Breach in the Shield: Unveiling the Vulnerabilities of Large Language Models

Runpeng Dai[1*] **Run Yang**[2*] **Fan Zhou**[3] **Hongtu Zhu**[1†]
[1]University of North Carolina at Chapel Hill [2]BiliBili
[3]Shanghai University of Finance and Economics
{runpeng, htzhu}@email.unc.edu
yangrun@bilibili.com
zhoufan@mail.shufe.edu.cn

Abstract

Large Language Models and Vision-Language Models have achieved impressive performance across a wide range of tasks, yet they remain vulnerable to carefully crafted perturbations. In this study, we seek to pinpoint the sources of this fragility by identifying parameters and input dimensions (pixels or token embeddings) that are susceptible to such perturbations. To this end, we propose a stability measure called **FI**, **F**irst order local **I**nfluence, which is rooted in information geometry and quantifies the sensitivity of individual parameter and input dimensions. Our extensive analysis across LLMs and VLMs (from 1.5B to 13B parameters) reveals that: (I) A small subset of parameters or input dimensions with high FI values disproportionately contribute to model brittleness. (II) Mitigating the influence of these vulnerable parameters during model merging leads to improved performance.

1 Introduction

Large Language Models (LLMs) and Vision Language Models (VLMs) have revolutionized the field of Natural Language Processing (NLP), exhibiting remarkable proficiency across a variety of tasks (Gong et al., 2024; Zheng et al., 2025b; Luo et al., 2025) and modalities (Liu et al., 2025a,b; Zheng et al., 2025a; He et al., 2025). These modern LLMs are massive in size, trained on vast amounts of data, and meticulously aligned to prevent generating harmful content (Perez et al., 2022; Zhou et al., 2025), leaking private information (Zhang et al., 2024), or exhibiting sexual or religious bias (Xie and Lukasiewicz, 2023).

Despite the enthusiasm for these integrative approaches, a critical issue remains: LLMs remain susceptible to both external and internal perturbations, affecting their reliability and performance.

Externally, LLMs are vulnerable to input perturbations, such as Embedding-Corrupted Prompts (Fort, 2023; Liu et al., 2024a). This susceptibility extends to visual inputs in VLMs, where adversarially optimized images can drastically alter model behavior (Qi et al., 2024). Beyond adversarial attacks, VLMs exhibit high sensitivity to perturbations in specific local regions of an image—a common issue, as user-uploaded images often suffer from blurring, masking, or low resolution. The vulnerability is highlighted in our case study of the Qwen-VL model. As depicted in Figure 1, masking the ten most sensitive pixels, which are unrelated to the question, resulted in incorrect model outputs.

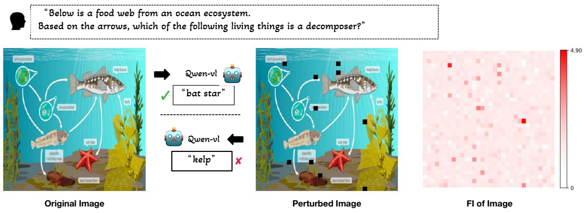

"Below is a food web from an ocean ecosystem.
Based on the arrows, which of the following living things is a decomposer?"
Qwen-vl
✓ "bat star"
Qwen-vl
"kelp" ✗
4.90
0
Original Image
Perturbed Image
FI of Image

Figure 1: A case study of the Qwen-VL model (Bai et al., 2023) on SCI-QA. The image on the far right visualizes the per-pixel FI values. Masking just 10 pixels with the highest FI values leads to a failure in producing the correct answer.

Internally, LLM stability is further challenged by parameter perturbations, often introduced through model merging and quantization. While these techniques improve deployment efficiency by reducing inference costs (Frantar and Alistarh, 2023; Ashkboos et al., 2024), they can also induce hallucinations and degrade performance (Men et al., 2024; Yu et al., 2024; Li et al., 2024). However, our findings reveal that parameter susceptibility varies significantly. As Figure 2 illustrates, randomly dropping 5% of parameters has a minimal impact on performance. In contrast, zeroing out just 1% of the parameters identified by our measure can drastically reduce accuracy, even below random guessing levels.

To pinpoint the sources of this fragility, we pro-

Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics
Volume 1: Long Papers, pages 3509–3521
March 24-29, 2026 ©2026 Association for Computational Linguistics

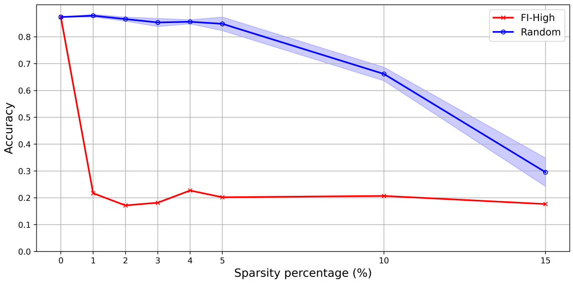


Figure 2: A case study of Qwen2.5 on MMLU-Geography. “FI-High” refers to zeroing out parameters with the highest FI values, whereas “Random” denotes random parameter removal.

pose a novel stability measure called **FI**, **F**irst order local **I**nfluence, to quantitatively assess the stability of LLMs against perturbations. Specifically, we construct a perturbation manifold that encompasses all perturbed models, along with its associated geometric properties. Our stability measure quantifies the degree of local influence of a perturbation on a given objective function within this manifold, thereby reflecting the stability of individual LLM components. We summarize the **advantages** of **FI** as follows:

1. **FI’s versatility** allows for effective stability assessment under both external and internal perturbations across various granularities—from individual parameters to input features like pixels and patches.

2. **FI effectively identifies vulnerabilities**. Our extensive studies validate its effectiveness in pinpointing fragile pixels in VLM vision inputs, vulnerable embedding dimensions of tokens in LLMs (Section 4), and salient model parameters (Subsection 5.1).

3. **FI offers insights into improving model robustness**. We further illustrate that understanding these vulnerabilities can lead to enhanced model resistance to perturbations. By focusing on model merging as an example, we show that safeguarding key parameters identified by high FI values can substantially reduce performance degradation during the merging process (Subsection 5.2).

2 Related Work

Recent efforts to evaluate LLM stability typically adopt a coarse-grained approach, aiming to assess the overall robustness of models under various perturbations. One line of work investigates how stability is influenced by sampling parameters, such as temperature, which affect output variability during generation (Atil et al., 2024; Ouyang et al., 2025). Another direction studies model sensitivity to input or parameter perturbations. For instance, Blanchet et al. (2024) analyze input-level robustness using optimal transport to quantify a model’s response to distributional shifts in prompts. On the other hand, Peng et al. (2024) focus on parameter-space perturbations, demonstrating that LLMs remain robust to weight changes up to a certain threshold, beyond which performance significantly degrades. They estimate a model’s robustness tolerance by injecting random perturbations into model weights and evaluating the performance drop.

Despite these contributions, fine-grained analyses—such as those examining the effect of individual input tokens, pixels, or specific model parameters—remain underexplored. Wei et al. (2024) take a step in this direction by leveraging pruning-based techniques, including SNIP (Lee et al., 2018) and Wanda (Sun et al., 2023), to identify critical neurons and low-rank structures that impact model safety and utility. However, there is still a lack of unified metrics or frameworks that assess stability with respect to both input- and parameter-level perturbations. Moreover, the downstream applications of such stability assessments remain largely unexamined.

