formalisations dans PutnamBench	Hors V0.1 ; comparaison possible pour H3a plus tard

Pourquoi Lean 4 + SymPy. Physlib, seule bibliothèque de physique formelle de cette ampleur trouvée en P0, n'existe qu'en Lean ; elle fournit déjà les unités typées (H3a, item U1), la mécanique hamiltonienne (F5) et les commutateurs (C1). Les prouveurs ouverts recensés en P0–P1 et les benchmarks audités en 2026 (miniF2F, ProofNet#, PutnamBench) visent Lean 4, ce qui rend possible une baseline à poids ouverts. Depuis la v4.35.0, la vérification de l'énoncé et des axiomes fait partie du cœur de Lean (`lake comparator`, `lake check`). SymPy tourne en local, en Python pur, sous licence permissive, et a déjà servi à l'audit des items N5. Le précédent de `polyrith`, devenu inutilisable à la fermeture de son serveur Sage distant, écarte toute dépendance à un calcul formel distant.

Pourquoi ni Rocq ni Isabelle en V0.1. Aucun des deux n'offre d'équivalent de Physlib pour la dynamique d'opérateurs, même si Isabelle dispose d'une bibliothèque d'unités. Nos recherches n'ont recensé aucun prouveur ouvert pour ces assistants, sans les avoir cherchés

spécifiquement. Enfin, une seconde chaîne d'outils coûterait trop cher à un fondateur seul qui travaille 10 heures par semaine. Comme PutnamBench existe aussi en Isabelle et en Rocq, une extension multi-assistants reste possible après la V0.1.

3. CHOIX FIGÉS

Composant	Choix	Version ou référence	Justification	Alternative ou repli
Chaîne Lean	`leanprover/lean4:v4.35.0`	`bench/lean/lean-toolchain`	Le protocole exige au moins la 4.20.0 (bogue de `apply?`) ; première version avec `lake comparator` et `lake check` dans le cœur ; Physlib a un tag correspondant	Aucune mise à jour pendant une campagne ; la v4.36.0 (16 nov.) est écartée jusqu'à la V0.2
Mathlib	Tag `v4.35.0`, requis par Physlib v4.35.0	Commit `56a005855ea17e0b0958f73ec801ae140b0a1679`	Hérité de Physlib, donc cohérent avec lui	—
Physlib	Tag `v4.35.0`	Commit `af04d60bcd7618af849ff511e0a19d77f709937e`	Unités typées, mécanique hamiltonienne ; `Physlib.Units.Dimension` compile	Énoncés N5 sans Physlib si un module pose problème
Options Lake	`autoImplicit = false`, `relaxedAutoImplicit = false`	`bench/lean/lakefile.toml`	Évite les énoncés vacus (document 04, §5.1)	—
Système de modules	En-tête `module`, recommandé par un avertissement de Mathlib v4.35	—	**Décision reportée à P2 (S3)**	Imports classiques, qui compilent aujourd'hui
SymPy	1.14.0, avec mpmath 1.3.0	PyPI : dernière version stable au 18 nov. ; la 1.15.0 n'en est qu'à la version candidate rc3 (17 nov.)	Version utilisée pour l'audit des items N5 ; pas de version candidate dans un protocole gelé	Passage à la 1.15 seulement entre deux campagnes
Python	3.14.7 (version locale)	Consignée au gel, dépendances épinglées dans un environnement virtuel	Environnement déjà en place	Version antérieure si une dépendance refuse la 3.14 (à vérifier au jour 1)
Réponses fermées	Math-Verify 0.9.0 (Apache-2.0), plus une équivalence numérique en points aléatoires	PyPI ; runtime ANTLR4 à épingler	Document 04, §5.2	Audit manuel de chaque désaccord
Vérificateur dans la boucle	REPL Lean au tag `v4.35.0` (Apache-2.0) + LeanInteract 0.11.6 (MIT)	Le REPL a un tag v4.35.0 ; LeanInteract couvre Lean v4.8.0-rc1 à v4.36.0-rc1	Garde Lean chargé, renvoie erreurs et `sorry`, fonctionne sous Windows	kimina-lean-server (MIT, Docker, sous WSL2)
Certification finale	`lake comparator` sous WSL2 dans le bac à sable `bwrap`, puis `lake check --paranoid` sur le lot final	`comparator.json` : `permitted_axioms = propext`, `Quot.sound`, `Classical.choice`	Énoncé gelé, axiomes et noyau vérifiés ; le code produit par le modèle n'est pas fiable, d'où le bac à sable	SafeVerify comme second avis seulement, s'il est porté (aucune branche v4.35)
Interdits dans le source	`sorry`, `admit`, `axiom`, `native_decide`, `implemented_by`, `extern`, `unsafe`, `debug.skipKernelTC`	Filtre textuel avant compilation	Couvre ce que les vérifications au niveau des `.olean` laissent passer	—

