# Stability Evaluation of Large Language Models via Distributional Perturbation Analysis

**Jiashuo Liu**[1], **Jiajin Li**[2], **Peng Cui**[1], **Jose Blanchet**[3]

[1]Tsinghua University, [2]University of British Columbia, [3]Stanford University
liujiashuo77@gmail.com, jiajin.li@sauder.ubc.ca
cuip@tsinghua.edu.cn, jose.blanchet@stanford.edu

## Abstract

The performance of Large Language Models (LLMs) can degrade when exposed to shifts such as changes in language style or domain-specific knowledge that is underrepresented in the training data. To ensure robust deployment, we propose a stability evaluation criterion based on distributional perturbations. Conceptually, this criterion measures the minimal perturbation required in the data to induce a specified deterioration in model performance. We employ optimal transport (OT) discrepancy with moment constraints on the *(sample, density)* space to quantify these perturbations. This allows our stability criterion to address both *data corruptions* and *sub-population shifts*, which are common in real-world LLM applications. To make this approach practical, we provide tractable convex formulations and computational methods tailored to different classes of loss functions used in LLMs. Empirically, we validate the utility of our stability criterion by testing LLMs on tasks such as jailbreak attempts and general question-answering tasks, demonstrating its effectiveness in assessing model robustness and providing insights into improving stability under diverse real-world scenarios.

## 1 Introduction

Large Language Models (LLMs) [14, 1, 6, 17] have emerged as powerful tools for a wide range of natural language processing (NLP) tasks, from question answering [18] and summarization [7] to code generation [10] and conversational agents [16]. Their ability to handle complex language patterns and vast amounts of information has led to widespread adoption in both academic and industrial applications. However, despite these advancements, LLMs still exhibit significant weaknesses when confronted with real-world challenges, particularly when the data distribution they encounter shifts from the training distribution [11].

Distribution shifts can take many forms, such as changes in language style [4], introduction of domain-specific knowledge [15], or the presence of adversarial inputs [8]. For example, when an LLM trained on general-purpose data is faced with legal or medical terminology that was underrepresented in its training set, its performance often degrades. Similarly, shifts in linguistic patterns, such as informal or colloquial language, can cause the model to generate inaccurate or irrelevant responses. Even more critically, adversarial attacks, where inputs are deliberately crafted to manipulate the model into producing incorrect or harmful outputs, expose the fragility of LLMs. This sensitivity to shifts in input distribution can have serious implications in real-world applications, where LLMs must reliably handle dynamic and diverse inputs across various domains.

Conventional evaluation metrics for LLMs, which typically rely on the assumption of independent and identically distributed ($i.i.d.$) data, fail to capture the challenges posed by such distributional shifts. Metrics like accuracy offer limited insights into how well an LLM can generalize under

NeurIPS 2024 Workshop on Red Teaming GenAI.

varying real-world conditions. In scenarios such as adversarial testing—where inputs are deliberately manipulated to bypass content filters or exploit weaknesses in the model—or domain-specific shifts, such as transitioning from general conversational language to highly technical medical or legal jargon, traditional metrics like accuracy are likely to fail to capture the true robustness of a model. For example, an LLM trained primarily on everyday text may perform well in general question-answering tasks but could struggle when faced with precise medical diagnoses or legal contract analysis, leading to critical errors. Similarly, in jailbreak attempts, where inputs are crafted to trick the model into generating harmful or inappropriate content, conventional metrics like accuracy do not reflect the model's susceptibility to these manipulations. These kinds of real-world distribution shifts highlight the limitations of standard evaluation approaches.

To address these limitations, we propose a stability evaluation framework that goes beyond conventional metrics by quantifying an LLM's sensitivity to distributional perturbations. Our framework is based on the concept of minimal perturbation: the smallest change in the input data required to induce a specified deterioration in the model's performance. By employing optimal transport (OT) methods, we are able to quantify both data corruptions and sub-population shifts, two of the most prevalent types of distribution shifts in real-world settings. This approach enables us to evaluate not only how well a model performs under $i.i.d.$ conditions, but also how stable and resilient it remains when faced with shifts in input data.

Our stability evaluation offers a comprehensive assessment of model robustness, taking into account a broad range of potential distribution shifts. By adjusting the ratio of different perturbation types and setting varying risk thresholds, our framework provides insights into the trade-offs that LLMs face between performance on standard tasks and resilience to adversarial or shifted data. This allows for a deeper understanding of the model's capabilities and limitations in dynamic, real-world environments, where data distributions are rarely static or predictable. In this paper, we present the theoretical foundation of our stability evaluation criterion and demonstrate its practical utility through empirical validation on tasks such as jailbreak attempts and general question answering tasks.

