The Source Code of Law: AI Agents Rewrite and Red-Team 30,000 Articles of French Law

Alexandre Masmejean* Independent
Nathalie Gouailhardou Independent
Kevin Primicerio Independent
kevin.primicerio@gmail.com

31 October 2026 (revised 10 November 2026)

Abstract

Turning a political programme into enforceable law means amending a vast, interconnected corpus: statutes, decrees, tax annexes and future versions that often contradict one another. We study whether a policy-configurable pipeline of AI agents can make this translation coherent, verifiable and far cheaper than manual drafting. Case study: five major French codes in full (Labour, the General Tax Code and its four annexes, Social Security, Town Planning, Construction and Housing), 30,317 articles from the official LEGI database as of 18 October 2026, rewritten according to reforms inspired by David Lisnard's 2027 programme, in an independent project. The work ran in three passes and an open public review: a first draft of every article (14,115 rewritten, 16,199 retained with a justification), a complete coherence re-reading, then a separate red team that applied 1,285 corrections and measured reliability with blind random audits. On two fresh samples totalling 1,264 articles, audited blind on both texts with the same protocol, the rewritten law has **54% fewer gross defects** than the law in force (95% CI 40–64%): $3.5\% \pm 1.0$ against $7.5\% \pm 1.5$ (paired test, $p \approx 2 \times 10^{-10}$). The project cost about US$1,400 in cash, against an order-of-magnitude estimate of €6 to 36 million for the same work by lawyers. The programme is a separate input: the coherence checks do not depend on it, and we invite other programmes to be run through the same pipeline. These results, produced and measured by AI agents alone, are released for verification by lawyers and citizens.

Key result. Measured the same way on the same randomly drawn articles, the AI-rewritten law has **54% fewer gross defects** than the French law in force: 3.5% of articles against 7.5% (1,264 articles, 95% CI of the reduction 40–64%). **The rate mainly measures coherence (references, versions, contradictions, transitions); it judges neither compliance with EU or constitutional law (F9, "to be confirmed") nor the merits of the reforms.** It was **2,500 to 10,000 times cheaper** at the midpoints (Table 9 ranges run from 1,300 to 28,000, excluding human direction time, and lawyer hours are assumptions) than the same work by lawyers (order-of-magnitude estimates, §7.7). No article has been reviewed by a lawyer: these texts are a verifiable working basis, not a bill ready to be tabled.

1 Introduction

Legibility as a legal requirement. Accessibility and intelligibility of the law are an objective of constitutional value in France [2]. The Conseil d'État has for decades criticised the growth and instability of legal texts [3].

*Corresponding author: alex@masmej.com. Known as Alex Masmej on X: https://x.com/AlexMasmej.

The problem is not only the volume. A single rule is spread across several texts:

- a statute (*partie législative*);
- its implementing decrees (*partie réglementaire*);
- for taxation, the four annexes of the CGI;
- consolidated future versions already published but not yet in force.

When one of these texts changes and the others do not, contradictions appear. Some are inherited from the official sources; others come from reform.

Origin of this work. On 19 October 2026 the first author published a short thread in French [4]. It answered a public challenge and announced that AI agents had produced, overnight, rewritten versions of several French codes, published on five thematic websites. The thread invited lawyers, economists, employees and entrepreneurs to criticise the proposals article by article.

The reception was much larger than expected: about 120,000 views, 1,190 likes, 201 reposts and 293 bookmarks when last checked. The criticism that followed was more useful still. It made clear that a fast first draft was not enough, and that a credible demonstration needed three things:

(i) the complete codes, not a selection;

(ii) a systematic search for contradictions;

(iii) an honest measurement of what remains wrong.

This paper documents that second, slower phase.

Why now: the 2027 presidential election. France will elect its next President in 2027, at the end of the current five-year term [1]. Parties will put forward reforms of taxation, employment, pensions, housing and business regulation. A reform promise is not yet law: to take effect, it must be translated into precise amendments to statutes and regulations across a vast, interconnected corpus. That translation is today slow, costly and hard to verify from outside.

The 2027 election therefore offers an immediate, concrete use case for AI-assisted legislative engineering, whoever wins and however extensive the subsequent reforms turn out to be. We do not claim that reform on this scale will happen; we examine whether AI agents can make the translation much cheaper, faster, more systematic and easier to audit.

Research question. *Can a general-purpose, policy-configurable multi-agent system translate ambitious political reform objectives into coherent, verifiable amendments across a national legal corpus, at dramatically lower cost than traditional manual drafting?*

We break it into three aims:

- **Legibility.** Produce, for every article in scope, a complete proposed text set side by side with the official source, with a written justification.
- **Coherence.** Show how AI agents can find and resolve contradictions in a large, interconnected legal corpus, and how far that goes. Reforming a code is easy to state and hard to do consistently: removing an obligation from a statute is meaningless if a decree, a tax annex or a future consolidated version keeps imposing it.
- **Configurability.** Separate the political programme, treated as an input, from the rewriting and verification infrastructure, and state what in that infrastructure still depends on it.

Contributions.

- A four-layer architecture that separates *policy specification* (which objectives to implement) from *legislative transformation* (how to translate them into amendments), *consistency verification* (contradictions, broken references, incompatible provisions, implementation errors) and *independent evaluation* (§4.1). We state precisely which parts of the current infrastructure still depend on the chosen programme.
- A complete rewrite of five major codes and the four CGI annexes (30,317 entries), in three passes and an open public review, versioned and public. Official sources and proposals are kept apart, and a comparative reader and downloads are provided (§3–§4).
- A red-teaming protocol for AI-produced legislation:
  - frozen, hashed snapshots;
  - an objection register with priorities and natures;
  - re-examination on diffs;
  - correction under an explicit alignment criterion;
  - a guarded writer that cannot silently overwrite (§5).
- A measurement method: eight blind, stratified random audits, with a taxonomy of ten defect families (F1–F10). The rate has a confidence interval, not an anecdote (§6).
- A paired baseline: the same audit run on the official law, on the same articles. The rewritten law has 34% fewer gross defects than the law it replaces on 1,879 articles at the time of the audits (4.6% against 7.0%), and **54% fewer** on two fresh samples totalling 1,264 articles drawn at the end of the work (3.5% against 7.5%) (§7.3, §7.4).
- A cost comparison: about US$1,400 in cash, US$3,500 to 5,100 at API prices, against order-of-magnitude estimates of €6 to 36 million for the same work by lawyers, 2,500 to 10,000 times more at the midpoints (Table 9 ranges run from 1,300 to 28,000, excluding human direction time, and lawyer hours are assumptions; §7.7).
- Empirical findings on what AI agents do well and badly on legal coherence (§7–§8):
  - mechanical sweeps reach a plateau;
  - correction rounds can create new defects;
  - a large part of the residual error comes from texts outside the corpus.

