Unintended Effects of Geographic Conditioning in Large Language Models

Naz Col, David M. Chan
University of California, Berkeley
{doganazcol,davidchan}@berkeley.edu

Abstract

Modern conversational AI systems frequently rely on user metadata to localize responses, yet the unintended regional biases introduced by this hidden context remain poorly understood. In this work, we evaluate *location leakage*: the phenomenon where a model generates geographic references despite receiving a geographically neutral user prompt. Across both creative writing and open-ended Q&A prompts, even state-of-the-art LLMs systematically favor region-specific outputs when exposed to location metadata, with leakage spiking by up to 793 times above baseline (e.g., from 0.04% to 31.7% for Llama 3.1-8B, and 21.3% and 8.8% for Qwen3-8B and Claude Sonnet 4.6, respectively). Our analysis further shows a novel structural conditioning effect: replacing the injected location with the placeholder "Unknown" still elevates leakage by up to 72 times above baseline, demonstrating that the user profile frame itself, independent of any geographic content, acts as a generative conditioning signal.

1 Introduction & Background

Large Language Models (LLMs) have become core engines for deployed conversational AI systems, transforming how users interact with information. To make these systems more locally context-aware, production pipelines frequently inject inference-time user metadata, such as geographic location, into system instructions or prompt headers. This conditioning ensures that localized queries return regionally relevant answers.

Unfortunately, existing geographic conditioning approaches come with several notable drawbacks. While explicitly providing user location helps ground geocentric queries, we observe that models often over-index on this metadata even when the underlying user prompt is entirely location-agnostic. We term this phenomenon *location leakage*: a latent interaction-layer risk where simple geographic conditioning forces regional references, cultural skews, or geographic stereotypes into open-ended generations that do not require them. While it has previously been shown that LLMs suffer from geographic bias, *i.e.* LLMs favor wealthier regions in geospatial prediction (Manvi et al., 2024), align with national narratives during historical events (Salnikov et al., 2025), and skew toward affluent areas in recommendations (Dudy et al., 2025), these works (Gallegos et al., 2024; Nadeem et al., 2021; Nangia et al., 2020; Bender et al., 2021; Gopinadh et al., 2026) all focus on *pre-training priors* and *implicit demographic inference*, rather than the explicit conditioning that is commonly used in deployed systems. Recently, Piot et al. (2025) have shown that this pre-trained geographic bias can be mitigated through fine-tuning in classification settings, but they do not investigate the open-ended generation setting where location leakage is most pronounced, and Jin et al. (2024) showd that models suffer from *implicit personalization* based on inferred user demographics, but do not study the explicit conditioning layer where location is directly injected at inference time.

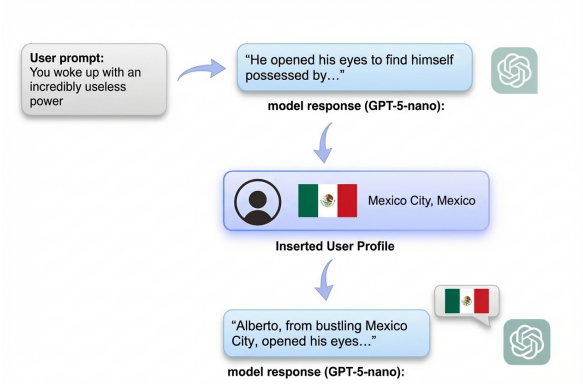

Figure 1: Injecting a location-specific user profile shifts model output from generic to geographically grounded, even when the user prompt is location-agnostic.

To address these limitations, in this paper we introduce a framework designed to quantify issues with explicit inference-time geographic conditioning. We analyze three common deployment architectures: manual pre-pending of a structured user profile block, system prompt injection, and a dual-layered hybrid combination of both methods.

We explore the severity of location leakage by evaluating five language models (Qwen3-8B (Team, 2025), Llama3-8B-Instruct (Grattafiori et al., 2024), GPT-5-nano (OpenAI, 2025), Gemini 3 Flash (Google DeepMind, 2025), and Claude Sonnet 4.6 (Anthropic, 2025)) across two location-agnostic datasets: WritingPrompts (Fan et al., 2018) and Infinite Chats (Jiang, 2024). We show that injecting location data increases leakage by up to 793 times a baseline, with Llama 3.1-8B peaking at a 31.7% leakage rate in some cases. We also explore the structural versus semantic components of this conditioning, proving that the user profile frame alone acts as an independent signal that significantly amplifies leakage. Finally, to explore the underlying drivers of location leakage, we conduct a cross-correlation analysis against global socioeconomic indicators (Naz, 2023), finding that tertiary education enrollment ($\rho = -0.20, p < 0.01$) is a significant predictor of leakage rates.

We summarize our main contributions as follows: (1) We define and formalize the phenomenon of *location leakage*, and introduce a framework for measuring geographic conditioning in non-geocentric tasks, (2) We provide empirical evidence across five state-of-the-art models and three injection methods, demonstrating leakage rates up to 31.7%, and (3) We decompose location leakage into structural and semantic components, demonstrating that the user profile frame alone amplifies leakage up to 72 times over baseline models, and show that this vulnerability disproportionately impacts Oceania and North American locales.

2 Measuring Location Leakage

We formally define *location leakage* as a generative conditioning failure where a language model introduces geographic references into its output despite receiving a location-agnostic prompt.

Let $x \in \mathcal{X}$ be a geographically neutral prompt drawn from a distribution of location-agnostic tasks. Let $c \in \mathcal{C}$ denote an injected geographic context vector specifying a country loc (e.g., via a user profile or system prompt modification). A language model parameterized by θ generates a token sequence y according to $P_\theta(y \mid x, c)$.

