Freeing the Law with LOCUS: A Local Ordinance Corpus for the United States

Anonymous Author(s)
Affiliation
Address
email

Abstract

Progress in legal AI increasingly depends on access to authoritative legal text at scale. Yet one of the most consequential layers of American law remains largely absent from existing machine-readable corpora: local ordinances. Local codes govern zoning, housing, business licensing, public health, noise, animal control, and many other domains of everyday regulation, but they are fragmented across vendor platforms designed for human browsing rather than bulk research access. We introduce LOCUS—the **L**ocal **O**rdinance **C**orpus for the **U**nited **S**tates—a comprehensive corpus and county-harmonized access layer for U.S. municipal and county ordinance codes. The raw corpus, available for release to researchers, represents nearly all publicly available municipal and county ordinance codes. The resulting raw corpus contains codes from 9,239 cities and counties. A smaller county-harmonized LOCUS access layer provides coverage for the largest 2,309 of 3,144 U.S. counties, accounting for a majority of the population. We use OCR to handle the myriad of document formats that have kept the law from being a public resource. We release the corpus with coverage metadata to support reproducibility, downstream legal AI research, and the incremental expansion of machine-readable access to local law. We train a collection of ModernBERT-based classifiers and scorers to facilitate analyzing U.S. local law among several dimensions, such as opacity and paternalism, that have not previously been studied at this scale. LOCUS-v1 and its derivative models are available at: https://huggingface.co/datasets/LocalLaws/LOCUS-v1

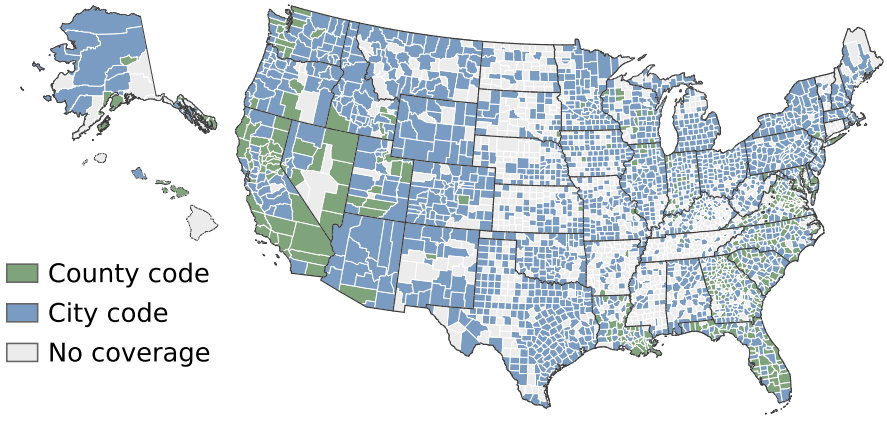

Figure 1: LOCUS represents the longest digitally available code—city or county—for each county.

Submitted to 40th Conference on Neural Information Processing Systems (NeurIPS 2026). Do not distribute.

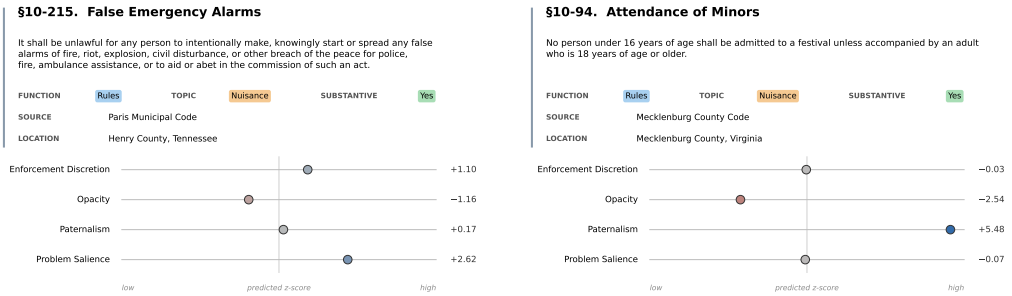

Figure 2: Two example ordinances with predicted scores (in standard units) on four axes (*opacity, enforcement discretion, paternalism*, and *salience.*) produced by ModernBERT regressors (§5) and function/topic labels produced by ModernBERT classifiers (§4.4). Together they demonstrate the per-ordinance analysis enabled by LOCUS.

1 What it means to "free the law"

Legal AI systems increasingly operate over statutes, cases, regulations, contracts, and administrative materials [Chalkidis et al., 2022, Henderson et al., 2022, Guha et al., 2023]. This expansion has been accompanied by domain-specific resources for case law [Zheng et al., 2021], contracts [Hendrycks et al., 2021, Koreeda and Manning, 2021, Tuggener et al., 2020], and statutory reasoning [Holzenberger et al., 2020]. Despite this, they still lack systematic access to one of the most consequential layers of American law: local ordinances. These codes govern zoning, housing, building permits, business licensing, public health, noise, signs, animal control, and other domains of everyday regulation. For many questions faced by residents, businesses, landlords, and local governments, the relevant legal text is not only federal or state law, but a municipal or county code.

Local law is not merely another collection of statutes. It is a layered system of legal authority. State statutes, county ordinances, municipal codes, home-rule provisions, charters, preemption doctrines, and issue-specific delegations can all interact. Whether a state rule, county rule, or municipal rule controls is often not obvious in the abstract and may depend on the legal domain. This makes local law a particularly important setting for legal AI: a useful system must not only retrieve text, but identify the relevant jurisdictional layer and reason about overlap, delegation, and conflict among sources of authority.

We introduce **LOCUS-v1**, a large-scale corpus and county-harmonized access layer for U.S. local ordinances. The first release of LOCUS adopts a deliberately transparent simplification: for each U.S. county, we record the most substantial available local code among the county ordinance code and the ordinance code of the county's largest municipality, using document length as a reproducible proxy for local-law coverage. This representation does not purport to decide which local authority controls every legal question. Rather, it provides a common geographic substrate on which local legal text can be searched, compared, and connected to population, geographic, Census, and policy data.

