

Understanding Pre-training and Fine-tuning from Loss Landscape Perspective

Huanran Chen^{1,2}, Yinpeng Dong^{1,2}, Zeming Wei³, Yao Huang^{1,2},
Yichi Zhang^{1,2}, Hang Su¹, Jun Zhu¹

¹Tsinghua University, ²RealAI, ³Peking University
huanran.chen@outlook.com

Abstract

Recent studies have revealed that the loss landscape of large language models resembles a basin, within which the models perform nearly identically, and outside of which they lose all their capabilities. In this work, we conduct further studies on the loss landscape of large language models. We discover that pre-training creates a "basic capability" basin, and subsequent fine-tuning creates "specific capability" basins (e.g., math, safety, coding) within the basic capability basin. We further investigate two types of loss landscapes: the most-case landscape (i.e., the landscape along most directions) and the worst-case landscape (i.e., the landscape along the worst direction). We argue that as long as benign fine-tuning remains within the most-case basin, it will not compromise previous capabilities. Similarly, any fine-tuning (including the adversarial one) that stays within the worst-case basin would not compromise previous capabilities. Finally, we theoretically demonstrate that the size of the most-case basin can bound the size of the worst-case basin and the robustness with respect to input perturbations. We also show that, due to the over-parameterization property of current large language models, one can easily enlarge the basins by five times.

1 Introduction

Large language models (LLMs) have garnered significant attention in recent years for their remarkable performance across diverse applications [52, 4, 23, 42]. These models typically undergo a pre-training phase with extensive datasets to acquire foundational knowledge, followed by multiple alignment stages using high-quality, domain-specific data to activate specialized capabilities [10, 52, 53]. In this work, we investigate the intriguing *alignment brittleness* phenomenon. In particular:

- Why does fine-tuning with benign data sometimes compromise capabilities acquired through prior alignment [58, 22, 50, 9, 46, 31, 41, 44]?
- Why does fine-tuning with adversarial data, even for just a few steps, rapidly destroy all capabilities of large language models [58, 59, 67, 33, 39, 32, 34, 16]?
- Why are large language models easily jailbroken in white-box settings, and how does this relate to the above issues [77, 57, 14, 3]?

We propose that these questions may be partially explained by the loss landscape of large language models [40]. We examine two types of loss landscapes: the **most-case landscape**, which reflects capacity degradation when parameters move along most directions, and the **worst-case landscape**, which captures degradation along the most detrimental direction.

As shown in Fig. 1, the **most-case landscape** resembles a basin, within which models perform nearly identically and outside of which they rapidly lose all capabilities [54]. This aligns with

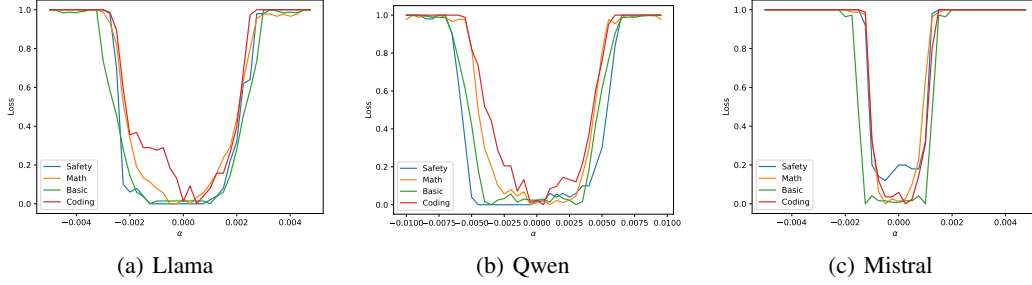


Figure 1: The most-case loss landscape of different models. Specific benchmarks and visualization details are provided in Sec. 3.2. As shown, the loss landscape of LLMs resembles a basin, within which models perform nearly identically and outside of which they lose all capabilities. The basic capability basin, endowed during pre-training, is consistently the largest. Fine-tuning creates specific capability basins (e.g., math, safety, coding) within this basic capability basin.

recent findings that large language models can resist common noise perturbations [49, 29, 64]. The pre-training phase establishes a "basic capability basin" that endows the model with fundamental language comprehension and conversational abilities. Subsequent alignment stages sequentially create specific capability basins (e.g., coding, math, safety) within this basic capability basin. Based on this observation, we propose that *as long as benign fine-tuning remains within a specific capability basin, the parameters remain in that basin, preserving prior capabilities*.

As illustrated in Fig. 2, the **worst-case landscape** is consistently sharp, such that even a small fine-tuning step can move parameters outside the basin, resulting in the loss of all capabilities. This phenomenon resembles prior explanations for adversarial examples: in high-dimensional spaces, there exists a direction that causes rapid degradation, despite most directions being safe [65, 27, 47]. The parameter dimensions of large language models are significantly larger than those of earlier smaller models, making the worst-case direction potentially more detrimental. Furthermore, we propose that a lack of robustness to worst-case parameter perturbations implies vulnerability to input perturbations, such as jailbreaking. Let \mathbf{W} denote the embedding layers. Given that the embedding layers of current large language models are onto transformations (i.e., \mathbf{W} is column full-rank) [11], if there exists a perturbation $\delta_{\mathbf{W}}$ such that the model with weights $\mathbf{W} + \delta_{\mathbf{W}}$ is not robust, then there always exists an input perturbation $\delta_{\mathbf{x}}$ such that the model with input $\mathbf{x} + \delta_{\mathbf{x}}$ is not robust, as $\mathbf{W}\mathbf{x} + \delta_{\mathbf{W}}\mathbf{x}$ and $\mathbf{W}\mathbf{x} + \mathbf{W}\delta_{\mathbf{x}}$ can yield the same vector [74]. This explains the vulnerability of large language models to both jailbreaking [77] and fine-tuning attacks [58].

Through exploratory preliminary studies, we find that one can construct a smooth model such that *the size of the worst-case basin is theoretically lower bounded by the size of the most-case basin*. This implies that we can derive a theoretical upper bound on performance degradation for *any fine-tuning*, even along the worst direction, as well as for input jailbreaking. Combined with our conjecture that benign fine-tuning preserves capabilities within the most-case basin, we conclude that enlarging the most-case basin enhances benign fine-tuning, mitigates harmful fine-tuning, and improves robustness against input jailbreaking. Additionally, we demonstrate that *the basins can be expanded five times with ease*, likely due to the over-parameterization property of neural networks [8, 2]. We hope our work sheds light on the relationships among loss landscapes, robustness to benign and harmful fine-tuning, and jailbreaking, and that our proposed theoretical lower bounds and optimization strategies inspire future large-scale studies on pre-training and fine-tuning.