3 Stability Measure of Large Language Models

In this section, we propose a new metric called FI to quantify the stability of large language models against local perturbations. Considering the autoregressive nature of LLMs, we first develop FI for single-step generation and discuss its theoretical and computational properties in detail. We then show how FI can be naturally extended to sequence generation tasks. Finally, we compare FI to existing stability measures, highlighting its unique advantages.

3.1 FI Metric

Problem formulation. Consider an LLM parameterized by θ, with input data x, which may consist of text or, for visual language models, a combination of text and images. Given x, the model generates a probability distribution over its vocabulary to predict the next token, which can be framed as a classification problem with K classes, where K represents the vocabulary size.

However, vocabulary sizes are typically large (Bai et al., 2023; Dubey et al., 2024), and predictions are often concentrated on a small subset of tokens. Instead of using the entire vocabulary, it is more efficient to focus on a relevant subset based on the task. For example, in multiple-choice questions, probabilities are restricted to the choices “A”, “B”, “C”, or “D”. Classes can also be defined semantically, such as categorizing tokens as “neutral” or “notorious” in toxicity detection (Gehman et al., 2020).

With appropriately defined classes, the predicted probability for class $y \in \{1, \ldots, K\}$ is denoted as $P(y|x, \theta)$, satisfying $\sum_{y=1}^{K} P(y|x, \theta) = 1$. Let $\omega \in \mathbb{R}^d$ be a perturbation vector varies in an open subset Ω. ω can be applied to a subset of the model parameters θ and locations within the input data x. We denote the output of the perturbed model under this perturbation as $P(y|x, \theta, \omega)$.

Perturbation Manifold and FI Since our primary interest lies in examining the behavior of $P(y|x, \theta, \omega)$ as a function of ω near $\omega_0 = 0$, we shift focus from θ to ω. We introduce the perturbation manifold as defined in (Zhu et al., 2007) and (Zhu et al., 2011).

Definition 3.1. *Define the d-dimensional perturbation manifold $\mathcal{M} = \{P(y|x, \theta, \omega) : \omega \in \Omega\}$, which encompasses all perturbed models. Assume that for all $\omega \in \Omega$, the perturbed models $\{P(y = i|x, \theta, \omega)\}_{i=1}^{K}$ are positive and sufficiently smooth. The tangent space T_ω of $\mathcal{M}$ at ω is spanned by the partial derivatives of the log-likelihood function $\ell(\omega|y, x, \theta) = \log P(y|x, \theta, \omega)$ with respect to ω, specifically $T_\omega = span\{\frac{\partial}{\partial \omega_i} \ell(\omega|y, x, \theta)\}_{i=1}^{d}$.*

The metric g_ω on $\mathcal{M}$ can be defined with the metric tensor G_ω. Consider two tangent vectors at ω given by $v_j(\omega) = h_j^\top \partial_\omega \ell(\omega|y, x, \theta) \in T_\omega$, where h_1 and h_2 are the weights on the basis. Their inner product is defined as:

$$\langle v_1(\omega), v_2(\omega) \rangle_{g_\omega} = \sum_{y=1}^{K} v_1(\omega) v_2(\omega) P(y|x, \theta, \omega).$$

The metric tensor G_ω is given by:

$$\sum_{y=1}^{K} \partial_\omega \ell(\omega|y, x, \theta) \partial_\omega^\top \ell(\omega|y, x, \theta) P(y|x, \theta, \omega).$$

Subsequently, the norm of $v_j(\omega)$ under metric g_ω is $\|v_j\|_{g_\omega} = \sqrt{h_j^\top G_\omega h_j}$. Let $C(t) = P(y|x, \theta, \omega(t))$ be a smooth curve on the manifold $\mathcal{M}$ connecting two points $\omega_1 = \omega(t_1)$ and $\omega_2 = \omega(t_2)$. Then, the distance between ω_1 and ω_2 along the curve $C(t)$ is given by:

$$\begin{aligned} & S_C(\omega_1, \omega_2) \\ &= \int_{t_1}^{t_2} \sqrt{\|\partial_t \log P(y|x, \theta, \omega(t))\|_{g_\omega}}\, dt \\ &= \int_{t_1}^{t_2} \sqrt{\frac{d\omega(t)^T}{dt} G_{\omega(t)} \frac{d\omega(t)}{dt}}\, dt. \end{aligned}$$

With the Perturbation manifold $\mathcal{M}$ and respective metric g_ω defined, we are ready to propose the metric that quantifies the stability of large language models (LLMs) against various types of local perturbations. Let $f(\omega)$ be the objective function of interest for sensitivity analysis, in our case being $-\log P(y_{pred}|x, \theta, \omega)$, we can define the following (first-order) local influence metric FI:

Definition 3.2. *Given the perturbation manifold $\mathcal{M}$ and its metric, the first-order local stability measure of $f(\omega)$ at $\omega(0) = \omega_0$ is defined as*

$$\mathbf{FI}_\omega(\omega_0) = \max_{C} \lim_{t \to 0} \frac{[f(\omega(t)) - f(\omega(0))]^2}{S_C^2(\omega(t), \omega(0))}. \quad (1)$$

The ratio in Equation 1 measures the amount of change introduced to the objective function relative to the distance of the perturbation on the perturbation manifold. Thus, Equation 1 can be naturally interpreted as the maximum local ratio of change among all possible perturbation curves $C(t)$.

Computation of FI. As we will show, Theorem A.1 in Appendix A.3 regarding diffeomorphic reparameterization invariance enables us to derive an easy-to-compute solution for Equation 1, while addressing the low-dimensionality problem inherent in LLMs.

Theorem 3.3. *If G_ω is positive definite, the **FI** measure has the following closed-form:*

$$\mathbf{FI}_\omega(\omega_0) = \nabla_{f(\omega_0)}^T G_{\omega_0}^{-1} \nabla_{f(\omega_0)}, \quad (2)$$

where

$$\nabla_{f(\omega_0)} = \frac{\partial f(\omega)}{\partial \omega}\Big|_{\omega=\omega_0}.$$

The detailed proof of Theorem 3.3 can be found in Appendix A.6. It is important to note that the closed form of FI in Theorem 3.3 depends on the positive definiteness of G_ω, which is not always guaranteed. This is due to the fact that the parameters in LLMs are often high-dimensional tensors with low-rank structures (Kaushal et al., 2023).

We apply the invariance result of Theorem A.1 in Appendix A.3 by transforming ω to a vector ν such that $G_\nu = \mathbf{I}_K$, where K is an integer. Specifically, we notice that $G_{\omega_0} = B_0^T B_0$, where

$$B_0 = \left[P(y=i|x,\theta,\omega)^{1/2}\partial_\omega \ell(\omega|y=i,x,\theta)\right]_{i\leqslant K}.$$

Let $r_0 = \mathrm{rank}(G_{\omega_0})$, we apply the compact SVD to $B_0 \in \mathbb{R}^{p\times K}$, which yields $B_0 = V_0\Lambda_0 U_0$, where $V_0 \in \mathbb{R}^{p\times r_0}$ and $U_0 \in \mathbb{R}^{r_0\times K}$ are semi-orthogonal matrices and $\Lambda_0 \in \mathbb{R}^{r_0\times r_0}$ is a diagonal matrix. Under the transformation $\nu = \Lambda_0 V_0^T \omega$, we have $\mathbf{FI}_\omega(\omega_0) = \mathbf{FI}_\nu(\nu_0)$, which can be expressed as

$$\nabla_{f(\omega_0)}^\top (V_0 R_0)^\top \Lambda_0^{-2} (V_0 R_0) \nabla_{f(\omega_0)},$$

where the equality holds by applying the chain rule to G_ν.

