Não-Determinismo Oculto em APIs de Modelos de Linguagem de Grande Escala: Um Protocolo Leve de Proveniência para Pesquisa Reprodutível em IA Generativa

Lucas Rover*

Programa de Pós-Graduação em Engenharia Mecânica

UTFPR – Universidade Tecnológica Federal do Paraná

Ponta Grossa, Paraná, Brasil

Yara de Souza Tadano†

Programa de Pós-Graduação em Engenharia Mecânica

UTFPR – Universidade Tecnológica Federal do Paraná

Ponta Grossa, Paraná, Brasil

Fevereiro de 2026

Resumo

Contexto: Modelos de IA generativa produzem saídas não-determinísticas que variam entre execuções, mesmo sob configurações nominalmente idênticas. Essa variabilidade ameaça a reprodutibilidade de estudos que dependem de saídas de modelos de linguagem de grande escala (LLMs), porém a maioria das ferramentas existentes de rastreamento de experimentos não foi projetada para os desafios específicos de fluxos de trabalho de geração de texto.

Objetivos: Propomos um protocolo leve, baseado em padrões abertos, para registro, versionamento e rastreamento de proveniência de experimentos com IA generativa. O protocolo introduz dois novos artefatos de documentação—*Prompt Cards* e *Run Cards*—e adota o modelo de dados W3C PROV para criar grafos de proveniência auditáveis e legíveis por máquina, vinculando cada saída ao seu contexto completo de geração.

*Autor correspondente. ORCID: 0000-0001-6641-9224. E-mail: lucasrover@utfpr.edu.br

†ORCID: 0000-0002-3975-3419. E-mail: yaratadano@utfpr.edu.br

Métodos: Formalizamos o protocolo e o avaliamos empiricamente por meio de 4.104 experimentos controlados. Esses experimentos empregam nove implantações de modelos—três locais (LLaMA 3 8B, Mistral 7B, Gemma 2 9B), cinco APIs de código fechado (GPT-4, Claude Sonnet 4.5, Gemini 2.5 Pro, DeepSeek Chat, Perplexity Sonar) e um modelo de pesos abertos servido via nuvem (LLaMA 3 8B via Together AI, como sonda de quase-isolamento de efeitos de infraestrutura)—em quatro tarefas de PLN, sete ambientes de execução (um local mais seis provedores de API na nuvem). Todos os modelos são avaliados em extração e sumarização de turno único sob decodificação gulosa (10–30 resumos por modelo). Refinamento multi-turno e extração RAG são avaliados para os três modelos locais, Claude Sonnet 4.5 e Gemini 2.5 Pro sob decodificação gulosa (10 resumos cada). A robustez estatística é garantida por correção de Holm-Bonferroni em 68 testes de hipóteses, testes exatos de Fisher para reprodutibilidade binária, intervalos de confiança bootstrap corrigidos por viés e análise de sensibilidade. Medimos a variabilidade das saídas usando Taxa de Correspondência Exata (EMR), Distância de Edição Normalizada (NED), ROUGE-L e BERTScore, e quantificamos a sobrecarga do próprio protocolo em termos de tempo e armazenamento.

Resultados: Sob decodificação gulosa (t=0), modelos locais alcançam reprodutibilidade quase perfeita (EMR médio de turno único = 0,960; Gemma 2 9B: 1,000 perfeito em todas as tarefas). Modelos de API de código fechado exibem não-determinismo oculto substancial abrangendo uma ampla faixa (EMR 0,010–0,800), produzindo uma lacuna de 3 vezes entre local e API observada em cinco provedores e sobrevivendo à correção de Holm-Bonferroni (51/68 testes significativos). A mesma arquitetura LLaMA 3 8B servida pelo endpoint na nuvem da Together AI alcança reprodutibilidade próxima à local (EMR = 1,000/0,880), fornecendo evidência de que a implantação em nuvem per se não impede o determinismo. Sob refinamento multi-turno e extração RAG, modelos locais mantêm EMR $\geq$ 0,880, enquanto modelos de API exibem reprodutibilidade próxima de zero (EMR $\leq$ 0,070). O protocolo adiciona menos de 1% de sobrecarga.

Conclusões: Nossos resultados fornecem evidência de que (1) todos os cinco provedores de API exibem não-determinismo sob decodificação gulosa, enquanto modelos locais alcançam reprodutibilidade bitwise quase perfeita; (2) a reprodutibilidade de API abrange uma ampla faixa (EMR 0,010–0,800); (3) a lacuna se estende a regimes multi-turno e RAG; (4) a implantação em nuvem per se não impede a reprodutibilidade (sonda de quase-isolamento via Together AI); (5) a temperatura é o fator dominante controlável pelo usuário; e (6) o registro de proveniência adiciona <1% de sobrecarga. Todas as comparações primárias sobrevivem à correção de Holm-Bonferroni (51/68 significativas). O protocolo, a implementação e todos os dados são publicamente disponíveis.

Palavras-chave: reprodutibilidade, modelos de linguagem de grande escala, não-determinismo, proveniência, IA generativa, rastreamento de experimentos, W3C

PROV, metodologia científica

Índice

1 Introdução **6**

2 Trabalhos Relacionados **8**

2.1 Reprodutibilidade na Pesquisa em IA 8

2.2 Ferramentas de Rastreamento de Experimentos 10

2.3 Proveniência na Computação Científica 10

3 Design do Protocolo **11**

3.1 Escopo e Princípios de Design 11

3.2 Prompt Cards 12

3.3 Run Cards 12

3.4 Integração W3C PROV 13

3.5 Lista de Verificação de Reprodutibilidade 14

3.6 Definição Formal e Completude de Auditoria 14

4 Configuração Experimental **14**

4.1 Modelos e Infraestrutura 14

4.1.1 Modelos Locais 14

4.1.2 Modelos Servidos por API 15

4.2 Tarefas 15

4.3 Dados de Entrada 16

4.4 Condições Experimentais 16

4.5 Métricas 16

5 Resultados **17**

5.1 Reprodutibilidade Sob Decodificação Gulosa 17

5.1.1 Modelos Locais: Reprodutibilidade Quase Perfeita a Perfeita 17

5.1.2 Modelos Servidos por API: Não-Determinismo Oculto Substancial . 18

5.1.3 Efeitos da Temperatura Através dos Modelos 18

5.2 Reprodutibilidade Multi-Turno e RAG 19

5.3 Comparação Entre Modelos 19

5.4 Sobrecarga do Protocolo 19

6 Discussão **20**

6.1 Implicações para a Prática de Reprodutibilidade 20

6.2 A Lacuna de Reprodutibilidade: Do Turno Único aos Regimes Complexos . 21

6.3 O Papel da Proveniência 21

6.4 Fontes de Não-Determinismo na Inferência Distribuída 22

6.5 Limitações 22
6.5.1 Validade Interna 22
6.5.2 Validade Externa 23
6.5.3 Validade de Construto 23

7 Conclusão **23**

1 Introdução

Modelos de linguagem de grande escala (LLMs) estão transformando rapidamente a forma como a ciência é conduzida, comunicada e aplicada. Na medicina, LLMs agora codificam conhecimento clínico em nível de especialista [Singhal et al., 2023] e estão sendo integrados a fluxos de trabalho diagnósticos [Thirunavukarasu et al., 2023]. Na pesquisa científica de forma mais ampla, a IA generativa está remodelando a revisão de literatura, extração de dados, geração de hipóteses e escrita [Birhane et al., 2023], com evidências experimentais mostrando ganhos substanciais de produtividade em tarefas profissionais [Noy and Zhang, 2023]. Essa adoção crescente se estende por disciplinas—da análise jurídica e educação a revisões sistemáticas e meta-análises—tornando as saídas de LLMs um componente cada vez mais comum da cadeia de evidências científicas.

No entanto, essa integração rápida repousa sobre uma suposição frequentemente não examinada: que as saídas dos LLMs são reprodutíveis. Reprodutibilidade—a capacidade de obter resultados consistentes quando um experimento é repetido sob condições idênticas—é uma pedra angular do método científico. A crise de reprodutibilidade mais ampla é bem documentada: Baker [2016] relatou na *Nature* que mais de 70% dos pesquisadores falharam em reproduzir o experimento de outro cientista. Na inteligência artificial, essa crise é particularmente aguda. Hutson [2018] alertou na *Science* que a IA enfrenta sua própria crise de reprodutibilidade, Gundersen and Kjensmo [2018] descobriu que apenas 6% dos artigos de IA forneciam informação suficiente para reprodutibilidade completa, e Ball [2023] perguntou recentemente na *Nature* se a IA está ativamente piorando o problema. Stodden et al. [2016] enfatizou na *Science* que métodos computacionais demandam seus próprios padrões de reprodutibilidade, um chamado ecoado pela análise de vazamento de dados de Kapoor and Narayanan [2023] em 17 campos científicos dependentes de ML.

