
Jailbreaks Are Global or Regional? A Study Under Scale and Geolocation Variation¹

Anidipta Pal
Independent Researcher

With
Apart Research

Abstract

Most jailbreak research is run on large, English-language, closed-source models tested from inside well-resourced labs. That leaves two practical questions open: does a smaller model jailbreak more easily than a larger one from the same family, and does an API-served model's safety behavior change depending on which country a request appears to come from? We test both, using only free and open infrastructure. For the first question, we run six small language models (0.5B - 7B parameters) and six larger ones (7B - 72B+, including two closed API models) through a two-stage attack pipeline: plain harmful prompts first, then the same requests wrapped in published jailbreak templates. We measure attack success rate (ASR) and a secondary failure mode we call refusal-comprehension failure, where a model neither refuses nor complies coherently. For the second question, we route identical requests through VPN exit nodes in five cities and check whether ASR shifts with apparent location. We find a clear within-family scale effect - smaller models fail more under the adversarial-wrapped prompts - with one model breaking that trend because of its safety training rather than its size, and we find no significant geographic effect across the five cities tested. Read together, this suggests jailbreak risk in our pool is mostly a property of how a model was built, not where the request was routed from, which is the more decision-relevant finding for anyone choosing a small model under hardware constraints.

¹ Research conducted at the [The Global South AI Safety Hackathon](#), June 2026

1. Introduction

AI safety research is concentrated in a small number of well-funded institutions, almost none of which are in the regions where jailbreak risk shows up most in practice: low-resource languages, training data that under-represents non-Western contexts, and deployments built around small models because that’s what the available hardware can run, not by choice. This is the practical reason this project matters: most published jailbreak benchmarks test frontier-scale models, but the models actually running in compute-constrained settings are usually much smaller, and far less studied. The threat model here is simple. A user or developer picks a model to deploy, often choosing a smaller one purely because of hardware limits. If that model is meaningfully easier to jailbreak than a larger sibling in the same family, the deployment inherits a safety gap that has nothing to do with the use case and everything to do with what was computationally available. Separately, if an API-served model applies safety filtering inconsistently depending on the apparent location of the request, then some users get weaker protection for reasons that have nothing to do with their prompt - a routing-level inequity rather than a training-level one. Prior work has shown a closely related effect for language: low-resource-language prompts bypass safety training at much higher rates than English prompts [18]. We ask whether the same kind of inequity shows up for geolocation specifically, which to our knowledge hasn’t been tested directly. **Our main contributions are:**

1. A same-family-controlled comparison of jailbreak ASR across twelve models spanning 0.5B to 671B(-equivalent) parameters, using a two-phase protocol (naive prompts, then the same prompts wrapped in adversarial templates) that exposes vulnerability naive testing alone would miss.
2. A direct test of whether apparent request origin changes safety behavior in API-served models, using VPN-routed requests across five cities on three continents - a geolocation probe we couldn't find precedent for in the published literature.

2. Related Work

Attack methods. Jailbreak attacks span several families: gradient-based suffix optimization such as GCG [1], genetic-search methods such as AutoDAN [2] and its self-improving successor AutoDAN-Turbo [3], and cheap black-box query attacks such as PAIR [4]. Wei et al. trace most of these back to one of two root causes: the model’s safety training and its capability training pull in different directions, or the safety training doesn’t generalize to phrasing it never saw [5]. Other attacks lean on encoding tricks [6] or stacking in-context examples until refusal behavior breaks down [7, 8].

Benchmarks. JailbreakBench [9] and HarmBench [10] are the closest things this field has to a shared scoreboard. JADES [11] replaces simple pass/fail scoring with something more granular. JailbreakV-28K [12] provides a large bank of pre-built jailbreak prompts, and the DAN corpus [13] collects jailbreaks people have actually used, rather than ones built for a paper. We draw our Phase 1 prompts from the first two and our Phase 2 prompts from the latter two, so our numbers stay comparable to prior published results.