The need for such a dataset arises because local law is public but not practically available as a national research corpus. Georgetown Law Library [2026], the most applied-to law school in the United States comments, "*there is unfortunately no single source where you can find a comprehensive collection of all municipal codes.*" U.S. local codes are fragmented across commercial vendor platforms designed for in-browser reading rather than bulk research access. Vendors expose different navigation structures, print workflows, dynamically generated PDFs, and jurisdiction indexes. No central registry maps every county or municipality to its hosting platform, and no vendor provides a complete machine-readable index of all jurisdictions it hosts. As a result, constructing a national corpus requires discovering where each code lives, extracting it through platform-specific workflows, validating the resulting artifacts, and harmonizing them to a common unit of analysis.

We leave full issue-specific hierarchy and conflict modeling to later releases and benchmark tasks. This staged design reflects both the legal complexity of determining controlling authority and the need to preserve uncontaminated evaluation settings for future legal-reasoning benchmarks.

LOCUS enables a new class of legal AI and empirical legal studies applications. At the retrieval layer, it supports search and question answering over local rules whose terminology varies substantially across jurisdictions. At the representation layer, it enables structured extraction of regulated activities, permits, fees, penalties, effective dates, and cross-references. At the reasoning layer, it creates a foundation for benchmarks that test whether systems can navigate multiple layers of law, identify the relevant jurisdictional authority, and reason about state-local or county-municipal overlap. By making local law observable at national scale, LOCUS turns a fragmented body of public legal authority into infrastructure for legal retrieval, regulatory extraction, comparative policy analysis, and legal-domain language model evaluation.

We provide a summary of our corpus (§3), decision points necessary to create it (§4), evaluations of the corpus (§5), and a discussion of how this can improve our understanding of the legal system (§ 6).

2 Related Work

Studying the law has been important in society for centuries [Holmes, 1897]. In the Information Age, the law has become both immediately accessible but increasingly complicated. We are not the first to create corpora for legal NLP [Steinberger et al., 2006, Aletras et al., 2016, Livermore et al., 2017, Harvard Law School Library Innovation Lab, 2018]. Neural network era corpora such as ECHR [Chalkidis et al., 2019] and pile of law [Henderson et al., 2022] contain case law, court and administrative opinions, and legal codes but not the local law. The 162 tasks in LegalBench [Guha et al., 2023] draw heavily from contracts and merger agreements and none involve local ordinances.

Access to the law is a historical challenge which has been reshaped in part by the internet. Georgia v. Public.Resource.Org, Inc., No. 18-1150 (decided April 27, 2020) [Supreme Court of the United States, 2020] upheld that laws, statutes, and court decisions are public domain, in so far as digital content goes. Since that time the rise of large language models and other modern techniques has enabled intelligent data processing on an unprecedented scale; standardizing over 9,239 one-thousand page documents would not have been feasible several years ago. Local laws have been understudied in part due to data access that we hope LOCUS will resolve.

3 Properties of LOCUS

Our corpus benefits both the technical and social science communities by providing valuable data and insight. We discuss the harmonized LOCUS access layer and additional data provided for researchers.

3.1 A County-Harmonized Access Layer

LOCUS adopts a transparent simplification: for each U.S. county, it identifies the most substantial available local code among the county ordinance code and the ordinance code of the county's largest municipality. This design does not purport to determine which layer of law controls in every doctrinal context. Instead, it provides a reproducible substrate for retrieval, comparison, and future benchmarks on state–county–municipal legal reasoning.

Figure 3 summarizes our publicly released corpus: 2,211,516 chunks of text, out of which the majority are judged to be substantive laws in nature. We define substantive as concerned with rules or enforcement, rather than any text that is purely structural, process-oriented, or purely context; the majority of our annotations are rules. These substantive laws deal with four major categories: buildings, business licensing, zoning, and nuisance. The remainder, roughly a third of the laws are categorized with near 90% precision as other. We investigate the headers of these chunks and find that other constitutes topics such as government, employment matters, and animal regulation (this last category makes Alaska have a disproportionately large share of 'other' chunks). Table 1 provides examples that illustrate the diversity of laws.

3.2 Additional Data for Researchers

In addition to the released data, we collect an additional 7,000 documents of other cities and counties. We intend to make this data available to researchers with signed release similar to MIMIC [Johnson

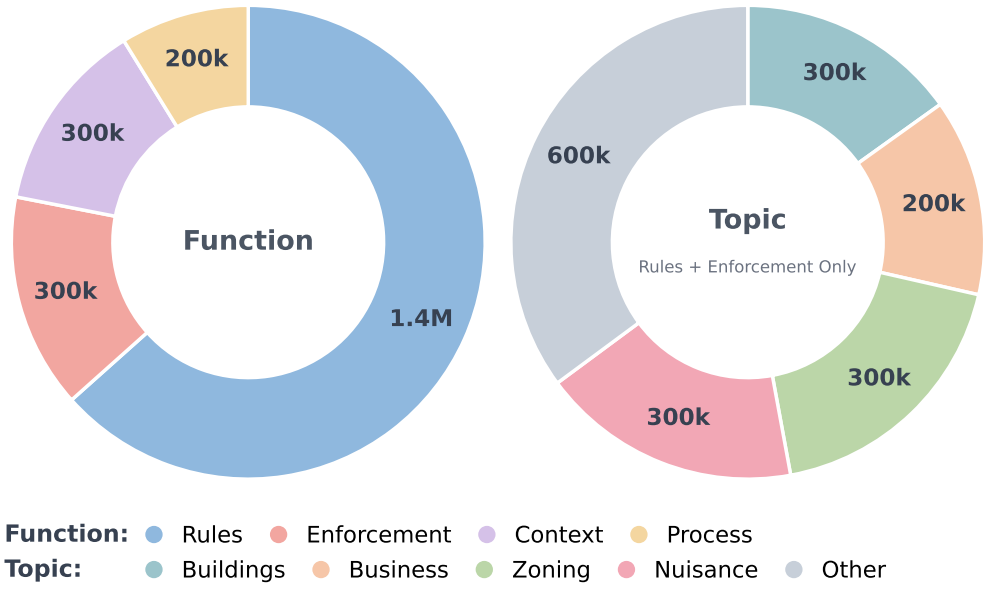

Figure 3: We annotate our corpus at the chunk level along its *Function*, and the substantive laws {Rules and Enforcement} according to the *Topic* referenced. Table 1 provides example texts.