## 2 LLM Stability Evaluation Framework

**Notations.** Throughout this paper, we let $\mathbb{R}$ denote the set of real numbers, $\mathbb{R}_+$ denote the subset of non-negative real numbers. We use capitalized letters for random variables, e.g., $X, Y, Z$, and script letters for the sets, e.g., $\mathcal{X}, \mathcal{Y}, \mathcal{Z}$. For any close set $\mathcal{Z} \subset \mathbb{R}^d$, we define $\mathcal{P}(\mathcal{Z})$ as the family of all Borel probability measures on $\mathcal{Z}$. For $\mathbb{P} \in \mathcal{P}(\mathcal{Z})$, we use the notation $\mathbb{E}_{\mathbb{P}}[\cdot]$ to denote expectation with respect to the probability distribution $\mathbb{P}$. For the prediction problem, the random variable of data points is denoted by $Z = (X, Y) \in \mathcal{Z}$, where $X \in \mathcal{X}$ and $Y \in \mathcal{Y}$ are both natural language / text, denoting the input questions and the answer, respectively. $f_\beta : \mathcal{X} \to \mathcal{Y}$ denotes the language model parameterized by $\beta$. The loss function is denoted as $\ell : \mathcal{Y} \times \mathcal{Y} \to \{0, 1\}$, and $\ell(f_\beta(X), Y)$ is abbreviated as $\ell(\beta, Z)$. We use $(\cdot)_+ = \max(\cdot, 0)$. We adopt the conventions of extended arithmetic, whereby $\infty \cdot 0 = 0 \cdot \infty = 0/0 = 0$ and $\infty - \infty = -\infty + \infty = 1/0 = \infty$.

### 2.1 OT Discrepancy for Language Data

We begin by presenting the OT discrepancy with moment constraints, as proposed in [2, Definition 2.1], based on which we develop our stability evaluation framework.

**Definition 2.1** (OT discrepancy with moment constraints)**.** *If $\mathcal{Z} \subseteq \mathbb{R}^d$ and $\mathcal{W} \subseteq \mathbb{R}_+$ are convex and closed sets, $c : (\mathcal{Z} \times \mathcal{W})^2 \to \mathbb{R}_+$ is a lower semicontinuous function, and $\mathbb{Q}, \mathbb{P} \in \mathcal{P}(\mathcal{Z} \times \mathcal{W})$, then the OT discrepancy with moment constraints induced by $c$, $\mathbb{Q}$ and $\mathbb{P}$ is the function $\mathbb{M}_c : \mathcal{P}(\mathcal{Z} \times \mathcal{W})^2 \to \mathbb{R}_+$ defined through*

$$\mathbb{M}_c(\mathbb{Q}, \mathbb{P}) = \begin{cases} \inf & \mathbb{E}_\pi[c((Z, W), (\hat{Z}, \hat{W}))] \\ \text{s.t.} & \pi \in \mathcal{P}((\mathcal{Z} \times \mathcal{W})^2) \\ & \pi_{(Z,W)} = \mathbb{Q},\ \pi_{(\hat{Z},\hat{W})} = \mathbb{P} \\ & \mathbb{E}_\pi[W] = 1 \quad \pi\text{-a.s,} \end{cases}$$

*where $\pi_{(Z,W)}$ and $\pi_{(\hat{Z},\hat{W})}$ are the marginal distributions of $(Z, W)$ and $(\hat{Z}, \hat{W})$ under $\pi$.*

**Remark.** *The core idea is to map the original sample space $\mathcal{Z}$ into a higher-dimensional space $\mathcal{Z} \times \mathcal{W}$, which combines both samples and densities. In this extended space, the additional random*

*variable $W$ represents the "density" or "probability mass," allowing it to be adjusted through optimal transport methods. These adjustments are constrained by the requirement that the expected density remains constant at one. Therefore, the transportation cost function $c((z,w),(\hat{z},\hat{w}))$ quantifies changes in both the sample values ($\hat{z} \to z$) and their associated densities ($\hat{w} \to w$).*

In order to measure the discrepancy between two distribution of language data (e.g. sentences), we design the transportation cost function $c(\cdot,\cdot)$ as:

$$c((z,w),(\hat{z},\hat{w})) = \theta_1 \cdot w \cdot \underbrace{\left( \underbrace{\frac{\Phi(x)^T \Phi(\hat{x})}{\|\Phi(x)\|\|\Phi(\hat{x})\|}}_{\text{cosine similarity}} \cdot \underbrace{\max\left(\frac{\#\text{Token}(x)}{\#\text{Token}(\hat{x})}, \frac{\#\text{Token}(\hat{x})}{\#\text{Token}(x)}\right)}_{\text{token number ratio}} \right)}_{\text{perturbation distance}} + \theta_2 \cdot \underbrace{(\phi(w) - \phi(\hat{w}))_+}_{\text{reweighting distance}}. \tag{1}$$