Composant	Choix	Version ou référence	Justification	Alternative ou repli
Modèle généraliste	`nvidia/nemotron-3-ultra-550b-a55b`	Via OpenRouter (`:free` , servi par Nvidia), vérifié le 18 nov. ; catalogue NVIDIA inaccessible (vérification téléphonique)	§4 et encadré « Jour 1 »	À redéfinir : Kimi K3.1 Pro n'est pas gratuit sur OpenRouter ; candidats : Gemma 4 31B (OpenRouter) ou `gpt-oss-120b` (Groq)
Prouveur ouvert	`Goedel-LM/Goedel-Prover-V2-8B`	Hugging Face, Apache-2.0	§4	`project-numina/Kimina-Prover-Distill-8B` (Apache-2.0)
Sources du benchmark	miniF2F (google-deepmind), ProofNet#, PutnamBench Lean ; items rédigés par AIM²P	§6	Licences permissives, audit de 2026 disponible	miniF2F-v2c pour N3, une fois sa licence confirmée

> **Écart déclaré par rapport au document 04**
>
> Le document 04 (§5.2, point 4) plaçait SafeVerify dans la boucle. SafeVerify n'a pas de branche pour Lean v4.35 : dans la boucle, il est remplacé par le REPL, le filtre de source et `#print axioms` . La définition de la réussite vérifiée ne change pas : acceptation par le noyau contre l'énoncé gelé, avec les seuls axiomes standard. C'est la certification finale par `lake comparator` qui en décide. Les médianes citées au document 04 (10,1 s et 95,7 s) viennent d'un autre environnement et ne valent pas pour ce portable.

4. MODÈLES DES BASELINES SOUS CONTRAINTE DE GRATUITÉ

Budget. Pour un modèle, le plan d'expérience compte au plus 3 000 générations, soit environ 2 500 avec l'arrêt au premier succès, et environ 12 000 tokens par génération, soit à peu près 30 millions de tokens (estimations des notes P1).

Modèle généraliste recommandé : `nvidia/nemotron-3-ultra-550b-a55b` . C'est la seule route gratuite qui puisse plausiblement tenir ce budget en quelques jours. Les conditions d'essai de NVIDIA (§2.3) excluent tout stockage ou usage du contenu après la session, ce qui en fait les meilleures conditions d'usage des données parmi les offres gratuites ; d'après une source secondaire, aucune carte bancaire n'est demandée. Comme il s'agit du modèle de NVIDIA sur le catalogue de NVIDIA, c'est le moins exposé à un passage en accès réservé aux partenaires. Ses poids sont ouverts (`openmdw-1.1`), il est public depuis juin 2026 et dispose d'une seconde route gratuite (OpenRouter `:free`). Il accepte `temperature` , `seed` et `reasoning_effort` (métadonnées OpenRouter). **Limite :** sa force en mathématiques et en Lean n'est pas vérifiée de façon indépendante. Les chiffres secondaires se contredisent, et il est absent du classement global de MathArena.

Repli : `moonshotai/kimi-k3.1-pro` . C'est le meilleur modèle ouvert du classement global de MathArena (54,3 % ± 5,8 %). Mais son dépôt a été créé le 2 novembre 2026, il compte environ 2 780 milliards de paramètres, sa licence est propre, et, selon une source secondaire, des modèles Moonshot ont déjà été retirés de l'accès libre du catalogue NVIDIA.