2 Alignment Brittleness of LLMs

The alignment of LLMs plays a crucial role in ensuring adherence to safety and ethical standards in their applications [5, 68, 36]. During the early stages of LLM advancement, researchers developed various paradigms, like reinforcement learning [20, 6] or red-teaming [55, 48], to build their alignment, which was initially believed to be sufficient to solve the alignment problems. However, a series of recent discoveries revealed that the current alignment of LLMs is shallow and superficial [69, 58, 72, 77, 57, 75]. Though performing aligned values in regular deployments, their alignment ability may be easily broken during fine-tuning or reasoning phases, particularly in the following three aspects:

Normal fine-tuning¹. Supervised Fine-Tuning (SFT) on task-specific datasets has become a common paradigm for applying pre-trained base LLMs in various practical scenarios. However, even after achieving desirable alignment after pre-training, these base models may significantly forget their alignment after fine-tuning. For example, the Llama-2-7b-Chat model raises their harmfulness response rate from 5.5% to 31.8% after fine-tuning on the Alpaca dataset with only one epoch [58].

Adversarial fine-tuning. Furthermore, this superficial alignment can be easily undermined by a small amount of adversarial data. By exploiting the inherent vulnerabilities in the alignment of large language models (LLMs), malicious actors can use a minimal number of harmful examples, such as instructions that encourage harmful behavior or shift identities. For instance, fine-tuning GPT-3.5-turbo on a maliciously crafted dataset containing just 10 harmful examples for 5 epochs can raise the harmfulness rate from 1.8% to 88.8% [58], effectively dismantling the safety mechanisms of sufficiently aligned LLMs.

Input-space jailbreaking. Finally, LLMs after fine-tuning still suffer from input-space attacks, which are known as jailbreaking attacks. For example, adversaries can utilize optimization-based methods to induce the model to answer harmful outputs, even in black-box settings [70, 12, 3, 16]. Similarly, adversaries can exploit other input modules like vision inputs in VLLMs to induce the model to generate any target harmful content [21, 73, 56], with an attack success rate of 100%.

Overall, these threads of discoveries suggest that the current alignment of LLMs is still overly brittle, posing significant concerns regarding their trustworthiness in real-world applications.

3 A Closer Look at the Loss Landscape of LLMs

In this work, we explore these unaddressed questions through the perspective of loss landscapes, which has achieved notable success in understanding various dynamics of conventional neural networks [40, 54].

3.1 Visualization of the Loss Landscape

The loss landscape visualizes the performance degradation of a neural network with respect to parameter perturbations. Formally, let f_{θ} denote a language model with parameters $\theta \in \mathbb{R}^d$, and let $J_{\mathcal{D}}$ represent the benchmark functional on a dataset \mathcal{D} , defined as $J_{\mathcal{D}}(f_{\theta}) = \mathbb{E}_{x \in \mathcal{D}}[J(f_{\theta}(x))]$. This functional takes the language model f_{θ} as input and returns a benchmark value on \mathcal{D} , characterizing its specific capabilities. The loss landscape is thus a visualization of the high-dimensional function $J_{\mathcal{D}} : \mathbb{R}^d \rightarrow \mathbb{R}$.

Benchmarks. To assess the diverse capabilities of a given model, we adopt the following benchmarks: MMLU [30] for basic language proficiency, GSM8K [17] for mathematical reasoning, HumanEval [24] for coding ability, and AdvBench [77] for safety performance. Since the values of different benchmarks are not directly comparable, we normalize each landscape to the interval $[0, 1]$ and invert benchmarks where higher values indicate better performance, unifying them such that lower values indicate better performance for consistent visualization.

Models. We visualize the loss landscape of three cutting-edge models, including Llama-3.1-8B [23], Qwen-2.5-7B [71] and Mistral-8B-2410 [37]. We also analyze the relationship between loss landscape and model size in Appendix D.2.

However, since the function $J_{\mathcal{D}}(\theta)$ is high-dimensional, directly visualizing a d -dimensional landscape is computationally expensive and unintuitive [40]. To address this challenge, researchers typically visualize the loss landscape along a specific direction $\delta \in \mathbb{R}^d$, reducing the problem to visualizing a single-variable function [40, 28, 35, 62]:

$$L(\alpha) = J_{\mathcal{D}}(\theta + \alpha\delta). \quad (1)$$

In this work, we investigate two types of loss landscapes defined by the choice of direction (δ):

¹This is what previous work called ‘‘Benign Fine-tuning’’. In this paper, benign fine-tuning refers to another type of fine-tuning defined in Sec. 3.4

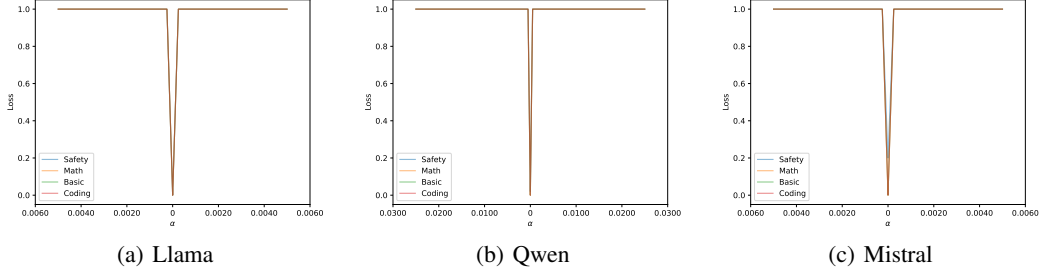


Figure 2: The worst-case loss landscape of the three models. As shown, moving even a small distance along the worst-case direction rapidly degrades all capabilities of LLMs. (Due to all curves reaching the maximum loss at the smallest scale, they completely overlap.)