FI for sequence generation. Sequence generation is essentially multiple rounds of next-token generation, where the l-th token $y^{(l)}$ is generated given the initial input z and previously generated tokens $\boldsymbol{y}^{(l)} = \{y^{(1)}, \ldots, y^{(l-1)}\}$. We define the FI measure for generating the l-th token $y^{(l)}$ given the initial input z by averaging out the randomness from the preceding steps $\mathbf{FI}_l(z) = \mathbb{E}_{\boldsymbol{y}^{(l)}}[\mathbf{FI}(\{z, \boldsymbol{y}^{(l)}\}, \theta, \omega)|z]$.

To formulate an overall measure for sequence generation, we aggregate these per-token FI measures. Since sequences generated by LLMs can vary in length, we propose two methods to handle this heterogeneity. The first approach sets a fixed horizon L and computes the mean FI over these rounds

$$\mathbf{FI}_{\text{seq}}^L(z) = \frac{1}{L}\sum_{l=1}^{L} \mathbf{FI}_l(z). \tag{3}$$

Alternatively, inspired by the concept of average discounted rewards in reinforcement learning (Liu et al., 2018), we consider sequences of potentially infinite length and propose a discounted FI measure with discount factor γ

$$\mathbf{FI}_{\text{seq}}^{\infty,\gamma}(z) = (1-\gamma)\sum_{l=0}^{\infty} \gamma^l \cdot \mathbf{FI}_l(z).$$

By taking the expectation over the distribution of z, we obtain the average FI for sequence generation in both cases $\mathbb{E}_{P_z}[\mathbf{FI}_{\text{seq}}^L(z)]$ and $\mathbb{E}_{P_z}[\mathbf{FI}_{\text{seq}}^{\infty,\gamma}(z)]$, respectively.

3.2 Other Measures & Discussion

We note that several alternative methods can also serve as stability measures for LLMs. We provide their explicit formulations and compare them with FI.

Jacobian Norm (Novak et al., 2018):

$$\|\partial_\omega f(y_{pred}, \omega)\|_2$$

SNIP (Lee et al., 2018):

$$\|\omega \odot \partial_\omega f(y_{pred}, \omega)\|_2$$

Both measures focuses solely on y_{pred}, while neglecting the probabilities assigned to other choices. For example, consider two output distributions: (0.9, 0.05, 0.05, 0.02) and (0.3, 0.25, 0.25, 0.2). In both cases, the model selects option A. However, the second distribution is more unstable, as a small perturbation in the probabilities could lead to a different prediction. In contrast, FI measure accounts for both the probability and gradient across all possible choices.

Saliency map (Simonyan et al., 2013):

$$\begin{cases} 0, & \text{if } \dfrac{\partial f(y_{\text{pred}}, \omega)}{\partial \omega} < 0\text{or} \displaystyle\sum_{y \neq y_{\text{pred}}} \dfrac{\partial f(y, \omega)}{\partial \omega} > 0 \\ -\dfrac{\partial f(y_{\text{pred}}, \omega)}{\partial \omega} \displaystyle\sum_{y \neq y_{\text{pred}}} \dfrac{\partial f(y, \omega)}{\partial \omega}, & \text{otherwise} \end{cases}$$

Saliency maps consider the gradients with respect to all possible choices. However, they lose significant information by zeroing out many of these gradients.

To this end, we highlight the unique advantages of FI. **Effectiveness:** A quantitative comparison of these measures is provided in Section 4 and 5, while their computational complexities are discussed in Appendix A.2. **Theoretical rigor:** In particular, only FI possesses a reparameterization invariance property (see Appendix A.3), which further distinguishes it by enhancing interpretability.

4 External Perturbations Analysis

In this section, we first demonstrate the effectiveness of FI in identifying vulnerable locations in both vision and language inputs through guided attack. Then, we conclude the section with a finding from cross-modal analysis.

Identify Fragile Pixels We conduct the attack process on the MMbench dataset (Liu et al., 2024b), a comprehensive benchmark designed to evaluate various multimodal capabilities of VLMs. For a fair comparison, we identify the top 10 pixels using different stability measures and assess the model's performance after masking out the corresponding pixels.

Model	Method	Action Recognition	Attribute Recognition	Celebrity Recognition	Function Reasoning
Qwen VL	FI (Ours)	**0.320**	**0.402**	**0.673**	**0.411**
	Jacobian	0.668	0.587	0.906	0.604
	Saliency	0.782	0.525	0.873	0.639
	Random	0.812	0.550	0.881	0.683
	Original	0.814	0.549	0.882	0.686
Qwen2.5 VL-3B	FI (Ours)	**0.720**	**0.735**	**0.780**	**0.723**
	Jacobian	0.731	0.752	0.797	0.755
	Saliency	0.745	0.761	0.797	0.774
	Random	0.882	0.931	0.957	0.928
	Original	0.890	0.946	0.959	0.930
Qwen2.5 VL-7B	FI (Ours)	**0.768**	**0.750**	**0.796**	**0.723**
	Jacobian	0.778	0.768	0.815	0.755
	Saliency	0.792	0.777	0.815	0.774
	Random	0.891	0.944	0.951	0.925
	Original	0.890	0.946	0.959	0.930

Table 1: Accuracy on the MMBench dataset after masking out top ten pixels in images identified by different measures.

Identify Vulnerable Embedding Dimensions We conduct attack on pure-text LLMs to verify the effectiveness of our approach in identifying vulnerable embedding dimensions. Specifically, we follow the token embedding attack methods proposed in (Liu et al., 2024a) and (Fort, 2023).

More concretely, we compute the stability measure for each embedding dimension and select the top 0.1% most sensitive dimensions (ω) as identified by the metrics. We then apply a gradient-based attack strategy following (Fort, 2023), perturbing the selected dimensions in the direction of $-\nabla_{\omega} \log P(y_{\text{pred}} \mid x, \theta)$.

From both Table 1 and Table 2, we observe the following: (I) Stability measures are effective in identifying vulnerable input dimensions (i.e., pixels in images and dimensions in embeddings). Notably, LLMs are generally robust to random perturbations and such perturbations rarely lead to significant performance degradation. In contrast, perturbations guided by stability measures consistently result in substantial drops in performance. (II) Among all the stability measures evaluated, FI proves to be the most effective: masking pixels or perturbing dimensions identified by FI leads to the largest observed decline in performance.

Model	Method	Business	Geo	Culture	Law	Average
Pythia 1B	Saliency	0.278	0.272	0.210	0.243	0.251
	Jacobian	0.273	0.264	0.201	0.241	0.245
	Random	0.301	0.368	0.237	0.246	0.288
	FI (ours)	**0.270**	**0.261**	**0.195**	**0.236**	**0.240**
	SNIP	0.297	0.281	0.226	0.242	0.262
	Original	0.303	0.370	0.240	0.247	0.290
Qwen2.5 3B	Saliency	0.677	0.637	0.632	0.560	0.627
	Jacobian	0.665	0.641	0.625	0.560	0.623
	Random	0.805	0.781	0.781	0.672	0.760
	FI (ours)	**0.656**	**0.620**	**0.610**	**0.547**	**0.608**
	SNIP	0.783	0.663	0.665	0.563	0.669
	Original	0.810	0.800	0.785	0.673	0.767
Qwen2.5 7B	Saliency	0.756	0.789	0.709	0.725	0.745
	Jacobian	0.764	0.782	0.717	0.720	0.746
	Random	0.852	0.884	0.802	0.735	0.818
	FI (ours)	**0.748**	**0.780**	**0.705**	**0.713**	**0.737**
	SNIP	0.757	0.791	0.710	0.727	0.746
	Original	0.856	0.890	0.810	0.737	0.823

Table 2: Comparison of accuracy in the MMLU dataset after perturbing the same number of dimensions in the embedding space identified using different measures.