The project at a glance. Table 1 summarises the whole undertaking: three passes over the five codes and the four tax annexes, an open public review, and the measurements reported here.

AI-only results. This paper deliberately reports what AI agents achieved on their own. Drafting, review, correction and measurement were all done by AI agents, under human direction and arbitration but without legal experts. We release the results so that lawyers and citizens can verify and harden them (see the call for contributions at the end).

A case study: David Lisnard's programme. The first implementation uses reforms inspired by David Lisnard's 2027 programme [5]. This choice provides a concrete and ambitious set of policy assumptions with which to test a general pipeline: fewer taxes and constraints; exemptions, favourable derogations, acquired rights and transitions preserved; never a replacement formality. It is the project's historical origin, and we disclose it explicitly.

Stage	Dates (Oct. 2026)	Agents	What it did
Pass 1: drafting	18–23	OpenAI Codex (`gpt-6-astra`)	Read every article in full and wrote a complete proposal: 14,115 rewritten, 16,199 retained with a written justification.
Pass 2: coherence	23–29	OpenAI Codex (`gpt-6-astra`, then `gpt-6.1-sol`)	Re-read all 30,314 proposals against their sources and counterparts, contradictions first; 2,991 review jobs.
Pass 3: red team	26–31	Claude Code (Opus 5.5, Fable 5.1)	About 570 agent runs: 303 objections raised and 243 retained; 1,285 corrections on 1,211 articles; full reading of the 2,740 rewritten tax articles.
Public review	from 27	External contributor, readers	16 issues and 2 pull requests on GitHub (two of the issues on this paper), nine of the first eleven pointing at a real defect; triaged as they came; all answered and closed at the date of this version.
Measurement	29–31	Blind AI auditors	Eight random audits (3,143 articles read blind) and three paired comparisons with the official law.

Table 1: The project at a glance.

This project is an independent initiative. It is not commissioned, reviewed or endorsed by David Lisnard or by his movement, Nouvelle Énergie, and the published websites say so. This paper makes no scientific claim that these positions are preferable to others.

We distinguish four questions, which belong to different layers:

- **Policy specification:** which political objectives should be implemented? This is the input, here the chosen programme; the question is not a scientific one.
- **Legislative transformation:** how do AI agents translate those objectives into amendments to existing law?
- **Consistency verification:** how do they detect contradictions, broken references, incompatible provisions and implementation errors?
- **Independent evaluation:** how can the resulting corpus be evaluated without privileging a particular political philosophy?

Taking a complete, ambitious programme as the target shows that a large-scale reform can be drafted end to end with AI while ending up less defective than the law in force (§7.4). With another programme, whether economically liberal, centrist, socialist, conservative or otherwise, the policy objects, the orientation of the drafting instruction and the red team's alignment criterion would change; the guards, the validation and the coherence audits would not (§4.1, Table 3). The red team was instructed to fix contradictions and loopholes, *never* to change political choices.

2 Related work

Large language models are increasingly evaluated on legal reasoning, e.g. LegalBench [7]. Work on making legislation machine-readable comes in two forms:

- programming languages for statutes, such as Catala, applied to French tax law [8];
- "rules as code" initiatives [9].

Both formalise existing law faithfully. Our goal is different: we *rewrite* the law according to a reform programme, and then check the coherence of the result.

Red-teaming, the adversarial search for failures by an independent party, is standard practice for AI systems [10]. Here it is turned against AI-produced legislation.

We are not aware of prior work that rewrites complete national codes end to end and publishes a measured defect rate. We place this work in *legislative engineering*: treating a programme as a specification, the legal corpus as the object to transform, and the coherence of the result as a property to measure.

3 Corpus and scope

The corpus is frozen at **18 October 2026**. It is built from the official LEGI database of the Direction de l'information légale et administrative (DILA) [6]: the stock of 13 July 2025, plus 466 daily updates.

The working identity of an article is the pair (code identifier, version identifier `LEGIARTI`), together with a hash of its source and its validity dates. Numbers alone are never trusted, because they are recycled. The database holds 135,904 historical versions; the scope is the 30,317 entries in force or scheduled (Table 2).

Instrument	Entries	Rewritten	Retained	Reserves
Labour Code	11,593	4,296	7,297	0
General Tax Code (CGI)	2,375	1,387	988	0
Social Security Code	7,427	3,178	4,247	2
Town Planning Code	2,459	1,530	929	0
Construction and Housing Code	4,126	2,205	1,921	0
CGI annex I	71	42	29	0
CGI annex II	674	388	286	0
CGI annex III	1,115	799	316	0
CGI annex IV	477	290	186	1
Total	30,317	14,115	16,199	3

Table 2: Scope at the end of the second pass. "Retained" articles are kept unchanged with a specific justification. The three "reserves" are documentary gaps (a missing figure, missing original forms, and a conflict in the official consolidation) that were not filled by guesswork. At commit 55621278, `progress.json` counts 14,273 drafted and 16,041 retained articles after pass-3 requalifications, plus the 257 proposals on future versions.

Out of scope are texts that the corpus cites but does not contain. Examples are the Monetary and Financial Code, the Commercial Code, non-codified statutes, ministerial orders, and the Code of taxes on goods and services (CIBS), into which much of VAT is being recodified on 1 January 2027. This boundary turns out to matter (§8).