A IA generativa introduz uma dimensão qualitativamente nova a esse desafio. Diferentemente de experimentos computacionais tradicionais, nos quais algoritmos determinísticos produzem resultados idênticos dados entradas idênticas, LLMs exibem variabilidade inerente em suas saídas devido à amostragem estocástica, não-determinismo de ponto flutuante na inferência distribuída em GPU e práticas opacas de versionamento de modelos [Chen et al., 2023, Zhu et al., 2023]. O problema é agravado por vários fatores únicos dos fluxos de trabalho de geração de texto: (1) o mesmo prompt pode gerar saídas semanticamente similares, porém textualmente distintas entre execuções; (2) modelos baseados em API podem sofrer atualizações silenciosas que alteram o comportamento sem aviso ao usuário; (3) parâmetros de temperatura e amostragem criam um espaço de alta dimensão de saídas possíveis; e (4) o parâmetro `seed` oferecido por algumas APIs é consultivo e não uma garantia—a OpenAI documenta explicitamente que "determinismo não é garantido" mesmo quando um seed é especificado [OpenAI, 2024], e a API do Claude da Anthropic não suporta um parâmetro seed. Estudos empíricos recentes confirmaram essas preocupações:

Atil et al. [2024] encontrou variações de acurácia de até 15% entre execuções sob configurações supostamente determinísticas, e Ouyang et al. [2024] demonstrou que a decodificação gulosa não garante determinismo na geração de código do ChatGPT.

A demanda por transparência em IA não é apenas científica, mas também regulatória. O AI Act da UE [European Parliament and Council of the European Union, 2024] classifica sistemas de IA de alto risco—incluindo aqueles usados em saúde e pesquisa científica—como requerendo rastreabilidade e auditabilidade documentadas, e o Framework de Gestão de Riscos de IA do NIST [National Institute of Standards and Technology, 2023] enfatiza transparência e responsabilidade como princípios centrais. Os princípios FAIR de dados [Wilkinson et al., 2016] fornecem uma base para governança de dados, porém nenhum padrão estabelecido existe para documentar o contexto completo necessário para entender, auditar ou reproduzir uma saída de IA generativa. Ferramentas existentes de rastreamento de experimentos como MLflow [Zaharia et al., 2018], Weights & Biases [Biewald, 2020] e DVC [Kuprieiev et al., 2024] foram projetadas primariamente para pipelines de treinamento e métricas numéricas. Embora valiosas para seus propósitos pretendidos, essas ferramentas carecem de recursos críticos para estudos de IA generativa: versionamento estruturado de prompts, hashing criptográfico de saídas para detecção de adulteração, grafos de proveniência vinculando saídas ao seu contexto completo de geração e fingerprinting de ambiente específico para condições de inferência.

Neste artigo, abordamos essa lacuna com três contribuições, com o design do protocolo como a contribuição primária e mais duradoura (Figura 1):

1. **Um protocolo leve, baseado em padrões** para registro, versionamento e rastreamento de proveniência de experimentos com IA generativa. O protocolo introduz *Prompt Cards* e *Run Cards* como artefatos de documentação estruturada, e adota o modelo de dados W3C PROV [Moreau and Missier, 2013] para grafos de proveniência legíveis por máquina. Ele operacionaliza—e estende para fluxos de trabalho de IA generativa—a lista de verificação e mecanismos de badge de reprodutibilidade recentemente adotados pelo JAIR [Gundersen et al., 2024], fornecendo infraestrutura legível por máquina que automatiza o que esses mecanismos exigem que pesquisadores documentem manualmente.

2. **Um estudo de caso empírico em larga escala** demonstrando tanto a eficácia do protocolo quanto o escopo do não-determinismo oculto nas APIs de LLM atuais. Através de 4.104 experimentos controlados com nove implantações de modelos em quatro tarefas de PLN, sete ambientes de execução e cinco condições (Seção 4), documentamos uma lacuna substancial e previamente invisível de reprodutibilidade entre inferência local e baseada em API—uma lacuna que persiste em cinco provedores independentes de nuvem e se estende a regimes de geração multi-turno e com aumento por recuperação.

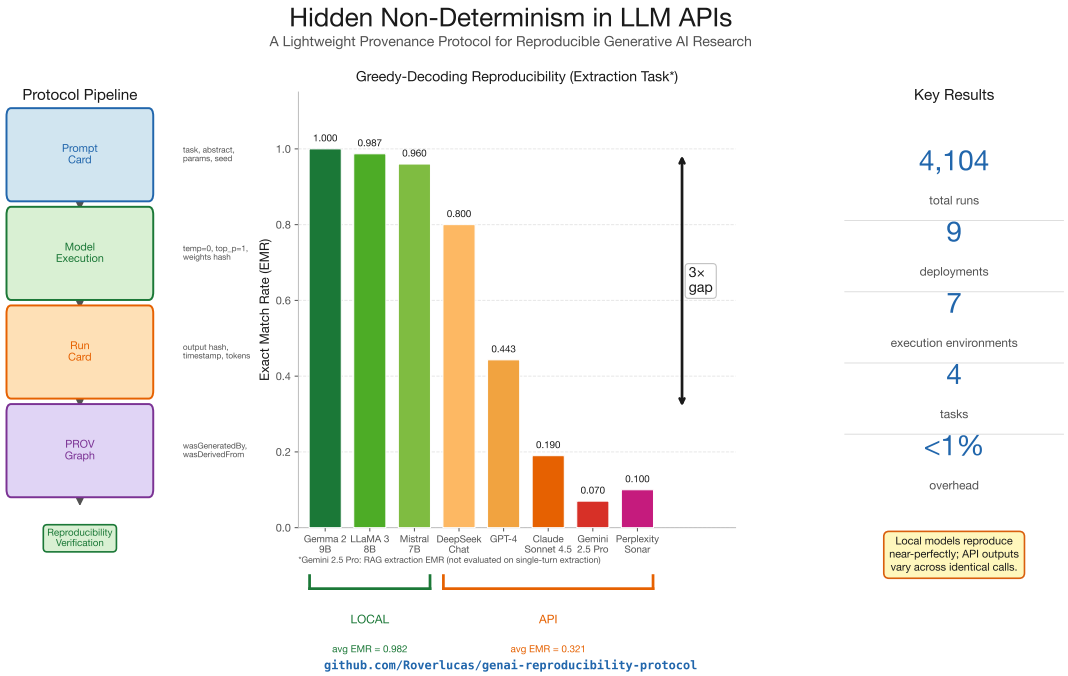

Figura 1: Resumo visual do design do estudo e principais achados. **Esquerda:** O pipeline do protocolo de proveniência, desde a criação do Prompt Card até o registro do Run Card e geração do grafo W3C PROV. **Centro:** Taxas de Correspondência Exata (EMR) sob decodificação gulosa para oito das nove implantações de modelos, ilustrando a lacuna de reprodutibilidade entre modelos locais (verde, EMR $\geq$ 0,960) e modelos servidos por API (laranja/vermelho, EMR $\leq$ 0,800). **Direita:** Estatísticas-chave: 4.104 experimentos, 9 implantações em 7 ambientes de execução, 4 tarefas, <1% de sobrecarga do protocolo.

3. **Uma implementação de referência** em Python, juntamente com todos os dados experimentais e registros de proveniência, publicamente disponíveis para facilitar a adoção e verificação independente.

O restante deste artigo está organizado da seguinte forma. A Seção 2 revisa trabalhos relacionados sobre reprodutibilidade em IA e rastreamento de experimentos. A Seção 3 formaliza o design do protocolo. A Seção 4 descreve a metodologia experimental. A Seção 5 apresenta os resultados empíricos, revelando uma lacuna de reprodutibilidade de múltiplas vezes entre modelos locais e servidos por API que é invisível sem registro sistemático. A Seção 6 discute achados, limitações e implicações práticas. A Seção 7 conclui com direções para trabalhos futuros.