Effect of Prompting on Pixel Vulnerability While the significant impact of prompt design on VLM performance is well-recognized (Zhou et al., 2022), and carefully crafted prompts are known to even jailbreak these models (Shayegani et al., 2023; Zhou et al., 2024), a quantitative analysis of this cross-modal influence – specifically, how prompting affects the processing and stability of visual input – remains largely unexplored.

Our study aims to bridge this gap by investigating how varying prompt instructions influence the sensitivity of VLMs to visual perturbations. Specifically, we examine two types of prompts:

- Aggressive Prompts: Designed to encourage the model to consider every detail in the image, potentially increasing sensitivity to noise.
- Safe Prompts: Intended to focus the model on salient entities and relationships, potentially enhancing robustness by ignoring irrelevant details.

We computed the FI value for each pixel and visualized the resulting distributions under different prompt settings, as illustrated in Figure 3. Our main findings are as follows:

(I) Prompt choice has a substantial impact on the stability of individual pixels within the image. As shown on the left of Figure 3, aggressive prompts shift the FI distribution toward higher values, resulting in a marked increase in both the mean and maximum FI values. This suggests that the model becomes more sensitive to pixel-level perturbations throughout the image. In contrast, safe prompts significantly shift the FI distribution

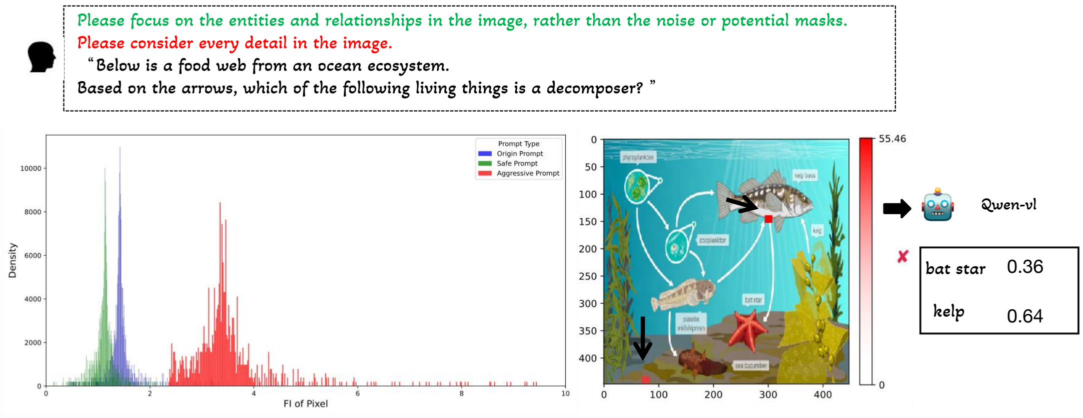


Figure 3: A case study utilizing FI for cross-modal analysis. In the same example, the bottom-left image shows how Aggressive and Safe prompts affect the FI distribution on the image.

toward lower values, indicating reduced sensitivity and improved stability against perturbations in less relevant regions.

(II) Vulnerability remains even with careful prompt design. Although safe prompts generally reduce FI values, they do not fully guarantee model stability, as outliers with large FI values persist. As shown in the right column of Figure 3, even when applying the safe prompt, masking out the two pixels with the highest FI values still leads to incorrect model predictions. This result underscores the persistent challenge of achieving robustness in VLMs and demonstrates the effectiveness of the FI measure for identifying vulnerable regions.

Our findings contribute to the growing body of literature on cross-modal interactions in VLMs, offering a stability-centric perspective that complements existing behavioral and attributional analyses. Importantly, this framework can inform the development of more robust multimodal systems and prompt design strategies for safety-critical applications.

5 Internal Perturbations Analysis

In this section, we first conduct a parameter sparsification experiment to demonstrate the effectiveness of the FI. We then apply the FI measure to mitigate parameter interference during model merging, showcasing its potential for guiding LLM improvement.

5.1 Parameter Sparsification

We conduct experiments on multiple-choice problems from MMLU (Hendrycks et al., 2020) and sequence generation tasks from Alpaca-Eval (Dubois et al., 2024) to examine how these perturbations impact two key capabilities of large models: knowledge retention and instruction-following. Details of both experimental setups are provided in Appendix A.

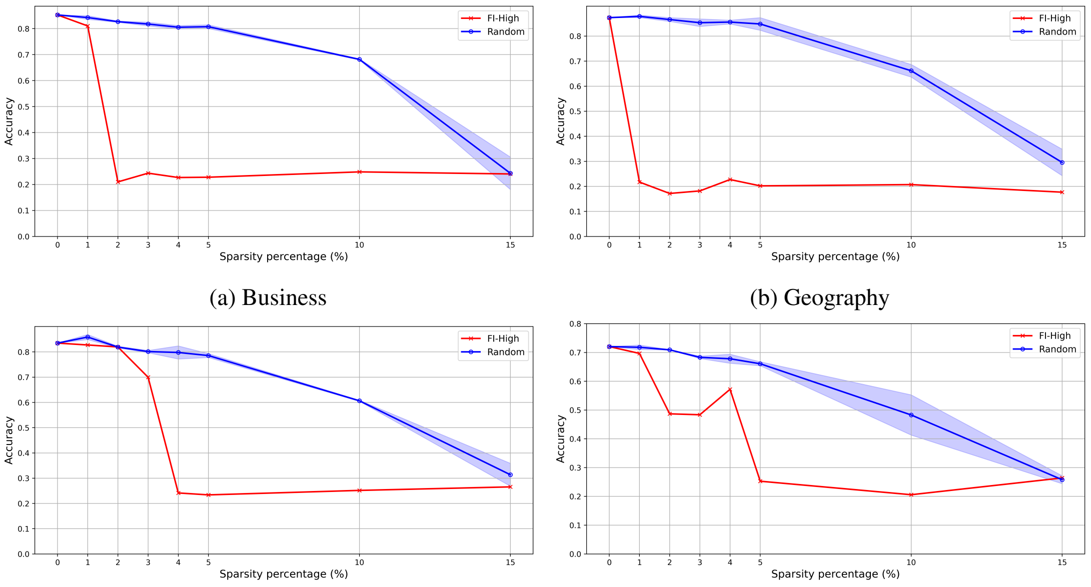
As shown in Figure 2 and 4, sparsifying (zeroing out) just 2–3% of the high-FI parameters significantly degrades the model's knowledge capacity, leading to catastrophic forgetting and hallucinations, with performance dropping by up to 75%. A similar trend is observed in Table 6 at around the 10% sparsity level and Table 7 at around the 15% sparsity level. In contrast, models remain relatively robust against random sparsification, often exhibiting nearly identical behavior even after 5% sparsification.

These findings demonstrate FI's effectiveness in identifying fragile parameters and further support the inherent structure within the parameter matrix, aligning with recent observations on model brittleness (Ma et al., 2023; Wei et al., 2024; Yu et al., 2024).