Here, $\theta_1$ and $\theta_2$ satisfy $\theta_1 \geq 0, \theta_2 \geq 0$, and $1/\theta_1 + 1/\theta_2 = c$, where $c > 0$ is a constant. Generally, $\theta_1$ and $\theta_2$ controls the relative strength of data perturbations (e.g., change the input questions to a different style) and sub-population shifts (e.g., change the relative ratios of different topics of questions).

**Data Perturbation Distance** To quantify the extent of data perturbation from a sentence $x$ to $\hat{x}$, we introduce two key metrics: semantic similarity and editing distance. For semantic similarity, we compute the distance in the text embedding space, using the OpenAI embedding model [1], denoted as $\Phi(\cdot)$ throughout this paper. To further capture the "editing" distance, we combine cosine similarity with the token count ratio, where a larger ratio indicates a higher cost for perturbing the data. And the token count of a sentence, represented as $\#\text{Token}(\cdot)$, is determined by the tokenizer. These metrics together provide a comprehensive measure of the similarity between two sentences. Furthermore, to ensure that the semantic meaning remains unchanged from $x$ to $\hat{x}$, we utilize GPT-4 as a judge. We input both $x$ and $\hat{x}$, asking whether they represent the same question. If GPT-4 determines they do not, the cost function is set to infinity. Intuitively, our cost function penalizes both changes in semantic meaning and increases in sentence length during data perturbation.

**Sub-population Shift Distance** To measure the strength of sub-population shifts (i.e. reweighting), we use $(\phi(w) - \phi(\hat{w}))_+$ to capture the differences in probability mass, where $\phi : \mathbb{R}_+ \to \mathbb{R}_+$ is a convex function satisfying $\phi(1) = 0$. Throughout this paper, we choose $\phi(w)$ as $\phi(w) = w \log w - w + 1$, which is associated with the Kullback–Leibler (KL) divergence.

### 2.2 OT-based Stability Evaluation Criterion

Unlike conventional LLM evaluation metrics that primarily focus on $i.i.d.$ settings—such as computing average scores across test samples—our stability evaluation assesses the model's *sensitivity* to potential distribution shifts. This approach provides a more accurate reflection of an LLM's performance in real-world applications.

To evaluate the stability of a given large language model $f_\beta$ on data drawn from the distribution $\mathbb{P}_0 \in \mathcal{P}(\mathcal{Z})$, we formally introduce the OT-based stability evaluation criterion as:

$$\mathfrak{R}(\beta, r) = \left\{ \begin{array}{ll} \inf\limits_{\mathbb{Q} \in \mathcal{P}(\mathcal{Z} \times \mathcal{W})} & \mathbb{M}_c(\mathbb{Q}, \hat{\mathbb{P}}) \\ \text{s.t.} & \mathbb{E}_{\mathbb{Q}}[W \cdot \ell(\beta, Z)] \geq r. \end{array} \right. \tag{P}$$

Here, the reference measure $\hat{\mathbb{P}}$ is selected as $\mathbb{P}_0 \otimes \delta_1$, with $\delta_1$ denoting the Dirac delta function,[2] $\mathbb{M}_c(\mathbb{Q}, \hat{\mathbb{P}})$ represents the OT discrepancy with moment constraints between the projected distribution $\mathbb{Q}$ and the reference distribution $\hat{\mathbb{P}}$, the transportation cost function $c$ is chosen as Equation 1, $\ell(\beta, z)$ denotes the prediction risk of model $f_\beta$ on sample $z$, and $r > 0$ is the pre-defined risk threshold.

To sum up, we evaluate a model's stability under distribution shifts by quantifying the *minimum* level of perturbations required for the model's performance to degrade to a predetermined risk threshold.

[1] https://platform.openai.com/docs/guides/embeddings

[2] This implies that the sample weights are almost surely equal to one with respect to the reference distribution, as we lack any prior information about them.

The magnitude of perturbations is determined through the use of the OT discrepancy with moment constraints and the cost function $c$, see definition 2.1.