Prouveur ouvert : Goedel-Prover-V2-8B (Apache-2.0). C'est le seul petit prouveur ouvert entraîné explicitement à l'auto-correction guidée par le compilateur. Il l'a été sur 2 tours alors que B3 en autorise 5 : les invites de révision resteront donc compactes (dernière tentative et erreurs). Il tourne sous vLLM sur Kaggle (fp16 sur 2×T4 en parallélisme tensoriel, ou version 8 bits sur une T4), **par tours synchrones** : le tour k est généré sur Kaggle pour tous les problèmes, vérifié sur le portable, puis les invites du tour k+1 sont construites. Cela fait au plus 6 tours par exécution, sans Lean sur Kaggle. Leanstral 1.5, le prouveur ouvert le plus fort, est inutilisable gratuitement : son API a été retirée le 30 septembre 2026, et ses 119 milliards de paramètres ne tiennent ni sur Kaggle ni sur le portable.

Règles de bascule, pré-enregistrées.

1. Kimi remplace Nemotron seulement si le pilote du jour 1, mené sur des items de développement, montre que moins de 5 % des tentatives B0 de Nemotron compilent, et si Kimi est resté accessible pendant au moins deux semaines.
2. Si l'accès NVIDIA est décompté en crédits (1 000 crédits selon une source secondaire), aucune campagne ne démarre. On demande d'abord une extension, que des messages de forum disent possible ; à défaut, on applique les réductions de périmètre du document 04 (§7), ou l'on emprunte la route OpenRouter `:free` pour les seuls items publics N3–N4 (environ 20 jours à 50 requêtes par jour, estimation).

3. Kimina-Prover-Distill-8B remplace Goedel si Goedel ne se charge pas sous vLLM sur Kaggle, ou si le débit mesuré ne permet pas de tenir le budget en trois semaines de quota.
4. Toute bascule est décidée avant le gel de S8. Si un modèle disparaît ou change en cours de campagne, la campagne de ce modèle est arrêtée, signalée et relancée en entier avec le repli, sous une version datée. On ne mélange jamais deux modèles dans une même campagne.

Options gratuites évaluées (vérifié le 18 novembre 2026).

Option	Limite déterminante	Usage des données	Durée pour un modèle (estimation)	Verdict
Catalogue d'API NVIDIA	Environ 40 requêtes par minute selon des messages de forum ; décompte en crédits non confirmé	Ni stockage ni usage après la session (conditions d'essai, §2.3)	1 à 3 jours si seul le débit est limité ; impossible avec 1 000 crédits (environ 40 % du plan)	**Retenu**, sous réserve du jour 1
Notebooks Kaggle (prouveur)	Environ 30 h de GPU par semaine (P100 ou 2×T4), selon une source secondaire	Notebooks privés ; conditions à relire	1 à 2 semaines de quota	**Retenu** pour le prouveur
Google AI Studio, offre gratuite	Limites non publiées et « non garanties » ; environ 20 requêtes par jour pour Gemini 3.8 Flash et 500 pour Flash-Lite, selon un forum	Contenus utilisés pour améliorer les produits, relecture humaine possible ; identifiants dépréciés redirigés en silence (`gemini-2.5-pro` vers `gemini-3.8-flash` à partir du 1er décembre 2026)	Environ 125 jours (3.8 Flash), 5 jours (Flash-Lite)	Rejeté pour les items privés ; pilotes sur items publics au plus
Groq (`openai/gpt-oss-120b`)	200 000 tokens par jour, 8 000 par minute (tableau de l'offre gratuite, libellé ambigu)	Aucune conservation par défaut	Environ 150 jours	Rejeté ; tests de fumée
OpenRouter `:free`	50 requêtes par jour sans achat de crédits	Selon le point d'accès ; ceux de NVIDIA « peuvent entraîner », selon une source secondaire	Environ 50 jours	Route de secours pour Nemotron, items publics seulement ; jamais le routeur `openrouter/free`
API Mistral	Le mode gratuit n'inclut plus l'accès à l'API	—	—	Exclu ; l'API Leanstral (`labs-leanstral-1-5`) a été retirée le 30 septembre 2026
GitHub Models	Service retiré le 30 juillet 2026	—	—	Disparu
Cloudflare Workers AI	10 000 « neurons » par jour, soit environ 14 générations de gpt-oss-120b (estimation)	Non confirmé	Environ 180 jours	Rejeté
SambaNova Cloud	20 requêtes et 200 000 tokens par jour	Non confirmé	Environ 125 jours	Rejeté
Cohere (clé d'essai)	1 000 appels par mois	Non confirmé	Environ 2,5 mois	Rejeté
Cerebras ; Hugging Face Inference Providers	Moyen de paiement vérifié exigé ; aucun crédit mensuel gratuit	—	—	Exclus
Colab gratuit ; Hugging Face ZeroGPU	Ressources non garanties ; 5 minutes de GPU par jour	—	—	Appoint pour Colab ; ZeroGPU rejeté
Portable (CPU)	3 à 10 tokens/s pour un 8B quantifié (estimation)	Local	Environ 1 400 h	Tests de fumée seulement