5.2 FI-Guided Parameter Protection in Model Merging

Model merging is a technique for acquiring domain-specific knowledge by combining models from different domains, thereby reducing the computational cost of additional fine-tuning (see (Yang et al., 2024c) for a review). However, a persistent challenge is that merging parameters introduces perturbations that can hinder a model's ability to retain previously learned information. To address this, we use FI to identify parameters susceptible to forgetting and exclude them from the merging process.

We demonstrate that FI can be seamlessly inte-



(a) Business

(b) Geography

(c) Culture

(d) Law

Figure 4: Performance of Qwen2-7B on the MMLU dataset under varying levels of parameter sparsification. "FI-High" denotes sparsifying parameters with the highest FI values, while "Random" refers to random parameter sparsification.

grated into mainstream model merging methods, including **Average Merging** (Wortsman et al., 2022), **Task Arithmetic** (Ilharco et al., 2022), and **TIES** (Yadav et al., 2024). Additionally, we include **DARE** (Yu et al., 2024) as a competing baseline for completeness.

We consider merging two models, A and B, both fine-tuned from the same base model. Let θ_A, θ_B, and θ_{Base} denote the parameters of models A, B, and the base model, respectively. We first introduce the merging methods and then demonstrate how FI can be integrated into these methods to mitigate perturbation effects.

Average Merging Average merging obtains the merged model by averaging θ_A and θ_B, resulting in parameters $\theta_{\text{Avg}} = \frac{\theta_A+\theta_B}{2}$.

Task Arithmetic Task arithmetic constructs "task vectors" by subtracting a base model from each task-specific model and then merges these vectors linearly before adding back the base model $\theta_{\text{Task}} = \theta_{\text{Base}} + \gamma(\delta_A + \delta_B)$, where $\delta_A = \theta_A - \theta_{\text{Base}}$ and similarly for δ_B.

Both Average Merging and Task Arithmetic modify all parameters in models A and B, potentially degrading performance by disturbing their most sensitive parameters. To address this, we employ a protection strategy that preserves these vulnerable parameters while merging only the less critical ones. Specifically, we identify the top $k\%$ of high-FI parameters in both models and record their locations in Θ_A and Θ_B. Then, for each layer in both θ_{Task} and θ_{Avg}, we revert parameters at locations in $\Theta_A \cap \Theta_B^{\complement}$ to their original values from θ_A, and parameters at locations in $\Theta_B \cap \Theta_A^{\complement}$ to their original values from θ_B.

TIES (**Tr**I**m**, **E**lect **Sign**) operates in two steps. First, it sets a fraction of the "task vectors" δ_A and δ_B to zero. Then, for each remaining entry, it retains the weight from the vector with the larger absolute value.

FI-guided protection can be incorporated into both steps. In the first step, we protect δ_A at locations Θ_A and δ_B at Θ_B from being trimmed. In the second step, entries within Θ_A are preserved as δ_A, while those in Θ_B remain as δ_B, regardless of their absolute values.

We merged **Qwen2.5-Math-7B** (Yang et al., 2024a) and **HuatuoGPT-o1-7B** (Chen et al., 2024), as both models are fine-tuned from **Qwen2.5-7B** (Yang et al., 2024b). We evaluate the performance of the merged models on math and health subjects within the MMLU benchmark (Hendrycks et al., 2020).

From Table 3, we observe the following: (1) FI-guided protection generally enhances the performance of the merged models in both domains. For example, the Average model merging method with FI-guided protection yields approximately a

1% improvement in both the Math and Health domains. (2) Furthermore, TIES with FI protection applied in its first stage performs the best among all merging methods.

	FI-protect	Math	Health	Mean
Qwen2.5 Math-7B	/	0.616	/	/
Huatuo o1-7B	/	/	0.724	/
Average	Without	0.534 (-8.2%)	0.514 (-21.0%)	0.524
	With	0.543 (-7.3%)	0.522 (-20.2%)	0.533
Task	Without	<u>0.577</u> (-3.9%)	0.597 (-12.7%)	<u>0.587</u>
	With	0.573 (-4.3%)	<u>0.598</u> (-12.6%)	0.586
TIES	Without	0.565 (-5.1%)	0.596 (-12.8%)	0.581
	With I	**0.583** (-3.3%)	**0.606** (-11.8%)	**0.595**
	With II	0.566 (-5.0%)	0.601 (-12.3%)	0.584
DARE Task	/	0.573 (-4.3%)	0.589 (-13.5%)	0.581
DARE TIES	/	0.560 (-5.6%)	0.588 (-13.6%)	0.574

Table 3: Performance of merging Qwen2.5-Math-7B and HuatuoGPT-o1-7B. The "Mean" column reports the average accuracy across tasks. Blue and cyan percentages indicate the performance drop for the "Without" and "With" variants compared to the original model, respectively.

Figure 5 uses average merging as an example. The results indicate that as the percentage of protected parameters increases, the performance of the merged models initially improves but later declines, highlighting a trade-off in FI-guided protection. Protecting a small proportion of parameters with the highest FI helps mitigate performance degradation caused by parameter conflicts. However, a high percentage of protection may lead to forgetting issues in both domains. To determine the optimal protection percentage, we conduct a hyperparameter search on the validation set. More details can be found in Appendix D.

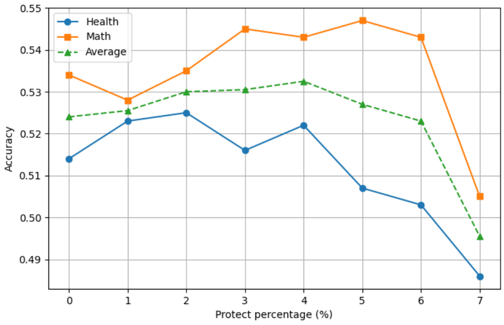


Figure 5: Accuracy of average-merged models with FI-guided protection across both domains for different protection percentages k.

6 Conclusion & Discussion

In summary, we introduce a stability measure, FI, to systematically identify the fragility of LLMs and VLMs. Through experiments under both internal and external perturbations, we demonstrate the effectiveness of our proposed method.

Our work constitutes an initial attempt to leverage sensitivity measures for improving model performance, focusing primarily on their application to model merging at the inference stage. While our study provides insights into the potential of such measures, we believe that further research is warranted to explore their utility in enhancing model training.

7 Limitations

Our method relies on gradient information and is not applicable to "black-box" models that do not expose internal parameters or gradients to users. In such cases, text-only approaches like Influence Function (Koh and Liang, 2017) are more suitable.

References

Saleh Ashkboos, Maximilian L Croci, Marcelo Gennari do Nascimento, Torsten Hoefler, and James Hensman. 2024. Slicegpt: Compress large language models by deleting rows and columns. *arXiv preprint arXiv:2401.15024*.

Berk Atil, Alexa Chittams, Liseng Fu, Ferhan Ture, Lixinyu Xu, and Breck Baldwin. 2024. Llm stability: A detailed analysis with some surprises. *arXiv preprint arXiv:2408.04667*.

Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. 2023. Qwen-vl: A frontier large vision-language model with versatile abilities. *arXiv preprint arXiv:2308.12966*.

Jose Blanchet, Peng Cui, Jiajin Li, and Jiashuo Liu. 2024. Stability evaluation via distributional perturbation analysis. *arXiv preprint arXiv:2405.03198*.