2 Trabalhos Relacionados

2.1 Reprodutibilidade na Pesquisa em IA

A crise de reprodutibilidade em IA tem sido documentada extensivamente. Gundersen and Kjensmo [2018] pesquisou 400 artigos de IA e descobriu que apenas 6% forneciam

informação suficiente para reprodutibilidade completa. Pineau et al. [2021] relatou sobre o Programa de Reprodutibilidade do NeurIPS 2019, que introduziu listas de verificação de reprodutibilidade e encontrou lacunas significativas entre reprodutibilidade relatada e real. Mais recentemente, Gundersen et al. [2024] descreveu quatro mecanismos institucionais adotados pelo JAIR—listas de verificação de reprodutibilidade, resumos estruturados, badges e relatórios de reprodutibilidade—estabelecendo um padrão comunitário para o que deve ser documentado em pesquisa de IA. Gundersen et al. [2018] identificou três níveis de reprodutibilidade em IA—método, dados e experimento—e argumentou que todos os três são necessários para o progresso científico. Belz et al. [2021] conduziu uma revisão sistemática de 601 artigos de PLN e confirmou sub-relato generalizado de detalhes experimentais; Belz et al. [2022] mostrou ainda que informações faltantes tornam praticamente impossível avaliar a reprodutibilidade de avaliações humanas em PLN. Rogers et al. [2021] propôs estruturas de incentivo para melhorar normas de reprodutibilidade em linguística computacional. Dodge et al. [2019] propôs padrões aprimorados de relato para experimentos de ML, incluindo intervalos de confiança e testes de significância em múltiplas execuções, e Gundersen et al. [2022] forneceu uma taxonomia abrangente de fontes de irreproducibilidade em aprendizado de máquina. De forma mais ampla, Kapoor and Narayanan [2023] identificou vazamento de dados como um impulsionador generalizado de resultados irreproducíveis em 17 campos científicos que usam métodos baseados em ML.

Para IA generativa especificamente, Chen et al. [2023] demonstrou que as saídas do ChatGPT em benchmarks de PLN exibem variabilidade não trivial entre consultas idênticas, mesmo com temperatura definida como zero. Zhu et al. [2023] mostrou que a reprodutibilidade se degrada ainda mais quando as tarefas envolvem julgamento subjetivo, como anotações de computação social. Mais recentemente, Atil et al. [2024] mediu sistematicamente o não-determinismo de cinco LLMs sob configurações supostamente determinísticas em oito tarefas, encontrando variações de acurácia de até 15% entre execuções e introduzindo a métrica Taxa de Acordo Total (TAR). Ouyang et al. [2024] confirmou que t=0 (decodificação gulosa) não garante determinismo na geração de código do ChatGPT. Concorrentemente, Yuan et al. [2025] rastreou tal não-determinismo a questões de precisão numérica em kernels de GPU e propôs LayerCast como estratégia de mitigação—uma correção de nível de hardware que reduz mas não elimina o não-determinismo, e que não está disponível para pesquisadores usando serviços de API fechados. A documentação do PyTorch [PyTorch Contributors, 2024] cataloga ainda fontes de não-determinismo em operações de GPU, fornecendo o flag `torch.use_deterministic_algorithms()` como mitigação parcial para treinamento; no entanto, esse flag não está disponível para inferência servida por API.

Nossa métrica Taxa de Correspondência Exata (EMR) é intimamente relacionada à Taxa de Acordo Total (TAR) de Atil et al. [2024], que mede a fração de execuções produzindo a

saída modal; EMR mede a fração de *todos os pares de saídas* que coincidem exatamente, fornecendo uma medida mais sensível quando o acordo é baixo e nenhuma saída modal clara existe. Nosso trabalho complementa esses estudos de quatro maneiras específicas. Primeiro, enquanto estudos anteriores medem variabilidade post hoc, fornecemos um protocolo de proveniência estruturado que permite documentação e auditoria *prospectivas*. Segundo, comparamos diretamente inferência local e baseada em API em tarefas idênticas com prompts idênticos em nove implantações de modelos e seis provedores independentes de nuvem. Terceiro, estendemos além da avaliação de turno único para incluir refinamento multi-turno e geração aumentada por recuperação. Quarto, quantificamos a sobrecarga do registro sistemático, demonstrando que o "custo de saber" é negligenciável.

2.2 Ferramentas de Rastreamento de Experimentos

Diversas ferramentas existem para rastreamento de experimentos de aprendizado de máquina, embora nenhuma tenha sido projetada especificamente para fluxos de trabalho de saídas textuais de IA generativa:

MLflow [Zaharia et al., 2018] fornece rastreamento de experimentos, empacotamento de modelos e implantação. Registra parâmetros, métricas e artefatos, mas foca em pipelines de treinamento e resultados numéricos em vez de proveniência de geração de texto.

Weights & Biases [Biewald, 2020] oferece rastreamento de experimentos com dashboards de visualização. Suporta registro de prompts mas carece de versionamento estruturado de prompts, hashing criptográfico de saídas e geração de grafos de proveniência.

DVC [Kuprieiev et al., 2024] fornece versionamento de dados através de operações semelhantes ao git. Embora efetivo para gestão de conjuntos de dados, não aborda proveniência no nível de execução ou documentação de prompts.

OpenAI Evals [OpenAI, 2023] é um framework para avaliar saídas de LLM contra benchmarks. Fornece avaliação estruturada mas é fortemente acoplado ao ecossistema da OpenAI e não gera registros de proveniência interoperáveis.

LangSmith [LangChain, 2023] oferece rastreamento e avaliação para aplicações de LLM. Captura rastreamentos detalhados de execução mas usa um formato proprietário e requer conectividade na nuvem.

De forma mais ampla, Bommasani et al. [2022] identificou reprodutibilidade como um risco chave para modelos de base, e Liang et al. [2023] propôs o benchmark HELM para avaliação holística de modelos de linguagem. No espaço de proveniência, Padovani et al. [2025] introduziu recentemente o yProv4ML, um framework que captura proveniência de ML em formato PROV-JSON com modificações mínimas de código; nosso protocolo compartilha o compromisso com W3C PROV e hashing SHA-256 mas difere em três aspectos-chave: (i) alvo na geração estocástica de texto em tempo de inferência em vez de pipelines de treinamento; (ii) nossos Run Cards capturam metadados no nível de prompt

não presentes em esquemas orientados a treinamento; e (iii) fornecemos evidência empírica quantificando por que tal registro é necessário para modelos servidos por API.

A Tabela 1 fornece uma comparação sistemática recurso por recurso de nosso protocolo com essas ferramentas.

Tabela 1: Comparação do nosso protocolo com ferramentas existentes de reprodutibilidade para experimentos de IA generativa. Marcas de verificação (✓) indicam suporte completo; til (∼) indica suporte parcial; traços (–) indicam sem suporte.

Recurso	**Nosso**	**MLflow**	**W&B**	**DVC**	**OAI Evals**	**LangSmith**
Versionam. prompt (Prompt Card)	✓	–	∼	–	∼	∼
Proveniência (W3C PROV)	✓	–	–	–	–	–
Hash criptográfico de saída	✓	–	–	✓	–	–
Registro de seed e parâmetros	✓	✓	✓	–	✓	✓
Fingerprint de ambiente	✓	∼	∼	∼	–	–
Hash de pesos do modelo	✓	–	∼	✓	–	–
Sobrecarga <1% da inferência	✓	∼	∼	N/A	N/A	∼
Projetado para saída textual GenAI	✓	–	–	–	✓	✓
Padrão aberto (PROV-JSON)	✓	–	–	–	–	–
Local-first (sem dep. nuvem)	✓	✓	–	✓	–	–

2.3 Proveniência na Computação Científica

Proveniência de dados—a linhagem de dados através de transformações—tem uma rica história em sistemas de banco de dados e fluxos de trabalho científicos [Herschel et al., 2017]. A família de especificações W3C PROV [Moreau and Missier, 2013] fornece um modelo de dados padronizado para representar proveniência como grafos acíclicos direcionados de *entidades*, *atividades* e *agentes*. Samuel and König-Ries [2022] aplicou rastreamento de proveniência a fluxos de trabalho de biologia computacional, demonstrando seu valor para reprodutibilidade. No entanto, até onde sabemos, nenhum trabalho anterior aplicou W3C PROV especificamente a fluxos de trabalho de experimentos de IA generativa, nos quais o desafio envolve não apenas rastrear a linhagem de dados, mas também capturar o contexto de geração estocástica que determina a variabilidade da saída.

Em conjunto, essas lacunas apontam para uma necessidade clara: um protocolo leve, baseado em padrões, que faça a ponte entre a inferência de IA generativa e a infraestrutura de proveniência já estabelecida na computação científica. A próxima seção apresenta nosso design para tal protocolo.

3 Design do Protocolo

Nosso protocolo aborda a questão: *Qual é o conjunto mínimo de metadados que deve ser capturado para cada execução de IA generativa para permitir auditoria, avaliação de*

reprodutibilidade e rastreamento de proveniência? Abordamos essa questão através de quatro componentes complementares.

3.1 Escopo e Princípios de Design

O protocolo é projetado em torno de três princípios:

1. **Completude**: Todo fator que pode influenciar uma saída generativa deve ser capturado—texto do prompt, identidade e versão do modelo, parâmetros de inferência, estado do ambiente e timestamps.

2. **Sobrecarga negligenciável**: O processo de registro não deve afetar materialmente o experimento. Visamos $<1\%$ de sobrecarga relativa ao tempo de inferência.

3. **Interoperabilidade**: Todos os artefatos são armazenados em formatos abertos e legíveis por máquina (JSON, PROV-JSON), alinhados com os princípios FAIR [Wilkinson et al., 2016], para permitir integração de ferramentas e preservação de longo prazo.