4 Drafting pipeline (first and second passes)

4.1 Policy layer

Figure 1 summarises the whole pipeline. All substantive decisions are stored as explicit, versioned policy objects, separate from the legal text: per-domain coordinations, site summaries, and a net-costing method for tax measures. Examples:

- a single open-ended employment contract with a 15-day notice period on each side;
- corporate tax triggered by distribution;

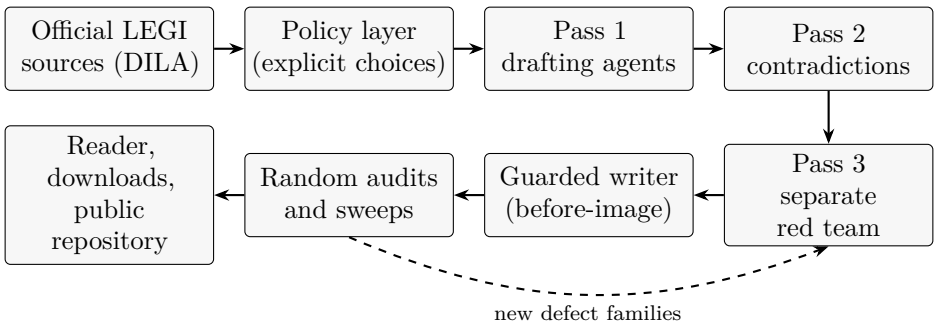


Figure 1: Overview of the pipeline. Every write goes through the same guarded writer; audits feed new defect families back to the red team.

- abolition of inheritance taxes, while gifts remain taxed;
- abolition of production taxes;
- shorter planning-permit deadlines.

The rest of the pipeline treats them as inputs.

Editorial invariants apply throughout:

- official sources are never overwritten;
- no rate, date, equivalence or facsimile is invented;
- favours are never removed to make two texts artificially consistent;
- an obligation removed in a statute must not survive in a decree, a form or a cross-reference.

Replacing the programme. Table 3 states what in the current pipeline depends on the programme and what does not.

Component	Depends on the programme	What would have to change
Policy objects (coordinations, summaries, costing)	Yes	Rewrite them for the new programme. They were written under the author's direction, not extracted automatically from a programme.
Drafting instruction to the agents	Partly	Replace the orientation sentence (today: "cut taxes and constraints further"); the rest (read in full, return the complete article, invent nothing) is general.
Red team's alignment criterion	Yes	Restate it from the new programme. The principle (fix contradictions and loopholes, send arbitrations back) stays.
Guards, guarded writer, validation	No	Nothing.
Audits, families F1–F7 and F9	No	Nothing. Family F8 ("contrary to the orientation") is recomputed against the new programme.
Official-law baseline	No	Nothing: it reads no programme.

Table 3: What depends on the programme in the current pipeline.

The paired audit of the official law (§7.3) shows that the evaluation works without any programme. In the two fresh samples (§7.4), no defect was classified F8, on either text: the headline rate rests only on families that do not depend on the programme. No other programme has been implemented yet, however: the portability of the transformation itself is not measured.

4.2 Article-level workers

Each article (or batch of related articles) is processed by a non-interactive coding agent. The agent receives:

- the complete source XML, including notes (*NOTA*), tables and versions;
- the applicable policies;
- the explicit cross-reference targets.

It must return the *complete* proposed article, never a summary, together with a reason and review notes. It assigns one of two statuses, `rewrite_draft` or `retained_unchanged`. Repeals and partial reductions are typed (`full_repeal`, `normative_reduction`). Every removed provision is listed, so that a reduction is auditable.

The first pass and most of the second ran on an OpenAI model (`gpt-6-astra`, medium effort) through the Codex command-line agent. The last 1,079 re-readings of the second pass (28–29 October) ran on `gpt-6.1-sol` at high effort.

4.3 Second pass: contradictions first

From 23 October, the priority shifted from further cuts to *contradictions and inconsistencies*, within an article, between articles and across the nine instruments. All 30,314 prior drafts were re-read against their complete sources and counterparts. Each worker classified every tension it found as one of:

- an effective contradiction;
- a required coordination;
- a compatible transition;
- a difference of scope;
- a point that cannot be verified.

4.4 Guards

Nothing is accepted on the word of a model. An answer is accepted only after the following checks:

- source, NOTA and validity dates;
- content validation, including allowed markup and complete tables;
- a before-image check on five canonical fields (source hash, status, proposal, reason, notes), both in the database and in the batch file that owns the article;
- ownership, registry and archive;
- an independent whole-corpus validation.

Native execution evidence (model, effort, session) is also checked.

These guards are not decoration. They caught real divergences: three second-pass jobs were refused because the article had changed under them. The guards also made the third pass possible, because every correction could be applied through the same writer.

The second pass ended on 29 October with all 30,314 identities accepted. A post-hoc structural audit then listed 94,059 points to examine, 1,412 of them high-priority, on 792 articles. These are leads, not established contradictions.

5 Separate red team (third pass)

The third pass was run with a different system, Anthropic's Claude Code (Claude Opus 5.5; the last cycle, audit 8 and its fixes, ran on Claude Fable 5.1). A coordinating agent dispatched about 570 sub-agent runs over six days. It worked as a reviewer *separate from the drafting agents*, under strict rules:

- read-only access to production, on frozen snapshots whose files are all hashed;
- exact quotations with version identifiers for every claim;
- no invented source, date or "evidence of resolution";
- no bypass of access controls;
- an explicit statement that an AI review is not a lawyer's validation.

5.1 Objection register

The first wave covered 31 thematic dossiers (for example future versions, decrees versus statutes, tax annexes, and the transitions of the 2026 occupational-injury reform). Five priority domains followed: distribution-based corporate tax, inheritance, the SRU social-housing quotas, the CVAE production tax, and the payroll tax.

Each objection records:

- a priority: P0 (grave and certain effect), P1 (probable effect) or P2 (limited effect);
- a nature: missing coordination, inconsistency introduced by the project, legal risk, source conflict, inventory anomaly, documentary reserve or simple alert;
- the exact passages, the reasoning and a *minimal correction*;
- what must be preserved: rights, favours and transitions.

The red team classified 1,943 structural alerts of the static audit one by one, including every high-priority alert still live on the final text (1,324). Duplicates were merged with a written reason (46 merges).

5.2 Re-examination on diffs

The second pass was still running when the red team started. The register was therefore re-examined twice, against an intermediate snapshot and then the final one, using a file-level diff. On the final text, 136 objections whose articles had changed were re-read:

- 58 were resolved by the second pass (including 17 P1);
- 30 were reformulated;
- 48 were maintained.

The final register holds **243 objections**: 3 P0, 83 P1 and 157 P2, of which 188 are confirmed and 55 still to be confirmed.

5.3 From review to correction

After the review, the first author made the red team the sole writer, with one standing instruction: issues may fix contradictions and loopholes, never political choices. A correction is applied only if it does one of three things:

(a) make a decided reform effective, for example by carrying it onto a published future version that would otherwise restore the old rule;

(b) remove a contradiction;

(c) close a loophole that contradicts the declared object of the project.