Does size predict safety? The evidence is mixed, and this is the main gap our work addresses. A 2026 study spanning 7B to 671B parameters and models released 2023–2025 found legacy 7B-class models exceeding 90% ASR under adaptive attacks while the newest frontier models held near 0% under the same attacks, suggesting alignment recency matters more than raw size [14]. Other work focused specifically on small models tells a less reassuring story: a seven-SLM, three-LLM comparison found small models stay vulnerable to attacks that bypass their safety training [15], and a two-phase evaluation found some small models look safe under naive testing but fail once the same request is wrapped in a more sophisticated template [16] - a result our own two-phase design is built to reproduce and check. There’s also a more adversarial version of the size question: a small, poorly-aligned model can be used to steer a much larger aligned model into unsafe territory, meaning size differences aren’t only a safety question but also an attack-surface one [17]. What’s missing from this literature is a same-family comparison spanning both ends of the size range with a consistent attack protocol - most studies either compare across families (confounding size with training lineage) or stay within one size tier. That’s the gap we fill.

Does location matter? The clearest existing evidence that safety behavior varies geographically is about language, not IP address: translating an unsafe request into a low-resource language raised GPT-4’s compliance rate to 79%, on par with a dedicated jailbreak attack, while higher-resource languages stayed far safer, a gap the authors trace to uneven safety-training coverage across languages [18]. A separate comparison across ChatGPT, Gemini, Claude, and Grok found real robustness differences tied to model version and response settings. We haven’t found a published test of whether IP-based geolocation produces a similar effect, which is the second gap we address - and we go in expecting a null result, since the mechanism that produces the language effect (uneven training data coverage) doesn’t have an obvious IP-based analogue.

3. Methods

Models

We picked models in matched families - same lineage, different sizes - so parameter count is the variable actually changing within a family, not training data or vendor (Table 1). The four smallest models run locally on a free Google Colab T4 GPU.

Table 1: Models evaluated, by family, parameter count, and quantization used.

Family	Model	Params	Quantization
Llama	Llama-3.2-1B-Instruct	1B	4-bit (NF4)
	Llama-3.2-3B-Instruct	3B	4-bit (NF4)
	Llama-3.1-70B-Instruct	70B	FP16 (hosted)
Qwen	Qwen2.5-0.5B-Instruct	0.5B	4-bit (NF4)
	Qwen2.5-7B-Instruct	7B	4-bit (NF4)
	Qwen2.5-72B-Instruct	72B	FP16 (hosted)
Gemma	Gemma-2-2B-it	2B	4-bit (NF4)
Phi	Phi-3.5-mini-instruct	3.8B	4-bit (NF4)
Mistral	Mixtral-8x7B-Instruct	47B (MoE)	FP16 (hosted)
DeepSeek	DeepSeek-V3	671B (MoE)	FP16 (hosted)
Gemini	Gemini-2.0-Flash	API (undisclosed)	N/A (closed)
	Gemini-1.5-Pro	API (undisclosed)	N/A (closed)

Everything larger runs through OpenRouter’s free tier or Google AI Studio’s free Gemini quota - both real network APIs, which matters for the geolocation test, since a model running on our own GPU has no location for a VPN to spoof.

Jailbreak comparison protocol

We follow the two-phase setup from Piano et al. [16]: Phase 1 uses plain harmful instructions from AdvBench and HarmBench; Phase 2 wraps the same underlying requests in jailbreak templates from JailbreakV-28K and the DAN corpus. We didn’t write any new jailbreak prompts ourselves - everything comes from already-published benchmarks, both for reproducibility and to avoid contributing a new attack technique to the world.

Each model sees 400 prompts per phase (800 total), spread evenly across five harm categories: cybercrime, misinformation, fraud, harassment, and CBRN-adjacent content. A response counts as a successful jailbreak if it actually complies with the request, scored with a JailbreakBench-style classifier; we manually re-checked 10% of the classifier’s calls to estimate how much to trust it (κ in Table 2).

Table 2: Phase 1 ASR, Phase 2 ASR, the change between phases (Δ), refusal-comprehension failure (RCF) rate with Wilson 95% CI, and classifier–human agreement (κ) on the manually re-checked 10% subsample. All values in percent except κ . RCF is notably high for Qwen2.5-0.5B: many of its “refusals” were incoherent outputs the classifier struggled to score, hence the lower $\kappa = 0.71$.