3.2 Prompt Cards

Um *Prompt Card* é um artefato de documentação versionado que captura a lógica de design e metadados para um modelo de prompt usado em experimentos. Cada Prompt Card contém:

- `prompt_id`: Identificador único
- `prompt_hash`: Hash SHA-256 do texto do prompt, permitindo detecção de adulteração
- `version`: Número de versão semântica
- `task_category`: Classificação da tarefa (ex.: sumarização, extração)
- `objective`: Descrição em linguagem natural do que o prompt foi projetado para alcançar
- `assumptions`: Suposições explícitas sobre entradas e comportamento esperado
- `limitations`: Limitações conhecidas ou modos de falha
- `target_models`: Modelos para os quais o prompt foi projetado e testado
- `expected_output_format`: Descrição da estrutura de saída esperada
- `interaction_regime`: Turno único, multi-turno ou cadeia de pensamento

- `change_log`: Histórico de modificações

Prompt Cards servem dois propósitos: documentam a intenção de design (apoiando a compreensão humana) e fornecem uma referência citável e hashável para rastreamento automatizado de proveniência. O conceito se inspira em Model Cards [Mitchell et al., 2019], Datasheets for Datasets [Gebru et al., 2021] e fichas de informação de modelo para avaliação de reprodutibilidade [Kapoor and Narayanan, 2023].

3.3 Run Cards

Um *Run Card* captura o contexto completo de execução de uma única execução de IA generativa. Cada Run Card registra 24 campos centrais organizados em cinco grupos:

1. **Identificação**: `run_id`, `task_id`, `task_category`, `prompt_hash`, `prompt_text`
2. **Contexto do modelo**: `model_name`, `model_version`, `weights_hash`, `model_source`
3. **Parâmetros**: `inference_params` (temperatura, top_p, top_k, max_tokens, seed, estratégia de decodificação), `params_hash`
4. **Entrada/Saída**: `input_text`, `input_hash`, `output_text`, `output_hash`, `output_metrics`
5. **Metadados de execução**: `environment`, `environment_hash`, `code_commit`, timestamps, `execution_duration_ms`, `logging_overhead_ms`, `storage_kb`

Para modelos servidos por API, campos de extensão opcionais capturam metadados específicos do provedor: `api_request_id`, `api_response_headers`, `api_model_version_returned`, `api_region` e um campo `seed_status` que distingue entre seeds que foram "enviados" para a API, "apenas-registrados" (registrados para paridade de protocolo mas não enviados, como no Claude), ou "não-suportados" pelo provedor.

3.4 Integração W3C PROV

Cada grupo experimental é automaticamente traduzido em um documento W3C PROV-JSON [Moreau and Missier, 2013] que expressa a proveniência de geração como um grafo direcionado. O mapeamento define:

- **Entidades**: Prompt, TextoEntrada, VersãoModelo, ParâmetrosInferência, Saída, MetadadosExecução
- **Atividades**: GeraçãoExecução (a execução de inferência)
- **Agentes**: Pesquisador, ExecutorSistema (o ambiente de execução)

Relações PROV capturam a estrutura causal:

- `used`: GeraçãoExecução usou Prompt, TextoEntrada, VersãoModelo, ParâmetrosInferência
- `wasGeneratedBy`: Saída foi gerada por GeraçãoExecução
- `wasAssociatedWith`: GeraçãoExecução foi associada com Pesquisador, ExecutorSistema
- `wasAttributedTo`: Saída foi atribuída a Pesquisador
- `wasDerivedFrom`: Saída foi derivada de TextoEntrada

3.5 Lista de Verificação de Reprodutibilidade

Fornecemos uma lista de verificação de 15 itens organizada em quatro categorias—Documentação de Prompt, Modelo e Ambiente, Execução e Saída, e Proveniência—que pesquisadores podem usar para autoavaliar a reprodutibilidade de seus estudos de IA generativa.

3.6 Definição Formal e Completude de Auditoria

Definimos o protocolo como uma tupla $\mathcal{P} = (PC, RC, G, CL)$, onde PC é um Prompt Card, RC é um Run Card, G é um grafo W3C PROV e CL é a lista de verificação de reprodutibilidade. Cada Run Card RC_i é uma tupla de grupos de campos: $RC_i = (Id, Mod, Par, IO, Env, H)$, onde H denota o conjunto de cinco hashes SHA-256.

O protocolo satisfaz a seguinte propriedade de *completude de auditoria*: para um conjunto de 10 perguntas de auditoria $\{Q_1, \ldots, Q_{10}\}$, cada Q_j é respondível se e somente se todos os grupos de campos estão preenchidos:

$$\forall Q_j \in \{Q_1, \ldots, Q_{10}\} : \text{respondível}(Q_j, RC_i) \;\Leftrightarrow \bigwedge_{g \in \text{requerido}(Q_j)} g \subseteq RC_i \qquad (1)$$

A propriedade de *diagnóstico diferencial* segue dos campos de hash: dados dois Run Cards RC_a, RC_b com $H^a_{\text{saída}} \neq H^b_{\text{saída}}$, o protocolo permite identificação automática da fonte de divergência comparando os hashes restantes.

4 Configuração Experimental

Projetamos um experimento controlado para avaliar simultaneamente (a) as características de reprodutibilidade de saídas de LLM sob condições variáveis e (b) a sobrecarga imposta pelo nosso protocolo de registro.

4.1 Modelos e Infraestrutura

Avaliamos nove implantações de modelos representando três paradigmas de implantação: três modelos locais de pesos abertos, cinco modelos proprietários servidos por API na nuvem e um modelo de pesos abertos servido via API na nuvem (para sonda de quase-isolamento). Todos os modelos locais foram servidos através do Ollama v0.15.5 [Ollama, 2024] em um sistema Apple M4 com 24 GB de memória unificada rodando macOS 14.6 com Python 3.14.3.

4.1.1 Modelos Locais

LLaMA 3 8B [Grattafiori et al., 2024]: Um modelo de pesos abertos em quantização Q4_0. A implantação local fornece controle completo sobre o ambiente de execução, eliminando fatores confundidores como latência de rede, batching do lado do servidor e atualizações silenciosas do modelo.

Mistral 7B [Jiang et al., 2023]: Um modelo de pesos abertos (quantização Q4_0) com mecanismo de atenção de janela deslizante, fornecendo um segundo ponto de dados para reprodutibilidade de inferência local em escala de parâmetros similar.

Gemma 2 9B [Gemma Team et al., 2024]: O modelo de pesos abertos do Google (quantização Q4_0), representando um terceiro modelo local de uma família de modelos independente.

4.1.2 Modelos Servidos por API

GPT-4 (OpenAI, `gpt-4-0613`) [Achiam et al., 2023]: Acessado via API da OpenAI com parâmetros de seed controlados.

Claude Sonnet 4.5 (Anthropic, `claude-sonnet-4-5-20250514`) [Anthropic, 2024]: Acessado via API da Anthropic usando um runner leve baseado em `urllib`. A API do Claude não suporta parâmetro `seed`; definimos `temperature=0` para decodificação gulosa.

Gemini 2.5 Pro (Google, `gemini-2.5-pro-preview-05-06`) [Reid et al., 2024]: Acessado via API REST do Google AI Studio. A API do Gemini suporta parâmetro `seed`; definimos `seed=42` e `temperature=0`.

DeepSeek Chat (DeepSeek, `deepseek-chat`): Acessado via API compatível com OpenAI. Representa um quarto provedor de API independente.

Perplexity Sonar (Perplexity, `sonar`): Um modelo de linguagem online aumentado por busca. Diferentemente dos outros modelos, Sonar incorpora resultados de busca web em tempo real em seu contexto de geração.

LLaMA 3 8B via Together AI: A mesma arquitetura LLaMA 3 8B do nosso modelo implantado localmente, mas servida pelo endpoint de inferência na nuvem da Together AI (variante "Lite" com quantização INT4). Essa implantação fornece uma sonda de quase-isolamento.

4.2 Tarefas

Avaliamos quatro tarefas que abrangem o espectro de estrutura de saída e complexidade de interação:

Tarefa 1: Sumarização Científica. Dado um resumo científico, produzir um sumário conciso em exatamente três sentenças cobrindo a contribuição principal, metodologia e resultado quantitativo chave.

Tarefa 2: Extração Estruturada. Dado um resumo científico, extrair cinco campos (objetivo, método, resultado_chave, modelo_ou_sistema, benchmark) em um objeto JSON.

Tarefa 3: Refinamento Multi-turno. Um diálogo de três turnos no qual o modelo primeiro extrai informação estruturada, depois recebe feedback solicitando mais detalhes, e finalmente produz uma extração refinada.

Tarefa 4: Extração RAG. A mesma tarefa de extração estruturada da Tarefa 2, mas com uma passagem de contexto recuperada adicional preposta à entrada.