Conséquences pour la reproductibilité.

- **Identifiants non datés.** Les identifiants du catalogue NVIDIA ne portent pas de date. Pour chaque appel, on journalise le champ `model` renvoyé, l'éventuel `system_fingerprint` , les tokens et la latence. Un jeu de contrôle fixe de 5 items de développement est rejoué chaque

jour de campagne, et l'on compare les sorties et leurs longueurs. Les 3 répétitions d'un modèle s'enchaînent dans une fenêtre courte, et chaque réponse est sauvegardée.

- **Usage des données et items privés.** Les items privés (N1, N2, N5) ne passent que par des routes sans conservation (NVIDIA, §2.3) ou par du calcul que l'on contrôle (portable, notebooks Kaggle privés). Ils ne passent jamais par l'offre gratuite de Gemini, par les points d'accès OpenRouter susceptibles d'entraîner sur les données, ni par AXLE (service hébergé). La clause §2.6(a) des conditions NVIDIA, qui demande de ne pas soumettre d'« informations confidentielles », reste à apprécier pour des items inédits (§9).
- **Dérive de versions des prouveurs.** Les prouveurs ont été entraînés sur Lean 4.9 à 4.15, alors que la v4.35.0 retire notamment `Lean.reduceBool`, `Lean.reduceNat` et `Lean.trustCompiler`. Une partie du gain de B3 viendra donc de la correction de lemmes renommés plutôt que d'erreurs logiques. Chaque échec est classé : `unknown identifier/constant`, `deprecated`, `type mismatch`, `unsolved goals`, `timeout` ou autre. Les gains de B3 sont rapportés avec et sans ces corrections de noms. La catégorie H-c du document 04 sera scindée en identifiants inventés et identifiants renommés, selon une règle de classement à fixer en P2.

> **Réglages de la campagne P3, validés par le fondateur le 18 novembre 2026**
>
> Ces réglages sont figés dans `harness/config/p3_baselines.toml` avant le premier appel sur les 40 problèmes figés :
>
> - **Échantillonnage :** température 0,6 et top_p 0,95, pour que les 5 échantillons partagés par B0, B1 et B2 diffèrent ; graine dérivée par échantillon et envoyée au fournisseur.
> - **Longueur :** `max_tokens` = 32 000, raisonnement compris (le pilote a mesuré 11 806 tokens sur un problème de développement).
> - **Fournisseur :** Nvidia imposé, sans bascule silencieuse ; aucune invite système ; effort de raisonnement par défaut.
> - **Sélection B2 :** pour les réponses fermées, vote majoritaire entre les 5 réponses, à équivalence SymPy et numérique près, sans accès à la clé. Pour les preuves, B2 = B1 par construction.
> - **Correction :** un verdict « à auditer » (désaccord symbolique/numérique) compte comme un échec jusqu'à l'audit manuel. Délai de `lake comparator` : 600 s.
> - **Volume et contrôle :** R = 3, k = 5, soit 600 requêtes sur les 40 problèmes ; deux problèmes de développement (DEV-03, DEV-04) sont rejoués chaque jour pour détecter un changement silencieux du modèle.