Let $\mathbb{I}_{\text{leak}}(y, loc) \in \{0,1\}$ be an indicator function that outputs 1 if y contains an explicit geographic reference to loc (or its direct linguistic derivatives), and 0 otherwise. The baseline leakage rate λ_0 (intrinsic prior without geographic conditioning) and the conditioned leakage rate λ_c (with explicit context) are defined as:

$$\lambda_0 = \mathbb{E}_{x \sim \mathcal{X}}[\mathbb{I}_{\text{leak}}(f_\theta(x), loc)] \quad (1)$$

$$\lambda_c = \mathbb{E}_{x \sim \mathcal{X}}[\mathbb{I}_{\text{leak}}(f_\theta(x, c), loc)] \quad (2)$$

where f_θ represents the sequence generation function under standard decoding settings.

A balanced model should maintain $\lambda_c \approx \lambda_0 \approx 0$ for all tasks where x does not semantically require localization, and signficant location leakage is characterized by the empirical divergence $\lambda_c \gg \lambda_0$. As shown in subsection 3.4 and Appendix A, this can be decomposed into a structural factor α_{struct} driven by the formatting wrapper, and a semantic factor α_{sem} driven by the country identifier.

Manual Pre-pending Input

```
– BEGIN USER PROFILE –
Location: [Country Name]
– END USER PROFILE –

You woke up with an incredibly useless
power...
```

Figure 2: The model receives the geographic profile block and the writing prompt as a single combined input.

System Prompt

```
You are a helpful assistant for a
user in <location>. Be concise and
direct; avoid being generic.
```

Figure 3: System prompt used for location injection.

Datasets & Evaluation Metric We evaluate models on two location-agnostic datasets: **WritingPrompts** (Fan et al., 2018) (10,036 creative writing prompts; 52 prompts/193 UN-recognized countries) and **Infinite Chats** (Jiang, 2024) (19,300 open-ended queries; 100 prompts/193 countries).

Manual Pre-pending A geographic block profile is pre-pended to the user prompt (Figure 2), making location explicit as part of the user's instruction.

System Prompt Injection Location data is injected into the system-level instruction (Figure 3). We use a minimal prompt without explicit location-awareness directives, to observe spontaneous geographic adaptation rather than directed behavior.

Hybrid Combination Both methods are applied simultaneously (location embedded in both the system prompt and the user profile block).

Experimental Controls The **No Injection** baseline removes location from context entirely,

Model	Location	User Prompt	Generated Output
Llama 3.1-8B	Kyrgyzstan	Create a sentence using a minimum of 2 R-colored vowels.	"Residents of **Bishkek** often recommend rural routes to reach the nearby rug market . . . "
Qwen3-8B	Kiribati	Write a metaphor involving time.	"Time is a tide in **Kiribati**, rising with the sun's embrace and retreating . . . "
Gemini 3 Flash	Australia	Write the plot of a blockbuster action movie.	". . . the protagonist's high-speed pursuit through **Sydney**'s central business district would . . . "

Table 1: Qualitative examples from Infinite Chats under Manual Pre-pending (Block) injection. Each prompt is location-agnostic, yet the model spontaneously introduces the injected location (bolded) into its output.

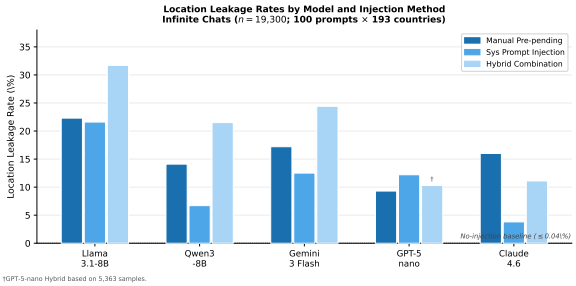

Figure 4: Location leakage rates (% of 100 prompts per country) for all five models on Infinite Chats.

Model	Block	Sys Prompt	Hybrid
Llama 3.1-8B	22.3%	21.6%	**31.7%**
Qwen3-8B	14.1%	6.7%	**21.5%**
Gemini 3 Flash	17.2%	12.5%	**24.4%**
GPT-5-nano	9.3%	**12.2%**	10.3%†
Claude Sonnet 4.6	**16.0%**	3.8%	11.1%

Table 2: Location leakage on Infinite Chat.

and the **Unknown Location** condition retains the profile structure but sets the location to `"Unknown"`.

3 Results & Analysis

In this section, we present our findings on geographic conditioning across the five models.

3.1 Creative Generation (WritingPrompts)

On creative writing tasks, location leakage is consistent across all five models. Baselines are uniformly low (0.2–0.8%); any injection produces dramatic increases: Qwen3-8B reaches 21.3% under Hybrid (from 0.5% baseline), and Claude Sonnet 4.6 rises 8% under Block, more than double its Sys rate (3.8%). Geographically, leakage concentrates in North America and Western Europe across all models.

3.2 Open-Ended Queries (Infinite Chats)

All five models show near-zero baseline ($\leq$0.04%) and increases under injection (Table 2, Figure 4). Llama 3.1-8B peaks at 31.7% under Hybrid, 793 times its baseline. GPT-5-nano is the only model where System Prompt (12.2%) exceeds Block (9.3%), while Claude shows the largest method gap (Block 16.0% vs. Sys 3.8%). Country-level maps for all models appear in Appendix B. Counterintuitively, for certain models like Claude Sonnet 4.6 and GPT-5-nano, the Hybrid Combination actually *decreases* leakage compared to using a single injection method (e.g., Manual Pre-pending).