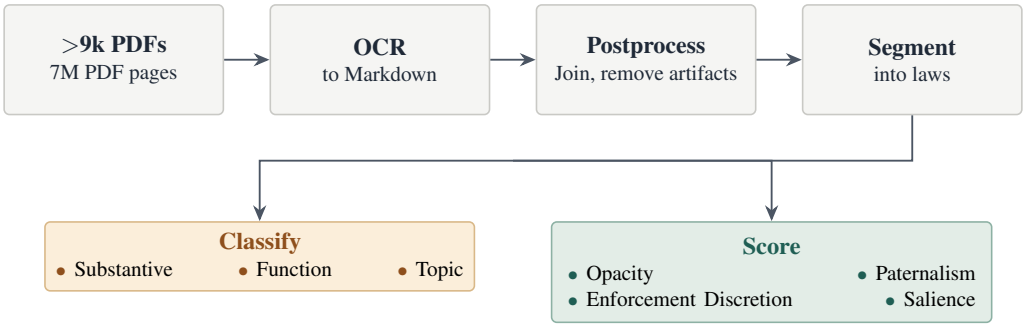

Figure 4: **Processing pipeline.** A corpus of more than 9,000 PDFs (7M pages total) is OCR'd into Markdown, cleaned, and segmented into individual laws. Each segment is then independently *classified* for function, topic, and substance and *scored* on four normative dimensions.

et al., 2016, 2023]. Given current LLM ingestion policies, we believe this is necessary for any future evaluation of local law coverage by foundational models [Dahl et al., 2024].

4 Constructing LOCUS

An overview of the pipeline is shown in Figure 4.

4.1 Collecting the data

The original raw corpus contains 9,239 valid PDFs totaling approximately ∼80 GB. Constructing LOCUS required solving a coupled systems and legal-data problem across thousands of jurisdictions. Our pipeline uses browser automation and vendor-specific download logic to collect municipal and county codes from major hosting platforms. The construction process surfaced several nontrivial failure modes, including server-side PDF assembly limits, filename collisions among non-unique municipality names, hidden interface thresholds, 15 second crawl delays, anti-bot measures, and multi-county consolidated cities. Addressing these failures required targeted recovery techniques

Label	Representative example
Function	
RULES	No direct seller shall engage in direct sales within the city without receiving a permit for that purpose as provided herein.
ENFORCEMENT	Any Code Enforcement Officer may issue notices of violation and administrative citations, inspect public and private property, and enforce any available administrative remedies.
STRUCTURAL	Park Regulations 973.01 Adoption and purpose. 973.02 Powers. 973.03 Enforcement. 973.04 Application to concessions.
CONTEXT	The purpose of this notice and review procedure is to notify the public of the permit review process for development proposed in areas having identified significant resources and functional values.
PROCESS	Application for a license required by this division shall be filed with the city clerk on forms provided for that purpose.
Topic	
BUILDINGS	The registered design professional shall submit sufficient technical data to substantiate the proposed alternative engineered design and prove that the performance meets the intent of this code.
BUSINESS	No direct seller shall engage in direct sales within the city without receiving a permit for that purpose as provided herein.
ZONING	Where a change of use of an existing structure requires additional parking or other requirements applicable to a new use, a site plan shall be submitted for review.
NUISANCE	Bike paths may be used only for the operation of bicycles and pedestrian use.
OTHER	Monies appropriated for salaries, wages and related benefits shall not be used for general operations, capital outlay, or other purposes without recommendation from the Mayor and specific approval of a majority of the council.

Table 1: Representative examples for the five *Function* labels and the five merged *Topic* labels in LOCUS. All items in the topic group are annotated as *Rules* or *Enforcement* in their function.

rather than a single generic scraper. Furthermore, we manually collect self-hosted or pdf-restricted codes for cities and counties which are not covered by this methodology.

4.2 Identifying salient laws

Given the huge amount of data, and the diversity of its content and format, we employ a two-level zero-shot approach as the initial labeling approach. Given that our data is being ingested in thousands of different formats after OCR, we need to remove structural content (i.e., stray headers, table of contents) and identify the substantive chunks.

After preliminary investigation of Anthropic and Gemini, we settle on OpenAI's GPT-5.4 as a fast and reliable annotator for this data [OpenAI, 2026]. After comparing a 500 sample of 5.4 mini and nano, we select nano as a cost-effective and only marginally worse option for large-scale annotation. Inspired by LLM-as-a-Judge [Zheng et al., 2023], we evaluate the 5.5% of annotations deemed most challenging with a much more expensive GPT-5.4 model. The model agrees on 64,977 out of 108,889 predictions. The more advanced model often decreases its predictions of rules in favor of process and enforcement. Crucially, no models hesitated in identifying structural content, which was ultimately removed from our release. We intend to maintain this dataset and hope to get support from the LLM and law communities in improving these annotations as we update the corpus. Ideally, direct evaluation by lawyers and judges would enable us to exceed the limitations of LLM-as-a-Judge.

4.3 OCR and Processing

The ordinances are stored in diverse layouts and formats, including single- and double-column layouts, born-digital, exported, and scanned documents, etc. To best handle this diversity, the pipeline

for building LOCUS starts by running optical character recognition (OCR) to convert every image of a page to Markdown.