3.2 Most-case Loss Landscape

The **most-case loss landscape** uses a uniformly random direction $\delta \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, visualizing the single-variable function:

$$L(\alpha) = J_{\mathcal{D}}(\theta + \alpha\delta), \quad \delta \sim \mathcal{N}(\mathbf{0}, \mathbf{I}). \quad (2)$$

Empirically, we observe that different directions $\delta \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ yield nearly identical results. Since $\mathcal{N}(\mathbf{0}, \mathbf{I})$ represents uniformly random directions, this suggests that most directions produce similar landscapes². Therefore, we designate this as the **most-case loss landscape**.

General Geometry. As shown in Fig. 1, the most-case loss landscape for each capability resembles a basin, within which the models perform nearly identically, and outside of which they rapidly lose all capabilities [54]. The pre-training stage creates a "basic capability basin" that endows the model with fundamental language comprehension and conversational abilities. Subsequent alignment stages sequentially establish specific capability basins (e.g., coding [24], math [17], safety [77]) within this basic capability basin. Based on this observation, we argue that *as long as subsequent benign fine-tuning remains within the basin of a specific capability, the parameters will remain within this basin and thus will not compromise those capabilities*.

Model- and Data-Specific Geometry. As demonstrated, some basins are sufficiently large to match the size of the basic capability basin (e.g., safety in Llama and Qwen), while others are smaller (e.g., coding in Llama and Qwen). This suggests that, in these models, coding capabilities are more likely to be forgotten than other capabilities during benign fine-tuning. The size of subsequent alignment basins is model-dependent and hyperparameter-dependent. For instance, the safety basin matches the size of the basic capability basin in Llama and Qwen, but it is significantly smaller in Mistral, indicating that Mistral may be more prone to compromising safety when fine-tuned on new datasets.

3.3 Worst-case Loss Landscape

The **worst-case loss landscape** identifies the steepest direction δ that compromises model capabilities. Formally, the worst-case direction δ is determined by:

$$\delta = \arg \max_{\delta} L(\theta + b\delta) \text{ s.t. } \|\delta\|_2^2 = \mathbb{E}[\|\mathcal{N}(\mathbf{0}, \mathbf{I})\|_2^2] \quad (3)$$

The constraint $\|\delta\|_2^2 = \mathbb{E}[\|\mathcal{N}(\mathbf{0}, \mathbf{I})\|_2^2]$ ensures that the perturbation norm matches that used in Sec. 3.2, enabling direct comparison between the loss landscapes in Fig. 1 and Fig. 2 for each model. The hyperparameter b is only designed to facilitate the identification of the worst-case δ , and does not affect the final direction. Equation (3) is solved by optimizing $L(\theta + b\delta)$ using SGD and projecting the norm of δ to unity at each step [47].

General Geometry. As illustrated in Fig. 2, the worst-case loss landscape resembles a cliff, regardless of the model or capability evaluated. This indicates that moving a short distance along the worst-case direction rapidly degrades all model capabilities. This phenomenon aligns with prior explanations

²This can be validated through hypothesis testing; see the Appendix D.3 for details.

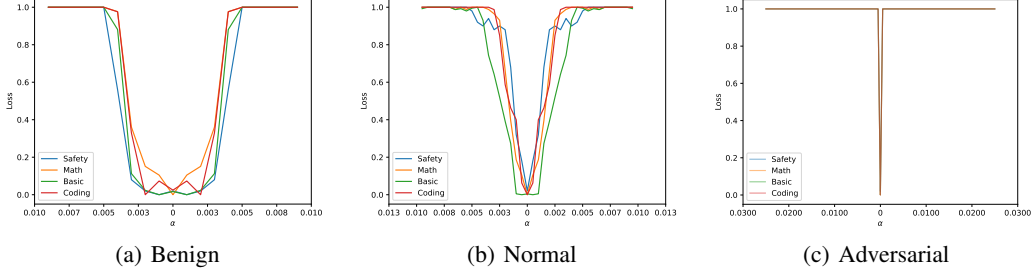


Figure 3: The SFT-case loss landscapes for three different datasets using Qwen2.5-7B.

for adversarial examples: in high-dimensional spaces, there exists a direction that causes rapid degradation, despite most directions being safe [65, 27, 47]. The parameter dimensions of large language models are significantly larger than those of earlier smaller models, making the worst-case direction potentially far more detrimental. This explains why prior adversarial fine-tuning, using only 10 samples and one epoch, can severely compromise the safety capabilities of a model [57].

3.4 SFT-case Loss Landscape

Section 3.2 demonstrates that most directions do not lead to performance degradation within a certain range, whereas Section 3.3 shows that a worst-case direction exists that rapidly degrades all capabilities. The direction of supervised fine-tuning (SFT) naturally lies between these extremes: it may not preserve all capabilities as effectively as the most-case direction, but it does not degrade as quickly as the worst-case direction.

Settings. To visualize the loss landscape along the SFT direction, we select $\delta = \frac{\theta_{sft} - \theta_0}{\|\theta_{sft} - \theta_0\|_2}$. $\sqrt{\mathbb{E}[\|\mathcal{N}(\mathbf{0}, \mathbf{I})\|_2^2]}$. This normalization and rescaling ensure that the perturbation norm matches those used in Sections 3.2 and 3.3, enabling direct comparison of the loss landscapes.

SFT Configurations. We investigate three types of supervised fine-tuning (SFT) using Qwen2.5-7B [71]. *Benign fine-tuning* employs a dataset similar to the original training data. We achieve this by selecting θ_0 as Qwen2.5-7B and θ_{sft} as its officially fine-tuned version, Qwen2.5-7B-1M [71]. *Normal fine-tuning* uses a dataset with a distributional gap from the original data. We achieve this by following the setup in Section 4.4 of [58], i.e., fine-tuning on the Alpaca dataset [76] for one epoch. *Adversarial fine-tuning* utilizes the adversarial AdvBench dataset, fine-tuning for only 10 steps, following the setup in Section 4.2 of [58].

Results. As shown in Fig. 3(a), the loss landscape along the benign fine-tuning direction resembles the most-case landscape. It preserves safety within the most-case basin and loses capability when moving outside this basin. In Fig. 3(b), when there is a distributional gap between the SFT dataset and the original dataset, the loss landscape becomes narrower and sharper, indicating that the fine-tuning direction does not align with the most-case directions. In Fig. 3(c), when fine-tuning on the adversarial dataset, the model rapidly loses all capabilities, responding only with phrases like “Sure, here is” without providing factual answers to questions. Thus, while some SFT configurations align closely with the most-case direction (e.g., Fig. 3(a)), others deviate and degrade capabilities more rapidly (e.g., Fig. 3(b)). This variation clearly depends on the dataset and hyperparameters.