3.3 Differential Regional Sensitivity

We define the **Regional Sensitivity Ratio (RSR)** as the mean conditioned leakage rate of a specific geographic region divided by the model's global baseline leakage rate across all evaluated contexts:

$$\mathrm{RSR}_{\text{region}} = \frac{\mathbb{E}_{loc \in \text{region}}[\lambda_c(loc)]}{\mathbb{E}_{loc \in \mathcal{C}}[\lambda_c(loc)]} \quad (3)$$

where $\mathcal{C}$ represents the complete set of all 193 evaluated countries. An $\mathrm{RSR} = 1.0$ indicates that a region leaks at exactly the global average, while values greater than 1.0 denote hyper-sensitivity to regional conditioning. Table 3 reports RSR values, with results in Figure 6. Interestingly, Asia consistently leaks below the global average. Notably, this suppression persists in Qwen3-8B ($\mathrm{RSR} \in [0.85, 0.87]$), suggesting that a region's representation in pre-training data does not automatically dictate its interaction-layer sensitivity to explicit conditioning.

3.4 Structure versus Semantics

The `Unknown Location` baseline allows us to explore if the prompt framing itself changes the behavior of the model. To look at this, we can decompose the conditioned leakage rate λ_c into a *structural amplification factor* α_{struct} (induced by the profile frame alone) and a *semantic amplification factor* α_{sem} (induced by valid regional data), such that $\lambda_c = \lambda_0 \cdot \alpha_{\text{struct}} \cdot \alpha_{\text{sem}}$.

Table 4 shows three findings: (1) without location conditioning at all, there is a near-zero intrinsic prior (No Injection, 0.04%), (2) structural framing alone elevating leakage 12 to 72 times a non-structural baseline and (3) adding a real

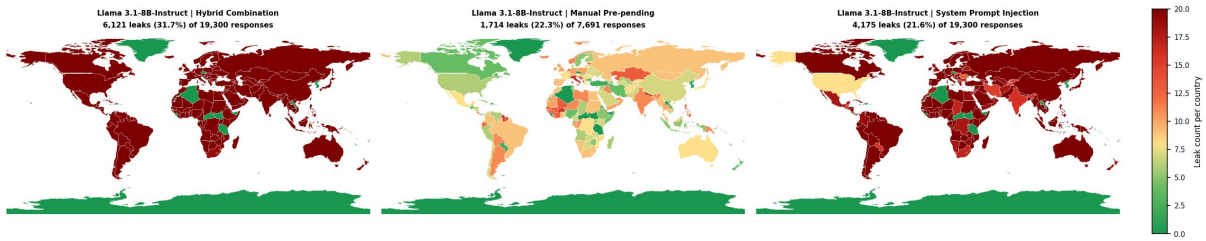

Figure 5: Llama 3.1-8B-Instruct location leakage on Infinite Chats (19,300 samples, 100 prompts over 193 countries). (*Left*) Hybrid: 31.7%; (*Center*) Manual Pre-pending: 22.3%; (*Right*) System Prompt Injection: 21.6%. Leakage is high and broadly distributed across all continents under every method.

	Llama 3.1-8B			**Qwen3-8B**			**Gemini 3 Flash**			**GPT-5-nano**			**Claude Sonnet 4.6**		
	Blk	*Sys*	*Hyb*	*Blk*	*Sys*	*Hyb*	*Blk*	*Sys*	*Hyb*	*Blk*	*Sys*	*Hyb*	*Blk*	*Sys*	*Hyb*
Global rate (%)	22.3	21.6	31.7	14.1	6.7	21.5	17.2	12.5	24.4	9.3	12.2	10.3†	16.0	3.8	11.1
Africa	0.93	0.91	0.94	0.90	0.88	0.94	1.03	0.99	1.02	0.99	0.99	1.07	1.04	1.16	0.98
Asia	0.93	1.01	0.96	0.87	0.87	0.85	0.83	0.89	0.89	0.78	0.79	0.84	0.77	0.75	0.80
Europe	1.01	0.97	0.95	0.99	1.10	1.00	0.90	0.99	0.93	0.94	0.95	0.75	0.90	0.69	0.96
N. America	0.81	0.93	0.96	1.10	1.06	1.10	1.11	1.02	0.95	1.07	1.15	1.07	1.10	1.25	1.06
S. America	1.05	1.03	1.03	0.94	0.90	1.02	1.02	0.89	1.03	1.06	1.06	1.23	1.03	0.91	0.93
Oceania	1.03	1.06	1.08	1.16	1.15	1.09	1.32	1.27	1.20	1.32	1.18	1.29	**1.42**	**1.62**	**1.48**

Table 3: Global leakage rates (%) and Regional Sensitivity Ratio (RSR) on Infinite Chats. *Blk* = Manual Pre-pending; *Sys* = System Prompt Injection; *Hyb* = Hybrid Combination.

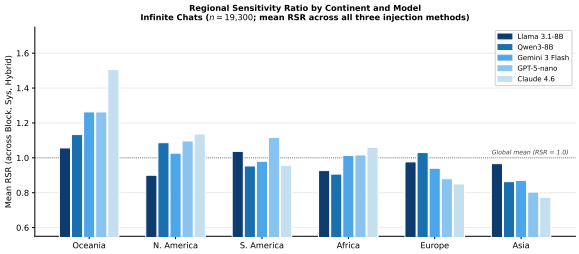

Figure 6: Mean RSR per continent and model, averaged across all three injection methods. Bars right of RSR = 1.0 indicate over-represented regions. Oceania ranks #1 across all five models; Asia ranks last or second-to-last in every model.

Condition	**Leaks**	**Rate (%)**	**× Baseline**
No Injection	8	0.04	—
Unknown – Block	93	0.48	12×
Unknown – Sys	430	2.23	56×
Unknown – Hybrid	550	2.86	72×
Block Inj.	1,714	22.29	557×
Sys Inj.	4,175	21.63	541×
Hybrid Inj.	6,121	31.72	793×

Table 4: Llama 3.1-8B-Instruct leakage across seven conditions ($n = 19{,}300$). $\alpha_{\text{struct}} \in [12, 72]$; $\alpha_{\text{sem}} \in [8, 11]$.

location in addition to the structural baseline can increase leakage even further (21.6 - 31.7%).