We accomplish this with LightOnOCR-2-1B [Taghadouini et al., 2026], an open 1B parameter vision-language model (VLM) based on Qwen-3 [Bai et al., 2025] finetuned on 16MM PDF pages that scores highly on a standard OCR benchmark, OlmOCR-Bench [Poznanski et al., 2025]. LightOnOCR-2-1B generates Markdown text from a page image. We find that this model is robust to the diversity of the raw ordinances, consistently generating correct text in natural reading order.

The rest of our post-processing pipeline consumes the unified Markdown output to stitch together laws across pages. We strip artifacts such as repeated headers, footers, and page numbers, and merge content that crosses pages such as paragraphs and tables. The next stage of this post-processing pipeline is to segment the joined content into individual laws, identifying section and subsection headers.

The final step of our post-processing pipeline is to classify the substantivity, function, and topic of each extracted law. We discuss the construction of these classifiers in 4.4. Each is trained on the roughly 100M parameter ModernBERT-base [Warner et al., 2025] encoder, which enables us to efficiently run inference on every law. Segments that are classified as purely structural, rather than containing any laws, are omitted from the dataset.

The raw ordinances are contained in roughly 7M pages. We are able to scale our OCR pipeline on Modal[1]. Given the relatively small size of LightOnOCR-2-1B and Modal's batch inference support, we were able to efficiently run the entire pipeline and process documents across all formats at roughly $0.30 per 1,000 pages.

4.4 Annotating the Law

To organize the ordinances, we develop three classifiers: **substantivity**, **function**, and **topic**. A breakdown of the label space and selected examples are shown in Table 1.

We build these classifiers by sampling 100,000 laws from the pipeline discussed in the previous section and using GPT-5.4-nano to annotate each of them. The resulting labels are used to train a ModernBERT classifier [Warner et al., 2025], which can be efficiently used for inference across the rest of the dataset. The classifiers are trained using 80,000 samples for training, 10,000 for parameter sweeps, and finally evaluated on a 10,000 instance subset.

From this collection, LOCUS-v1 derives a county-harmonized release that records a representative local-law artifact for each covered county, together with the structured metadata from the classifiers.

4.5 Creating a Harmonized Access Layer

Our access layer illustrated in Figure 1 is built by a simple algorithm run on all the codes: for every county in the United States, is there an existing county code and an existing city code, ideally from the largest city in that county? If both exist, pick the longest by page length. This is an imperfect process but length of code and population of jurisdiction were correlated.[2] By doing this, we are able to provide a code for counties *representing* 94% of the United States by population. Since for example the second order city or the population living in the county outside the city are not captured by this, this access layer applies to a smaller literal population than the full data.

5 A Dimensional Analysis of Local Laws

By linking the text of the laws to the locales in which they apply, LOCUS-v1 opens the door for new types of analysis. In addition to the function and topic metadata, we annotate each ordinance in LOCUS-v1 with dimensional data. We consider four dimensions:

1. **Enforcement Discretion** (*highly discretionary* to *non-discretionary*) — how much selective judgment does the law leave to officials?

[1] https://modal.com

[2] Counties run on average slightly shorter than cities, but we opted for an easily interpretable selection algorithm rather than introducing weights; this did not dramatically impact the final selection as certain states, such as Maryland, have much more powerful counties than cities.

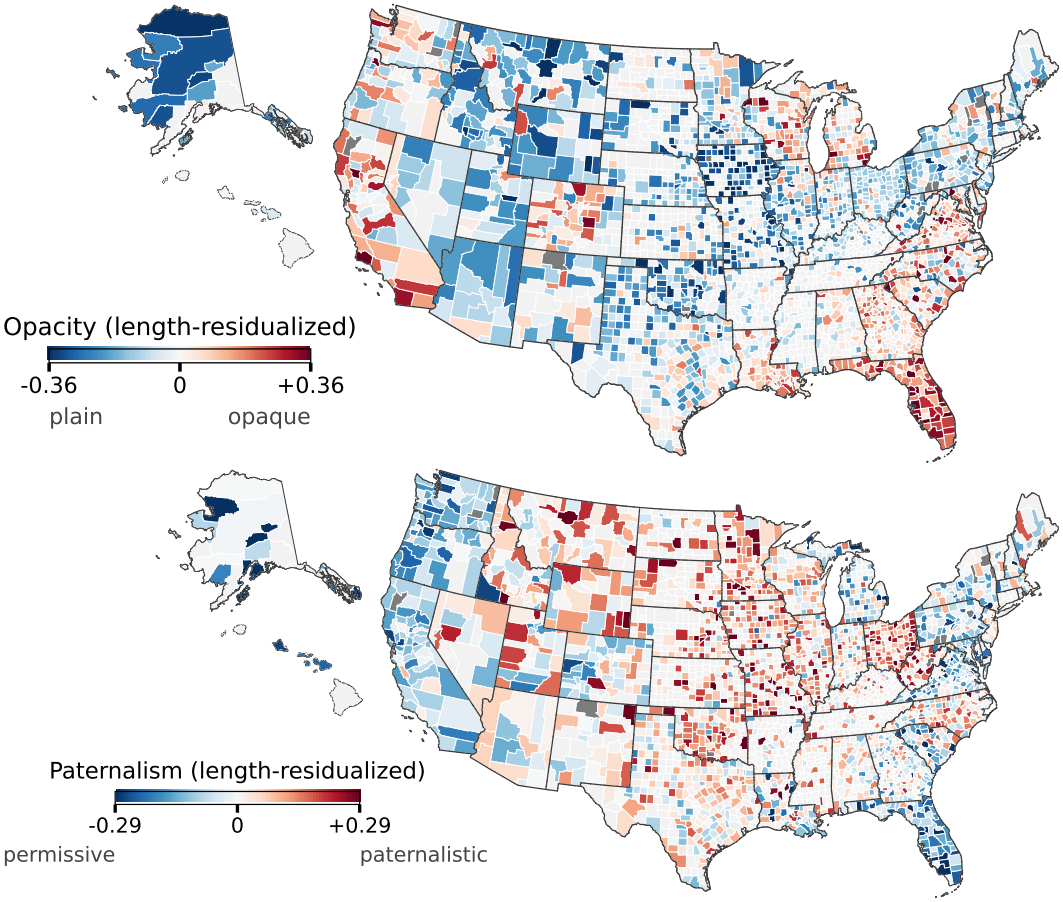

Figure 5: The opacity and paternalism of laws varies across the country. LOCUS facilitates studying the laws for macro trends such as discovering that Florida law is opaque but not paternalistic.