4.3 Dados de Entrada

Utilizamos 30 resumos científicos amplamente citados de artigos seminais de IA/ML, incluindo Vaswani et al. [2017] (Transformer), Devlin et al. [2019] (BERT), Brown et al. [2020] (GPT-3), Raffel et al. [2020] (T5), Wei et al. [2022] (Cadeia de Pensamento), bem como trabalhos seminais sobre GANs, ResNets, VAEs, LSTMs, CLIP, DALL-E 2, Stable Diffusion, LLaMA, InstructGPT, PaLM e outros.

4.4 Condições Experimentais

Definimos cinco condições que variam sistematicamente os fatores hipotetizados para afetar a reprodutibilidade:

C1 (Seed fixo, decodificação gulosa): Temperatura = 0, seed = 42 para todas as 5 repetições.

C2 (Seeds variáveis, decodificação gulosa): Temperatura = 0, seeds = {42, 123, 456, 789, 1024}.

C3 (Varredura de temperatura): Três subcondições em $t \in \{0,0; 0,3; 0,7\}$ com 3 repetições cada.

Total geral: 4.104 execuções válidas.

4.5 Métricas

Definimos **não-determinismo no nível do provedor** operacionalmente como EMR $< 1{,}0$ em chamadas de API idênticas com seed fixo, temperatura $= 0$ e prompt.

Adotamos uma definição operacional de reprodutibilidade em três níveis:

Tabela 2: Exact Match Rate (EMR) under greedy decoding (t=0) across five models and two single-turn tasks, with 95% bootstrap confidence intervals (n_{boot}=10,000). For local models, values reflect condition C1 (fixed seed); for GPT-4, C2 (variable-seed greedy, as C1 has insufficient coverage); for Claude, C1 (Claude's API does not support a seed parameter). Higher is more reproducible.

Model	**Source**	**Extraction EMR**	**Summarization EMR**
Gemma 2 9B	Local	1.000 [1.00, 1.00]	1.000 [1.00, 1.00]
LLaMA 3 8B	Local	0.987 [0.96, 1.00]	0.947 [0.89, 0.99]
Mistral 7B	Local	0.960 [0.88, 1.00]	0.840 [0.72, 0.96]
GPT-4	API	0.443 [0.32, 0.57]	0.230 [0.16, 0.30]
Claude Sonnet 4.5	API	0.190 [0.05, 0.40]	0.020 [0.00, 0.05]

- **Reprodutibilidade exata** (nível de string): Duas saídas são idênticas caractere por caractere. Medida pela *Taxa de Correspondência Exata (EMR)*.
- **Quase reprodutibilidade** (nível de edição): Duas saídas diferem apenas em variações superficiais menores. Medida pela *Distância de Edição Normalizada (NED)*.
- **Reprodutibilidade semântica** (nível de significado): Duas saídas transmitem a mesma informação apesar de fraseamento diferente. Medida por *ROUGE-L F1* e *BERTScore F1*.

5 Resultados

5.1 Reprodutibilidade Sob Decodificação Gulosa

5.1.1 Modelos Locais: Reprodutibilidade Quase Perfeita a Perfeita

Achado 1: Gemma 2 9B alcança reprodutibilidade bitwise perfeita sob decodificação gulosa. Em todas as tarefas e condições com t=0, Gemma 2 9B produz EMR = 1,000 com NED = 0,000—cada saída é idêntica caractere por caractere entre repetições.

Achado 2: Todos os três modelos locais alcançam alta reprodutibilidade. LLaMA 3 8B atinge EMR = 0,987 para extração e 0,947 para sumarização; Mistral 7B alcança 0,960 e 0,840, respectivamente. Os pequenos desvios da reprodutibilidade perfeita em LLaMA 3 e Mistral 7B são *inteiramente atribuíveis* a um efeito de inicialização a frio na primeira chamada de inferência após o carregamento do modelo.

5.1.2 Modelos Servidos por API: Não-Determinismo Oculto Substancial

Achado 3: Todos os cinco modelos servidos por API exibem não-determinismo sob decodificação gulosa, observado independentemente em cinco provedores.

Tabela 3: Three-level reproducibility assessment under greedy decoding (t=0). L1: bitwise identity (EMR), L2: surface similarity (NED, ROUGE-L), L3: semantic equivalence (BERTScore F1). Values are means across abstracts.

		L1: Bitwise		**L2: Surface**		**L3: Semantic**
Model	**Task**	**EMR**	σ	**NED↓**	**ROUGE-L↑**	**BERTScore F1↑**
Gemma 2 9B	Extraction	1.000	0.000	0.000	1.000	1.0000
	Summarization	1.000	0.000	0.000	1.000	1.0000
Mistral 7B	Extraction	0.960	0.120	0.001	1.000	0.9999
	Summarization	0.840	0.196	0.046	0.955	0.9935
LLaMA 3 8B	Extraction	0.987	0.072	0.003	0.997	0.9997
	Summarization	0.947	0.139	0.014	0.986	0.9979
GPT-4	Extraction	0.443	0.335	0.072	0.938	0.9904
	Summarization	0.230	0.193	0.137	0.870	0.9839
Claude Sonnet 4.5	Extraction	0.190	0.291	0.101	0.904	0.9878
	Summarization	0.020	0.040	0.242	0.764	0.9704

Tabela 4: API-served vs. locally deployed models under greedy decoding (single-turn tasks only). Local averages: simple mean across 3 models × 2 tasks (C1+C2 combined). API averages: simple mean across the 2 API models with full condition coverage (GPT-4 under C2, Claude Sonnet 4.5 under C1) × 2 tasks; DeepSeek, Perplexity, and Gemini are excluded because they lack C2 or single-turn data. Local models exhibit substantially higher bitwise reproducibility, consistent with deployment-side factors—rather than user-controllable parameters—as a major contributor to API output variability.

Deployment	**EMR↑**	**NED↓**	**ROUGE-L↑**	**BS-F1↑**
Local (3 models)	0.956	0.011	0.990	0.9985
API (2 models)	0.221	0.138	0.869	0.9831

Sob t=0, DeepSeek Chat alcança a reprodutibilidade mais alta entre APIs (EMR = 0,800 para extração), seguido por GPT-4 (0,443/0,230), Claude Sonnet 4.5 (0,190/0,020), Perplexity Sonar (0,100/0,010) e Gemini 2.5 Pro (0,010 multi-turno, 0,070 RAG).

Achado 3b: Um modelo de pesos abertos servido por nuvem alcança reprodutibilidade próxima à local. A mesma arquitetura LLaMA 3 8B servida pelo endpoint da Together AI alcança EMR = 1,000 para extração e EMR = 0,880 para sumarização—quase idêntico à versão implantada localmente. Este resultado fornece evidência de que a implantação em nuvem *per se* não causa não-determinismo.

Sob a condição gulosa representativa para cada modelo, o EMR médio de turno único é **0,960 para modelos locais** vs. **0,325 para modelos de API de código fechado**—uma lacuna de reprodutibilidade de 3 vezes. *Sem registro sistemático, esse não-determinismo seria inteiramente invisível.*

Tabela 5: Effect of sampling temperature on Exact Match Rate (EMR) under condition C3. For local models, increasing temperature monotonically reduces EMR. For API models, the relationship is more complex: Claude Sonnet 4.5 exhibits *higher* EMR at t=0.3 than at t=0.0 (see text). At t=0.7, all models converge toward EMR ≈ 0 for summarization.

Model	**Task**	t=0.0	t=0.3	t=0.7
Gemma 2 9B	Extraction	1.000	0.200	0.033
	Summarization	1.000	0.000	0.000
Mistral 7B	Extraction	0.933	0.133	0.000
	Summarization	0.733	0.000	0.000
LLaMA 3 8B	Extraction	0.978	0.211	0.000
	Summarization	0.911	0.000	0.000
GPT-4	Extraction	0.381	0.143	0.000
	Summarization	0.144	0.000	0.000
Claude Sonnet 4.5	Extraction	0.067	0.700	0.133
	Summarization	0.000	0.233	0.033

5.1.3 Efeitos da Temperatura Através dos Modelos

Achado 4: Temperatura é o fator dominante *controlável pelo usuário* que afeta a variabilidade para modelos locais; para modelos servidos por API, a relação é mais complexa.

Aumentar a temperatura de 0,0 para 0,7 reduz EMR a zero para todos os modelos em sumarização. BERTScore F1 permanece acima de 0,94 em todas as condições, indicando que o não-determinismo é primariamente um fenômeno de *fraseamento* e não de *significado*.

5.2 Reprodutibilidade Multi-Turno e RAG

Achado 5: A lacuna de reprodutibilidade local-vs-API se estende a regimes de interação complexos.