A correction is refused if it does any of the following:

- create or raise a tax;
- add a formality or a condition;
- remove an exemption or an acquired right;
- reverse an explicit choice of the author;
- settle an open political question.

Drafting agents write corrections only in a workspace, as complete article texts with typed metadata. A checker rebuilds each row exactly as the production applier would and runs the project's own validator in a temporary directory.

The coordinator then applies the corrections with the guarded tool. That tool checks the before-image, archives the old text, revalidates the whole corpus and rolls back on failure. An ownership map gives each article to a single group. Cross-group needs are expressed as requests, so two agents never write the same article.

5.4 External contributions

During the review, an external contributor (the third author) filed 13 issues and two pull requests on the public repository, and three more issues after it: one on a code and two on this paper. The issues on the codes were produced with his own AI agents and covered labour-law cross-references, decrees that survive repealed statutes, penalties, works-council committees, a real-estate wealth-tax exemption and a safeguard in planning litigation. Each was triaged against the register on the current text. Four asked for a policy change or the removal of a favour; they were explained and closed as not planned. Two were withdrawn by their author. The 18 correction rows applied on 4 November in response to the later reports are not counted in the totals of §7.

6 Measuring reliability

A register says what was found, not what remains. We therefore measured the residual defect rate directly.

Sampling. Each audit draws about 314 articles at random. The draw is stratified by instrument and by level (statute or regulation), and each stratum gets at least six articles. Successive audits exclude every article read before. Seeds are published.

Blind reading. Ten agents per audit read their share "as a lawyer would", on the production text: the article, its future versions, its three most important links, and its implementing texts or legal basis. They do not consult the register before qualifying an article.

Verdicts. Each article is classified as:

- **G** (gross): a lawyer would notice it and find it abnormal;
- **m** (minor): wrong notes, clumsy but harmless wording, or a reasonable doubt;
- **0**: nothing to report.

Each G or m needs an exact quotation. Percentages are simple proportions of the articles read, with normal-approximation 95% intervals. Because the allocation is close to proportional, stratum-weighted rates (the published `audit_rate.py`) differ from them by at most 0.25 point for gross defects; on the fresh samples of §7.4 they give 3.6% against 7.7%, a 53.5% reduction. The interval of a relative reduction uses the log method for a ratio of paired proportions. Defects are classified into ten families:

F1	A statute removes a formality or an obligation, but a decree or an annex still organises it.
F2	A penalty survives the obligation it punishes.
F3	Dead, shifted or wrong cross-reference.
F4	A published future version, with no proposal, restores the old rule; or a false claim that an article is repealed.
F5	Contradiction between two provisions (other than F1).
F6	Missing or faulty transition (retroactivity, loss of acquired rights).
F7	Defective text (truncated, broken numbering, amputated table).
F8	Contrary to a declared orientation (a tax or formality re-created, a favour removed without decision).
F9	Higher-law risk (Constitution, EU law), always "to be confirmed".
F10	Notes or reasons only (always minor).

Table 4: Defect families used by auditors and sweeps.

Family sweeps. After each audit, we turned its gross defects into mechanical lead generators and swept the whole corpus for them. Examples:

- a decree citing a repealed statute;
- a penalty citing a removed article;
- a reference to the n-th paragraph of an article that a proposal or future version has shifted;
- a current version with a proposal whose next published version has none;
- a decree that still uses the distinctive words of what its statute removed;
- a chapter where only some articles are repealed.

Leads are not findings. Each lead is qualified by an agent that reads the texts in full, as a false positive, already fixed, G or m. A correction is drafted for every confirmed G.

Baseline. To know whether a residual rate is good or bad, we ran the same audit on the *official* text of the same 1,879 articles, in the same packets, with the same criterion. A source-only mode of the reading tool hid the project's proposals, reasons and notes, and the packets did not say whether an article had been rewritten. Defects seen on only one side were then checked against both texts by separate verifier agents (§7.3).

Systematic reading. Finally, we read every rewritten article of the CGI and its annexes, where the audits showed the highest rates. The 2,740 articles not already audited were grouped by their deepest section, so that each reader sees a whole chapter. They were split into 132 packets of about 21 articles. Each reader qualifies every article and drafts corrections for its gross defects.

7 Results

7.1 Corrections applied

Over six days, the third pass applied **1,285 correction rows to 1,211 distinct articles**:

- 1,028 replacements of existing proposals;
- 257 new proposals on published future versions, which would otherwise have restored the pre-reform text when they enter into force.

Each row was validated by the project's own validator and archived with before and after hashes. The project's 321 tests pass after every round. Table 5 breaks the rows down by source.

Source of corrections	Leads or objections	Rows applied
P0/P1 objections and first GitHub issues	86 objections	395
P2 objections	157 objections	201
Sweep 1 (dead references, future versions)	1,127 leads → 154 G	145
Sweep 2 (statute removes, decree keeps)	1,117 leads → 33 G	36
Audit 1–3 defects, notes pilot, later issues	–	≈ 90
Round 4 (audit 4 + sweep)	918 leads → 45 G	64
Round 5 (audit 5 + tax-aware sweep)	548 leads → 114 G	97
Round 6 (audit 6 + whole chapters)	365 leads → 25 G	110
Systematic reading, CGI and annexes	2,740 articles → 96 G	81
Defects of the paired sample still present	52 articles → 28 G	27
Defects of the first fresh sample still present	28 articles → 28 G	36

Table 5: Corrections applied by the third pass, by source. Counts come from the per-round reports; the "≈" line groups small fix sets.

7.2 Residual defect rate

Auditors also flagged higher-law risks (F9), published in the verdicts; we do not qualify them, since that requires a legal analysis the pipeline does not perform.

Audit (state of the text)	Sample	G	G rate	G or m
1 (after P0/P1 corrections)	309	12	3.9% ± 2.2	21.0%
2 (after sweep 1)	314	9	2.9% ± 1.8	22.6%
3 (after sweep 2)	314	12	3.8% ± 2.1	17.8%
4 (after P2 objections)	314	8	2.5% ± 1.7	14.6%
5 (after round 4)	314	10	3.2% ± 1.9	22.6%
6 (after round 5)	314	13	4.1% ± 2.2	15.3%
Pooled, audits 1–6	1,879	64	**3.4% ± 0.8**	
7 (fresh sample, current text)	632	15	2.4% ± 1.2	20.7%
8 (second fresh sample, current text)	632	9	1.4% ± 0.9	12.3%

Table 6: Eight blind random audits on disjoint samples (raw verdicts). Each sample excludes all previously read articles. Audits 7 and 8 are the rewritten-text side of the fresh paired audits (§7.4); their auditors were told explicitly that a defect inherited from the official text counts. Audit 8 ran on a different model (Fable 5.1).