2. **Opacity** (*opaque* to *intelligible*) — how hard is it for an ordinary person to know what is required?
3. **Paternalism** (*paternalistic* to *externality oriented*) — is it protecting the actor from themself or protecting others/the public?
4. **Problem Salience** (*highly salient* to *unimportant*) —how strongly does it represent the issue as important, urgent, or threatening?

Examples of laws occupying these dimensions are given in Figure 2. For instance, the law preventing minors under the age of 16 from attending a festival unless accompanied by an adult is scored as highly paternalistic, intelligible, with neutral discretion and salience.

Our core intuition is that these dimensions are continuous and that they can better be used to order and measure laws rather than categorize them. Accurate models of dimensions allow us to bring into focus particular aspects of the law. Incorporating all laws onto the same set of axes enables analysis both within individual bodies of law (i.e., within a single city), but also for comparative analysis across bodies.

5.1 Building LOCUS Scorers

For our dimensional analysis, we fine-tune a ModernBERT-base with a linear regression head for each dimension to score a law. For each dimension, we generate 10,000 scores using 200,000 pairwise LLM-as-a-judge match-ups between ordinances. During each match, we ask the LLM to compare the two ordinances along a specific dimension, and return which better exemplifies that dimension. The model outputs A, B, or Tie. Order can produce bias in pairwise judgement [Liu et al., 2024],

so every (A, B) comparison pair is also judged in reverse order (B, A). Pairwise comparison aligns better with human judgement than direct/numeric scoring [Liu et al., 2024], motivating us to use it for dimensional scoring. Each ordinance's match history is used to compute its latent score along each axis using the Bayesian skill rating system, TrueSkill [Herbrich et al., 2006]. This gives us a total ordering over the sampled ordinances, along with an underlying mean, μ.

To train the regression model, we normalize the scores to their z-score by subtracting out the dimension's mean and dividing its standard deviation. For each dimension, we split the 10,000 scored ordinances into a training set (n=8,000), validation set (n=1,000), and test set (n=1,000). We fine-tune a ModernBERT regression model to predict the normalized TrueSkill score, using mean-squared error as our loss function. To evaluate the model, we compute Pearson correlation on the test set. This technique is inspired by the methodology behind Havelock.ai, an AI-powered orality detector that scores text on how oral or literate it is [Weisenthal, 2026].

The dataset for each dimension is constructed using a fixed 10,000 ordinance sample, and 200,000 pairwise comparisons using GPT-5.4-nano. We report the Pearson correlation coefficient of the trained BERT models versus the TrueSkill values in Figure 6. Each dimension has a correlation of between 0.82 and 0.94, implying the BERT-based scorers largely capture the dynamics of the TrueSkill model. We provide the prompts plus a sample of high- and low-scoring laws along each dimension in Appendix A. We also provide a website to view the TrueSkill scores for the 10,000 laws along each dimension.[3] We can use these scores to analyze the laws and correlate them with real-world values of interest, discussed in the next section.

5.2 Analysis

Figure 5 demonstrates the importance of studying this at a nationwide rather than a single case level. For example, counties are notably more opaque than cities on average and Florida is more than twice as opaque as any other state. Studying multiple dimensions in tandem can unlock new insights into unique laws; opacity and paternalism are only weakly correlated across sections (Pearson r=0.11 on n=2,211,516).

Finding interesting needles in this haystack of laws can be facilitated through this evaluation. For example, curfews are detected with paternalism and a subsequent analysis of the data provides insight into curfew distribution for minors across the United States. Headers containing 'possession' and 'alcoholic' are associated with paternalistic laws while 'definitions' and 'variances' are associated with opaque ones.

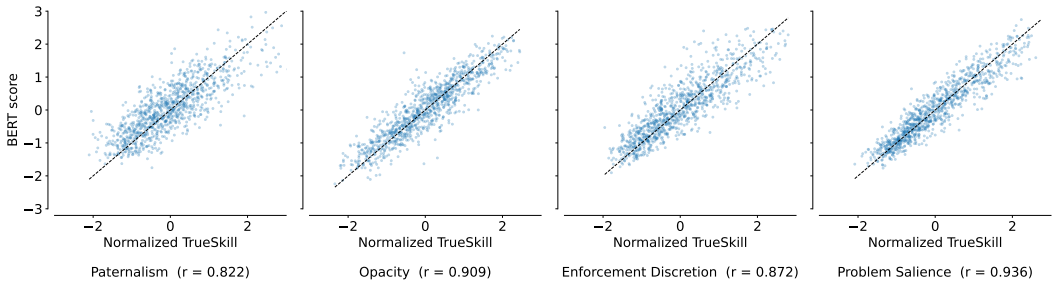

Figure 6: The Pearson correlation between the predicted BERT scores and the normalized TrueSkill scores on 4 distinct test sets (1,000 ordinances per dimension).

6 Discussion, Limitations, and Future Work

LOCUS-v1 is designed as an access layer, not as a final theory of local legal authority. Its county-harmonized release adopts a transparent simplification: for each county, we select the most substantial available local code among the county code and the code of the county's largest municipality. This design makes local law searchable and comparable on a national geographic substrate, but it does not determine which rule controls for a particular person, parcel, business, or legal question. In local

[3] https://locallaws–locus-leaderboards-web.modal.run

law, authority is layered. State statutes, home-rule provisions, county ordinances, municipal codes, charters, preemption doctrines, and issue-specific delegations may all matter. LOCUS therefore should be understood as infrastructure for retrieval, comparison, and benchmark construction rather than as a substitute for doctrine-sensitive legal analysis.