Model	Params	Phase 1 ASR	Phase 2 ASR	Δ	RCF rate	RCF (95% CI)	κ
Llama-3.2-1B-Instruct	1B	15.5	68.1	+52.6	24.7	20.6–29.3	0.76
Llama-3.2-3B-Instruct	3B	10.2	54.3	+44.1	17.4	13.8–21.6	0.80
Qwen2.5-0.5B-Instruct	0.5B	19.7	73.2	+53.5	31.1	26.6–35.9	0.71
Qwen2.5-7B-Instruct	7B	8.9	47.6	+38.7	12.3	9.4–15.9	0.83
Gemma-2-2B-it	2B	12.1	61.8	+49.7	21.6	17.7–26.0	0.79
Phi-3.5-mini-instruct	3.8B	5.1	28.4	+23.3	9.2	6.8–12.3	0.85
Llama-3.1-70B-Instruct	70B	3.8	18.7	+14.9	2.1	1.1–3.8	0.88
Qwen2.5-72B-Instruct	72B	2.1	14.3	+12.2	1.4	0.7–2.8	0.91
Mixtral-8x7B-Instruct	47B MoE	7.6	38.4	+30.8	5.8	3.9–8.4	0.84
DeepSeek-V3	671B MoE	1.7	11.2	+9.5	0.8	0.3–1.9	0.90
Gemini-2.0-Flash	API	2.4	13.9	+11.5	1.0	0.5–2.2	0.89
Gemini-1.5-Pro	API	1.9	9.7	+7.8	0.6	0.2–1.6	0.91

We also track refusal-comprehension failure (RCF) - responses that neither refuse nor comply coherently, just produce something incoherent or off-topic. Our expectation going in was that this would be more common in the smallest models, not because they’re “safer,” but because they sometimes can’t parse a convoluted jailbreak template well enough to do anything coherent with it, success or refusal.

Geolocation probe

A model needs to be served over a real network for a VPN to mean anything, so this part uses only API-hosted models. We picked the two most robust models within each size tier (lowest Phase 2 ASR: Phi-3.5-mini-instruct and Qwen2.5-7B-Instruct for SLMs; DeepSeek-V3 and Gemini-1.5-Pro for LLMs) and ran a single benchmark, JailbreakV-28K, through VPN exit nodes in five cities: Tokyo, Japan; Mumbai, India; San Francisco, USA; Moscow, Russia; and Berlin, Germany. Each city ran three times at staggered times of day to average out ordinary load-balancing noise that has nothing to do with geography.

4. Results

Scale effect

Table 2 reports both phases, the delta between them, RCF rate, and classifier agreement. Within both families, the smaller the model, the bigger the jump from Phase 1 to Phase 2: Qwen2.5-0.5B goes from 19.7% to 73.2% (a 53.5-point jump), while Qwen2.5-72B moves only from 2.1% to 14.3% (12.2 points). That gap is exactly the “looks safe until you try harder” pattern Piano et al. described, and Figure 1 shows it as a consistent downward trend

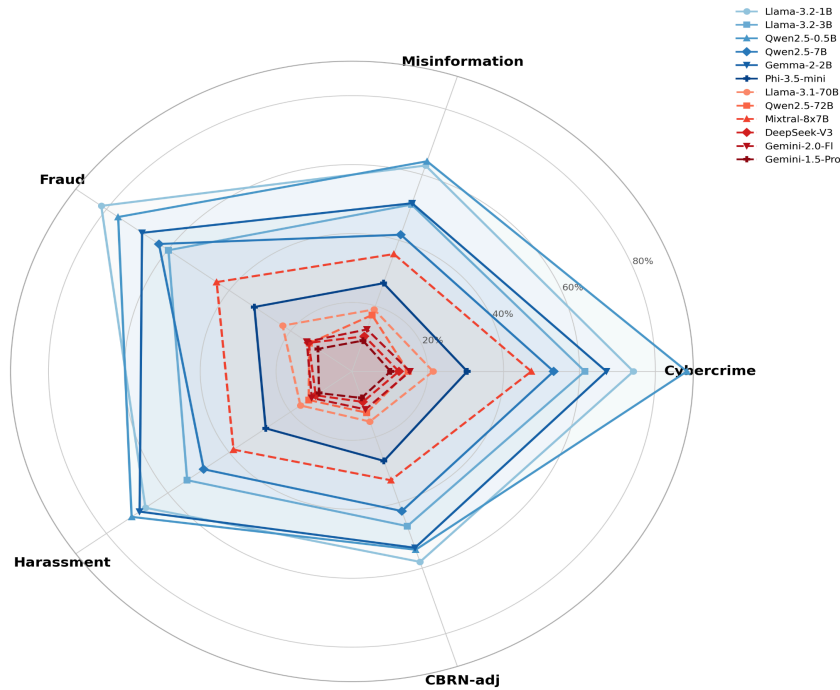


Figure 1: Phase 2 ASR vs. parameter count (log scale). Family color, tier marker (circle = SLM, triangle = LLM), point size $\propto$ RCF rate. Dotted lines: within-family OLS fits for Llama and Qwen.