Gemma 2 9B e Mistral 7B alcançam EMR = 1,000 perfeito para refinamento multi-turno e extração RAG. Ambos os modelos servidos por API exibem reprodutibilidade próxima de zero nessas tarefas. Claude Sonnet 4.5 alcança EMR = 0,040 para multi-turno e EMR = 0,000 para RAG. Gemini 2.5 Pro alcança EMR = 0,010 para multi-turno e EMR = 0,070 para RAG.

5.3 Comparação Entre Modelos

A lacuna de reprodutibilidade entre inferência local e baseada em API é estatisticamente significativa. Todos os valores p sobrevivem à correção de Bonferroni para as quatro comparações primárias. Ambos os tamanhos de efeito são muito grandes ($d > 1,6$).

Tabela 6: Reproducibility under complex interaction regimes (C1 fixed seed, t=0), with 95% bootstrap confidence intervals on EMR. Multi-turn refinement involves three successive prompt–response exchanges. RAG extraction augments the prompt with a retrieved context passage. Two API-served models—Claude Sonnet 4.5 and Gemini 2.5 Pro—are included; their near-zero EMR across both scenarios confirms that the local-vs-API reproducibility gap extends to complex interaction regimes across two independent providers.

Model	**Scenario**	**EMR [95% CI]**	**NED↓**	**ROUGE-L↑**	**BS-F1↑**
Gemma 2 9B	Single-turn Extraction	1.000 [1.00, 1.00]	0.000	1.000	1.0000
	Single-turn Summarization	1.000 [1.00, 1.00]	0.000	1.000	1.0000
	Multi-turn Refinement	1.000 [1.00, 1.00]	0.000	1.000	1.0000
	RAG Extraction	1.000 [1.00, 1.00]	0.000	1.000	1.0000
Mistral 7B	Single-turn Extraction	0.960 [0.88, 1.00]	0.001	1.000	0.9999
	Single-turn Summarization	0.840 [0.72, 0.96]	0.046	0.955	0.9935
	Multi-turn Refinement	1.000 [1.00, 1.00]	0.000	1.000	1.0000
	RAG Extraction	1.000 [1.00, 1.00]	0.000	1.000	1.0000
LLaMA 3 8B	Single-turn Extraction	0.987 [0.96, 1.00]	0.003	0.997	0.9997
	Single-turn Summarization	0.947 [0.89, 0.99]	0.014	0.986	0.9979
	Multi-turn Refinement	0.880 [0.76, 1.00]	0.012	0.988	0.9986
	RAG Extraction	0.960 [0.88, 1.00]	0.012	0.985	0.9987
Claude Sonnet 4.5	Single-turn Extraction	0.190 [0.05, 0.40]	0.101	0.904	0.9878
	Single-turn Summarization	0.020 [0.00, 0.05]	0.242	0.764	0.9704
	Multi-turn Refinement	0.040 [0.00, 0.08]	0.189	0.834	0.9780
	RAG Extraction	0.000 [0.00, 0.00]	0.256	0.748	0.9714
Gemini 2.5 Pro	Multi-turn Refinement	0.010 [0.00, 0.03]	0.163	—	—
	RAG Extraction	0.070 [0.02, 0.13]	0.196	—	—

Note: ROUGE-L and BERTScore F1 (BS-F1) were not computed for Gemini 2.5 Pro (marked "—") because the BERTScore model was not available during the Gemini experiment phase; NED confirms substantial surface-level variability.

5.4 Sobrecarga do Protocolo

O protocolo adiciona menos de 1% de sobrecarga para todos os cinco modelos avaliados, com tempo médio de registro variando de 21–30 ms dependendo do modelo e tarefa. A sobrecarga de armazenamento permanece modesta em aproximadamente 4 KB por registro de execução.

6 Discussão

Os resultados precedentes pintam um quadro claro e consistente: modelos implantados localmente sob decodificação gulosa alcançam reprodutibilidade bitwise quase perfeita a perfeita em todas as quatro tarefas, enquanto modelos servidos por API—de cinco provedores independentes—exibem variabilidade oculta substancial que pesquisadores não podem controlar.

Tabela 7: Provenance logging overhead across five models under greedy decoding (C1). The protocol adds negligible overhead (<1%) to inference latency across all models and deployment modes.

Model	Source	Mean Inference (ms)	Mean Overhead (ms)	Overhead (%)
Gemma 2 9B	Local	181,579.3	30.6	0.234
Mistral 7B	Local	13,931.3	27.3	0.281
LLaMA 3 8B	Local	7,524.8	26.7	0.456
GPT-4	API	4,519.7	24.5	0.564
Claude Sonnet 4.5	API	4,359.3	26.5	0.727

6.1 Implicações para a Prática de Reprodutibilidade

Nossos resultados geram várias recomendações acionáveis:

Use decodificação gulosa com modelos locais para máxima reprodutibilidade. Gemma 2 9B alcançou EMR = 1,000 *perfeito* em todas as tarefas sob decodificação gulosa.

Não-determinismo de API é observado em todos os cinco provedores. Pesquisadores usando *qualquer* modelo servido por API nunca devem assumir reprodutibilidade sem verificação.

Prefira formatos de saída estruturados quando possível. A reprodutibilidade consistentemente mais alta da tarefa de extração em todas as nove implantações demonstra que restrições de formato de saída melhoram diretamente a reprodutibilidade.

Inclua execuções de aquecimento para modelos locais. Adicionar uma única chamada de aquecimento descartada antes da coleta de dados elimina o efeito de inicialização a frio.

Registre de forma abrangente; o custo é negligenciável. A menos de 1% de sobrecarga e aproximadamente 4 KB por execução, não há razão prática para não aplicar registro abrangente.

6.2 A Lacuna de Reprodutibilidade: Do Turno Único aos Regimes Complexos

A lacuna de reprodutibilidade de 3 vezes entre modelos locais (EMR = 0,960) e modelos de API de código fechado (EMR = 0,325) persiste em todos os cinco provedores e quatro tarefas.

6.3 O Papel da Proveniência

Os grafos W3C PROV gerados pelo nosso protocolo servem múltiplos propósitos:

1. **Comparação automatizada**: Comparando grafos PROV de duas execuções, pode-se automaticamente identificar quais fatores diferiram.

2. **Rastreamento de linhagem**: Quando saídas são usadas como entradas para processos posteriores, a cadeia de proveniência pode ser estendida para rastrear qualquer resultado final até seu contexto completo de geração.

3. **Conformidade**: Para domínios regulados (saúde, jurídico, finanças), documentos PROV fornecem a trilha de evidências formais exigida por padrões de auditoria [National Institute of Standards and Technology, 2023] e regulações emergentes como o AI Act da UE [European Parliament and Council of the European Union, 2024].

6.4 Fontes de Não-Determinismo na Inferência Distribuída

Nossos experimentos estabelecem *que* modelos servidos por API exibem não-determinismo sob decodificação gulosa, enquanto modelos locais de GPU única não. Seis mecanismos bem documentados na inferência distribuída de GPU podem independentemente produzir saídas não-determinísticas mesmo sob decodificação gulosa:

1. **Aritmética de ponto flutuante não-associativa** [Higham, 2002]: A causa raiz fundamental.

2. **Acumulação em precisão mista** [Micikevicius et al., 2018]: Formatos de precisão reduzida (FP16 ou BF16).

3. **Paralelismo tensorial e não-determinismo de all-reduce** [Shoeybi et al., 2019]: Distribuição entre múltiplas GPUs.

4. **Não-determinismo de kernel de atenção**: FlashAttention [Dao et al., 2022] introduz não-determinismo através de sua estratégia de paralelização [Golden et al., 2024].

5. **Batching dinâmico e escalonamento de requisições** [Yu et al., 2022, Kwon et al., 2023]: Composições de batch diferentes levam a padrões diferentes de acesso à memória.

6. **Decodificação especulativa** [Leviathan et al., 2023]: Um componente estocástico adicional na amostragem de aceitação/rejeição.

Por que modelos locais escapam desses mecanismos. Nossa implantação local (GPU única Apple M4, servidor Ollama, uma requisição por vez) elimina os mecanismos (3)–(6) inteiramente. A reprodutibilidade quase perfeita dos nossos modelos locais (EMR médio = 0,960) é, portanto, uma *consequência prevista* da ausência desses mecanismos de sistemas distribuídos.

6.5 Limitações

6.5.1 Validade Interna

Tamanho amostral. LLaMA 3 usa 30 resumos por condição, enquanto os modelos mais novos usam 10 resumos. Com $n = 30$, o poder estatístico excede 0,999 para todas as comparações primárias. Uma análise de subamostra balanceada confirmou que a lacuna observada é robusta à equalização do tamanho amostral.

Lacuna de versionamento de código. As 330 execuções iniciais (LLaMA 3 8B e GPT-4, 7 de fevereiro de 2026) foram executadas antes da inicialização do repositório git; esses registros contêm `code_commit: "no-git-repo"`. As 3.774 execuções restantes (92% do total) incluem hashes de commit git válidos.