5. CHAÎNE DE VÉRIFICATION

Étape	Outil	Où	Coût mesuré ou estimé	Ce qui est vérifié
Filtre de source	Recherche textuelle des interdits (§3)	WSL2 (Ubuntu)	Négligeable	Absence de `sorry`, `axiom`, `native_decide`, etc.
Boucle	LeanInteract + REPL v4.35.0, environnement Mathlib/Physlib gardé chargé	WSL2 (Ubuntu 26.04) ; Windows écarté, environ 30 fois plus lent (voir ci-dessous)	1 à 5 s par preuve typique avec réutilisation de l'en-tête, soit 1 à 6 h par modèle ; 2 à 3 REPL Mathlib au plus avec 16 Go (estimations)	Erreurs, `sorry` restants, buts non résolus, renvoyés au LLM avec les verdicts SymPy
Contrôle d'axiomes	`#print axioms`	Idem	Mesuré : le test de fumée ne rapporte que `propext`, `Classical.choice`, `Quot.sound`	Liste blanche
Certification	`lake comparator`, seulement sur les preuves acceptées par le REPL	WSL2, bac à sable `bwrap`	Mesuré le 18 novembre 2026 sur le portable : 13 à 20 s par cas, compilation du défi et de la solution comprise	Même énoncé que le fichier gelé, axiomes permis, acceptation par le noyau
Lot final	`lake check --paranoid` (noyaux externes fournis : leanchecker-paranoid, lean4lean, nanoda, con-leche)	WSL2	Une fois par campagne	Rejeu indépendant par le noyau

Pourquoi WSL2 et pourquoi garder Lean chargé. Sur le portable, l'élaboration à froid du fichier de fumée, qui importe la trigonométrie de Mathlib et `Physlib.Units.Dimension`, a pris 171 s sous Windows mais **5,9 s sous WSL2** (Ubuntu 26.04, même manifeste, même verdict d'axiomes). L'installation complète (Lean, cache Mathlib, compilation) a pris environ 8 min sous WSL2 contre environ 26 min sous Windows. La boucle tourne donc entièrement sous WSL2. Même à 6 s par fichier, recompiler à chaque tentative resterait coûteux à grande échelle : Lean est gardé chargé dans le REPL. Recompiler un fichier à chaque tentative coûterait donc de l'ordre de 120 à 360 heures pour 2 500 à 7 500 vérifications (simple calcul d'ordre de grandeur). L'installation initiale a pris environ 26 minutes, dont environ 14 pour récupérer les dépendances et le cache de Mathlib et environ 5 pour la compilation. Chaque vérification a un délai maximal, car les tactiques lourdes

(`nlinarith` , `decide`) dominent les temps ; un dépassement au stade de la certification compte comme un échec de certification et il est rapporté.

Ce que vérifie `lake comparator` . Son aide (`lake comparator --help` , vérifiée en local dans Lean v4.35.0) indique qu'il établit trois points : chaque théorème nommé de la solution prouve le même énoncé que le défi, n'utilise aucun axiome hors de la liste permise, et est accepté par le noyau. Le projet non fiable est construit dans un bac à sable `bwrap` (bubblewrap), disponible seulement sous Linux. Pour les items dont l'énoncé contient une définition à compléter (`definition_names`), la fidélité doit en plus être relue par un humain.

> **`lake comparator` testé sous WSL2 (18 novembre 2026)**
>
> Environnement : WSL2, Ubuntu 26.04.1, bubblewrap 0.11.1, elan 4.4.0, Lean v4.35.0, même `lake-manifest.json` que sous Windows. Trois cas jouets sur un même énoncé (`bench/lean/AMPBench/Demo` , configurations dans `bench/lean/comparator/`), exécutés dans le bac à sable `bwrap` , sans l'option `--inadvisably-no-sandbox` :
>
> - preuve correcte : **acceptée** (code 0, « Lean default kernel accepts the solution »), 20 s ;
> - preuve contenant `sorry` : **refusée** (code 1, « Illegal axiom detected: 'sorryAx' »), 13 s ;
> - énoncé modifié (≤ au lieu de =) : **refusée** (code 1, « Challenge and solution theorem statement do not match »), 13 s.
>
> Les contre-exemples `axiom` ajouté et `native_decide` restent à tester au jour 1.