**Remark** (Effect of $\theta_1$ and $\theta_2$)**.** (i) When $\theta_1 = +\infty$, the stability criterion $\mathfrak{R}(\beta, r)$ only counts the sub-population shifts, as any data sample corruptions are not allowed. In this scenario, our proposed stability criterion can be reduced to the one recently introduced in [5] and [9]. (ii) When $\theta_2 = +\infty$, the stability criterion $\mathfrak{R}(\beta, r)$ only takes the data corruptions into account instead. (iii) The most intriguing scenario arises when both $\theta_1$ and $\theta_2$ have finite values. These parameters, $\theta_1$ and $\theta_2$, hold a pivotal role in adjusting the balance between data corruptions and sub-population shifts within our stability criterion, which allows us to simultaneously consider both types of distribution shifts. By manipulating the values of $\theta_1$ and $\theta_2$, we can achieve a versatile representation of a LLM's resilience across a wide range of distributional perturbation directions. This adaptability carries significant implications when evaluating the robustness of LLMs in diverse and ever-evolving real-world environments.

### 2.3 Dual Reformulation

**Proposition 2.1** (Dual reformulations)**.** *Denote $Z = (X, Y)$, and suppose that $\mathcal{W} = \mathbb{R}_+$. When choosing the cost function as Equation 1 with $\phi(w) = w \log w - w + 1$, the dual problem of Problem P admits:*

$$\sup_{h \geq 0} hr - \theta_2 \log \mathbb{E}_{\mathbb{P}_0} \left[ \exp \left( \frac{\ell_{h,\theta_1}(\hat{Z})}{\theta_2} \right) \right]; \tag{2}$$

*where $\ell_{h,\theta_1}(\cdot)$ is defined as*

$$\ell_{h,\theta_1}(\hat{z}) := \max_{z \in \mathcal{Z}} h \cdot \ell(\beta, z) - \theta_1 \cdot \left( \frac{\Phi(x)^T \Phi(\hat{x})}{\|\Phi(x)\| \|\Phi(\hat{x})\|} \cdot \max\left( \frac{\textit{\#Token}(x)}{\textit{\#Token}(\hat{x})}, \frac{\textit{\#Token}(\hat{x})}{\textit{\#Token}(x)} \right) \right). \tag{3}$$

Furthermore, we can derive the formulation of the most sensitive distribution of a given LLM.

**Remark** (Structure of the most sensitive distribution)**.** *We express $\mathbb{Q}^\star$ as follows: $\mathbb{Q}^\star = \frac{1}{n} \sum_{i=1}^n \delta_{(z_i^\star, w_i^\star)}$, where each $(z_i^\star, w_i^\star) \in \mathcal{Z} \times \mathbb{R}_+$ satisfies the conditions:*

$$z_i^\star = \arg\max_{z \in \mathcal{Z}} h^\star \ell(\beta; z) - \theta_1 \cdot \left( \frac{\Phi(x)^T \Phi(\hat{x}_i)}{\|\Phi(x)\| \|\Phi(\hat{x}_i)\|} \cdot \max\left( \frac{\textit{\#Token}(x)}{\textit{\#Token}(\hat{x}_i)}, \frac{\textit{\#Token}(\hat{x}_i)}{\textit{\#Token}(x)} \right) \right), \quad \forall i \in [n].$$

*Note that $z = (x, y)$ denotes both the input and the desired output. And we have*

$$w_i^\star \propto \exp \left( \frac{\ell_{h^\star,\theta_1}(\hat{z}_i)}{\theta_2} \right), \quad \forall i \in [n],$$

*where $h^\star$ and $\alpha^\star$ are the optimal solution of problem (2.1). Therefore, it becomes evident that the most sensitive distribution encompasses both aspects of shifts: the transformation from $\hat{z}_i$ to $z_i^\star$ and the reweighting from $\frac{1}{n}$ to $w_i^\star$. Our cost function enables a versatile evaluation of LLM stability across a range of distributional perturbation directions. This approach yields valuable insights into the behavior of a LLM in potential shifts and underscores the importance of incorporating both types of distributional perturbation in stability evaluation.*

### 2.4 Optimization

For LLMs, computing Equation (3) presents a significant challenge because gradient information with respect to the input $\hat{z}$ is not accessible through modern LLM APIs. Moreover, given our focus on evaluating the "generalization" ability of LLMs, we aim to avoid token-level perturbations that render the perturbed samples unnatural. Instead, our goal is to ensure that the perturbed samples remain natural, which adds an additional layer of difficulty to our optimization problem.

Therefore, we restrict the input space $\mathcal{X}$ to be discrete, and then calculate the maximizer $x^\star$ in the discrete space $\hat{\mathcal{X}}$.