In the following section, we propose that, regardless of the dataset or the model’s sensitivity to hyperparameters, the size of the most-case basin can consistently bound performance degradation during any fine-tuning or jailbreaking attacks.

4 Theoretical Benefits

To begin with, we first formally define the size of most-case basin.

Definition 4.1 A model f_θ is said to have σ -basin on benchmark $J_{\mathcal{D}}$, if it noised version $f_{\theta+\epsilon}$ where $\epsilon \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I})$ perform nearly the same as original version f_θ , i.e.,

$$J_{\mathcal{D}}(f_\theta) - \mathbb{E}_{\epsilon \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I})}[J_{\mathcal{D}}(f_{\theta+\epsilon})] \leq \epsilon. \quad (4)$$

Definition 4.1 is a necessary condition for a model to have most-case landscape looks like Fig. 1. In other words, if a model f_θ have loss landscape that within range σ the model perform nearly the same, then it must have σ -basin, satisfy Definition 4.1.

In the following section, we show that as long as a model have σ -basin, then we can have a (loose) guarantee the performance degradation during *any fine-tuning* and jailbreak attacks.

4.1 Any Alignment Can Be Bounded by Average-case Alignment

This is achieved through the concept of randomized smoothing [18, 60, 38, 14]. Since the model performs nearly identically within a σ -basin, for any input x , instead of returning $f_\theta(x)$, we can sample $\epsilon \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I})$ and return $f_{\theta+\epsilon}(x)$. Thus, the benchmark value for this model is $J_{\mathcal{D}}(f_{\theta+\epsilon})$.

The following theorem demonstrates that, for any bounded benchmark $J_{\mathcal{D}} : \mathbb{R}^d \rightarrow [0, 1]$ ³, regardless of how sensitive f is to its parameters, the smoothed model $\mathbb{E}_{\epsilon \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I})}[J_{\mathcal{D}}(f_{\theta+\epsilon})]$ is at most $\frac{1}{\sqrt{2\pi}\sigma}$ -Lipschitz. Consequently, when θ_0 is updated to θ_{sft} , the benchmark value changes by at most $\frac{1}{\sqrt{2\pi}\sigma} \|\theta_{sft} - \theta_0\|_2$.

Theorem 4.2 (Weak Law of Randomized Smoothing [60, 13]) For any benchmark $J_{\mathcal{D}} : \mathbb{R}^d \rightarrow [0, 1]$, the function $\mathbb{E}_{\epsilon \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I})}[J_{\mathcal{D}}(f_{\theta+\epsilon})]$ is at most $\frac{1}{\sqrt{2\pi}\sigma}$ -Lipschitz, i.e.:

$$|\mathbb{E}_{\epsilon \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I})}[J_{\mathcal{D}}(f_{\theta_{sft}+\epsilon})] - \mathbb{E}_{\epsilon \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I})}[J_{\mathcal{D}}(f_{\theta_0+\epsilon})]| \leq \frac{1}{\sqrt{2\pi}\sigma} \cdot \|\theta_{sft} - \theta_0\|_2. \quad (5)$$

Thus, we can bound the performance degradation as:

$$\mathbb{E}_{\epsilon \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I})}[J_{\mathcal{D}}(f_{\theta_{sft}+\epsilon})] \geq \mathbb{E}_{\epsilon \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I})}[J_{\mathcal{D}}(f_{\theta_0+\epsilon})] - \frac{1}{\sqrt{2\pi}\sigma} \cdot \|\theta_{sft} - \theta_0\|_2. \quad (6)$$

Intuition. The Lipschitz constant with respect to parameters equals the maximum gradient norm with respect to parameters. Although the gradient of a neural network cannot be bounded, the Gaussian-smoothed form transfers the gradient operator from the neural network to the probability density function of the Gaussian distribution, thereby bounding the maximum gradient norm.

The works in [18, 60, 14] provide a stronger version of Theorem 4.2 by considering the maximum Lipschitz constant at each point rather than across the entire input space, as presented in Theorem 4.3. Consequently, Theorem 4.3 consistently provides a tighter bound than Theorem 4.2.

Theorem 4.3 (Strong Law of Randomized Smoothing [18, 60, 14]) For any benchmark $J_{\mathcal{D}} : \mathbb{R}^d \rightarrow [0, 1]$, we have:

$$\mathbb{E}_{\epsilon \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I})}[J_{\mathcal{D}}(f_{\theta_{sft}+\epsilon})] \geq \Phi \left(\Phi^{-1} \left(\mathbb{E}_{\epsilon \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I})}[J_{\mathcal{D}}(f_{\theta_0+\epsilon})] \right) - \frac{\|\theta_{sft} - \theta_0\|_2}{\sigma} \right), \quad (7)$$

where Φ is the cumulative distribution function of the standard Gaussian distribution $\mathcal{N}(0, 1)$, i.e.,

$$\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^x \exp \left(-\frac{s^2}{2} \right) ds.$$

The term $\mathbb{E}_{\epsilon \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I})}[J_{\mathcal{D}}(f_{\theta+\epsilon})]$ represents the expected benchmark value when sampling $\epsilon \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I})$ and evaluating the benchmark on $f_{\theta+\epsilon}$. The bound in Theorem 4.3 provides a robust guarantee of performance stability during fine-tuning. In practice, one typically samples a single ϵ and evaluates $f_{\theta+\epsilon}$, rather than computing the expectation via extensive Monte Carlo sampling. Empirically, sampling a single instance yields results comparable to multiple samples. Theoretically, the variance introduced by single-sample evaluation can be bounded using the well-known concentration phenomenon, which states that a random variable is unlikely to deviate significantly from its expectation.

³Without loss of generality, any benchmark with a bounded output range can be normalized to this interval to obtain a corresponding certified bound.