Junying Chen, Zhenyang Cai, Ke Ji, Xidong Wang, Wanlong Liu, Rongsheng Wang, Jianye Hou, and Benyou Wang. 2024. Huatuogpt-o1, towards medical complex reasoning with llms. *arXiv preprint arXiv:2412.18925*.

R Dennis Cook. 1986. Assessment of local influence. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 48(2):133–155.

Laurent Dinh, Razvan Pascanu, Samy Bengio, and Yoshua Bengio. 2017. Sharp minima can generalize for deep nets. In *International Conference on Machine Learning*, pages 1019–1028. PMLR.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Yann Dubois, Balázs Galambosi, Percy Liang, and Tatsunori B Hashimoto. 2024. Length-controlled alpacaeval: A simple way to debias automatic evaluators. *arXiv preprint arXiv:2404.04475*.

Stanislav Fort. 2023. Scaling laws for adversarial attacks on language model activations. *arXiv preprint arXiv:2312.02780*.

Elias Frantar and Dan Alistarh. 2023. Sparsegpt: Massive language models can be accurately pruned in one-shot. In *International Conference on Machine Learning*, pages 10323–10337. PMLR.

Samuel Gehman, Suchin Gururangan, Maarten Sap, Yejin Choi, and Noah A Smith. 2020. Realtoxicityprompts: Evaluating neural toxic degeneration in language models. *arXiv preprint arXiv:2009.11462*.

Xinyu Gong, Jianli Zhang, Qi Gan, Yuxi Teng, Jixin Hou, Yanjun Lyu, Zhengliang Liu, Zihao Wu, Runpeng Dai, Yusong Zou, and 1 others. 2024. Advancing microbial production through artificial intelligence-aided biology. *Biotechnology Advances*, page 108399.

Yicheng He, Chengsong Huang, Zongxia Li, Jiaxin Huang, and Yonghui Yang. 2025. Visplay: Self-evolving vision-language models from images. *arXiv preprint arXiv:2511.15661*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. 2022. Editing models with task arithmetic. *arXiv preprint arXiv:2212.04089*.

Ayush Kaushal, Tejas Vaidhya, and Irina Rish. 2023. Lord: Low rank decomposition of monolingual code llms for one-shot compression. *arXiv preprint arXiv:2309.14021*.

Pang Wei Koh and Percy Liang. 2017. Understanding black-box predictions via influence functions. In *International conference on machine learning*, pages 1885–1894. PMLR.

Namhoon Lee, Thalaiyasingam Ajanthan, and Philip HS Torr. 2018. Snip: Single-shot network pruning based on connection sensitivity. *arXiv preprint arXiv:1810.02340*.

Junyi Li, Jie Chen, Ruiyang Ren, Xiaoxue Cheng, Wayne Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. 2024. The dawn after the dark: An empirical study on factuality hallucination in large language models. *arXiv preprint arXiv:2401.03205*.

Chris Yuhao Liu, Yaxuan Wang, Jeffrey Flanigan, and Yang Liu. 2024a. Large language model unlearning via embedding-corrupted prompts. *arXiv preprint arXiv:2406.07933*.

Qiang Liu, Lihong Li, Ziyang Tang, and Dengyong Zhou. 2018. Breaking the curse of horizon: Infinite-horizon off-policy estimation. *Advances in neural information processing systems*, 31.

Rui Liu, Dian Yu, Lei Ke, Haolin Liu, Yujun Zhou, Zhenwen Liang, Haitao Mi, Pratap Tokekar, and Dong Yu. 2025a. Stable and efficient single-rollout rl for multimodal reasoning. *arXiv preprint arXiv:2512.18215*.

Rui Liu, Dian Yu, Tong Zheng, Runpeng Dai, Zongxia Li, Wenhao Yu, Zhenwen Liang, Linfeng Song, Haitao Mi, Pratap Tokekar, and 1 others. 2025b. Vogue: Guiding exploration with visual uncertainty improves multimodal reasoning. *arXiv preprint arXiv:2510.01444*.

Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, and 1 others. 2024b. Mmbench: Is your multi-modal model an all-around player? In *European conference on computer vision*, pages 216–233. Springer.

Yingfeng Luo, Tong Zheng, Yongyu Mu, Bei Li, Qinghong Zhang, Yongqi Gao, Ziqiang Xu, Peinan Feng, Xiaoqian Liu, Tong Xiao, and 1 others. 2025. Beyond decoder-only: Large language models can be good encoders for machine translation. *arXiv preprint arXiv:2503.06594*.

Xinyin Ma, Gongfan Fang, and Xinchao Wang. 2023. Llm-pruner: On the structural pruning of large language models. *Advances in neural information processing systems*, 36:21702–21720.

Xin Men, Mingyu Xu, Qingyu Zhang, Bingning Wang, Hongyu Lin, Yaojie Lu, Xianpei Han, and Weipeng Chen. 2024. Shortgpt: Layers in large language models are more redundant than you expect. *arXiv preprint arXiv:2403.03853*.

Roman Novak, Yasaman Bahri, Daniel A Abolafia, Jeffrey Pennington, and Jascha Sohl-Dickstein. 2018. Sensitivity and generalization in neural networks: an empirical study. *arXiv preprint arXiv:1802.08760*.

Shuyin Ouyang, Jie M Zhang, Mark Harman, and Meng Wang. 2025. An empirical study of the non-determinism of chatgpt in code generation. *ACM Transactions on Software Engineering and Methodology*, 34(2):1–28.

Sheng Y Peng, Pin-Yu Chen, Matthew Hull, and Duen H Chau. 2024. Navigating the safety landscape: Measuring risks in finetuning large language models. *Advances in Neural Information Processing Systems*, 37:95692–95715.

Ethan Perez, Saffron Huang, Francis Song, Trevor Cai, Roman Ring, John Aslanides, Amelia Glaese, Nat McAleese, and Geoffrey Irving. 2022. Red teaming language models with language models. *arXiv preprint arXiv:2202.03286*.

Xiangyu Qi, Kaixuan Huang, Ashwinee Panda, Peter Henderson, Mengdi Wang, and Prateek Mittal. 2024. Visual adversarial examples jailbreak aligned large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 21527–21536.

Erfan Shayegani, Yue Dong, and Nael Abu-Ghazaleh. 2023. Jailbreak in pieces: Compositional adversarial attacks on multi-modal language models. *arXiv preprint arXiv:2307.14539*.

Hai Shu and Hongtu Zhu. 2019. Sensitivity analysis of deep neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 4943–4950.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2013. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*.

Mingjie Sun, Zhuang Liu, Anna Bair, and J Zico Kolter. 2023. A simple and effective pruning approach for large language models. *arXiv preprint arXiv:2306.11695*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, and 1 others. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Boyi Wei, Kaixuan Huang, Yangsibo Huang, Tinghao Xie, Xiangyu Qi, Mengzhou Xia, Prateek Mittal, Mengdi Wang, and Peter Henderson. 2024. Assessing the brittleness of safety alignment via pruning and low-rank modifications. In *Proceedings of the 41st International Conference on Machine Learning*, pages 52588–52610.

Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, and 1 others. 2022. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *International conference on machine learning*, pages 23965–23998. PMLR.

Zhongbin Xie and Thomas Lukasiewicz. 2023. An empirical analysis of parameter-efficient methods for debiasing pre-trained language models. *arXiv preprint arXiv:2306.04067*.

Prateek Yadav, Derek Tam, Leshem Choshen, Colin A Raffel, and Mohit Bansal. 2024. Ties-merging: Resolving interference when merging models. *Advances in Neural Information Processing Systems*, 36.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, and 1 others. 2024a. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*.