**Sample Perturbation** Specifically, for each of the input $\hat{x}$, we generate $K$ candidates $\hat{x}_1, \ldots, \hat{x}_K$ that satisfy: (i) they remain similar semantic meanings as original; (ii) they remain natural languages instead of some specific symbols, and include them in the discrete space $\hat{\mathcal{X}}$. To accomplish this, we use the tree-of-attack framework from [8, 3]. In this approach, an assisted LLM is tasked with rephrasing the input sentence into more challenging variations while preserving its semantic meaning. These rephrased questions are then evaluated by an evaluator (e.g., GPT-4) to determine if they induce a high prediction error (e.g., if the target model $f_\beta(\cdot)$ provides an incorrect answer). This process is carried out iteratively. To empirically validate the effectiveness of this approach, we measure the ratio of perturbed samples that successfully elicit incorrect answers for each type of target LLM. Additionally, we calculate the average number of queries required to achieve each successful perturbed sample. As shown in Table 1, the ratio is high and the number of average queries is low, indicating the efficiency of our sample perturbation part.

Based on the $K$ candidates generated above, our optimization problem becomes:

$$\mathfrak{R}(\beta, r) = \sup_{h \geq 0} hr - \theta_2 \log \mathbb{E}_{\mathbb{P}_0}\left[\exp\left(\frac{\ell_{h,\theta_1}(\hat{Z})}{\theta_2}\right)\right]$$

$$\ell_{h,\theta_1}(\hat{z}) := \max_{x \in \{\hat{x}_1, \ldots, \hat{x}_K\}} h \cdot \ell(f_\beta(x), \hat{y}) - \theta_1 \cdot \left(\frac{\Phi(x)^T \Phi(\hat{x})}{\|\Phi(x)\| \|\Phi(\hat{x})\|} \cdot \max\left(\frac{\#\text{Token}(x)}{\#\text{Token}(\hat{x})}, \frac{\#\text{Token}(\hat{x})}{\#\text{Token}(x)}\right)\right),$$

which becomes a convex optimization problem and is easy to solve. Note that for the loss function $\ell(\cdot, \cdot)$, we adopt the methods used in [8, 3], which uses GPT-4 to evaluate the answer, leading to a 0/1-loss.

Table 1: **Fraction of Adversarial Samples Achieved as per the GPT4-turbo.** For each method and target LLM, we report (1) the fraction of adversarial samples found on the Jailbreak and Alpaca dataset, and (2) the average number of queries sent to the target LLM for each adversarial sample. We use Vicuna-13B-V1.5 as the assisting LLM. In each column, the best results are bolded. Note that Llama2 models are extremely hard to jailbreak, which is also found in [8, 3]. And we exclude them for the jailbreak task.

| Dataset | Metric | Vicuna-13B | Vicuna-7B | Llama2-13B | Llama2-7B | Mistral-7B | DeepSeek-7B | ChatGLM2-6B | Qwen2-7B |
|---|---|---|---|---|---|---|---|---|---|
| Jailbreak | Adversarial % | 84.2% | 85.2% | - | - | 85.8% | 83.9% | 86.4% | 81.8% |
| | # Avg. Queries | 2.65 | 4.06 | - | - | 2.39 | 3.58 | 2.99 | 3.00 |
| Alpaca | Adversarial % | 80.2% | 82.1% | 82.3% | 84.9% | 73.8% | 87.7% | 83.7% | 82.4% |
| | # Avg. Queries | 4.47 | 3.52 | 4.82 | 4.39 | 6.20 | 4.25 | 3.09 | 4.77 |