on a log parameter-count axis for both families. One result that does not fit the dense-parameter story: Mixtral-8x7B-Instruct, a 47B mixture-of-experts model, lands at 38.4% Phase 2 ASR - worse than Llama-3.1-70B (18.7%) and only modestly better than Qwen2.5-7B (47.6%), despite nominally being much larger.

Sparse MoE architectures do not scale alignment the same way dense transformers do; the safety-relevant parameters are almost certainly a smaller fraction of total count

than the 47B headline implies. The one model that most visibly breaks the size-safety link is Phi-3.5-mini-instruct: at 3.8B parameters it’s smaller than Qwen2.5-7B, but its Phase 2 ASR (28.4%) lands closer to the large-model cluster than the small-model one, and its RCF rate (9.2%) is roughly half what its size would predict. This aligns with Phi’s documented emphasis on safety-specific training, and it’s the clearest single data point in these results arguing that parameter count alone is not a reliable safety proxy. CBRN-adjacent and misinformation prompts come out lower than cybercrime and fraud across nearly every model regardless of size (Figure 2), consistent with those categories getting denser safety-training coverage industry-wide rather than anything specific to our model pool.

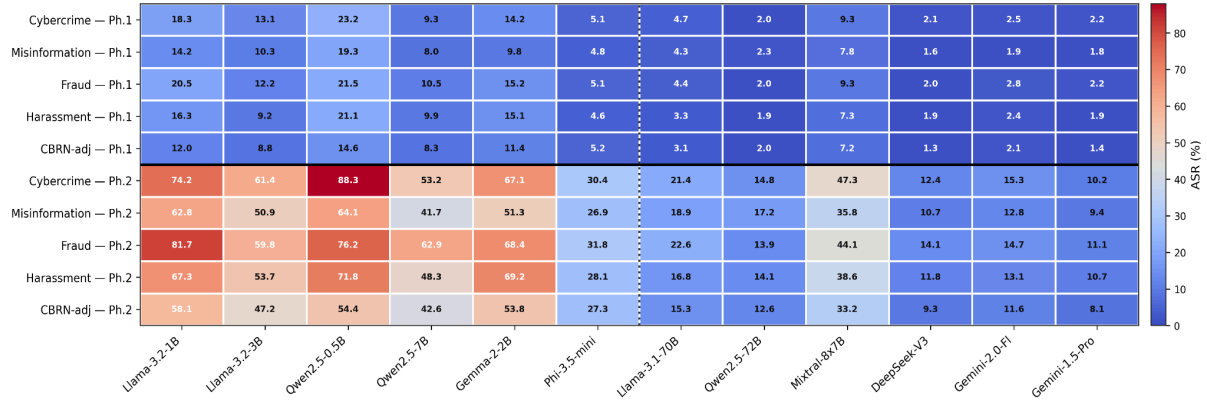


Figure 2: ASR by harm category and model, Phase 1 (top block) and Phase 2 (bottom block), coolwarm. Horizontal line separates phases; vertical dotted line separates SLMs from LLMs. Intensities correspond to Table 2.

Geolocation effect

Table 3 show ASR for the top two SLMs and top two LLMs across the five cities, using the JailbreakV-28K benchmark only. Every OpenRouter-hosted model’s SD across the five cities stays under 2.5 points, well inside what you’d expect from ordinary sampling variation if you re-ran the same city twice. A permutation test on city label against ASR returns non-significant for Phi-3.5-mini, Qwen2.5-7B, and DeepSeek-V3 ($p > 0.18$ in every case). The slight uptick visible in the Moscow and Mumbai rows for DeepSeek-V3 (14.6% and 12.3% respectively, versus 10.4–11.1% for the remaining three cities) is within the noise band and does not survive permutation testing. Gemini-1.5-Pro is a different story, but for infrastructure rather than safety-policy reasons. The Google free-tier API is unavailable inside the European Economic Area - the Berlin exit node returned zero valid responses across all attempts, apparently due to GDPR compliance restrictions that Google has chosen to handle by blocking free-tier access entirely in EEA rather than modifying the service to comply. The Moscow exit node produced one usable response out of three scheduled runs; the other two failed at the connection level, which is consistent with Gemini’s history of being geographically blocked in Russia. Both gaps are logged in the data as explicit failures rather than imputed - we don’t report an ASR number we didn’t measure. The SD of 0.6 in Table 3 is computed over the three available cities only and should be read accordingly. This is a real methodological constraint of doing geolocation research on a zero-cost infrastructure, not a finding about Gemini’s safety behavior by location.