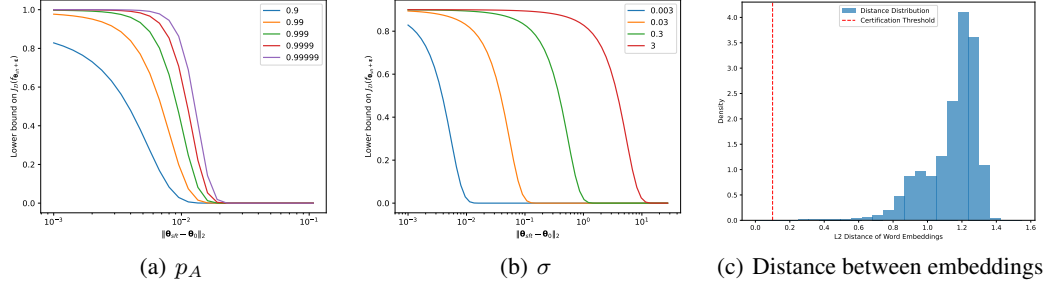


Figure 4: Lower bound guarantees. (a) The lower bound on the benchmark value of the smoothed fine-tuned model $J_{\mathcal{D}}(f_{\theta_{sft}+\epsilon})$ for varying benchmark values on the smoothed original model θ_0 , i.e., $p_A := \mathbb{E}_{\epsilon \sim \mathcal{N}(0, \sigma^2 I)} [J_{\mathcal{D}}(f_{\theta_0+\epsilon})]$, with $\sigma = 0.003$. (b) The lower bound on the benchmark value of the smoothed fine-tuned model $J_{\mathcal{D}}(f_{\theta_{sft}+\epsilon})$ for varying basin sizes σ , with $p_A = 0.9$. (c) Histogram of L2 distances between token embeddings.

Theorem 4.4 (Concentration of Gaussian, adapted from [66]) *With probability at least $1 - \delta$, we have:*

$$J_{\mathcal{D}}(f_{\theta+\epsilon}) \geq \mathbb{E}_{\epsilon \sim \mathcal{N}(0, \sigma^2 I)} [J_{\mathcal{D}}(f_{\theta+\epsilon})] - L\sigma \sqrt{2 \log \frac{1}{\delta}}, \quad (8)$$

where L is the Lipschitz constant of $J_{\mathcal{D}}$ with respect to θ .

By combining Theorem 4.3 and Theorem 4.4, we can lower bound the performance degradation when fine-tuning from θ_0 to $\theta_{sft} + \epsilon$ as follows:

Theorem 4.5 *For any benchmark $J_{\mathcal{D}} : \mathbb{R}^d \rightarrow [0, 1]$, with probability at least $1 - \delta$, we have:*

$$J_{\mathcal{D}}(f_{\theta_{sft}+\epsilon}) \geq \Phi \left(\Phi^{-1} \left(\mathbb{E}_{\epsilon \sim \mathcal{N}(0, \sigma^2 I)} [J_{\mathcal{D}}(f_{\theta_0+\epsilon})] \right) - \frac{\|\theta_{sft} - \theta_0\|_2}{\sigma} \right) - L\sigma \sqrt{2 \log \frac{1}{\delta}}. \quad (9)$$

Examples. Through hypothesis testing in Appendix D.3, we establish that Qwen2.5-7B [71] has a σ -basin with $\sigma = 0.003$ on the safety task using the AdvBench dataset, where $\mathbb{E}_{\epsilon \sim \mathcal{N}(0, \sigma^2 I)} [J_{\mathcal{D}}(f_{\theta_0+\epsilon})] \geq 0.9^4$. Based on this, we derive a lower bound on safety degradation as a function of $\|\theta_{sft} - \theta_0\|_2$. As shown in Fig. 4(a), for a $\sigma = 0.003$ basin, a higher performance on the original parameters, i.e., $\mathbb{E}_{\epsilon \sim \mathcal{N}(0, \sigma^2 I)} [J_{\mathcal{D}}(f_{\theta_0+\epsilon})]$, yields a stronger guarantee on the fine-tuned parameters θ_{sft} , i.e., $J_{\mathcal{D}}(f_{\theta_{sft}+\epsilon})$.

Increasing the Basin Size Improves the Guarantee. As illustrated in Fig. 4(b), enlarging the basin of a model while preserving the performance on the original model, i.e., $\mathbb{E}_{\epsilon \sim \mathcal{N}(0, \sigma^2 I)} [J_{\mathcal{D}}(f_{\theta_0+\epsilon})]$, linearly strengthens the theoretical guarantee. In other words, increasing σ results in a linear improvement in the performance degradation guarantee during fine-tuning.

4.2 Input-space Robustness Can Be Bounded by Average-case Alignment

In this section, we propose that the size of a basin also bounds performance degradation against jailbreaking attacks [77, 70].

Intuition. Let \mathbf{W} denote the embedding layers. Given that the embedding layers of current large language models are onto transformations (i.e., \mathbf{W} is column full-rank) [11], the activation after a weight perturbation, $\mathbf{W}\mathbf{x} + \delta_{\mathbf{W}}\mathbf{x}$, and the activation after an input perturbation, $\mathbf{W}\mathbf{x} + \mathbf{W}\delta_{\mathbf{x}}$, can yield the same vector [74]. Thus, if the model is robust to any weight perturbation $\|\delta_{\mathbf{W}}\|_2 \leq \epsilon_{\mathbf{W}}$, it is also robust to any input perturbation $\delta_{\mathbf{x}}$ such that $\mathbf{W}\delta_{\mathbf{x}} \in \{\delta_{\mathbf{W}}\mathbf{x} \mid \|\delta_{\mathbf{W}}\|_2 \leq \epsilon_{\mathbf{W}}\}$.

Theorem 4.6 *Let \mathcal{D}' be a modified version of \mathcal{D} where k tokens are substituted, i.e., each token \mathbf{e}_i is replaced with \mathbf{e}'_i in the set $\mathcal{C} = \{(\mathbf{e}_i, \mathbf{e}'_i)\}_{i=1}^k$. For any benchmark $J_{\mathcal{D}} : \mathbb{R}^d \rightarrow [0, 1]$, with probability*

⁴Unlike landscape visualizations, benchmark values here are higher-is-better.

at least $1 - \delta$, the performance degradation can be bounded by the embedding differences:

$$J_{\mathcal{D}'}(f_{\theta+\epsilon}) \geq \Phi \left(\Phi^{-1} \left(\mathbb{E}_{\epsilon \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I})} [J_{\mathcal{D}}(f_{\theta+\epsilon})] \right) - \frac{\sqrt{\sum_{i=1}^k \|\mathbf{W} \mathbf{e}_i - \mathbf{W} \mathbf{e}'_i\|_2^2}}{\sigma} \right) - L\sigma \sqrt{2 \log \frac{1}{\delta}}. \quad (10)$$

A straightforward application of this theorem involves comparing the modified and original inputs, calculating the equivalent weight differences, and applying the results from Sec. 4.1.