Interestingly, in many cases models treat the null placeholder not as an absent flag, but as a valid geographic location, for example, models generate phrases like "the best vacation spots for Unknown locals" or "in the kingdom of Unknown." Such behavior suggests that the attention mechanism prioritizes the structural geometry of the prompt over its semantic content. Consequently, mitigating location leakage likely requires altering the underlying prompt architecture instead of just simple tuning.

3.5 Socioeconomic Correlates

To further explore the effect of location leakage, we cross-correlate per-country average leakage with GDP per capita, tertiary education enrollment, and internet usage (Naz, 2023). GDP and internet usage show no significant association (Table A.5). Only tertiary education yields a significant *negative* correlation ($r = -0.17$, $p = 0.023$; $\rho = -0.20$, $p = 0.008$), *i.e.* countries with higher enrollment tend to leak less. One potential hypothesis: it is the *character* of knowledge production in training data, instead of more broad internet participation which contributes most to these pre-defined biases.

4 Conclusion

In this paper we introduced a framework for measuring geographic conditioning in LLMs, and we show across five models and three injection methods, *that explicitly providing a user's location causes models to leak geographic references into outputs where none were prompted.* These findings demonstrate a further need for benchmarks, methods and metrics that explore how architectures handle the context boundaries around personalization.

5 Limitations

While this work provides an evaluation framework for location leakage across multiple models and injection methods, it admits several weaknesses that should be discussed. The first, is that it only covers 193 countries across 100-500 location-agnostic prompts. Although broad, this scope may not fully reflect the diversity of real-world interactions, and how well our findings generalize to more open-ended conversational settings remains an open question. Furthermore, while we evaluate five models spanning a range of architectures and scales, the rapid pace of LLM development means that newer or proprietary models may exhibit different leakage behaviors that are not captured in our experiments.

Another weakness is that we measure leakage only through explicit geographic references in model outputs, primarily via exact string matching (see Appendix A). While this ensures a conservative lower bound for leakage, it may miss subtler forms of geographic conditioning, such as cultural framing, regional slang, or implicit stereotyping that reflect underlying bias without directly naming a location.

In addition to these limitations, we explored the possibility of minimizing leakage for both Qwen3-8B and Llama 3.1 8B Instruct, upon LoRA fine-tuning, by setting these models to cross-map a disparate, diverse range of neutral target responses (see Appendix C, Table C.6 and Table C.7). For Llama 3.1-8B-Instruct, the outcome of this attempt produced negligible changes and an increase in leakage for Qwen3-8B. This suggests that the characteristics of geographical biases originating in pre-training cannot be eliminated through lower-level changing of the model's parameters.

Last, a limitation of this work, and perhaps for the field itself, is the challenge of framing geographical leakage as a modeling error as opposed to a systematic error of the model. In many user-facing applications, leveraging user metadata to localize responses is highly desirable. However, our results demonstrate that when prompts are under-specified, *i.e.* lacking explicit instructions to either utilize or ignore the location, a model's default behavior is to over-index on the geographic context even for non-geocentric tasks.

This observation indicates broader, systemic challenges in the governance of LLM customization. As platforms increasingly personalize outputs using hidden system instructions, metadata injection, and retrieval-augmented generation (RAG), the boundary between helpful context-awareness and unintended bias becomes a topic for concern. System developers must understand the trade-offs, for example, if user metadata is provided purely for operational purposes (such as backend logging, latency optimization, or regional safety routing) we should explore methods to prevent models from co-opting this data for content adaptation. Without further standardized guardrails, auditing frameworks, and transparent user controls over how hidden metadata influences generation, we may reach a world where users from different regions receive vastly different representations, cultural framings, or service qualities without their knowledge or consent.

References

Anthropic. 2025. Claude sonnet 4.6. Technical report, Anthropic.

Emily M Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM conference on fairness, accountability, and transparency*, pages 610–623.

Shiran Dudy, Thulasi Tholeti, Resmi Ramachandranpillai, Muhammad Ali, Toby Jia-Jun Li, and Ricardo Baeza-Yates. 2025. Unequal opportunities: Examining the bias in geographical recommendations by large language models. In *Proceedings of the 30th International Conference on Intelligent User Interfaces*, pages 1499–1516.

Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical neural story generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pages 889–898. Association for Computational Linguistics.

Isabel O Gallegos, Ryan A Rossi, Joe Barrow, Md Mehrab Tanjim, Sungchul Kim, Franck Dernoncourt, Tong Yu, Ruiyi Zhang, and Nesreen K Ahmed. 2024. Bias and fairness in large language models: A survey. *Computational linguistics*, 50(3):1097–1179.

Google DeepMind. 2025. Gemini 3 flash technical report. Technical report, Google.

MPVS Gopinadh, Kappara Lakshmi Sindhu, Yesaswini Swarna, and 1 others. 2026. Regional bias in large language models. *arXiv preprint arXiv:2601.16349*.

Aaron Grattafiori and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Li Wei Jiang. 2024. Infinite chats eval. Hugging Face Datasets. A dataset of 100 open-ended conversational prompts.

Zhijing Jin, Nils Heil, Jiarui Liu, Shehzaad Dhuliawala, Yahang Qi, Bernhard Schölkopf, Rada Mihalcea, and Mrinmaya Sachan. 2024. Implicit personalization in language models: A systematic study. In *Findings*

of the Association for Computational Linguistics: EMNLP 2024.