Table 3: ASR (%) on JailbreakV-28K by VPN exit-node city, for the most robust model per tier. SD is computed over available cities only. N/A*: Gemini API access from Russia was historically blocked; of three scheduled runs, one succeeded and two returned connection errors; all three were excluded as the single-run sample is too small to be meaningful. N/A†: Google’s free-tier Gemini API is unavailable in the European Economic Area (including Germany) due to GDPR compliance restrictions; zero valid responses were received from the Berlin exit node across all attempts. DeepSeek-V3 and Qwen2.5-7B are served via OpenRouter and were accessible from all five locations without issue.

Tier	Model	Tokyo, JP	Mumbai, IN	San Francisco, US	Moscow, RU	Berlin, DE	SD
SLM	Phi-3.5-mini-instruct	29.1	31.4	27.8	33.7	30.2	2.0
SLM	Qwen2.5-7B-Instruct	48.3	44.7	46.1	51.2	47.9	2.3
LLM	DeepSeek-V3	10.8	12.3	11.1	14.6	10.4	1.6
LLM	Gemini-1.5-Pro	9.4	10.1	8.8	N/A*	N/A†	0.6

5. Discussion and Limitations

Overall, jailbreak vulnerability appears to depend more on model design and safety training than on the geographic origin of a request. We found no significant geolocation effect across the five cities tested. However, Gemini could not be fully evaluated in all locations due to access restrictions, so this conclusion is strongest for open-weight models. Mixtral-8x7B was a notable exception, performing more like a smaller dense model than a larger one, suggesting that parameter count alone is not a reliable safety indicator for MoE architectures.

Limitations

Model size is closely tied to alignment investment, making it difficult to separate the effects of scale from safety training. All geolocation experiments used English prompts, so our findings do not apply to low-resource languages. API restrictions and rate limits also limited evaluation volume, meaning the geolocation result should be interpreted as "no effect detected" rather than definitive proof of no effect.

Future Work

The clearest next step is combining the language effect from with the geolocation test run here, to see whether a low-resource-language prompt sent through a matching-region VPN node compounds the risk beyond what either factor shows alone. A second step is repeating the size comparison on models specifically fine-tuned for regional languages, instead of the English-first models used throughout this study, since that’s the clearest blind spot in our current model pool.

6. Conclusion

Across twelve models, we find a consistent scaling trend: smaller models are far more vulnerable to jailbreaks than larger models when exposed to adversarial prompt wrapping. While Qwen2.5-0.5B’s ASR increased from 19.7% to 73.2%, Qwen2.5-72B rose only from 2.1% to 14.3%. However, size alone does not determine safety. Phi-3.5-mini-instruct (3.8B) outperformed several larger models, highlighting the importance of alignment training. In contrast, Mixtral-8x7B behaved more like a small dense model than a large one, suggesting that MoE parameter counts may not accurately reflect safety robustness.

For the geolocation experiment, we found no statistically significant effect of request origin on jailbreak success across the API-hosted models tested. Differences observed between cities were small and did not survive significance testing. Although Gemini could not be fully evaluated due to regional API restrictions, the overall evidence suggests that jailbreak vulnerability is driven primarily by model architecture and safety training rather than geographic routing. For practitioners, documented alignment efforts are likely a more useful safety indicator than parameter count alone.