## 3 Experiments

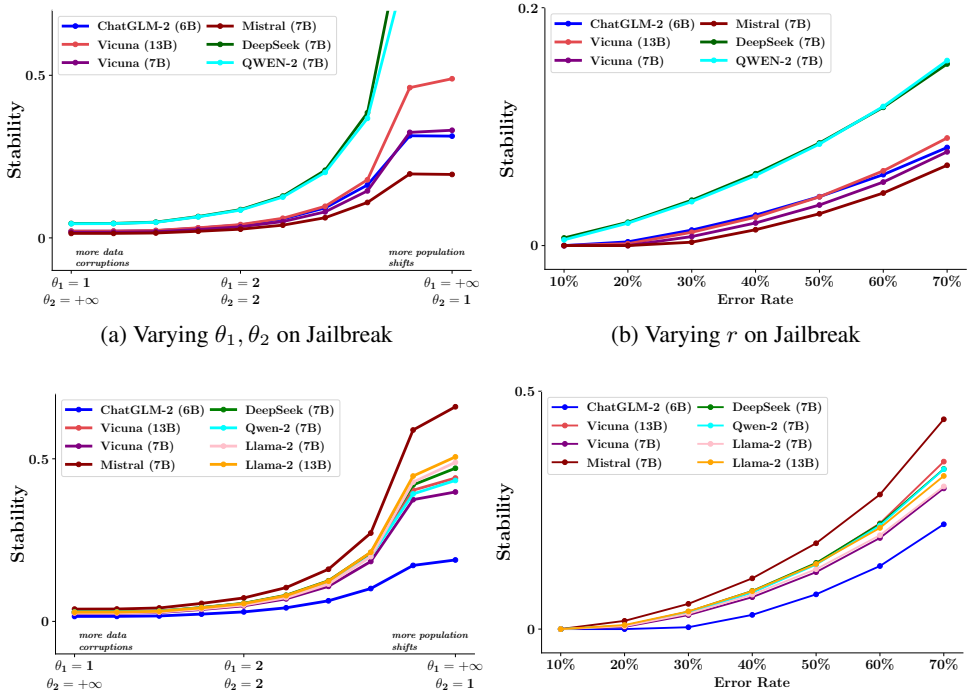
In this section, we evaluate the stability of open LLMs on two tasks, including jailbreak attempts and general QA challenges. Throughout this section, we use GPT-4 as the evaluator (for loss function $\ell(\cdot, \cdot)$), Vicuna-13B-v1.5 as the assisting LLM (for generating perturbed samples). The LLMs under evaluation include Vicuna-7B/13B-v1.5, Llama2-7B/13B, Mistral-7B-v0.2, DeepSeek-7B, ChatGLM-6B-v2, and Qwen-7B-v2.

**Jailbreak Task** For the jailbreak task, we use the forbidden question set from [12], which consists of 11 scenarios, including illegal activity, hate speech, malware generation, physical harm, economic harm, fraud, pornography, political lobbying, privacy violations, legal advice, and government decisions. Each scenario contains 30 forbidden questions.

**General QA Task** For the general QA task, we use the Alpaca [13] dataset, where we randomly sample 1,000 questions to evaluate the stability.

**Evaluation Setup** In our experiments, we adjust the relative ratios $\theta_1$ and $\theta_2$ of data perturbations and sub-population shifts, as well as the prediction risk threshold $r$ in our stability framework, to

compare the stability of different LLMs. Specifically, this involves two aspects: (i) for a fixed prediction error threshold $r$, we vary $\theta_1$ and $\theta_2$ to assess stability under different combinations of data corruptions and sub-population shifts; (ii) for a fixed ratio of $\theta_1$ and $\theta_2$, we vary the prediction error threshold to evaluate stability under different levels of difficulty. The results are shown in Figure 1.




(a) Varying $\theta_1, \theta_2$ on Jailbreak

(b) Varying $r$ on Jailbreak

(c) Varying $\theta_1, \theta_2$ on Alpaca

(d) Varying $r$ on Alpaca

Figure 1: Stability curve on Jailbreak dataset and Alpaca dataset. (a) and (c): we fix the error rate $r = 50\%$ and vary the choices of $\theta_1, \theta_2$; (b) and (d): we fix $\theta_1 = \theta_2 = 2$ and vary the error rate $r$.

From the results, we have the following findings:

**Evaluation should not rely on one single metric.** Evaluating LLMs is inherently complex and cannot be fully captured by simple metrics like accuracy or adversarial success rate. Our stability curve demonstrates that the ranking of LLMs shifts based on different values of $\theta_1$, $\theta_2$, and the error rate $r$. Furthermore, the rankings generated by our stability measure often differ from those derived from adversarial success rates alone. For example, there are many intersections between curves in Figure 1a-1c, indicating that under different scenarios, the stability of LLMs can change. Therefore, our stability metric offers a more comprehensive evaluation of LLM sensitivity across a wide range of potential distribution shifts.

**The success rate of jailbreak is important, but the quality of the adversarial samples is equally significant.** When measuring the stability w.r.t. data corruptions, different from the evaluation used in conventional jailbreak works, we need to look into the quality of the adversarial samples. In this work, we take into consideration both the length of adversarial prompts and the semantic similarity between adversarial prompts and the original ones. For example, the jailbreak success rate of Vicuna-13B (84.2%) is lower than ChatGLM2-7B (86.4%), indicating that Vicuna-13B is more stable. However, when setting the error threshold between 10% and 20%, as shown in Figure 1b, the stability of Vicuna-13B (red curve) is lower than ChatGLM2-6B (blue curve). This phenomenon suggests that stability should not be assessed solely based on success rate, as the quality of adversarial samples plays an equally important role. For instance, adversarial samples may vary significantly in terms of length or editing distance, both of which should be factored into the stability evaluation to provide a more accurate and nuanced assessment. This further highlights the need for a comprehensive stability metric in practical evaluations.