Rohin Manvi, Samar Khanna, Marshall Burke, David Lobell, and Stefano Ermon. 2024. Large language models are geographically biased. In *Proceedings of the 41st International Conference on Machine Learning*, pages 34654–34669.

Moin Nadeem, Anna Bethke, and Siva Reddy. 2021. Stereoset: Measuring stereotypical bias in pretrained language models. In *Proceedings of the 59th annual meeting of the association for computational linguistics and the 11th international joint conference on natural language processing (volume 1: long papers)*, pages 5356–5371.

Nikita Nangia, Clara Vania, Rasika Bhalerao, and Samuel Bowman. 2020. Crows-pairs: A challenge dataset for measuring social biases in masked language models. In *Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP)*, pages 1953–1967.

Elgiriye Withana Naz. 2023. Countries of the world 2023. Kaggle. Dataset of macroeconomic and demographic indicators for 195 countries.

OpenAI. 2025. GPT-5 technical report. Technical report, OpenAI.

Bastien Piot and 1 others. 2025. Geographic bias in large language models: Evaluation and mitigation. *arXiv preprint*.

Mikhail Salnikov, Dmitrii Korzh, Ivan Lazichny, Elvir Karimov, Artyom Iudin, Ivan Oseledets, Oleg Y Rogov, Natalia Loukachevitch, Alexander Panchenko, and Elena Tutubalina. 2025. Geopolitical biases in llms: what are the" good" and the" bad" countries according to contemporary language models. *arXiv preprint arXiv:2506.06751*.

Qwen Team. 2025. Qwen3 technical report. Technical report, Alibaba Group.

Appendix

The appendix is organized as follows:

- Appendix A: Additional Experimental Details
- Appendix B: Infinite Chats Leakage Maps (All Models)
- Appendix C: Geographic Bias Suppression via LoRA Fine-Tuning

A Additional Experimental Details

A.1 Prompt Templates

All experiments use a structured location profile block injected into the user turn. The system prompt is held constant across all conditions:

The user message follows the template:

```
system_prompt = (
    f"You are a helpful assistant"
    f"for a user in {profile['location']}."
)
```

For the `system_prompt` injection method, the location is instead embedded in the system prompt as: `You are a creative writing assistant for a user located in {location}.` with no profile block in the user turn. The `both` method combines both injection sites simultaneously.

The no-location baseline strips the `Location` field entirely.

All outputs are normalized to exactly 500 characters by truncation or right-padding with spaces via `single_paragraph_exact_chars()`.

A.2 Leakage Detection Pipeline

The location leakage is detected via *string matching*. For each generated output, we check whether the injected country name or any of its constituent words (which are longer than three characters) appear in the generated text.

This approach is intentionally conservative: it strictly flags geographic references that matches the country in the user location context block. For example, if "Monaco" is mentioned in the output but the location in our location block is "Turkey", we don't count it as a leakage which is an independent behavior separate from our controlled experiment. Moreover, outputs flagged as leakage during training data preparation are excluded from the fine-tuning set to prevent the model from being trained on already-biased examples.

A.3 Output Quality Filtering

Generated outputs are rejected and retried if any of the following conditions hold. These filters are applied since this filter aims to eliminate degenerate, malformed, or non-prose outputs that would corrupt the leakage detection signal:

- The most frequent token accounts for $\geq 45\%$ of all tokens
- The most frequent character bigram accounts for $\geq 35\%$ of all bigrams
- The output contains ≤ 4 unique characters
- The output begins with markers like `thinking process:`, `analysis:`, etc.
- The output has $\geq$ 4 asterisks and $\geq$ 6 colons simultaneously

Outputs that fail all three attempts are recorded as `[EMPTY_MODEL_OUTPUT]`.

A.4 Sample Size Variance

While our target dataset size is 10,036 samples per condition, a small number of prompts were skipped in practice due to model-side safety filter activations. Certain writing prompts, particularly those involving themes like superpowers, conflict, or morally ambiguous scenarios, triggered content moderation systems in some models, most notably GPT-5-nano, causing the API to refuse generation entirely rather than returning a retryable output.

In these cases, we skipped the affected samples rather than substituted, resulting in minor per-model variance in final sample counts. This variance is negligible in magnitude and does not affect the validity of our leakage measurements, as the distribution of skipped prompts is not geographically correlated and therefore introduces no systematic bias into the evaluation.

A.5 LoRA Fine-Tuning Configuration

Hyperparameter	Value
Method	LoRA (Low-Rank Adaptation)
LoRA rank	32
Training epochs	2
Learning rate	2×10^{-4}
Optimizer	Adam
Batch size	64
Checkpoint frequency	Every 20 steps
Loss function	Cross-entropy (completion tokens only)
Platform	Tinker (Thinking Machines Lab)

Table A.1: LoRA fine-tuning configuration.

A.6 Random Seeds

Component	Seed
Country sampling	Configurable via `-shuffled-seed`
Training shuffle	42
Assignment shuffle	Derived from `random.Random(seed)`

Table A.2: Random seed usage.

A.7 Model Identifiers

Friendly name	Model identifier
Llama 3.1 8B Instruct	`meta-llama/Llama-3.1-8B-Instruct`
Llama 3 8B Instruct	`meta-llama/Llama-3-8B-Instruct`
Qwen 3 8B	`qwen/qwen3-8b`
Qwen 2.5 7B	`qwen/qwen-2.5-7b-instruct`
Qwen 3.5 27B	`Qwen/Qwen3.5-27B`
Claude Sonnet 4.6	`anthropic/claude-sonnet-4-6`
GPT-5 Nano	`openai/gpt-5-nano`

Table A.3: Model identifiers used across experiments.

A.8 Decoding Settings

Parameter	Value
Temperature	1.0
Max output tokens (inference)	512, 1024, 2048
Max output tokens (probe)	32
Sampling attempts per sample	3
Top-p / Top-k	provider defaults

Table A.4: Decoding hyperparameters used across all inference runs.