6.5.2 Validade Externa

Nove implantações, sete ambientes de execução. Nossa avaliação cobre três modelos locais, cinco modelos de API de código fechado de provedores independentes e um modelo de pesos abertos servido por nuvem. No entanto, outros modelos podem exibir características diferentes.

Quatro tarefas. Nossa suíte de tarefas não cobre geração de código, raciocínio matemático ou escrita criativa.

Apenas inglês, domínio único. Nossos dados de entrada consistem em 30 resumos científicos em inglês de artigos de IA/ML.

6.5.3 Validade de Construto

Métricas no nível de superfície. Nossas métricas capturam similaridade textual em vez de semântica. Para preencher essa lacuna, reportamos BERTScore F1 ao lado de EMR: BERTScore permanece acima de 0,97 sob decodificação gulosa em todos os modelos, confirmando que saídas de API transmitem significado equivalente apesar da divergência lexical.

7 Conclusão

Apresentamos um protocolo leve para registro, versionamento e rastreamento de proveniência de experimentos com IA generativa, introduzindo Prompt Cards e Run Cards como novos artefatos de documentação fundamentados no modelo de dados W3C PROV. Através de 4.104 experimentos controlados com nove implantações de modelos em quatro tarefas de PLN, 30 resumos e sete ambientes de execução, demonstramos sete achados-chave:

1. **Não-determinismo de API é consistente em todos os cinco provedores avaliados.** Todos os cinco modelos de API exibem não-determinismo sob decodificação gulosa (t=0), enquanto todos os três modelos locais alcançam EMR médio = 0,960.

2. **Reprodutibilidade de API varia substancialmente entre provedores.** Dentro da categoria API, EMR varia de 0,800 (DeepSeek Chat) a 0,010 (Perplexity Sonar para sumarização).

3. **Modelos locais podem alcançar reprodutibilidade bitwise perfeita.** Gemma 2 9B atinge EMR = 1,000 em todas as quatro tarefas sob decodificação gulosa.

4. **A lacuna local-vs-API se estende a regimes de interação complexos.** Modelos locais alcançam EMR $\geq$ 0,880 para refinamento multi-turno e extração RAG, enquanto ambos os modelos de API testados nessas tarefas exibem EMR quase zero ($\leq$ 0,070).

5. **Temperatura é o fator dominante controlável pelo usuário para modelos locais.** Aumentar de t=0,0 para t=0,7 reduz EMR a zero para todos os cinco modelos avaliados em sumarização.

6. **Registro abrangente de proveniência adiciona sobrecarga negligenciável**: menos de 1% do tempo de inferência e aproximadamente 4 KB por execução.

7. **Implantação em nuvem é compatível com inferência quase determinística.** A mesma arquitetura LLaMA 3 8B servida pela Together AI alcança reprodutibilidade próxima à local (EMR = 1,000 para extração, 0,880 para sumarização).

Esses achados carregam uma implicação mais ampla: uma porção substancial da pesquisa publicada que depende de LLMs baseados em API de código fechado pode conter resultados não reprodutíveis sem o conhecimento dos autores. O protocolo fornece a infraestrutura para detectar, medir e documentar tal variabilidade—tornando o não-determinismo oculto visível onde quer que ele ocorra.

Olhando adiante, planejamos (i) estender a sonda de quase-isolamento a provedores de nuvem adicionais e tamanhos de modelos maiores; (ii) estender a cobertura de tarefas para geração de código, raciocínio matemático e fluxos de trabalho agênticos; e (iii) desenvolver pontuação automatizada de reprodutibilidade baseada em análise de grafos de proveniência.

A implementação de referência, todos os 4.104 registros de execução, documentos de proveniência e scripts de análise estão publicamente disponíveis para apoiar a adoção e verificação independente.

Declaração de Disponibilidade de Dados

A implementação de referência, todos os 4.104 registros de execução (JSON), documentos de proveniência PROV-JSON, Run Cards, Prompt Cards, dados de entrada, scripts de análise e figuras geradas estão publicamente disponíveis em:

https://github.com/Roverlucas/genai-reproducibility-protocol

Contribuições dos Autores

Seguindo o framework CRediT: **Lucas Rover**: Conceitualização, Metodologia, Software, Validação, Análise Formal, Investigação, Curadoria de Dados, Escrita – Rascunho Original, Escrita – Revisão e Edição, Visualização, Administração do Projeto. **Yara de Souza Tadano**: Supervisão, Conceitualização, Metodologia, Escrita – Revisão e Edição, Administração do Projeto.

Conflito de Interesses

Os autores declaram ausência de conflitos de interesse. Esta pesquisa foi conduzida independentemente na UTFPR sem financiamento externo de provedores comerciais de IA.

Uso de Ferramentas Assistidas por IA

Os autores utilizaram ferramentas assistidas por IA (Claude, Anthropic) durante a preparação deste manuscrito para edição de linguagem, suporte ao desenvolvimento de código e scripts de análise de dados. Todo conteúdo gerado por IA foi criticamente revisado, validado e revisado pelos autores, que assumem total responsabilidade pela precisão e integridade do manuscrito final.

Referências

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. GPT-4 technical report, 2023. arXiv preprint.

Anthropic. The Claude model family, 2024. URL https://www.anthropic.com/claude.

Berk Atil, Sarp Aykent, Alexa Chittams, Lisheng Fu, Rebecca J. Passonneau, Evan Radcliffe, Guru Rajan Rajagopal, Adam Sloan, Tomasz Tudrej, Ferhan Ture, Zhe Wu,

Lixinyu Xu, and Breck Baldwin. Non-determinism of "deterministic" LLM settings, 2024. arXiv preprint.

Monya Baker. 1,500 scientists lift the lid on reproducibility. *Nature*, 533(7604):452–454, 2016. doi: 10.1038/533452a.

Philip Ball. Is AI leading to a reproducibility crisis in science? *Nature*, 624(7990):22–25, 2023. doi: 10.1038/d41586-023-03817-6.

Anya Belz, Shubham Agarwal, Anastasia Shimorina, and Ehud Reiter. A systematic review of reproducibility research in natural language processing. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics*, pages 381–393. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.eacl-main.29.

Anya Belz, Craig Thomson, and Ehud Reiter. Missing information, unresponsive authors, experimental flaws: The impossibility of assessing the reproducibility of previous human evaluations in NLP. In *Proceedings of the 4th Workshop on Human Evaluation of NLP Systems*, pages 1–10. Association for Computational Linguistics, 2022. doi: 10.18653/v1/2022.humeval-1.1.

Lukas Biewald. Experiment tracking with weights and biases, 2020. URL https://wandb.com/.

Abeba Birhane, Atoosa Kasirzadeh, David Leslie, and Sandra Wachter. Science in the age of large language models. *Nature Reviews Physics*, 5(5):277–280, 2023. doi: 10.1038/s42254-023-00581-4.

Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ Altman, Simran Arber, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models, 2022. arXiv preprint.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901, 2020.

Yupeng Chen, Junyi Li, Xiaochuan Liu, and Yujiu Li. On the reproducibility of ChatGPT in NLP tasks, 2023. arXiv preprint.

Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. FlashAttention: Fast and memory-efficient exact attention with IO-awareness. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*, pages 4171–4186. Association for Computational Linguistics, 2019. doi: 10.18653/v1/N19-1423.

Jesse Dodge, Suchin Gururangan, Dallas Card, Roy Schwartz, and Noah A. Smith. Show your work: Improved reporting of experimental results. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, pages 2185–2194. Association for Computational Linguistics, 2019. doi: 10.18653/v1/D19-1224.

European Parliament and Council of the European Union. Regulation (EU) 2024/1689 laying down harmonised rules on artificial intelligence (AI Act), 2024. URL https://eur-lex.europa.eu/eli/reg/2024/1689/oj.

Timnit Gebru, Jamie Morgenstern, Briana Vecchione, Jennifer Wortman Vaughan, Hanna Wallach, Hal Daumé III, and Kate Crawford. Datasheets for datasets. *Communications of the ACM*, 64(12):86–92, 2021. doi: 10.1145/3458723.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, et al. Gemma 2: Improving open language models at a practical size, 2024. arXiv preprint.

Alicia Golden, Samuel Hsia, Fei Sun, Bilge Acun, Basil Hosmer, Yejin Lee, Zachary DeVito, Jeff Johnson, Gu-Yeon Wei, David Brooks, and Carole-Jean Wu. Is flash attention stable? *arXiv preprint arXiv:2405.02803*, 2024.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, et al. The LLaMA 3 herd of models, 2024. arXiv preprint.

Odd Erik Gundersen and Sigbjørn Kjensmo. State of the art: Reproducibility in artificial intelligence. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1): 1644–1651, 2018. doi: 10.1609/aaai.v32i1.11503.