**There is a tradeoff in stability between harmless and harmful questions.** As shown in Figure 1c and Figure 1d, Mistral-7B (dark red curve) performs exceptionally well on harmless question answering with Alpaca, demonstrating strong stability compared to other 7B models. However, as seen in Figure 1a, its stability is significantly weaker when handling harmful questions. This reveals a tradeoff in stability between harmless question-answer tasks and harmful jailbreak tasks. A likely explanation is Mistral-7B's proficiency in understanding complex semantic nuances, which may make it more susceptible to manipulation in role-playing scenarios commonly used in jailbreaks.

## 4 Conclusion

In this work, we propose an Optimal Transport (OT)-based stability criterion for large language models (LLMs) that addresses both data corruptions and sub-population shifts within a unified framework. By integrating these two forms of distributional changes, the proposed criterion offers a comprehensive assessment of LLM stability, enabling a more robust evaluation of model performance under realistic and diverse conditions. Future work could expand on this foundation by exploring additional tasks across varied domains and conducting larger-scale experiments to further validate the efficacy of the criterion. Investigating the behavior of LLMs under more extreme distribution shifts or in low-resource settings may also provide deeper insights. Moreover, incorporating advanced techniques for adaptive learning or domain adaptation could further enhance the model's stability across diverse environments.

## References


[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[2] Jose Blanchet, Daniel Kuhn, Jiajin Li, and Bahar Taskesen. Unifying distributionally robust optimization via optimal transport theory. *arXiv preprint arXiv:2308.05414*, 2023.

[3] Patrick Chao, Alexander Robey, Edgar Dobriban, Hamed Hassani, George J. Pappas, and Eric Wong. Jailbreaking black box large language models in twenty queries. *CoRR*, abs/2310.08419, 2023.

[4] Ge Gao, Alexey Taymanov, Eduardo Salinas, Paul Mineiro, and Dipendra Misra. Aligning llm agents by learning latent preference from user edits. *arXiv preprint arXiv:2404.15269*, 2024.

[5] Suyash Gupta and Dominik Rothenhaeusler. The s-value: evaluating stability with respect to distributional shifts. In *Advances in Neural Information Processing Systems 37*, 2023.

[6] Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. Mistral 7b. *arXiv preprint arXiv:2310.06825*, 2023.

[7] Hanlei Jin, Yang Zhang, Dan Meng, Jun Wang, and Jinghua Tan. A comprehensive survey on process-oriented automatic text summarization with exploration of llm-based methods. *arXiv preprint arXiv:2403.02901*, 2024.

[8] Anay Mehrotra, Manolis Zampetakis, Paul Kassianik, Blaine Nelson, Hyrum Anderson, Yaron Singer, and Amin Karbasi. Tree of attacks: Jailbreaking black-box llms automatically. *CoRR*, abs/2312.02119, 2023.

[9] Hongseok Namkoong, Yuanzhe Ma, and Peter W Glynn. Minimax optimal estimation of stability under distribution shift. *arXiv preprint arXiv:2212.06338*, 2022.

[10] Ansong Ni, Srini Iyer, Dragomir Radev, Veselin Stoyanov, Wen-tau Yih, Sida Wang, and Xi Victoria Lin. Lever: Learning to verify language-to-code generation with execution. In *International Conference on Machine Learning*, pages 26106–26128. PMLR, 2023.

[11] Yonadav Shavit, Sandhini Agarwal, Miles Brundage, Steven Adler, Cullen O'Keefe, Rosie Campbell, Teddy Lee, Pamela Mishkin, Tyna Eloundou, Alan Hickey, et al. Practices for governing agentic ai systems. *Research Paper, OpenAI, December*, 2023.

[12] Xinyue Shen, Zeyuan Chen, Michael Backes, Yun Shen, and Yang Zhang. "Do Anything Now": Characterizing and Evaluating In-The-Wild Jailbreak Prompts on Large Language Models. In *ACM SIGSAC Conference on Computer and Communications Security (CCS)*. ACM, 2024.

[13] Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Stanford alpaca: An instruction-following llama model. https://github.com/tatsu-lab/stanford_alpaca, 2023.

[14] Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

[15] Laslo Welz and Carsten Lanquillon. Enhancing large language models through external domain knowledge. In *International Conference on Human-Computer Interaction*, pages 135–146. Springer, 2024.

[16] Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Shaokun Zhang, Erkang Zhu, Beibin Li, Li Jiang, Xiaoyun Zhang, and Chi Wang. Autogen: Enabling next-gen llm applications via multi-agent conversation framework. *arXiv preprint arXiv:2308.08155*, 2023.