About 3.4% of articles (95% interval 2.6–4.2%) still carry a gross defect (Table 6, Figure 2). Put differently, a lawyer who reads 20 articles in their field has roughly an even chance of meeting at least one.

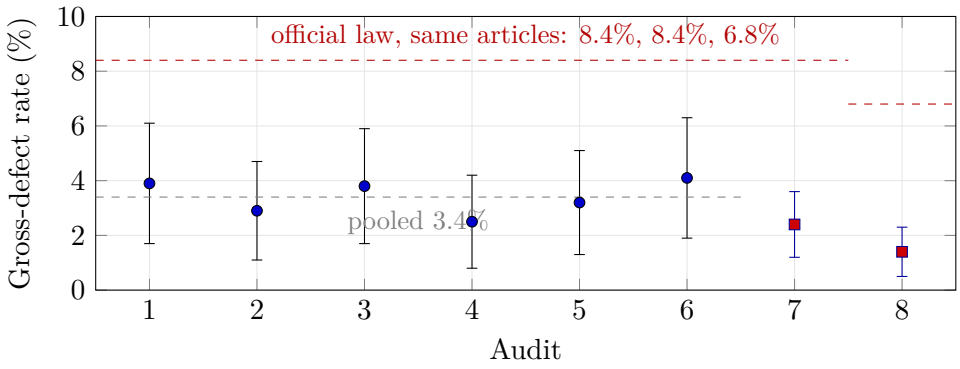


Figure 2: Measured gross-defect rate per audit, with 95% intervals. No decline can be detected from one round to the next. The red line is the same audit run on the official text of the same articles; 8.4% on the first two paired samples and 6.8% on the third (raw verdicts, before verification). The squares are audits 7 and 8, on fresh samples of the current text.

The rate shows no measurable decline from round to round, even though each round applied between 36 and 201 correction rows (Table 5). This is expected. If about 1,000 articles are defective, fixing 100 lowers the rate by about 0.3 points, and fixing 200 by less than 0.7, well below what a 314-article audit can detect (±2 points).

The composition of the defects did change:

- after the first sweep, dead references (F3) receded;
- contradictions between two proposals (F5) moved to the top;
- later audits concentrated on the tax annexes.

Of the gross defects found by audits, between half and three quarters were unknown to the register.

7.3 Baseline: the official law, same articles

Is 3.4% good or bad? We audited the official text of the same 1,879 articles with the same method (§6).

Raw verdicts. AI auditors found a gross defect in 158 official articles ($8.4\% \pm 1.3$), against 64 in the project audits ($3.4\% \pm 0.8$). Only 24 articles were flagged on both sides. Most official defects are dead or outdated references (family F3, 124 of 158): articles repealed years ago, bodies that were renamed, codes that were recodified.

Checking the disputed defects. The raw comparison is not enough. On the 966 articles that the project retained unchanged, the text is identical on both sides, yet the official-text auditors flagged 30 and the project auditors 15. Either the two sets of auditors were not equally strict, or the project auditors discounted defects inherited from the source.

We therefore had separate verifier agents check each of the 174 defects seen on only one side, against both texts, answering three questions: is it present in the official text, is it present in the proposed text, and is it really gross?

Result. After verification, the retained articles, whose text is the same on both sides, show the same rate (23 and 22 defects). This suggests that the verification removes the difference in strictness. On that basis (Table 7):

		Official law		Rewritten law	
Same articles, verified	Articles	G	Rate	G	Rate
All	1,879	131	7.0% ± 1.2	86	**4.6% ± 0.9**
Rewritten by the project	913	108	11.8%	64	7.0%
Retained unchanged (control)	966	23	2.4%	22	2.3%
Defects of the official law repaired by the rewrite					77
Defects introduced by the rewrite					32
Paired test (McNemar, exact)					$p \approx 2 \times 10^{-5}$

Table 7: Gross defects in the official law and in the rewritten law, on the same 1,879 randomly drawn articles, after verification of every defect seen on only one side. Project figures are at the time of each audit. Three defects whose presence in the rewritten text the verifier could not settle are counted as present.

- the rewritten law has **34% fewer gross defects** than the official law it replaces (95% CI 20–46%): 4.6% against 7.0%;
- among rewritten articles, the rate falls from 11.8% to 7.0%;
- for every defect the rewrite introduced, it repaired more than two in the official law (77 against 32).

One exception stands out. In the legislative part of the CGI, the sample has more verified defects after the rewrite than before (9 against 3): reforming corporate and income tax creates inconsistencies faster than it removes old ones. This is where the systematic reading (§7) was concentrated.

The project figures are those at the time of each audit. Every gross defect of this sample that was still present in the rewritten text has since been corrected (27 rows, Table 5). This sample can therefore no longer measure the current text without bias, so we drew a fresh one.

7.4 Two fresh paired audits of the current text

We drew 632 new articles, never sampled before, with the same strata and twice the allocation. Each was audited twice and blind: once on the current rewritten text (audit 7) and once on the official text, by 20 agents on each side. The rewritten-text auditors were told explicitly that a defect inherited from the official text counts, which makes this side stricter than in audits 1–6. The 52 defects seen on only one side were again checked against both texts.

The 28 defects still present in that sample were then corrected (36 rows). The audit was replicated on a second fresh sample of 632 articles (audit 8), with the same design and a different model (Claude Fable 5.1 instead of Claude Opus 5.5); 44 one-sided defects were checked. Audit 8 was read while those corrections were being applied, but the correction archive shows that none of the 63 rows applied since the first fresh sample was drawn touched an audit-8 article before it was read; the only one that later did was applied 28 minutes after the last audit-8 verdict on the rewritten text (the official-text side, which corrections cannot affect, finished two minutes before it). The two audits therefore measure the same state of the corpus on disjoint articles, which is why we pool them. The 16 defects found by audit 8 were left uncorrected, so that sample remains an untouched measurement.