Application to Certification Against Jailbreaking. Consider $\mathcal{D} = \{x\}$ containing a single input instance and J as a safety detector that returns values greater than 0.5 for safe outputs and less than 0.5 for harmful outputs. We can certify robustness against jailbreaking attacks by determining whether $J_{\mathcal{D}'}(f_{\theta+\epsilon})$ remains above 0.5 after modifying $\mathcal{D} = \{x\}$ to $\mathcal{D}' = \{x_{adv}\}$.

We derive the following conclusions for Qwen2.5-7B [71]:

Substituting Special Tokens Preserves Performance. In Qwen2.5-7B [71], special tokens such as "<s>", ".", "...", " ", and " " can be interchanged without compromising performance. Due to their small L2 distances, substituting these tokens, as evaluated in Theorem 4.6, results in equivalent parameter perturbations too small to significantly alter model outputs. Thus, the model exhibits robustness to these special or infrequently used tokens.

Substituting Tokens With or Without Prefix Spaces Has Limited Impact. Most large language models, including Qwen2.5-7B, use BPE tokenizers [61], where tokens with and without leading spaces are distinct (e.g., "hi" vs. " hi"). During decoding, the tokenizer automatically adjusts based on sentence position. Many certifiable substitution pairs include tokens with and without leading spaces, indicating minimal performance impact.

Current LLMs Are Generally Sensitive to Input Changes. As shown in Fig. 4(c), only a small fraction of token substitutions have negligible effects on model outputs. This flexibility allows models to generate diverse responses based on input variations but also introduces robustness vulnerabilities. Minor input changes can lead to significant output variations, potentially resulting in incorrect or unsafe responses [77, 14, 3].

4.3 Basin May Have Sufficient Expressive Power in the Future

Given the theoretical guarantees of fine-tuning within a basin, one might wonder whether subsequent fine-tuning can be constrained to a theoretically guaranteed region, thereby preserving all original capabilities and achieving continual learning without forgetting [15].

A primary concern with this constraint method is whether it limits the expressive power of the hypothesis set. Specifically, if the hypothesis set is constrained to functions where $\|\theta_{sft} - \theta_0\|_2 \leq O(\sigma)$, can it still effectively fit the fine-tuning dataset?

We propose that a large model with a parameter norm constraint of $O(\sigma)$ may exhibit greater expressive power than smaller models with significantly larger norms. This is because expressive power primarily depends on the number of parameters or layers, rather than the norm of the parameters. Thus, training a larger model could address concerns about expressive power.

Lemma 4.7 ([7]) *The upper and lower bounds of the VC dimension of an l -layer fully connected ReLU network are $O(dl \log d)$ and $\Omega(dl \log \frac{d}{l})$, respectively.*

Lemma 4.8 (Adapted from [51]) *The upper bound of the Rademacher complexity of a two-layer ReLU neural network, where each parameter has a value bound of $O(\sigma)$, is $O(d\sigma^2)$.*

5 Enlarging the Alignment Basin

Given that a larger basin provides stronger robustness guarantees against any fine-tuning and jail-breaking attacks, one might wonder whether the basin can be enlarged. In this section, we propose that the basin can be readily expanded, may due to the over-parameterization property of neural networks [43, 2].

Algorithm 1 Gaussian-augmented Optimizer for Basin Enlargement (GO optimizer)

Require: Model f_θ , dataset \mathcal{D} , perturbation variance σ^2 , base optimizer (e.g., SGD or Adam)

- 1: **for** each gradient step **do**
 - 2: Sample mini-batch $\{\mathbf{x}_i\}_{i=1}^B \sim \mathcal{D}$ and $\epsilon \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I})$.
 - 3: Compute gradient on perturbed parameters $\nabla_\theta L_{\text{train}} = -\sum_{i=1}^B \nabla_\theta \log p(\mathbf{x}_i | \theta + \epsilon)$
 - 4: Update parameters $\theta \leftarrow \text{Optimizer}(\theta, \nabla_\theta L_{\text{train}})$
 - 5: **end for**
 - 6: **return** θ
-

5.1 Gaussian-augmented Optimizer

As outlined in Definition 4.1, enlarging the basin size requires optimizing $\mathbb{E}_{\epsilon \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I})} [J_{\mathcal{D}}(f_{\theta+\epsilon})]$, ensuring that the model θ is robust to Gaussian perturbations. To this end, we define the loss function as the expected cross-entropy loss over perturbed parameters:

$$L_{\text{train}}(\mathbf{x}, \theta) = -\mathbb{E}_{\epsilon \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I})} [\log p(\mathbf{x} | \theta + \epsilon)]. \quad (11)$$

As detailed in Algorithm 1, the optimization process involves performing a forward pass with perturbed parameters $\theta + \epsilon$, computing the loss, calculating the gradient via backpropagation, and using the gradient to update the parameters with a standard optimizer.

5.2 Experimental Results

Settings. We utilize the UltraFeedback dataset [19], which comprises diverse corpora designed to endow models with foundational knowledge and conversational capabilities. Due to limited computational resources, we fine-tune a compact model, Qwen2.5-0.5B, using the Gaussian-augmented optimizer outlined in Algorithm 1 for four epochs.

Results. As shown in Fig. 5, the original Qwen2.5-0.5B model fails to maintain conversational abilities when its parameters θ are perturbed by Gaussian noise with a variance of 0.05, producing only incoherent outputs. In contrast, when trained using the Gaussian-augmented optimizer with $\sigma = 0.005$ or $\sigma = 0.01$, the model sustains basic conversational capabilities under the same noise perturbation. However, these models perform similarly to a model trained directly on the UltraFeedback dataset without pre-training; that is, using the Gaussian-augmented optimizer only during the fine-tuning phase, rather than both pre-training and fine-tuning, does not enable the perturbed model $f_{\theta+\epsilon}$ to perform comparably to the unperturbed model f_θ , since these parameters $\theta + \epsilon$ have not benefit from pre-training. This suggests that the Gaussian-augmented optimizer is most effective during the pre-training phase, enabling the model to retain capabilities across a broader parameter neighborhood. See Appendix G for further details.