The corpus itself shows why this distinction matters. City and county codes are not interchangeable legal objects. Across the raw corpus, county codes contain substantially more zoning material, while city codes contain more nuisance and public-order regulation. This pattern is consistent with a functional division of local authority: counties more often regulate land, development, and unincorporated territory, while cities more often regulate density, proximity, and everyday public order. For downstream users, this means that jurisdiction type is not merely provenance metadata. It is part of the substantive representation of local law. Models trained or evaluated on local codes should therefore preserve whether a text comes from a municipal or county source, even when the release is harmonized to a county-level unit of analysis.

LOCUS also reveals that local codes share a common representational architecture. When ordinances are ordered by their position in a code, topics tend to appear in a stable sequence: general provisions and governmental structure near the front, followed by business regulation, nuisance and public-order rules, zoning, and building regulation. This finding suggests that local law is not simply a bag of rules. It is organized through a recurring documentary form. That form matters for legal AI. Retrieval systems, chunking strategies, and benchmark designs that ignore position within a code may miss information embedded in the structure of codification itself.

At the same time, LOCUS documents the limits of any simple national harmonization. In much of the country, counties and cities follow the functional pattern described above. In the Northeast, however, the relationship changes: counties appear less zoning-heavy and more enforcement-oriented, consistent with a different institutional history in which towns and municipalities retain more primary land-use authority while counties often perform administrative, health, or enforcement functions. The implication is not that harmonization is impossible. Rather, it is that harmonization must be explicit about what it preserves and what it abstracts away. A county-level substrate is useful because counties form a mutually exclusive and exhaustive national geography, but the legal meaning of a county code is not constant across states and regions.

These limitations point directly to the next generation of legal AI benchmarks. A system that can answer questions about local law must do more than retrieve a plausible ordinance. It must identify the relevant layer of government, distinguish city from county authority, incorporate state-law context, recognize when multiple sources overlap, and reason about whether a retrieved text is actually controlling for the issue at hand. LOCUS-v1 provides the text, metadata, and geographic substrate needed to build those tasks while preserving a clean separation between corpus construction and future evaluations of legal reasoning.

More broadly, LOCUS shows that freeing the law is not only a problem of access. It is a problem of representation. Local ordinances were formally public before LOCUS, but they were not available as a national object of machine reading, systematic comparison, or computational legal analysis. Once made observable at scale, local law appears neither as an undifferentiated mass of rules nor as a set of isolated municipal idiosyncrasies. It has structure: a recurring architecture of codification, a functional division between jurisdictional forms, and regionally specific institutional variation. These are precisely the kinds of structure that legal AI systems must learn to respect if they are to move from text retrieval toward reliable reasoning over public authority.

References

Nikolaos Aletras, Dimitrios Tsarapatsanis, Daniel Preoţiuc-Pietro, and Vasileios Lampos. Predicting judicial decisions of the European Court of Human Rights: A natural language processing perspective. *PeerJ Computer Science*, 2:e93, 2016. doi: 10.7717/peerj-cs.93.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.

Ilias Chalkidis, Ion Androutsopoulos, and Nikolaos Aletras. Neural legal judgment prediction in English. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 4317–4323, 2019.

Ilias Chalkidis, Abhik Jana, Dirk Hartung, Michael Bommarito, Ion Androutsopoulos, Daniel Martin Katz, and Nikolaos Aletras. LexGLUE: A benchmark dataset for legal language understanding in English. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2022.

Matthew Dahl, Varun Magesh, Mirac Suzgun, and Daniel E. Ho. Large legal fictions: Profiling legal hallucinations in large language models. *Journal of Legal Analysis*, 16(1):64–93, 2024.

Georgetown Law Library. State legal research: General and multi-jurisdictional — local government. `https://guides.ll.georgetown.edu/statelegalresearch/localgovernment`, 2026. Last updated February 27, 2026; accessed May 5, 2026.

Neel Guha, Julian Nyarko, Daniel E. Ho, Christopher Ré, Adam Chilton, Aditya Narayana, Alex Chohlas-Wood, Austin Peters, Brandon Waldon, Daniel N. Rockmore, et al. Legalbench: A collaboratively built benchmark for measuring legal reasoning in large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Harvard Law School Library Innovation Lab. Caselaw Access Project. `https://case.law/`, 2018.

Peter Henderson, Mark S. Krass, Lucia Zheng, Neel Guha, Christopher D. Manning, Dan Jurafsky, and Daniel E. Ho. Pile of law: Learning responsible data filtering from the law and a 256GB open-source legal dataset. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Dan Hendrycks, Collin Burns, Anya Chen, and Spencer Ball. CUAD: An expert-annotated NLP dataset for legal contract review. In *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, 2021.

Ralf Herbrich, Tom Minka, and Thore Graepel. Trueskill™: a bayesian skill rating system. *Advances in neural information processing systems*, 19, 2006.

Oliver Wendell Holmes, Jr. The path of the law. *Harvard Law Review*, 10(8):457–478, March 1897.

Nils Holzenberger, Andrew Blair-Stanek, and Benjamin Van Durme. A dataset for statutory reasoning in tax law entailment and question answering. In *Proceedings of the Natural Legal Language Processing Workshop*, 2020.

Alistair E. W. Johnson, Tom J. Pollard, Lu Shen, Li-wei H. Lehman, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G. Mark. Mimic-iii, a freely accessible critical care database. *Scientific Data*, 3(1):160035, 2016. doi: 10.1038/sdata.2016.35.