The fresh samples confirm and sharpen the paired comparison (Table 8):

- the rewritten law has **54% fewer gross defects** than the official law (95% CI 40–64%): 3.5% against 7.5% on 1,264 articles;
- among rewritten articles, the rate falls from 14.5% to 6.0%, 58% fewer;

Fresh samples, verified	Articles	Official law G	Official law Rate	Rewritten law, at audit G	Rewritten law, at audit Rate	Repaired / introduced
Audit 7 (Opus 5.5)	632	49	7.8% ± 2.1	28	4.4% ± 1.6	28 / 7
Audit 8 (Fable 5.1)	632	46	7.3% ± 2.0	16	2.5% ± 1.2	32 / 2
Both	1,264	95	7.5% ± 1.5	44	**3.5% ± 1.0**	60 / 9
rewritten by the project	581	84	14.5%	35	6.0%	
retained unchanged (control)	683	11	1.6%	9	1.3%	
Paired test on both samples (McNemar, exact)					$p \approx 2 \times 10^{-10}$	

Table 8: Gross defects in the official law and in the rewritten law at the time of the audits, on two fresh random samples of 632 articles each, never sampled before, after verification of every defect seen on only one side.

- the rewrite repaired about seven defects of the official law for each one it introduced (60 against 9), against two to one in the first paired sample;
- the control agrees: retained articles, whose text is identical on both sides, show 11 and 9 defects;
- the two models agree on the official text (7.8% and 7.3%); on the rewritten text, Fable finds fewer defects (2.5% against 4.4%), within the intervals;
- taken alone, audit 8, the most recent sample and the one left uncorrected, gives about two thirds fewer defects (16 against 46, 2.5% against 7.3%); we keep the pooled 54% as the headline because it rests on twice as many articles and on two models.

Raw verdicts give 8.4% and 6.8% for the official text, against 2.4% and 1.4% for the rewritten text. Of the 44 defects found in the rewritten text, 35 were inherited from the official law and 9 were introduced by the rewrite. The 28 found by audit 7 have since been corrected; the 16 found by audit 8 remain open. The tax code, the exception in the first paired sample, shows 3 official and 5 rewritten defects on 96 articles here.

7.5 Sweeps: yield and plateau

Sweep yields fell as each family was exhausted. The "statute removes, decree keeps" sweep reached 3% G per lead, against 14% for the first sweep. Only re-qualification brought the yield back up: when earlier minor leads were re-read at the audit threshold, 56 of 166 turned out to be gross.

A pilot on 250 articles whose review notes announced an open coordination found that 60% of such notes were simply outdated. Its G rate was 4.8%, barely above random reading, so the full sweep was not run.

7.6 Systematic reading of the tax code

The chapter-by-chapter reading of the 2,740 rewritten CGI and annex articles gave:

- 96 gross defects (3.5%);
- 383 minor defects;
- 2,261 articles with nothing to report.

The 3.5% is far below the 9–12% that the pooled audits suggested for these strata. Those estimates rested on small samples (76 and 128 articles), and rounds 4 to 6 had concentrated on the tax annexes. We cannot exclude that a reader going through a whole chapter is somewhat less strict than a blind auditor.

Of the 96 defects, 86 are corrected (81 rows, 16 of them requested across groups). Ten remain open. Nine are VAT reforms written into CGI subdivisions that Ordinance No. 2025-1247 moves to the CIBS on 1 January 2027, so the reforms would have no effect; the CIBS is outside the corpus. The tenth is a political arbitration left to the author.

The reading consumed about 30 million tokens (excluding cache reads), roughly 11,000 tokens per article.

7.7 Cost

What was paid.

- **Drafting passes (OpenAI Codex):** about US$1,000 of API credits, plus a ChatGPT Pro subscription (US$200 per month).
- **Third pass (Claude Code):** a Claude Max subscription (US$200 per month), with no API credits billed. Its logged usage over six days is 32,046 model requests, 29.5 million output tokens, 5.1 billion cache-read tokens and 135 million cache-write tokens. That is about **US$2,500** at the list API prices of the two models used (Claude Opus 5.5 for 97% of the requests, Claude Fable 5.1 for the rest), consistent with Claude Code's own cost display. At the prices of the previous Opus generation (US$5 and US$25 per million input and output tokens, cache reads at US$0.50), the same usage would cost about US$4,100; we keep that figure as an upper bound.

Counting one month of each subscription, the cash outlay for the whole project was about **US$1,400**, under 5 cents per article. At API prices it would have been about US$3,500 (12 cents per article), and at most US$5,100 (17 cents) with the upper bound above. These figures exclude human time: a few days of direction and arbitration by one person.

What the same work would cost with lawyers. This is an order-of-magnitude estimate, not a measurement, and it depends on its assumptions, so we give two scenarios (Table 9). Both count only the work actually performed here: 14,115 rewrites and 16,199 justified retentions in the first pass, 30,314 re-readings in the second, and about 12,500 audit and sweep readings, 1,943 classified alerts and 1,285 corrections in the third. The *conservative* scenario assumes 2 hours per rewritten article, half an hour per reading and an hour and a half per correction. The *realistic* scenario assumes 6 hours per rewritten article, which leaves time to check its cross-references and implementing texts, 1 hour per retained article and per coherence reading, half an hour per audit reading, 1 hour per alert and 3 hours per correction. Costs use €100 per hour for in-house legal drafters and €250 for a blended law-firm rate, and US$1.10 per euro.

			Times the cost of the AI work	
Scenario	Lawyer-hours	Lawyer cost	at API prices	cash spent
Conservative	60,000	€6–15 M	1,300–4,700×	4,700–11,800×
Realistic	143,000	€14–36 M	3,100–11,300×	11,000–28,000×

Table 9: Order-of-magnitude cost of the same work by lawyers. 60,000 hours is 35 to 40 lawyer-years; 143,000 hours is about 85. The AI work cost about US$3,500 at API list prices (US$5,100 with the upper bound of §7.7, which gives the lower multiples) and about US$1,400 in cash.

Across both scenarios, the AI work is therefore **2,500 to 10,000 times cheaper**: about 2,500 times at the middle of the conservative scenario, and about 10,000 times in the realistic scenario at law-firm rates or against the cash actually spent. The comparison is one of cost, not of quality: the output of this project still awaits review by lawyers, which no AI measurement replaces.