An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, and 1 others. 2024b. Qwen2. 5-math technical report: Toward mathematical expert model via self-improvement. *arXiv preprint arXiv:2409.12122*.

Enneng Yang, Li Shen, Guibing Guo, Xingwei Wang, Xiaochun Cao, Jie Zhang, and Dacheng Tao. 2024c. Model merging in llms, mllms, and beyond: Methods, theories, applications and opportunities. *arXiv preprint arXiv:2408.07666*.

Le Yu, Bowen Yu, Haiyang Yu, Fei Huang, and Yongbin Li. 2024. Language models are super mario: Absorbing abilities from homologous models as a free lunch. In *Forty-first International Conference on Machine Learning*.

Xinyu Zhang, Huiyu Xu, Zhongjie Ba, Zhibo Wang, Yuan Hong, Jian Liu, Zhan Qin, and Kui Ren. 2024. Privacyasst: Safeguarding user privacy in tool-using large language model agents. *IEEE Transactions on Dependable and Secure Computing*.

Tong Zheng, Lichang Chen, Simeng Han, R Thomas McCoy, and Heng Huang. 2025a. Learning to reason via mixture-of-thought for logical reasoning. *arXiv preprint arXiv:2505.15817*.

Tong Zheng, Yan Wen, Huiwen Bao, Junfeng Guo, and Heng Huang. 2025b. Asymmetric conflict and synergy in post-training for llm-based multilingual machine translation. *arXiv preprint arXiv:2502.11223*.

Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. 2022. Learning to prompt for vision-language models. *International Journal of Computer Vision*, 130(9):2337–2348.

Yujun Zhou, Han Bao, Yue Huang, Kehan Guo, Zhenwen Liang, Pin-Yu Chen, Tian Gao, Werner Geyer, Nuno Moniz, Nitesh V Chawla, and 1 others. 2025. Emergent deceptive behaviors in reward-optimizing llms. In *Socially Responsible and Trustworthy Foundation Models at NeurIPS 2025*.

Yujun Zhou, Yufei Han, Haomin Zhuang, Kehan Guo, Zhenwen Liang, Hongyan Bao, and Xiangliang Zhang. 2024. Defending jailbreak prompts via in-context adversarial game. *arXiv preprint arXiv:2402.13148*.

Hongtu Zhu, Joseph G Ibrahim, Sikyum Lee, and Heping Zhang. 2007. Perturbation selection and influence measures in local influence analysis. 35:2565–2588.

Hongtu Zhu, Joseph G Ibrahim, and Niansheng Tang. 2011. Bayesian influence analysis: a geometric approach. *Biometrika*, 98(2):307–323.

A Appendices

A.1 Detail of Parameter sparsification experiment

Experiment on MMLU We conduct experiments on the multiple-choice problems from the MMLU (Hendrycks et al., 2020) dataset, using Qwen2-7B. We take the cross-entropy loss, *i.e.*, $f = -\log P(y = y_{\text{pred}}|x, \theta)$, as the target function, and calculate the FI value according to Theorem 3.3. In this setup, we treat the task as a 4-class classification problem with the possible classes being "A," "B," "C," and "D".

Experiment on Alpaca-Eval We use the Alpaca-eval validation set (Dubois et al., 2024), a widely adopted benchmark, and conduct experiments with various open-source models, including LLaMA2, LLaMA3 (Touvron et al., 2023), and Qwen2 (Bai et al., 2023), across different sizes. We report two metrics: ROUGE-1 (comparing to pre-sparsity responses) and length-control winning rate (LCWR), comparing to GPT-3.5 Turbo. Higher scores are better for both metrics.

To estimate the average FI for sequence generation, we use the fixed-context approach with $L = 5$. For each sample z, we estimate $\mathbf{FI}_l(z)$ by generating $N = 10$ responses, truncating them at position $l-1$. These truncated sequences are used to approximate the conditional expectation by computing the sample average. The per-token FI values are then aggregated using Equation 3 to obtain $\mathbf{FI}^L_{\text{seq}}(z)$, which is averaged across all samples to estimate the overall FI.

A.2 Computation complexity analysis

Let n denote the number of samples, p denote the dimension of perturbation ($p = 3$ for pixel-wise computations and $p = 1$ for parameter-wise computations), and d represent the total number of pixels or parameters.

Computational Complexity Analysis:

- **Jacobian-Norm**: $\mathcal{O}(npd)$, arising from gradient computation per pixel/parameter.
- **Saliency-Map**: Identical to Jacobian-Norm, $\mathcal{O}(npd)$.
- **FI-inverse**: $\mathcal{O}(np^3d + npd)$, with $\mathcal{O}(p^3)$ from inverse matrix computations and $\mathcal{O}(npd)$ from gradient calculations.
- **FI-cSVD (our method)**: $\mathcal{O}(np^2r_0d + npd)$, where $\mathcal{O}(p^2r_0)$ stems from the compact SVD used to compute matrix inversion efficiently.

1. **Parameter-wise stability**: Since individual parameters have dimension $p = 1$, the FI calculation reduces to scalar inversion, thus the complexity simplifies to $\mathcal{O}(npd)$, matching Jacobian-Norm and Saliency-Map.

2. **Pixel-wise stability (image data)**: Given that each pixel has dimension $p = 3$ (RGB), the FI calculation involves compact SVD for a 3×3 matrix. Theoretically, this makes our method about 9 times slower compared to baseline methods. However, in practical implementation, our approach is only approximately 2 times slower.

The table below presents the average time required to compute FI, Saliency Map, and Jacobian Norm for a single image using Qwen2VL-7B. All results are averaged over 100 images and measured on an A100-80G GPU.

Method	Time (s)
FI	0.3828
Saliency-Map	0.1964
Jacobian-Norm	0.1939

Table 4: Computation times for different methods.

A.3 Reparametrization Invariance of FI

The proposed FI measure has the property of transformation invariance.

Theorem A.1 (Reparametrization invariance)**.** *Suppose that ϕ is a diffeomorphism of ω. Then, $FI_\omega(\omega_0)$ is invariant with respect to any reparameterization corresponding to ϕ. Specifically, let*

$$\tilde{\omega}(t) = \phi \circ \omega(t), \quad \tilde{\omega}_0 = \phi(\omega_0),$$

we have $FI_{\tilde{\omega}}(\tilde{\omega}_0) = FI_\omega(\omega_0)$. The detailed proof can be found in (Shu and Zhu, 2019).

Theorem A.1 establishes that $FI_\omega(\omega_0)$ is invariant under any diffeomorphic (e.g., scaling and spinning) reparameterization of the original perturbation. This invariance property is not shared by other measures, such as Jacobian norm (Novak et al., 2018), Cook's local influence measure (Cook, 1986), and Sharpness (Novak et al., 2018).

For instance, consider a perturbation of the form $\alpha + \Delta\alpha$, where α is a subvector of $(x^\top, \theta^\top)^\top$. If we apply a scaling reparameterization $\alpha' =$

$K \odot \alpha$, where K is a scaling vector and $\odot$ denotes element-wise multiplication, then the Jacobian norms change:

$$\|J(\alpha)\|_F = \left[\sum_i \left(\frac{\partial f}{\partial \alpha_i}\right)^2\right]^{1/2} \neq \|J(\alpha')\|_F.$$

In contrast, the FI measure remains unchanged. Such a reparameterization does not alter the function itself but may affect the measure values, potentially weakening the correlation between perturbation and performance degradation. A similar discussion can be found in (Dinh et al., 2017).