Alistair E. W. Johnson, Lucas Bulgarelli, Lu Shen, Alvin Gow, Tom Pollard, Steven Horng, Leo Anthony Celi, and Roger Mark. Mimic-iv, a freely accessible electronic health record dataset. *Scientific Data*, 10(1):1, 2023. doi: 10.1038/s41597-022-01899-x.

Yuta Koreeda and Christopher D. Manning. ContractNLI: A dataset for document-level natural language inference for contracts. In *Findings of the Association for Computational Linguistics: EMNLP*, 2021.

Yinhong Liu, Han Zhou, Zhijiang Guo, Ehsan Shareghi, Ivan Vulić, Anna Korhonen, and Nigel Collier. Aligning with human judgement: The role of pairwise preference in large language model evaluators. In *Conference on Language Modeling (COLM)*, 2024.

Michael A. Livermore, Allen B. Riddell, and Daniel N. Rockmore. The supreme court and the judicial genre. *Arizona Law Review*, 59:837–901, 2017.

OpenAI. Introducing GPT-5.4. `https://openai.com/index/introducing-gpt-5-4/`, March 2026. Accessed: 2026-05-07.

Jake Poznanski, Luca Soldaini, and Kyle Lo. olmocr 2: Unit test rewards for document ocr. *arXiv preprint arXiv:2510.19817*, 2025.

Ralf Steinberger, Bruno Pouliquen, Anna Widiger, Camelia Ignat, Tomaž Erjavec, Dan Tufiş, and Dániel Varga. The JRC-Acquis: A multilingual aligned parallel corpus with 20+ languages. In *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC)*, 2006.

Supreme Court of the United States. Georgia v. public.resource.org, inc. Slip Opinion No. 18-1150, April 2020. URL `https://www.supremecourt.gov/opinions/19pdf/18-1150_7m58.pdf`. 590 U.S. 255, 140 S. Ct. 1498, 206 L. Ed. 2d 732.

Said Taghadouini, Adrien Cavaillès, and Baptiste Aubertin. Lightonocr: A 1b end-to-end multilingual vision-language model for state-of-the-art ocr. *arXiv preprint arXiv:2601.14251*, 2026.

Don Tuggener, Pius von Däniken, Thomas Peetz, and Mark Cieliebak. LEDGAR: A large-scale multi-label corpus for text classification of legal provisions in contracts. In *Proceedings of the 12th Language Resources and Evaluation Conference (LREC)*, 2020.

Benjamin Warner, Antoine Chaffin, Benjamin Clavié, Orion Weller, Oskar Hallström, Said Taghadouini, Alexis Gallagher, Raja Biswas, Faisal Ladhak, Tom Aarsen, et al. Smarter, better, faster, longer: A modern bidirectional encoder for fast, memory efficient, and long context finetuning and inference. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2526–2547, 2025.

Joe Weisenthal. Havelock ai. `https://havelock.ai`, 2026. Accessed: 2026-05-06.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Lucia Zheng, Neel Guha, Brandon R. Anderson, Peter Henderson, and Daniel E. Ho. When does pretraining help? assessing self-supervised learning for law and the CaseHOLD dataset of 53,000+ legal holdings. In *Proceedings of the 18th International Conference on Artificial Intelligence and Law (ICAIL)*, 2021.

A Scoring Prompts

We elicit pairwise judgments from GPT-5.4-nano using a single shared template, parameterized by a rubric for each axis.

Pairwise Comparison System Prompt

```
You are evaluating local laws along the dimension of "{{ axis }}"
{% include axis + ".tpl" %}
Read the following two laws and determine which has a GREATER degree of {{ axis
    }} according to the rubric above.
Respond with the winner and a one sentence explanation of why it is the winner.
-----
Law A
-----
{{ law_a["header"] }}
{{ text_a }}
-----
Law B
-----
{{ law_b["header"] }}
{{ text_b }}
-----
Respond in the following JSON format only:
‘‘‘json
```

```
{
    "winner": "A" or "B" or "Tie"
    "reasoning": "one sentence explanation"
}
```

Axis Rubric: Problem Salience

```
# Problem Salience
Problem salience measures how strongly the law represents the regulated issue
    as important, urgent, or threatening.
A high-salience law uses charged framing (crisis, epidemic, threat),
    findings/preambles emphasizing severity, or heightened penalties signaling
    gravity.
A low-salience law treats the issue as routine, technical, or administrative
    without rhetorical emphasis on stakes.
```

Axis Rubric: Paternalism vs. Externality Orientation

```
# Paternalism vs. Externality Orientation
Paternalism vs. externality orientation measures whether the law is primarily
    protecting the regulated actor from themself or protecting others/the
    public from the actor's conduct.
A highly paternalistic law targets self-regarding behavior (harms or risks
    borne mainly by the actor).
A law oriented toward externalities targets conduct whose harms fall on third
    parties or the public at large.
```

Axis Rubric: Opacity / Intelligibility

```
# Opacity / Intelligibility
Opacity / intelligibility measures how hard it is for an ordinary person to
    know what the law requires of them. A highly opaque law relies on dense
    cross-references, technical jargon, undefined terms, or convoluted
    structure that obscure the obligations.
A low-opacity (highly intelligible) law states its requirements in plain,
    self-contained language a layperson can readily understand.
```

Axis Rubric: Enforcement Discretion

```
# Enforcement Discretion
Enforcement discretion measures the degree to which a citizen's exposure to
    enforcement under a law depends on official choice rather than on the
    citizen's own conduct.
It is high when two factors compound: (1) breadth of exposure -- the pool of
    citizens potentially subject to the law is large because its triggering
    conditions are vague, evaluative, or so commonly met that many qualify as
    eligible targets; and (2) textual latitude -- the statute's language gives
    officials wide freedom over whether, when, against whom, and how to act.
A law that exposes a vast pool but mandates uniform enforcement leaves
    officials little real choice; a law that grants officials sweeping
    latitude but exposes no citizens to enforcement at all -- e.g., provisions
    concerning only internal government structure, personnel, contracting, or
    interagency procedure -- creates no opportunity for capricious wielding.
The score floor is reserved for laws1 that do not act on private parties; the
    score ceiling for laws under which many citizens stand exposed and
    officials choose freely among them.
```

B Annotation Prompt

We prompt gpt-5.4-nano for an initial zero-shot classification, and review anything evaluated flagged annotations (5.5%) with a second pass of gpt-5.4.