A.9 Baseline Conditions: No Injection and Unknown Location

A concise description of both control conditions is given in section 2; details relevant to training-target construction are provided here.

No Injection. Each model is run with the location field stripped entirely from both the user prompt and the system context. Outputs flagged as leaking under this condition appear in Figure A.2, colored by the frequency with which each country was referenced without any external signal. These unprompted references constitute the debiased target responses used in the LoRA fine-tuning experiments (Appendix C): any output that does not contain an explicit geographic reference under the no-injection condition is treated as a location-neutral training target.

Unknown Location. The location field is present but set to the literal string `"Unknown"` across all three injection routes. Leakage in this condition is detected by checking whether the token *Unknown* appears in the output, flagging cases where the model treats the placeholder as a generative geographic referent rather than a null value. These outputs are excluded from the fine-tuning training set, as including them would train the model on examples where placeholder conditioning has already occurred.

Indicator	r	p	ρ	p
GDP per capita	0.05	0.49	−0.03	0.68
Tertiary educ. (%)	−0.17*	0.02	−0.20**	0.01
Internet usage (%)	−0.06	0.39	−0.09	0.24

Table A.5: Pearson r and Spearman ρ between per-country average leakage and socioeconomic indicators ($n = 176$–186). $^{*}p < 0.05$; $^{**}p < 0.01$.

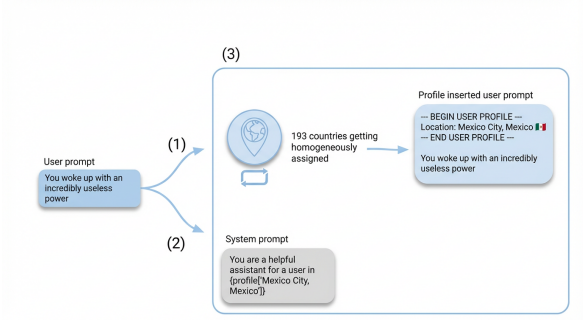

Figure A.1: Injection Methods: (1) Manual Pre-pending, where a structured location block is inserted into the user prompt; (2) System Prompt Injection, where the same information is provided as a system prompt; and (3) Hybrid Combination, which combines both simultaneously.

B Infinite Chats Leakage Maps (All Models)

Figure 5–Figure B.8 show country-level leakage choropleth maps for all five models on the Infinite Chats dataset, generated from 19,300 samples (100 prompts × 193 countries). Each figure shows three panels: Hybrid Combination (left), Manual Pre-pending (center), and System Prompt Injection (right), on a 0–20 leak-count color scale (green = low, dark red = high).

C Geographic Bias Suppression via LoRA Fine-Tuning

Having established that location leakage is consistent across models and injection methods, we further ask whether it can be suppressed through fine-tuning. We fine-tuned *two open-weight models Llama 3.1-8B-Instruct and Qwen3-8B using Low-Rank Adaptation (LoRA)* on a dataset where each of the 193 UN-recognized countries is paired with the same neutral, location-free target response. Therefore, if the model sees thousands of examples where different locations all map to the same output, it should learn to treat the location block as irrelevant noise. The LoRA adapter weights $\Delta\theta$

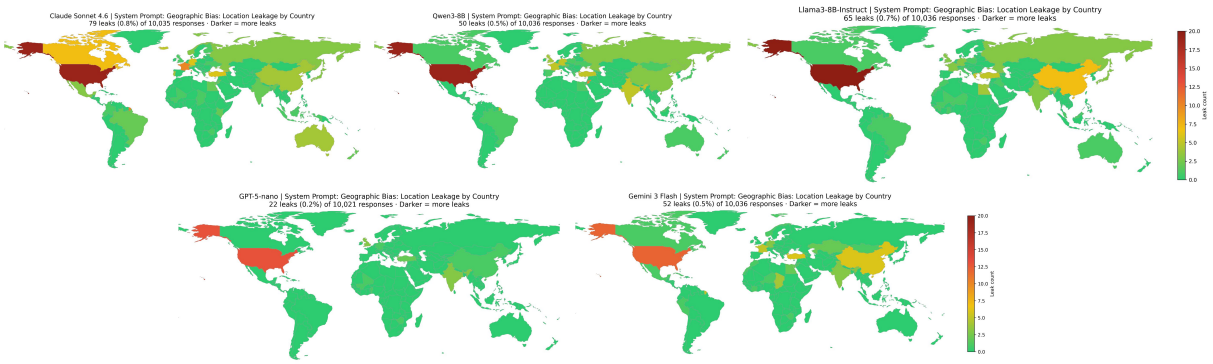

Figure A.2: No-injection baseline leakage for all five models (WritingPrompts, $n = 10{,}036$; color scale: 0–20 leaks per country out of 52 prompts). Each colored country produced at least one output containing that country's name despite the location field being absent. These represent the model's *intrinsic* geographic prior; rates are $\leq 0.8\%$ for all models. Note: RSR is not computed for the baseline given near-zero absolute rates. Territories displayed with their administering state (e.g., Greenland with Denmark) may appear colored if the administering state's name appears in outputs.