6. SOURCES DU BENCHMARK ET LICENCES

Niveau	Source	Licence	Contraintes
N1, N2 (12 items)	Rédigés par AIM²P, privés jusqu'à l'évaluation	Licence des données à fixer (§9) ; les documents du dépôt sont sous CC BY 4.0	Empreintes SHA-256 publiées au gel
N3 (8)	google-deepmind/miniF2F, ou miniF2F-v2c	Apache-2.0 (vérifié en P0). Pour v2c, les métadonnées GitHub du dépôt indiquent Apache-2.0 au 18 nov., mais la page arXiv mentionne CC BY 4.0, peut-être pour le seul article : **à confirmer** sur le fichier de licence	Audit de 2026 : 193 signalements et 8 défauts certifiés sur les 488 énoncés de v2c ; relecture item par item ; portage vers v4.35.0
N4 (8)	ProofNet# ; PutnamBench, version Lean	MIT ; Apache-2.0 pour les versions Lean et Isabelle, MIT pour la version Rocq	ProofNet# ne vise que Lean 4.7.0 à 4.16.0-rc2 et doit être porté. Pour PutnamBench, les énoncés informels sont utilisés avec la permission de la MAA : seuls les énoncés Lean sont redistribués, et **aucune preuve PutnamBench n'est publiée**
N5 (12)	Rédigés par AIM²P, en s'appuyant sur Physlib	Comme N1 ; Physlib est sous Apache-2.0	Privés jusqu'à l'évaluation
Écartés	ProverBench ; lanzarini-krylov-lean	Aucune licence de jeu de données ; aucune licence ouverte	—

Licences des composants réutilisés (métadonnées GitHub, PyPI, fichiers de licence ou cartes de modèles, consultés le 18 novembre 2026) :

- **Apache-2.0 :** Lean 4, Mathlib, Physlib, REPL Lean, SafeVerify, Math-Verify, Goedel-Prover-V2-8B, Kimina-Prover-Distill-8B.
- **MIT :** LeanInteract, lean-lsp-mcp, LeanSage, kimina-lean-server, bayes_evals.
- **Autres licences ouvertes :** SymPy (BSD à 3 clauses), Isabelle (licence de type BSD), Rocq (LGPL-2.1), SageMath (GPL-3.0 pour la distribution, utilisé comme outil externe et non redistribué).
- **Licences de modèles particulières :** Nemotron 3 Ultra (`openmdw-1.1`), Kimi K3.1 Pro (licence propre `kimi-k3.1-pro`), DeepSeek-Prover-V2-7B (DeepSeek Model License, avec restrictions d'usage ; code sous MIT). Leurs conditions doivent être relues avant toute publication des sorties.
- **Non confirmées :** aucune licence n'est détectée par GitHub pour numina-lean-agent, et le dépôt d'AxProverBase n'a pas été localisé. Aucun code n'en sera repris avant confirmation.