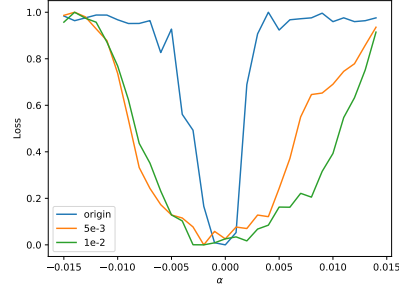


Figure 5: The loss landscape of original Qwen-2.5-0.5B and fine-tuned by GO optimizer with $\sigma = 0.005$ and $\sigma = 0.01$.

6 Conclusion

In this work, we explore the loss landscape of large language models to elucidate the alignment brittleness phenomenon. We demonstrate that the loss landscape of large language models resembles a basin, within which models perform nearly identically and outside of which they lose all capabilities. This property enables us to derive a theoretical lower bound on performance degradation during *any fine-tuning* and jailbreaking attacks within certain norm constraints. We also show that the basin can be readily expanded due to the over-parameterization property of neural networks. Despite these contributions, our exploration remains preliminary. We hope our work sheds light on the alignment brittleness phenomenon and motivates large-scale studies that apply the Gaussian-augmented optimizer during the pre-training phase of larger models, verifying the practical utility of theoretical guarantees, the practical relationship between the robustness of parameters against Gaussian perturbations and general fine-tuning.

References

- [1] Samuel K Ainsworth, Jonathan Hayase, and Siddhartha Srinivasa. Git re-basin: Merging models modulo permutation symmetries. *arXiv preprint arXiv:2209.04836*, 2022. 14
- [2] Zeyuan Allen-Zhu, Yuanzhi Li, and Zhao Song. A convergence theory for deep learning via over-parameterization. In *International conference on machine learning*, pages 242–252. PMLR, 2019. 2, 8
- [3] Maksym Andriushchenko, Francesco Croce, and Nicolas Flammarion. Jailbreaking leading safety-aligned llms with simple adaptive attacks. *arXiv preprint arXiv:2404.02151*, 2024. 1, 3, 8
- [4] Anthropic. The claude 3 model family: Opus, sonnet, haiku. 2024. 1
- [5] Usman Anwar, Abulhair Saparov, Javier Rando, Daniel Paleka, Miles Turpin, Peter Hase, Ekdeep Singh Lubana, Erik Jenner, Stephen Casper, Oliver Sourbut, et al. Foundational challenges in assuring alignment and safety of large language models. *Transactions on Machine Learning Research*, 2024. 2
- [6] Yuntao Bai et al. Constitutional ai: Harmlessness from ai feedback, 2022. 2
- [7] Peter L Bartlett, Nick Harvey, Christopher Liaw, and Abbas Mehrabian. Nearly-tight vc-dimension and pseudodimension bounds for piecewise linear neural networks. *Journal of Machine Learning Research*, 20(63):1–17, 2019. 8
- [8] Mikhail Belkin, Daniel Hsu, Siyuan Ma, and Soumik Mandal. Reconciling modern machine-learning practice and the classical bias–variance trade-off. *Proceedings of the National Academy of Sciences*, pages 15849–15854, 2019. 2
- [9] Federico Bianchi, Mirac Suzgun, Giuseppe Attanasio, Paul Röttger, Dan Jurafsky, Tatsunori Hashimoto, and James Zou. Safety-tuned llamas: Lessons from improving the safety of large language models that follow instructions. *arXiv preprint arXiv:2309.07875*, 2023. 1
- [10] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, pages 1877–1901, 2020. 1
- [11] Nicholas Carlini, Daniel Paleka, Krishnamurthy Dj Dvijotham, Thomas Steinke, Jonathan Hayase, A Feder Cooper, Katherine Lee, Matthew Jagielski, Milad Nasr, Arthur Conmy, et al. Stealing part of a production language model. *arXiv preprint arXiv:2403.06634*, 2024. 2, 7
- [12] Patrick Chao, Alexander Robey, Edgar Dobriban, Hamed Hassani, George J Pappas, and Eric Wong. Jailbreaking black box large language models in twenty queries. In *R0-FoMo: Robustness of Few-shot and Zero-shot Learning in Large Foundation Models*, 2023. 3
- [13] Huanran Chen, Yinpeng Dong, Shitong Shao, Zhongkai Hao, Xiao Yang, Hang Su, and Jun Zhu. Diffusion models are certifiably robust classifiers. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. 6
- [14] Huanran Chen, Yinpeng Dong, Zeming Wei, Hang Su, and Jun Zhu. Towards the worst-case robustness of large language models. *arXiv preprint arXiv:2501.19040*, 2025. 1, 6, 8, 15
- [15] Pin-Yu Chen, Han Shen, Payel Das, and Tianyi Chen. Fundamental safety-capability trade-offs in fine-tuning large language models. *arXiv preprint arXiv:2503.20807*, 2025. 8
- [16] Taiye Chen, Zeming Wei, Ang Li, and Yisen Wang. Scalable defense against in-the-wild jailbreaking attacks with safety context retrieval. *arXiv preprint arXiv:2505.15753*, 2025. 1, 3
- [17] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021. 3, 4
- [18] Jeremy Cohen, Elan Rosenfeld, and Zico Kolter. Certified adversarial robustness via randomized smoothing. In *International Conference on Machine Learning*, pages 1310–1320, 2019. 6, 15
- [19] Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Bingxiang He, Wei Zhu, Yuan Ni, Guotong Xie, Ruobing Xie, Yankai Lin, et al. Ultrafeedback: Boosting language models with scaled ai feedback. *arXiv preprint arXiv:2310.01377*, 2023. 9
- [20] Josef Dai, Xuehai Pan, Ruiyang Sun, et al. Safe rlhf: Safe reinforcement learning from human feedback. In *ICLR*, 2024. 2
- [21] Yinpeng Dong, Huanran Chen, Jiawei Chen, Zhengwei Fang, Xiao Yang, Yichi Zhang, Yu Tian, Hang Su, and Jun Zhu. How robust is google’s bard to adversarial image attacks? In *R0-FoMo: Robustness of Few-shot and Zero-shot Learning in Large Foundation Models*, 2023. 3
- [22] Yanrui Du, Sendong Zhao, Jiawei Cao, Ming Ma, Danyang Zhao, Fenglei Fan, Ting Liu, and Bing Qin. Towards secure tuning: Mitigating security risks arising from benign instruction fine-tuning. *arXiv preprint arXiv:2410.04524*, 2024. 1