Odd Erik Gundersen, Yolanda Gil, and David W. Aha. On reproducible AI: Towards reproducible research, open science, and digital scholarship in AI publications. *AI Magazine*, 39(3):56–68, 2018. doi: 10.1609/aimag.v39i3.2816.

Odd Erik Gundersen, Kevin Coakley, Christine Kirkpatrick, and Yolanda Gil. Sources of irreproducibility in machine learning: A review. *arXiv preprint*, 2022.

Odd Erik Gundersen, Malte Helmert, and Holger H. Hoos. Improving reproducibility in AI research: Four mechanisms adopted by JAIR. *Journal of Artificial Intelligence Research*, 81:1019–1041, 2024. doi: 10.1613/jair.1.16905.

Melanie Herschel, Ralf Diestelkämper, and Houssem Ben Lahmar. A survey on provenance: What for? what form? what from? *The VLDB Journal*, 26(6):881–906, 2017. doi: 10.1007/s00778-017-0486-1.

Nicholas J. Higham. *Accuracy and Stability of Numerical Algorithms*. SIAM, 2nd edition, 2002.

Matthew Hutson. Artificial intelligence faces reproducibility crisis. *Science*, 359(6377): 725–726, 2018. doi: 10.1126/science.359.6377.725.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. Mistral 7B, 2023. arXiv preprint.

Sayash Kapoor and Arvind Narayanan. Leakage and the reproducibility crisis in machine-learning-based science. *Patterns*, 4(9):100804, 2023. doi: 10.1016/j.patter.2023.100804.

Ruslan Kuprieiev, Dmitry Petrov, and Iterative. DVC: Data version control, 2024. URL https://dvc.org/.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with PagedAttention. In *Proceedings of the 29th ACM Symposium on Operating Systems Principles*, 2023.

LangChain. LangSmith: A platform for building production-grade LLM applications, 2023. URL https://smith.langchain.com/.

Yaniv Leviathan, Matan Kalman, and Yossi Matias. Fast inference from transformers via speculative decoding. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 19274–19286. PMLR, 2023.

Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, et al. Holistic evaluation of language models. *Transactions on Machine Learning Research*, 2023. URL https://openreview.net/forum?id=iO4LZibEqW.

Paulius Micikevicius, Sharan Narang, Jonah Alben, Gregory Diamos, Erich Elsen, David García, Boris Ginsburg, Michael Houston, Oleksii Kuchaiev, Ganesh Venkatesh, and Hao

Wu. Mixed precision training. In *International Conference on Learning Representations*, 2018.

Margaret Mitchell, Simone Wu, Andrew Zaldivar, Parker Barnes, Lucy Vasserman, Ben Hutchinson, Elena Spitzer, Inioluwa Deborah Raji, and Timnit Gebru. Model cards for model reporting. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, pages 220–229. ACM, 2019. doi: 10.1145/3287560.3287596.

Luc Moreau and Paolo Missier. PROV-DM: The PROV data model. W3C recommendation, World Wide Web Consortium, 2013. URL https://www.w3.org/TR/prov-dm/.

National Institute of Standards and Technology. Artificial intelligence risk management framework (AI RMF 1.0). Technical report, U.S. Department of Commerce, 2023.

Shakked Noy and Whitney Zhang. Experimental evidence on the productivity effects of generative artificial intelligence. *Science*, 381(6654):187–192, 2023. doi: 10.1126/science.adh2586.

Ollama. Ollama: Run large language models locally, 2024. URL https://ollama.com/.

OpenAI. OpenAI Evals: A framework for evaluating LLMs, 2023. URL https://github.com/openai/evals.

OpenAI. API reference: Create chat completion — seed parameter, 2024. URL https://platform.openai.com/docs/api-reference/chat/create. "If specified, our system will make a best effort to sample deterministically [...] Determinism is not guaranteed.".

Shuyin Ouyang, Jie M. Zhang, Mark Harman, and Meng Wang. An empirical study of the non-determinism of ChatGPT in code generation. *ACM Transactions on Software Engineering and Methodology*, 34(2):1–28, 2024. doi: 10.1145/3697010.

Gabriele Padovani, Valentine Anantharaj, and Sandro Fiore. yProv4ML: Effortless provenance tracking for machine learning systems. *SoftwareX*, 29:102028, 2025. doi: 10.1016/j.softx.2025.102028.

Joelle Pineau, Philippe Vincent-Lamarre, Koustuv Sinha, Vincent Larivière, Alina Beygelzimer, Florence d'Alché Buc, Emily Fox, and Hugo Larochelle. Improving reproducibility in machine learning research: A report from the NeurIPS 2019 reproducibility program. *Journal of Machine Learning Research*, 22(164):1–20, 2021. URL https://jmlr.org/papers/v22/20-303.html.

PyTorch Contributors. Reproducibility — PyTorch documentation, 2024. URL https://pytorch.org/docs/stable/notes/randomness.html. Documents sources of non-determinism in GPU operations and `torch.use_deterministic_algorithms()` flag.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140): 1–67, 2020. URL https://jmlr.org/papers/v21/20-074.html.

Machel Reid, Nikolay Savinov, Denis Teber, Ioana Bica, Zhitao Shen, Dmitry Lepikhin, et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context, 2024. arXiv preprint; Gemini 2.5 Pro builds upon this architecture.

Anna Rogers, Timothy Baldwin, and Kobi Leins. "just think about it": Incentivizing reproducibility in NLP. *Computational Linguistics*, 47(4):757–770, 2021. doi: 10.1162/coli_a_00420.

Sheeba Samuel and Birgitta König-Ries. A provenance-based semantic approach to support understandability, reproducibility, and reuse of scientific experiments. *Journal of Biomedical Semantics*, 13(1):1–30, 2022. doi: 10.1186/s13326-022-00263-z.

Mohammad Shoeybi, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, and Bryan Catanzaro. Megatron-LM: Training multi-billion parameter language models using model parallelism. *arXiv preprint arXiv:1909.08053*, 2019.

Karan Singhal, Shekoofeh Azizi, Tao Tu, S. Sara Mahdavi, Jason Wei, Hyung Won Chung, Nathan Scales, Ajay Tanwani, Heather Cole-Lewis, Stephen Pfohl, et al. Large language models encode clinical knowledge. *Nature*, 620(7972):172–180, 2023. doi: 10.1038/s41586-023-06291-2.

Victoria Stodden, Marcia McNutt, David H. Bailey, Ewa Deelman, Yolanda Gil, Brooks Hanson, Michael A. Heroux, John P.A. Ioannidis, and Michela Taufer. Enhancing reproducibility for computational methods. *Science*, 354(6317):1240–1241, 2016. doi: 10.1126/science.aah6168.

Arun James Thirunavukarasu, Darren Shu Jeng Ting, Kabilan Elangovan, Laura Gutierrez, Ting Fang Tan, and Daniel Shu Wei Ting. Large language models in medicine. *Nature Medicine*, 29(8):1930–1940, 2023. doi: 10.1038/s41591-023-02448-8.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837, 2022.

Mark D. Wilkinson, Michel Dumontier, IJsbrand Jan Aalbersberg, Gabrielle Appleton, Myles Axton, Arie Baak, Niklas Blomberg, Jan-Willem Boiten, Luiz Bonino da Silva Santos, Philip E. Bourne, et al. The FAIR guiding principles for scientific data management and stewardship. *Scientific Data*, 3:160018, 2016. doi: 10.1038/sdata.2016.18.

Gyeong-In Yu, Joo Seong Jeong, Geon-Woo Kim, Soojeong Kim, and Byung-Gon Chun. Orca: A distributed serving system for Transformer-Based generative models. In *16th USENIX Symposium on Operating Systems Design and Implementation*, pages 521–538. USENIX Association, 2022.

Jiayi Yuan, Hao Li, Xinheng Ding, Wenya Xie, Yu-Jhe Li, Wentian Zhao, Kun Wan, Jing Shi, Xia Hu, and Zirui Liu. Understanding and mitigating numerical sources of nondeterminism in LLM inference. In *Advances in Neural Information Processing Systems*, volume 38. Curran Associates, Inc., 2025. arXiv preprint.

Matei Zaharia, Andrew Chen, Aaron Davidson, Ali Ghodsi, Sue Ann Hong, Andy Konwinski, Siddharth Murching, Tomas Nykodym, Paul Ogilvie, Mani Parkhe, et al. Accelerating the machine learning lifecycle with MLflow. *IEEE Data Engineering Bulletin*, 41(4):39–45, 2018. URL http://sites.computer.org/debull/A18dec/p39.pdf.

Yiming Zhu, Peixian Zhang, Ehsanul Haq, Pan Hui, and George Buchanan. Can ChatGPT reproduce human-generated labels? a study of social computing tasks, 2023. arXiv preprint.