7. RISQUES ET CONTRÔLES

Risque	Contrôle
Une offre gratuite change ou disparaît en cours de campagne : les conditions NVIDIA sont modifiables à tout moment, des modèles sont déjà passés en accès réservé aux partenaires (source secondaire), et les quotas gratuits de Gemini ont été fortement réduits (source secondaire)	Campagnes courtes, réponses sauvegardées, repli pré-enregistré, campagne relancée en entier sous une version datée (§4)
Mise à jour silencieuse d'un modèle	Modèle renvoyé journalisé, jeu de contrôle quotidien, préférence pour les poids ouverts
Accès NVIDIA décompté en crédits	Porte du jour 1 ; réductions de périmètre pré-déclarées dans le document 04 (§7)
Fuite d'items privés	Règle de routage du §4 ; aucun service de vérification hébergé (AXLE) ; notebooks Kaggle privés
Gains de B3 gonflés par la dérive de versions des prouveurs	Taxonomie d'erreurs ; résultats rapportés avec et sans corrections de noms
Certification sans bac à sable	WSL2 et `bwrap` installés avant toute évaluation gelée ; mode sans bac à sable réservé au développement
Ressources du portable (16 Go, pas de GPU)	Lean gardé chargé, 2 à 3 REPL au plus, délais maximaux ; génération hors du portable
Traîne de durée de `lake comparator`	Lancé seulement sur les preuves acceptées par le REPL ; délai maximal ; dépassements rapportés
Vérification téléphonique refusée (NVIDIA, Kaggle)	**Constatée pour NVIDIA le 18 novembre 2026** ; routes sans téléphone retenues (OpenRouter ; Google Colab à tester)
Évolution de Lean et Mathlib (v4.36.0 publiée le 16 nov.)	Aucune mise à jour pendant une campagne ; nouvelle version datée sinon
Licence de miniF2F-v2c incertaine	google-deepmind/miniF2F par défaut
SymPy rend un résultat faux mais bien formé	Dispositif du document 04 (§4) : distinction entre verdict certifiable et heuristique, tests numérique et différentiel

8. CHECK-LIST « JOUR 1 »

À faire par le fondateur avant le gel (environ une demi-journée) :

1. **WSL2 : fait le 18 novembre 2026.** Ubuntu 26.04.1, bubblewrap 0.11.1, elan 4.4.0 et Lean v4.35.0 installés ; `bench/lean` copié dans `~/aim2p-bench`, cache Mathlib récupéré et projet compilé en environ 8 min.
2. `lake comparator` **: en partie fait.** Accepté pour une preuve correcte, refusé pour `sorry` et pour un énoncé modifié (voir §5). Restent les contre-exemples `axiom` ajouté et `native_decide`.
3. **Boucle :** installer le REPL au tag v4.35.0 et LeanInteract 0.11.6 ; chronométrer le chargement de Mathlib plus 20 preuves, et noter la RAM maximale avec 1, 2 puis 3 REPL.
4. **NVIDIA : échec le 18 novembre 2026.** Le code de vérification par SMS n'est jamais arrivé, malgré plusieurs essais. Route de remplacement : OpenRouter, avec le même modèle (encadré « Jour 1 »). Un nouvel essai NVIDIA avec un autre numéro reste possible.
5. **Kaggle : à risque**, car ses GPU exigent aussi une vérification téléphonique. Tester d'abord Google Colab (compte Google, GPU T4) avec Goedel-Prover-V2-8B quantifié en 4 bits ; à défaut, exécution sur processeur sur un périmètre réduit.
6. **Python :** créer un environnement virtuel avec SymPy 1.14.0, mpmath 1.3.0, Math-Verify 0.9.0 (et son runtime ANTLR4) et LeanInteract 0.11.6, et vérifier l'installation sous Python 3.14.7.
7. **Pilote :** travailler sur des items de développement ou publics uniquement (par exemple 10 énoncés miniF2F revérifiés sous v4.35.0 et des items de la réserve Krylov), **jamais sur les 40 items gelés**. Mesurer le taux de compilation B0 du modèle généraliste et du prouveur, et classer les erreurs.

8. **Registre de gel :** consigner les identifiants de modèles, les dates, les limites relevées dans chaque console, les paramètres d'échantillonnage et les empreintes de la chaîne d'outils (`lean-toolchain` , `lake-manifest.json` , versions Python).