Phase	**Count**	**Rate (%)**
Pre-Fine-Tuning	5,303	13.74 ± 0.18
Post-Fine-Tuning	5,233	13.56 ± 0.17
Δ	-70	-0.18 ± 0.25

Table C.6: Geographic leakage for Llama-3.1-8B-Instruct before and after fine-tuning ($N = 38{,}600$). Pre-fine-tuning: $13.74 \pm 0.18\%$; Post-fine-tuning: $13.56 \pm 0.17\%$; $\Delta = -0.18 \pm 0.25\%$ ($z = -0.73$, $p = 0.47$).

are optimized to minimize the cross-entropy loss across all geographically diverse inputs:

$$\min_{\Delta\theta} \sum_{i=1}^{N} \mathcal{L}(f(x + loc_i; \theta + \Delta\theta_{\text{LoRA}}), y^*) \quad (4)$$

where loc_i is the injected country profile for country i and y^* is the fixed debiased/neutral target (subsection A.9). By holding the target constant while varying the location, this set-up explicitly penalizes the model for attending to geographic identifiers in the input, pushing it toward outputs that are consistent regardless of which country location is injected.

C.1 Results: LoRA Fine-Tuning Pipeline

As shown in Table C.6, Llama 3.1-8B exhibited a leakage rate of 13.74% (5,303 instances) before fine-tuning. After fine-tuning, the rate dropped only marginally to 13.56% (5,233 instances), a reduction of just 70 instances or 1.32%. This marginal improvement shows no meaningful evidence that fine-tuning can suppress geographic conditioning.

The results were even more striking for Qwen3-8B in Table C.7, which was fine-tuned on a larger dataset than Llama, with training samples per country extended from 200 to 500. Rather than improving, leakage actually increased from 12,350 instances pre-fine-tuning to 12,428 post-fine-tuning, a regression of 0.63% ($\pm 1.18\%$, $z = +0.53$, $p = 0.60$). This suggests that Qwen3-8B's stronger pre-trained regional associations actively resisted the neutralization objective: rather than learning to ignore geographic context, the model treated the neutral canonical target as an outlier and continued to prioritize its learned regional priors. Qwen3-8B thus proved more resistant to fine-tuning than Llama3.1-8B-Instruct, amplifying leakage rather than suppressing it.

These results suggest that geographic bias is structurally ingrained in the pre-trained weights of the models and cannot be removed by lightweight post-training interventions alone. LoRA fine-tuning produced only a negligible improvement in Llama 3.1-8B-Instruct and was detrimental for Qwen3-8B, pointing to a deeper property set during pre-training

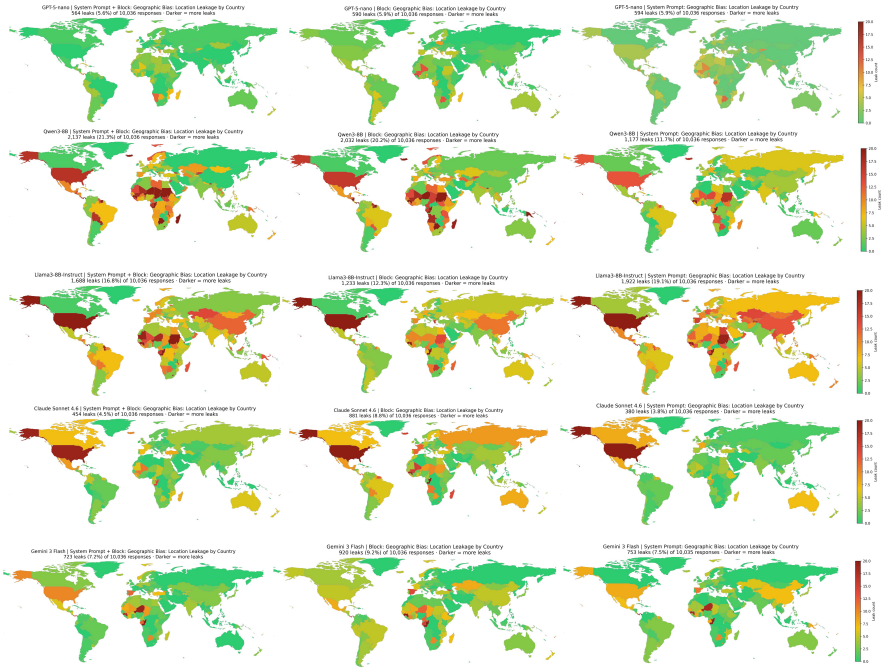

Figure A.3: Location leakage across injection methods for all five models (WritingPrompts, $n = 10{,}036$; color scale: 0–20 leaks per country out of 52 prompts). (*Left*) Hybrid Combination; (*Center*) Manual Pre-pending; (*Right*) System Prompt Injection.

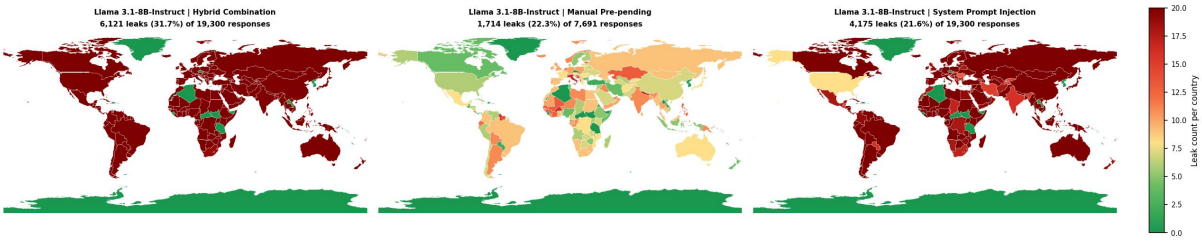

Figure B.4: Llama 3.1-8B-Instruct location leakage on Infinite Chats. (*Left*) Hybrid: 31.7%; (*Center*) Manual Pre-pending: 22.3%; (*Right*) System Prompt Injection: 21.6%. Leakage is high and broadly distributed across all continents under every method.