A.4 Detail of FI-guided protection in model merging

Hyper parameter	Search Ranges of Hyperparameters
Protecting ratio k	[1%, 2%, 3%, 4%, 5%, 6%, 7%, 8%, 9%, 10%]
Weight parameter γ in Task Arithmetic & TIES	[0.3, 0.4, 0.5, 0.6, 0.9, 1.0]

Table 5: Searched ranges of hyperparameters of model merging methods.

A.5 Additional experiment results on parameter sparsification

Model	Criteria	Full	6% Sparsity		8% Sparsity		10% Sparsity		12% Sparsity	
			FI	Random	FI	Random	FI	Random	FI	Random
Llama2	Rouge-1	1.0	0.52	0.59 ± 0.02	0.4	0.43 ± 0.06	0.18	0.68 ± 0.01	$\underline{0.05}$	0.19 ± 0.03
13B	LCWR	0.43	0.38	0.41 ± 0.03	0.29	0.34 ± 0.07	$\underline{0.09}$	0.42 ± 0.0	$\underline{0.01}$	0.08 ± 0.05
Llama3	Rouge-1	1.0	0.46	0.52 ± 0.04	0.21	0.41 ± 0.06	$\underline{0.09}$	0.25 ± 0.04	$\underline{0.04}$	0.12 ± 0.03
8B	LCWR	0.42	0.4	0.38 ± 0.01	0.12	0.30 ± 0.03	$\underline{0.0}$	0.12 ± 0.01	$\underline{0.0}$	0.01 ± 0.01
Llama2	Rouge-1	1.0	0.44	0.56 ± 0.01	0.25	0.45 ± 0.02	$\underline{0.06}$	0.33 ± 0.02	$\underline{0.0}$	0.21 ± 0.02
7B	LCWR	0.42	0.32	0.4 ± 0.0	0.12	0.35 ± 0.01	$\underline{0.0}$	0.19 ± 0.05	$\underline{0.0}$	0.1 ± 0.03
Qwen2	Rouge-1	1.0	$\underline{0.09}$	0.41 ± 0.05	$\underline{0.01}$	0.30 ± 0.09	$\underline{0.01}$	0.31 ± 0.06	$\underline{0.01}$	0.15 ± 0.02
7B	LCWR	0.41	$\underline{0.03}$	0.35 ± 0.03	$\underline{0.02}$	0.25 ± 0.1	$\underline{0.03}$	0.20 ± 0.05	$\underline{0.03}$	0.08 ± 0.02
Qwen2	Rouge-1	1.0	0.18	0.4 ± 0.13	0.16	0.32 ± 0.02	$\underline{0.05}$	0.28 ± 0.08	$\underline{0.05}$	0.23 ± 0.02
1.5B	LCWR	0.14	$\underline{0.03}$	0.07 ± 0.04	$\underline{0.04}$	0.02 ± 0.02	$\underline{0.0}$	0.04 ± 0.0	$\underline{0.0}$	0.02 ± 0.02

Table 6: Performance of Different Models Based on Criteria with Full Value and Sparsity Percentages.

Task	3% Sparsity		5% Sparsity		10% Sparsity		15% Sparsity	
	FI-High	Random	FI-High	Random	FI-High	Random	FI-High	Random
Action Recognization	0.89	0.91	0.89	0.90	0.68	0.79	0.25	0.68
Attribute Recognization	0.78	0.82	0.78	0.81	0.40	0.74	0.30	0.61
Celebrity Recognization	0.87	0.89	0.82	0.87	0.69	0.80	0.30	0.62
Functional Reasoning	0.83	0.88	0.82	0.83	0.73	0.80	0.40	0.71

Table 7: Performance comparison between FI-guided and random sparsification strategies on LLaVA-1.5-13B across tasks under varying sparsity levels.

A.6 Proof of Theorem 3.3

Proof. We apply Taylor expansion to $f(\omega(t))$ at the point $\omega(t)$:

$$f(\omega(t)) = f(\omega(0)) + \nabla_{f(\omega_0)}^T h_{\omega_0} t + \frac{1}{2}\left(h_{\omega_0}^T H_{f(\omega_0)} h_{\omega_0} + \nabla_{f(\omega_0)}^T d^2\omega(0)/dt^2\right) t^2 + o\left(t^2\right),$$

where $\nabla_{f(\omega_0)} = \partial f(\omega)/\left.\partial\omega\right|_{\omega=\omega_0}$ and $H_{f(\omega_0)} = \partial^2 f(\omega)/\left.\partial\omega\partial\omega^T\right|_{\omega=\omega_0}$. From the definition of S_C, $S_C^2(\omega_t, \omega_0)$ can be approximated as $S_C^2(\omega_t, \omega_0) = t^2 h_{\omega_0}^T G_{\omega_0} h_{\omega_0} + o\left(t^2\right)$. Based on l'H^opital's rule, the stability measure FI from Equation1 can be rewritten as:

$$\mathbf{FI}_\omega\left(\omega_0\right) = \max_{h_\omega} \frac{h_\omega^T \nabla_{f(\omega_0)} \nabla_{f(\omega_0)}^T h_\omega}{h_\omega^T G_{\omega_0} h_\omega}.$$

We then reparameterize ω to $\tilde{\omega} = G_{\omega_0}^{-1/2}\omega$. According to Theorem A.1, the stability measure **FI** remains invariant under this reparameterization

$$FI_\omega(\omega_0) = FI_{\tilde{\omega}}(\tilde{\omega}_0) = \arg\max_{h_{\tilde{\omega}}} \frac{h_{\tilde{\omega}}^\top G_{\omega_0}^{-1/2} \nabla_{f(\omega_0)} \nabla_{f(\omega_0)}^\top G_{\omega_0}^{-1/2} h_{\tilde{\omega}}}{h_{\tilde{\omega}}^\top h_{\tilde{\omega}}}.$$

The maximization problem is now in the form of a Rayleigh quotient, which attains its maximum when $h_{\tilde{\omega}}$ is proportional to $G_{\omega_0}^{-1/2}\nabla_{f(\omega_0)}$. Substituting back into the Rayleigh quotient, we find:

$$\begin{aligned}
\mathbf{FI}_\omega(\omega_0) &= \frac{\left(G_{\omega_0}^{-1/2}\nabla_{f(\omega_0)}\right)^T G_{\omega_0}^{-1/2} \nabla_{f(\omega_0)} \nabla_{f(\omega_0)}^T G_{\omega_0}^{-1/2} \left(G_{\omega_0}^{-1/2}\nabla_{f(\omega_0)}\right)}{\left(G_{\omega_0}^{-1/2}\nabla_{f(\omega_0)}\right)^T \left(G_{\omega_0}^{-1/2}\nabla_{f(\omega_0)}\right)} \\
&= \frac{\nabla_{f(\omega_0)}^T G_{\omega_0}^{-1} \nabla_{f(\omega_0)} \nabla_{f(\omega_0)}^T G_{\omega_0}^{-1} \nabla_{f(\omega_0)}}{\nabla_{f(\omega_0)}^T G_{\omega_0}^{-1} \nabla_{f(\omega_0)}} \\
&= \nabla_{f(\omega_0)}^T G_{\omega_0}^{-1} \nabla_{f(\omega_0)}.
\end{aligned}$$

This concludes the proof. □