9. DÉCISIONS RESTANT OUVERTES

1. **Confirmer Nemotron 3 Ultra comme LLM de l'analyse principale** après la porte du jour 1, ou choisir Kimi K3.1 Pro ou Mistral Large 4 sur le même catalogue.
2. **Choisir deux ou trois modèles.** Le couple minimal est Nemotron et Goedel ; un troisième modèle, décrit seulement à titre descriptif, alourdit le budget et entre en tension avec la règle anti-dérive.
3. **Fixer le périmètre du prouveur.** Il ne concerne que les items formels (N3, N4 et les items formels de N5). Reste à définir ce que deviennent pour lui B3 et l'ablation A2, puisqu'il ne produit ni dérivation informelle ni plan critiqué par SymPy.
4. **Items privés et OpenRouter : tranché le 18 novembre 2026.** Les 40 problèmes, inédits compris, sont envoyés via l'offre gratuite d'OpenRouter ; le risque d'entraînement de futurs modèles est déclaré dans `bench/FREEZE.md` .
5. **Volume sous 50 requêtes par jour :** étaler la campagne P3 (environ un mois pour le protocole complet), réduire le nombre de répétitions, ou ajouter une seconde route (Groq) pour un second modèle.
6. **Porter ou non SafeVerify vers v4.35** comme second avis.
7. **Adopter ou non l'en-tête** `module` pour les fichiers d'items (en P2, S3).
8. **Choisir la source de N3 :** google-deepmind/miniF2F ou miniF2F-v2c, une fois la licence de ce dernier confirmée.
9. **Fixer les licences d'AMP-Bench-mini et du code** (`bench/` , `harness/` , `engine/`). La licence CC BY 4.0 actuelle couvre les documents, pas le logiciel.
10. **Environnement de la boucle : tranché par la mesure.** Tout sous WSL2, car l'élaboration y est environ 30 fois plus rapide que sous Windows (5,9 s contre 171 s pour le fichier de fumée).

RÉFÉRENCES

- **Protocole AIM²P :** document 02, document 03, document 04 ; notes de recherche internes de P0 et de P1 (18 novembre 2026 : offres gratuites, prouveurs ouverts, outillage Lean), non publiées à ce stade.
- **Offres de modèles :** NVIDIA, liste des modèles ; NVIDIA, conditions d'essai ; Google AI Studio, limites, tarifs et conditions ; Groq ; OpenRouter ; Mistral, limites d'usage ; Leanstral 1.5 ; GitHub Models ; Cloudflare Workers AI ; SambaNova ; Cohere ; Hugging Face Inference Providers ; ZeroGPU ; Colab ; Kaggle ; MathArena.
- **Cartes de modèles :** Nemotron 3 Ultra ; Kimi K3.1 Pro ; Goedel-Prover-V2-8B ; Kimina-Prover-Distill-8B ; DeepSeek-Prover-V2-7B ; Leanstral 1.5.
- **Lean et vérification :** notes de version v4.35.0 ; Lean 4 ; Mathlib4 ; Physlib ; Comparator, archivé ; REPL ; LeanInteract ; SafeVerify ; kimina-lean-server ; lean-lsp-mcp ; LeanSage ; Kimina Lean Server (arXiv:2504.21230) ; AXLE (arXiv:2606.26442).
- **Calcul formel et évaluation :** SymPy et son étiquette `simplify` ; SageMath ; Math-Verify ; bayes_evals.
- **Autres assistants de preuve :** Rocq ; Isabelle ; AFP Physical_Quantities ; Sledgehammer ; SQIR.
- **Benchmarks :** google-deepmind/miniF2F ; miniF2F-v2 (arXiv:2511.03108) ; ProofNet# ; PutnamBench ; audit de 2026 (arXiv:2606.29493) ; Lean4PHYS (arXiv:2510.26094).

HISTORIQUE DES VERSIONS

- **v0.1 (18 novembre 2026) :** première version, livrable de la phase P1. Elle fige la chaîne Lean (v4.35.0, Mathlib et Physlib au tag v4.35.0), SymPy 1.14.0 et la chaîne de vérification (REPL et LeanInteract dans la boucle, `lake comparator` sous WSL2 pour la certification). Elle pré-sélectionne les modèles sous contrainte de gratuité, avec des règles de bascule, et déclare un écart par rapport au document 04 (SafeVerify retiré de la boucle). Elle s'appuie sur les notes de recherche P1 et sur des constats locaux du 18 novembre 2026. Encadré « Jour 1 » ajouté le même jour : route OpenRouter après l'échec de la vérification NVIDIA, pilote mesuré de bout en bout.