7.8 Examples of defects

The following examples, all corrected unless stated otherwise, show the kind of errors AI drafting produces and AI review can catch:

- *Re-creating a tax the project abolishes.* The inventory had selected a version of CGI 1609 *quinvicies* that never entered into force. Its proposal re-instituted the additional apprenticeship contribution that the proposed Labour Code L6242-1 abolishes.
- *A "reduced" rate above the normal rate.* After the corporate-tax reform, the reduced 24% rate for non-profit bodies (CGI 219 *bis*) exceeded the new normal rate of 20%.
- *Half-repealed chapters.* In CGI annex II, a correction round repealed some articles on approved management centres, whose legal basis (CGI 1649 *quater* C to O) had been repealed in 2025. Neighbouring articles still organised the approval procedure. The next audit caught it, and the whole chapter was then treated at once.
- *Future versions undoing the reform.* A Mayotte adaptation scheduled for 2028 (Social Security Code L759-23) applied "in the following wording" an article that the project rewrote, and so restored the rule the project removes.
- *A regulation against its statute.* A proposed tax decree dropped the prior accreditation of fiscal representatives, while the statute (CGI 1671) still required it. The coherent fix aligns the statute instead of restoring a formality.
- *A coherent cut that removes a protection (corrected after a report).* The rewrite of Urban Planning Code L600-8 removed the sanction on settlements that pay for the withdrawal of an appeal against a building permit (Ordinance No. 2013-638), without breaking any cross-reference. No audit had caught it; the third author reported it (issue #17), and the text in force was restored.

8 Discussion

What AI agents do well. At this scale, agents are tireless, complete and literal readers. They reproduce full articles, including tables of several hundred thousand characters. They follow cross-references through nine instruments and many versions, and they keep exact quotations. Combined with typed changes and guarded writes, this made it possible to rewrite and then repair a corpus that no individual could re-read in a week. It also made the work fully auditable: every change is archived with its before-image and its reason.

What they do badly.

- *The boundary of the corpus.* Many defects involve a text that was not provided: the CIBS, non-codified statutes, ministerial orders. An agent can only flag them as documentary reserves, and must not guess.
- *Time.* Legal texts have scheduled future versions. A reform written on a version that ends before the reform can take effect is void. A future version with no proposal silently restores the old rule.
- *Local coherence without global coherence.* A single change is valid in isolation, but its neighbours are left inconsistent. One of our own correction rounds created exactly such a half-repealed chapter. We fixed this by treating whole chapters at once and by running a regression check after each application: any article still citing an article repealed in the round.

- *Calibration.* Readers differ in how strictly they separate gross from minor. Early sweeps under-rated dead references as minor. Re-qualifying them at the audit threshold turned a third of them into gross defects.

Measurement changes the conversation. Without the random audits, the number of corrections, over 1,200, would have suggested steady progress. The audits show a plateau around 3–4%, and they explain it. Targeted sweeps help, but the remaining defects are scattered. Only systematic reading can bring the rate clearly down. The paired baseline turns the plateau into a result: the law in force is not defect-free either, and the rewrite, with its residual errors, is measurably more coherent than the texts it replaces. We consider a published, reproducible defect rate with an interval, compared with a baseline, the minimum standard for any claim about AI-drafted legislation.

Contradictions versus politics. Keeping the red team away from political choices was harder than expected. Many apparent contradictions have two coherent resolutions: removing a favour or extending it, restoring a formality or deleting its remnant. We resolved this with an explicit, published alignment criterion (§5), and we sent genuine arbitrations back to the author. A different programme would change the criterion, not the method.

Portability and neutrality. Two arguments support portability. First, most measured defects (dead references, future versions, unaligned decrees, contradictions) do not depend on the direction of the reform: they would arise with any programme that amends many articles. Second, the evaluation is applied to the official law with no programme at all. Two arguments call for caution. The drafting instructions and the alignment criterion carry the programme's orientation; and models may handle reforms that point in different directions differently. It is therefore reasonable to expect comparable defect detection from one programme to another. A comparative experiment would confirm it: run the same pipeline on at least two contrasting programmes, with the same evaluation protocol, and compare defect rates. We invite others to run it.

Expert review. The results are AI-only by design, and no lawyer has validated them yet. The pipeline makes expert review *cheaper*, not unnecessary: each article comes with its source, reason, notes, objections and audit trail. The public issue tracker turned out to be the most efficient channel. Nine of the first eleven external issues pointed at a real defect.

9 Limitations

- Results are AI-measured by design. Audit verdicts are model judgements and may be wrong in either direction; their accuracy is open to expert verification, which we invite.
- A coherent removal of a safeguard that the programme intends is counted as a defect only if the auditor judges it abnormal; F9 risks have no published rate.
- Auditor strictness varies. On identical texts, two sets of auditors disagreed by a factor of two before verification (§7.3). Single-audit rates should be read with this in mind. Audit 8 ran on a different model from audits 1–7; it agrees on the official text but finds fewer defects in the rewritten text, and a difference in strictness between models cannot be excluded.
- Partial self-evaluation. The model that made the corrections (Claude Opus 5.5) also ran audits 1–7, and both audit models come from the same vendor; a cross-audit of the same

sample by another vendor's model remains to be done, and the published verdicts and seeds let anyone run it.

- The lawyer-cost figures are orders of magnitude built from two stated sets of assumptions, not a quotation or a measurement.
- Only one programme has been implemented; the comparative experiment with other programmes remains to be run, and the drafting instructions and alignment criterion must be rewritten for each programme (Table 3).
- The corpus is frozen at 18 October 2026, and texts outside the nine instruments are not rewritten.
- Case law is not taken into account. Drafting and audits read the codes only, not the court decisions that interpret them, so a rewrite may unintentionally depart from a settled interpretation, and an audit may miss a defect that the courts have already identified.
- Audit samples of about 314 articles detect changes of about ± 2 points only, so per-stratum rates are imprecise.
- The drafting models and the reviewing model come from different vendors, which limits shared blind spots but does not remove them.
- Costs are given as paid, plus an API-equivalent estimate for the third pass. They exclude human time and are not a productivity benchmark.
- The economic and fiscal effects of the reform are outside the scope of this paper.

10 Conclusion and future work

We rewrote five French codes and the four CGI annexes end to end with AI agents, red-teamed the result with a separate pipeline, and measured what remains wrong. Measured against the official law on the same articles, with every disputed defect verified, the rewritten law has 34% fewer gross defects on 1,879 articles at the time of the audits, and **54% fewer** on two fresh samples totalling 1,264 articles drawn at the end of the work (3.5% against 7.5%). Four fifths of the defects found are inherited from the law in force. It cost about US$1,400 in cash, excluding human direction time, against an order-of-magnitude estimate of €6 to 36 million for the same work by lawyers (§7.7). The full trail is public.