[17] An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, et al. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*, 2024.

[18] Yuchen Zhuang, Yue Yu, Kuan Wang, Haotian Sun, and Chao Zhang. Toolqa: A dataset for llm question answering with external tools. *Advances in Neural Information Processing Systems*, 36:50117–50143, 2023.

## A Proof of Proposition 2.1

*Proof.* Now, we are trying to calculate the surrogate function with our proposed cost function $c$ in (1). Denote the distance function $\left(\frac{\Phi(x)^T\Phi(\hat{x})}{\|\Phi(x)\|\|\Phi(\hat{x})\|}\cdot\max(\frac{\#\text{Token}(x)}{\#\text{Token}(\hat{x})},\frac{\#\text{Token}(\hat{x})}{\#\text{Token}(x)})\right)$ as $d(\cdot,\cdot)$. Then, we have

$$\begin{aligned}\tilde{\ell}_c^{\alpha,h}(\beta,(\hat{z},\hat{w})) &= \min_{(z,w)\in\mathcal{Z}\times\mathcal{W}} \theta_1\cdot w\cdot d(z,\hat{z})+\theta_2(\phi(w)-\phi(\hat{w}))_+ - \alpha w - h\cdot w\cdot\ell(\beta,z)\\ &= \min_{z\in\mathcal{Z}}\theta_2\cdot\min_{w\in\mathbb{R}} -w\frac{h\cdot\ell(\beta,z)-\theta_1\cdot d(z,\hat{z})+\alpha}{\theta_2}+\phi(w)+\mathbb{I}_{\mathcal{W}}(w)\\ &= \min_{z\in\mathcal{Z}} -\theta_2\cdot(\phi+\mathbb{I}_{\mathcal{W}})^*\left(\frac{h\cdot\ell(\beta,z)-\theta_1\cdot d(z,\hat{z})+\alpha}{\theta_2}\right).\end{aligned}$$

where the first equality follows as $\hat{W}=1$ almost surely and $\phi(1)=0$, and the second equality holds due to the definition of conjugate functions.

When $\mathcal{W}=\mathbb{R}_+$ and $\phi(t)=t\log t-t+1$, we know its conjugate function $(\phi+\mathbb{I}_{\mathbb{R}_+})^*=\exp(t)-1$. Consequently, we obtain the following:

$$\begin{aligned}\tilde{\ell}_c^{\alpha,h}(\beta,(\hat{z},\hat{w})) &= \min_{z\in\mathcal{Z}} -\theta_2\cdot\exp\left(\frac{h\cdot\ell(\beta,z)-\theta_1\cdot d(z,\hat{z})+\alpha}{\theta_2}\right)+\theta_2\\ &= -\theta_2\cdot\exp\left(\frac{\max_{z\in\mathcal{Z}} h\cdot\ell(\beta,z)-\theta_1\cdot d(z,\hat{z})+\alpha}{\theta_2}\right)+\theta_2\\ &= -\theta_2\cdot\exp\left(\frac{\ell_{h,\theta_1}(\hat{z})+\alpha}{\theta_2}\right)+\theta_2.\end{aligned}$$

where the second equality follows from the fact the function $\exp(\cdot)$ is monotonically increasing. Hence, we can reformulate the dual problem as

$$\mathfrak{R}(\beta,r)=\sup_{h\in\mathbb{R}_+,\alpha\in\mathbb{R}} hr+\alpha-\theta_2\mathbb{E}_{\mathbb{P}_0}\left[\exp\left(\frac{\ell_{h,\theta_1}(\hat{Z})+\alpha}{\theta_2}\right)\right]+\theta_2. \tag{4}$$

Next, we will solve the supremum problem via $\alpha$ and the first-order condition reads

$$1-\exp\left(\frac{\alpha}{\theta_2}\right)\mathbb{E}_{\mathbb{P}_0}\left[\exp\left(\frac{\ell_{h,\theta_1}(\hat{Z})}{\theta_2}\right)\right]=0$$

and $\alpha^\star=-\theta_2\log\left(\mathbb{E}_{\mathbb{P}_0}\left[\frac{\ell_{h,\theta_1}(\hat{Z})}{\theta_2}\right]\right)$. Put all of them together, we get

$$\mathfrak{R}(\beta,r)=\sup_{h\in\mathbb{R}_+} hr-\theta_2\log\left(\mathbb{E}_{\mathbb{P}_0}\left[\exp\left(\frac{\ell_{h,\theta_1}(\hat{Z})}{\theta_2}\right)\right]\right).$$

□