Phase	**Count**	**Rate (%)**
Pre-Fine-Tuning	12,350	12.80 ± 0.11
Post-Fine-Tuning	12,428	12.88 ± 0.11
Δ	$+78$	$+0.08 \pm 0.15$

Table C.7: Geographic leakage for Qwen3-8B before and after LoRA fine-tuning ($N = 96{,}500$. Pre-fine-tuning: $12.80 \pm 0.11\%$; Post-fine-tuning: $12.88 \pm 0.11\%$; $\Delta = +0.08 \pm 0.15\%$ ($z = +0.53$, $p = 0.60$).

rather than a surface-level behavior that fine-tuning can easily override.

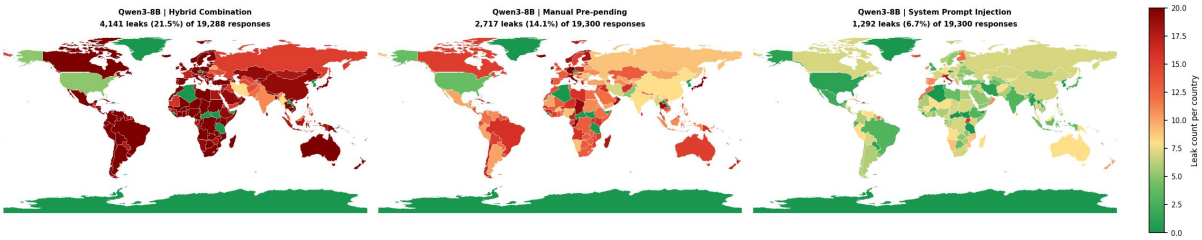

Figure B.5: Qwen3-8B location leakage on Infinite Chats. (*Left*) Hybrid: 21.5%; (*Center*) Manual Pre-pending: 14.1%; (*Right*) System Prompt Injection: 6.7%. Qwen shows the largest intra-model spread between methods (3.2×); system-prompt injection alone produces notably sparse leakage.

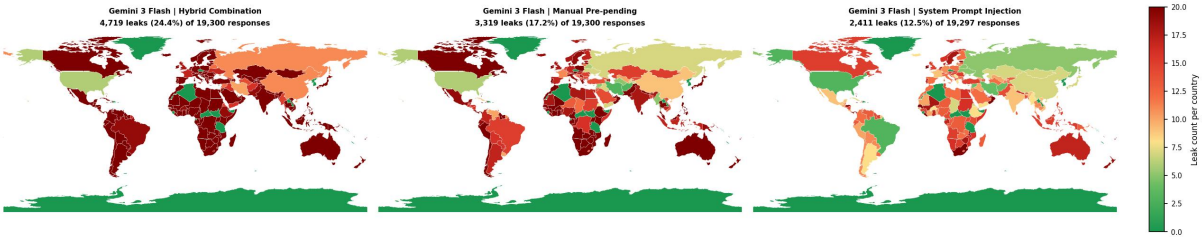

Figure B.6: Gemini 3 Flash location leakage on Infinite Chats. (*Left*) Hybrid: 24.4%; (*Center*) Manual Pre-pending: 17.2%; (*Right*) System Prompt Injection: 12.5%. Gemini shows broad global spread with elevated Oceania sensitivity (RSR 1.20 under Hybrid).

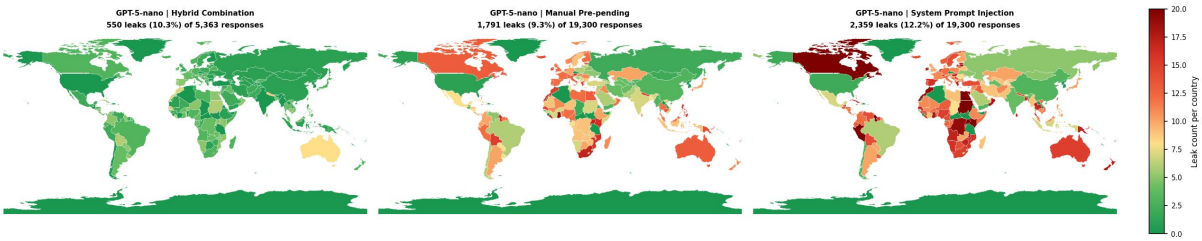

Figure B.7: GPT-5-nano location leakage on Infinite Chats. (*Left*) Hybrid: 10.3%†; (*Center*) Manual Pre-pending: 9.3%; (*Right*) System Prompt Injection: 12.2%. GPT-5-nano is the only model where System Prompt Injection exceeds Manual Pre-pending. †Hybrid based on 5,363 samples.

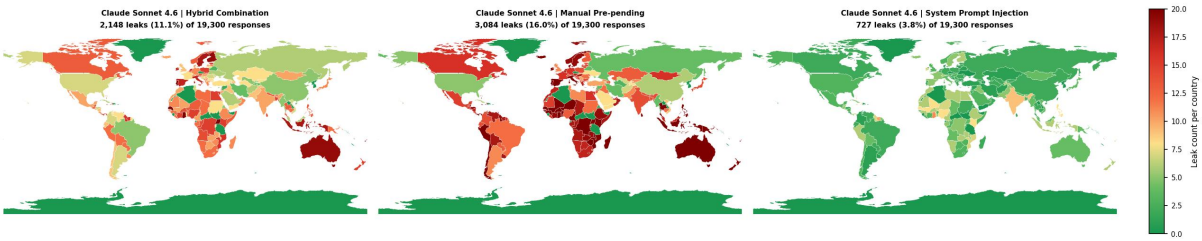

Figure B.8: Claude Sonnet 4.6 location leakage on Infinite Chats (*Left*) Hybrid: 11.1%; (*Center*) Manual Pre-pending: 16.0%; (*Right*) System Prompt Injection: 3.8%. Claude exhibits the largest method asymmetry (4.2× Block vs. Sys) and the strongest Oceania skew of all models (RSR up to 1.62 under System Prompt Injection).