Annotation Prompt

```
SYSTEM_PROMPT = (
    "You are a legal text classifier specializing in municipal and county "
    "codes. Return only a JSON object that matches the provided schema."
)

REVIEW_SYSTEM_PROMPT = (
    "You are a senior legal QA reviewer. Review the first-pass classification "
    "carefully, correct it when needed, and return only a JSON object that "
    "matches the provided schema. Treat Process as the label for operative "
    "administrative procedures, delegated authority, internal governance rules,
    "
    "meeting/quorum/voting rules, board composition, appointments, elections, "
    "hearings, notice requirements, and permitting workflows. Use Structural "
    "only for non-operative artifacts or formatting noise such as headers, "
    "tables of contents, HTML fragments, history/source notes, page markers, "
    "cross-reference lists, and similar text that does not itself state an "
    "operative rule or procedure."
)

USER_INSTRUCTIONS = """
Task: Classify the provided text by its primary legal function.

Allowed primary_function values:
- Context: defines terms, scope, or introductory intent.
- Rules: imposes permissions, obligations, or prohibitions.
- Process: describes administrative procedure, authority, or government
    structure.
- Enforcement: specifies penalties, violations, appeals, or exceptions.
- Structural: non-substantive artifacts such as section headers, tables of
  contents, HTML, history/source notes, page numbers, or formatting remnants.

Output rules:
- is_substantive must be 1 only for Rules or Enforcement. Otherwise use 0.
- primary_function must be exactly one of: Context, Rules, Process,
  Enforcement, Structural.
- sub_category must be null when is_substantive is 0.
- When is_substantive is 1, sub_category must be exactly one of:
  Land use, Noise/Nuisance, Housing, Business licensing, Public space,
  Building/Safety, Other.
- logic must be one short sentence.
""".strip()

REVIEW_INSTRUCTIONS = """
Task: Review a first-pass legal text classification and produce the final
corrected classification.

Rules:
- If the first-pass classification is correct, keep it and set review_outcome to
  "confirm".
- If the first-pass classification is incorrect, correct it and set
  review_outcome to "override".
- If the first-pass classification is missing or invalid, classify from scratch
  and set review_outcome to "fresh".
- Keep the same classification rules and category set as the first pass.
- Treat operative internal governance text as Process, not Structural. This
  includes board composition, quorum, voting, meeting procedures, delegation of
```

```
  authority, appointment/removal rules, election administration, hearings,
  notice, application steps, and similar administrative workflows.
- Use Structural only for non-operative artifacts/noise such as section headers,
  article labels, tables of contents, page markers, history/source notes, HTML,
  formatting remnants, or cross-reference lists that do not themselves contain
  operative requirements.
- Derive is_substantive from primary_function: use 1 only for Rules or
  Enforcement; use 0 for Context, Process, and Structural.
- review_logic must briefly explain why you confirmed, changed, or freshly
  classified the text.
""".strip()

CLASSIFICATION_SCHEMA = {
    "type": "object",
    "additionalProperties": False,
    "properties": {
        "is_substantive": {"type": "integer", "enum": [0, 1]},
        "primary_function": {
            "type": "string",
            "enum": [
                "Context",
                "Rules",
                "Process",
                "Enforcement",
                "Structural",
            ],
        },
        "sub_category": {
            "anyOf": [
                {
                    "type": "string",
                    "enum": [
                        "Land use",
                        "Noise/Nuisance",
                        "Housing",
                        "Business licensing",
                        "Public space",
                        "Building/Safety",
                        "Other",
                    ],
                },
                {"type": "null"},
            ]
        },
        "logic": {"type": "string"},
    },
    "required": [
        "is_substantive",
        "primary_function",
        "sub_category",
        "logic",
    ],
}
```

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We introduce our dataset and the dimensions of evaluation that it enables in our abstract. Section 3 describes our dataset and Section 5 describes our evaluation.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The Discussion session addresses limitations in two paragraphs (n-1 and n-2 in the section).

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [N/A]

Justification: No theoretical results.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We release our code on Github, our data on Hugging Face, and discuss our pipeline in Figure 4. Furthermore, all our prompts and model choices are provided in the appendix.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We release our dataset on Hugging Face and we release our code on Github for maximal reproducibility and transparency.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

Answer: [Yes]

Justification: Section 3 discusses our dataset. Our evaluation code is provided on Github and prompting decisions are meticulously documented in the appendix.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [N/A]

Justification: Our paper primarily details our dataset and its uses. Our Hugging Face datacard provides summary statistics of our columns.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Section 4.2 describes our annotation set-up. Section 4.3 describes our OCR pipeline.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We mitigate the harms of our research in so far as data is concerned. We do not have human test subjects. We maximize the transparency of our data in the interest of positive societal impact.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We discuss both our desired positive impacts and the challenges of working with data and creating an accurate harmonization in Section 6.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.

- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [Yes]

    Justification: We license the data under CC by NC 4.0 to minimize risk. Furthermore, we withhold a large set of laws for researchers to mimic the best practices of MIMIC [Johnson et al., 2023].

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All vendors allow data downloading under terms of use, and we follow any robots.txt guidance. Furthermore, fair use supports the collection of this data. We license the data under CC by NC 4.0 and release a fraction of the data as an access layer. Additionally, we cite the models that we use for annotation.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.

- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: Section 3 describes our dataset and our Hugging Face release provides full transparency of the LOCUS harmonization layer.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: We do not do research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: We do not work with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [Yes]

    Justification: We use LLMs as core methodology for our annotation (Section 4.2) and evaluation (Section 5.1).

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.