Next steps:

- systematic reading of the remaining rewritten articles: the housing and social security codes (about 5,500), then labour and town planning (about 5,900);
- bringing the CIBS into the corpus;
- in a second version, taking case law into account: linking each article to the decisions of the Conseil constitutionnel, the Conseil d'État and the Cour de cassation that interpret it (DILA open data and the Judilibre API), so that rewrites respect settled interpretations and audits check them;
- validation by practising lawyers, starting with the open objections and arbitrations;
- a comparative experiment: running the same pipeline on at least two contrasting programmes, with the same evaluation protocol.

Towards a policy-configurable pipeline. Table 10 describes the pipeline this work begins, and separates what is demonstrated here from what remains to be built.

Step	Status in this work
1. Ingest a party's programme	To build: done by hand, under the author's direction.
2. Convert it into structured policy specifications	Partly: explicit policy objects, written by hand.
3. Identify the relevant laws, regulations, cross-references and dependencies	Demonstrated on five codes and four annexes; out-of-corpus texts (CIBS, non-codified statutes) to be added.
4. Propose amendments across the affected corpus	Demonstrated: 14,115 articles rewritten.
5. Detect contradictions, unintended effects and ambiguities	Demonstrated and measured: red team, register, audits.
6. Produce a versioned, auditable, article-by-article proposal	Demonstrated: public repository, reader, before-images.
7. Estimate effort and list questions needing human judgement	Partly: costs measured, arbitrations sent back to the author.

Table 10: A policy-configurable pipeline: what is demonstrated and what remains.

France's 2027 election provides an immediate use case, but the scientific objective extends beyond any election or party: to explore whether AI can turn legislative reform from a costly, fragmented manual process into a reproducible, auditable and policy-configurable computational workflow. Competing programmes could then be translated into comparable, technically coherent legislative proposals, making the practical consequences of political choices easier to inspect and debate. Beyond this case study, the method is a template: an explicit policy layer, complete drafting, guarded writes, a separate red team and honest measurement. We hope it makes the law more legible, and that it gives anyone a concrete way to check whether a proposed reform is coherent.

Call for contributions

These texts are not foolproof: about 3% of articles still carry a gross defect, and no lawyer has validated them. Everything is public:

`https://github.com/AlexMasmej/liberer-la-france`

We invite lawyers, legal scholars, civil servants, French citizens and anyone interested to help harden the code. To report a defect, open an issue with:

- the article (code, number and `LEGIARTI` identifier);
- the exact passage;
- the problem and, if possible, a proposed correction.

Every issue is triaged against the register. Contradictions and loopholes are corrected, and the issue is closed with a link to the commit. Requests that would change a political orientation are explained and closed. Pull requests are welcome too.

We also invite researchers, legal scholars, political organisations and independent contributors to adapt the pipeline to other programmes: propose other policy configurations, write structured representations of other parties' programmes, and check whether the pipeline produces coherent amendments under different assumptions. The ambition is not to promote a candidate, but to build reusable infrastructure that could support the legislative work of any future government.

Data and code availability

The rewritten corpus, the official sources, policies, batches, the third-pass reports and register, the audit samples and verdicts, and the sweep and reading tools are public at `https://github.com/AlexMasmej/liberer-la-france`. A comparative reader and complete downloads are at `https://www.libererlaretraite.com/corpus/` and `https://www.libererlaretraite.com/telechargements/`.

Use of AI

AI systems did most of the drafting, reviewing and correction work described here, under human direction and arbitration. This manuscript was drafted with the assistance of Claude (Anthropic) from the project's records and reviewed by the authors.

Author contributions

Contributions are given with the CRediT taxonomy.

- **Alexandre Masmejean:** Conceptualization (conceived the project); Methodology (set the policy layer); Investigation (ran the drafting passes); Supervision and Project administration (directed the project and arbitrated all political choices); Writing – review & editing.
- **Nathalie Gouailhardou:** Conceptualization (proposed systematically comparing the error rates of the official legal texts with those of their AI-rewritten equivalents, the comparative evaluation of §7.3 and §7.4); Methodology (identified that assessing the official and rewritten texts differently would introduce confounding factors, and proposed applying the same evaluation methodology and criteria to both, which established a consistent baseline for a valid comparison).
- **Kevin Primicerio:** Investigation (independent external review of the published codes with his own AI agents); Validation (issues and acceptance tests, triaged and partly integrated); Writing – review & editing.

All authors have read and approved the final manuscript.

Acknowledgements

We thank Rand Hindi for the inspiration behind the red-teaming approach of the third pass. We thank Emmanuel Masmejean for his careful reading of the French version. We thank everyone who read, criticised and shared the initial thread.

References

[1] Constitution of 4 October 1958, Articles 6 and 7 (election of the President of the Republic by direct universal suffrage, for five years).

[2] Conseil constitutionnel, Décision n° 99-421 DC du 16 décembre 1999, *Loi portant habilitation du Gouvernement à procéder, par ordonnances, à l'adoption de la partie législative de certains codes*.

[3] Conseil d'État, *Simplification et qualité du droit*, Étude annuelle 2016, La Documentation française, 2016.

[4] A. Masmejean (Alex Masmej, `@AlexMasmej`), Thread on X, 19 October 2026, `https://x.com/AlexMasmej/status/2112088220804722917` (engagement figures read on 31 October 2026).

[5] D. Lisnard and Nouvelle Énergie, Programme for the 2027 French presidential election, `https://www.unenouvelleenergie.fr/`.

[6] Direction de l'information légale et administrative (DILA), LEGI: codes, lois et règlements consolidés, `https://www.data.gouv.fr/datasets/legi-codes-lois-et-reglements-consolides`.

[7] N. Guha et al., LegalBench: A collaboratively built benchmark for measuring legal reasoning in large language models, *NeurIPS Datasets and Benchmarks*, 2023. arXiv:2308.11462.

[8] D. Merigoux, N. Chataing and J. Protzenko, Catala: A programming language for the law, *Proceedings of the ACM on Programming Languages* 5(ICFP), 2021. arXiv:2103.03198.

[9] OECD Observatory of Public Sector Innovation, *Cracking the Code: Rulemaking for Humans and Machines*, OECD, 2020.

[10] D. Ganguli et al., Red teaming language models to reduce harms: Methods, scaling behaviors, and lessons learned, arXiv:2209.07858, 2022.