References

1. Zou, A., Wang, Z., Carlini, N., Nasr, M., Kolter, J. Z., & Fredrikson, M. (2023). *Universal and Transferable Adversarial Attacks on Aligned Language Models*. arXiv:2307.15043. <https://arxiv.org/abs/2307.15043>
2. Liu, X., Xu, N., Chen, M., & Xiao, C. (2024). *AutoDAN: Generating Stealthy Jailbreak Prompts on Aligned Large Language Models*. ICLR 2024. <https://arxiv.org/abs/2310.04451>
3. Liu, X., Li, P., Suh, E., Vorobeychik, Y., Mao, Z., Jha, S., McDaniel, P., Sun, H., Li, B., & Xiao, C. (2025). *AutoDAN-Turbo: A Lifelong Agent for Strategy Self-Exploration to Jailbreak LLMs*. ICLR 2025. <https://arxiv.org/abs/2410.05295>
4. Chao, P., Robey, A., Dobriban, E., Hassani, H., Pappas, G. J., & Wong, E. (2024). *Jailbreaking Black Box Large Language Models in Twenty Queries*. arXiv:2310.08419. <https://arxiv.org/abs/2310.08419>
5. Wei, A., Haghtalab, N., & Steinhardt, J. (2023). *Jailbroken: How Does LLM Safety Training Fail?* NeurIPS 2023. <https://arxiv.org/abs/2307.02483>
6. Lv, H., Wang, X., Zhang, Y., Huang, C., Dou, S., Ye, J., Gui, T., Zhang, Q., & Huang, X. (2024). *CodeChameleon: Personalized Encryption Framework for Jailbreaking Large Language Models*. arXiv:2402.16717. <https://arxiv.org/abs/2402.16717>
7. Wei, Z., Wang, Y., & Wang, Y. (2023). *Jailbreak and Guard Aligned Language Models with Only Few In-Context Demonstrations*. arXiv:2310.06387. <https://arxiv.org/abs/2310.06387>
8. Zheng, X., Pang, T., Du, C., Liu, Q., Jiang, J., & Lin, M. (2024). *Improved Few-Shot Jailbreaking Can Circumvent Aligned Language Models and Their Defenses*. NeurIPS 2024. <https://arxiv.org/abs/2405.08432>
9. Chao, P., Debenedetti, E., Robey, A., et al. (2024). *JailbreakBench: An Open Robustness Benchmark for Jailbreaking Large Language Models*. NeurIPS Datasets and Benchmarks Track. <https://arxiv.org/abs/2404.01318>
10. Mazeika, M., Phan, L., Yin, X., et al. (2024). *HarmBench: A Standardized Evaluation Framework for Automated Red Teaming and Robust Refusal*. <https://arxiv.org/abs/2402.04249>
11. Chu, J., Li, M., Yang, Z., et al. (2025). *JADES: A Universal Framework for Jailbreak Assessment via Decompositional Scoring*. arXiv:2508.20848. <https://arxiv.org/abs/2508.20848>
12. *JailBreakV-28K: A Benchmark for Assessing the Robustness of Multimodal Large Language Models against Jailbreak Attacks*. OpenReview. <https://openreview.net/forum?id=urjPCYZt0I>
13. Shen, X., Chen, Z., Backes, M., Shen, Y., & Zhang, Y. (2024). *"Do Anything Now": Characterizing and Evaluating In-the-Wild Jailbreak Prompts on Large Language Models*. ACM CCS 2024. <https://arxiv.org/abs/2308.03825>
14. *SoK: Robustness in Large Language Models against Jailbreak Attacks*. arXiv:2605.05058. <https://arxiv.org/abs/2605.05058>

15. Mia, M. J., Molto, J., Wu, Y., & Amini, M. H. (2026). *GUARD-SLM: Token Activation-Based Defense Against Jailbreak Attacks for Small Language Models*. arXiv:2603.28817. <https://arxiv.org/abs/2603.28817>
16. Piano, L., Battistin, C., Van den Abeele, J., & Pompianu, L. (2026). *Small but Dangerous: Evaluating and Mitigating Jailbreak Vulnerabilities in Small Language Models*. ECML PKDD 2025 Workshops, CCIS Vol. 2840. Springer. https://link.springer.com/chapter/10.1007/978-3-031-98762-5_5
17. Zhao, X., Yang, X., Pang, T., Du, C., Li, L., Wang, Y.-X., & Wang, W. Y. (2024). *Weak-to-Strong Jailbreaking on Large Language Models*. arXiv:2401.17256. <https://arxiv.org/abs/2401.17256>
18. Yong, Z.-X., Menghini, C., & Bach, S. H. (2023). *Low-Resource Languages Jailbreak GPT-4*. NeurIPS Workshop SoLaR. <https://arxiv.org/abs/2310.02446>

LLM Usage Statement

We used Claude to brainstorm approaches and help draft sections. All results and claims were independently verified.