- [23] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024. 1, 3
- [24] Mark Chen et al. Evaluating large language models trained on code. 2021. 3, 4
- [25] Jonathan Frankle, Gintare Karolina Dziugaite, Daniel Roy, and Michael Carbin. Linear mode connectivity and the lottery ticket hypothesis. In *International Conference on Machine Learning*, pages 3259–3269. PMLR, 2020. 14
- [26] Timur Garipov, Pavel Izmailov, Dmitrii Podoprikin, Dmitry P Vetrov, and Andrew G Wilson. Loss surfaces, mode connectivity, and fast ensembling of dnns. *Advances in neural information processing systems*, 31, 2018. 14
- [27] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*, 2015. 2, 5
- [28] Ian J Goodfellow, Oriol Vinyals, and Andrew M Saxe. Qualitatively characterizing neural network optimization problems. *arXiv preprint arXiv:1412.6544*, 2014. 3
- [29] Andrey Gromov, Kushal Tirumala, Hassan Shapourian, Paolo Gloriosi, and Daniel A Roberts. The unreasonable ineffectiveness of the deeper layers. *arXiv preprint arXiv:2403.17887*, 2024. 2, 14
- [30] Dan Hendrycks, Collin Burns, Steven Basart, Andrew Critch, Jerry Li, Dawn Song, and Jacob Steinhardt. Aligning ai with shared human values. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021. 3
- [31] Chia-Yi Hsu, Yu-Lin Tsai, Chih-Hsun Lin, Pin-Yu Chen, Chia-Mu Yu, and Chun-Ying Huang. Safe lora: The silver lining of reducing safety risks when finetuning large language models. *Advances in Neural Information Processing Systems*, 37:65072–65094, 2024. 1
- [32] Tiansheng Huang, Sihao Hu, Fatih Ilhan, Selim Furkan Tekin, and Ling Liu. Harmful fine-tuning attacks and defenses for large language models: A survey. *arXiv preprint arXiv:2409.18169*, 2024. 1
- [33] Tiansheng Huang, Sihao Hu, Fatih Ilhan, Selim Furkan Tekin, and Ling Liu. Lazy safety alignment for large language models against harmful fine-tuning. *arXiv preprint arXiv:2405.18641*, 2024. 1
- [34] Tiansheng Huang, Sihao Hu, and Ling Liu. Vaccine: Perturbation-aware alignment for large language models against harmful fine-tuning attack. *arXiv preprint arXiv:2402.01109*, 2024. 1
- [35] Daniel Jiwoong Im, Michael Tao, and Kristin Branson. An empirical analysis of deep network loss surfaces. 2016. 3
- [36] Jiaming Ji, Tianyi Qiu, Boyuan Chen, Borong Zhang, Hantao Lou, Kaile Wang, Yawen Duan, Zhonghao He, Jiayi Zhou, Zhaowei Zhang, et al. Ai alignment: A comprehensive survey. *arXiv preprint arXiv:2310.19852*, 2023. 2
- [37] Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, L  lio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timoth  e Lacroix, and William El Sayed. Mistral 7b. *arXiv preprint arXiv:2310.06825*, 2023. 3
- [38] Guang-He Lee, Yang Yuan, Shiyu Chang, and Tommi Jaakkola. Tight certificates of adversarial robustness for randomly smoothed classifiers. *Advances in Neural Information Processing Systems*, 32, 2019. 6
- [39] Chak Tou Leong, Yi Cheng, Kaishuai Xu, Jian Wang, Hanlin Wang, and Wenjie Li. No two devils alike: Unveiling distinct mechanisms of fine-tuning attacks. *arXiv preprint arXiv:2405.16229*, 2024. 1
- [40] Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, and Tom Goldstein. Visualizing the loss landscape of neural nets. *Advances in Neural Information Processing Systems*, 31, 2018. 1, 3
- [41] Mingjie Li, Wai Man Si, Michael Backes, Yang Zhang, and Yisen Wang. Salora: Safety-alignment preserved low-rank adaptation. *arXiv preprint arXiv:2501.01765*, 2025. 1
- [42] Aixiu Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024. 1
- [43] Chaoyue Liu, Libin Zhu, and Mikhail Belkin. Loss landscapes and optimization in over-parameterized non-linear systems and neural networks. *Applied and Computational Harmonic Analysis*, 59:85–116, 2022. 8
- [44] Qin Liu, Chao Shang, Ling Liu, Nikolaos Pappas, Jie Ma, Neha Anna John, Srikanth Doss, Llu  s Marqu  ez, Miguel Ballesteros, and Yassine Benajiba. Unraveling and mitigating safety alignment degradation of vision-language models. *arXiv preprint arXiv:2410.09047*, 2024. 1
- [45] Ekdeep Singh Lubana, Eric J Bigelow, Robert P Dick, David Krueger, and Hidenori Tanaka. Mechanistic mode connectivity. In *International Conference on Machine Learning*, pages 22965–23004, 2023. 14

- [46] Kaifeng Lyu, Haoyu Zhao, Xinran Gu, Dingli Yu, Anirudh Goyal, and Sanjeev Arora. Keeping llms aligned after fine-tuning: The crucial role of prompt templates. *arXiv preprint arXiv:2402.18540*, 2024. 1
- [47] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018. 2, 4, 5
- [48] Ninareh Mehrabi, Palash Goyal, Christophe Dupuy, Qian Hu, Shalini Ghosh, Richard Zemel, Kai-Wei Chang, Aram Galstyan, and Rahul Gupta. Flirt: Feedback loop in-context red teaming. In *EMNLP*, 2023. 2
- [49] Xin Men, Mingyu Xu, Qingyu Zhang, Bingning Wang, Hongyu Lin, Yaojie Lu, Xianpei Han, and Weipeng Chen. Shortgpt: Layers in large language models are more redundant than you expect. *arXiv preprint arXiv:2403.03853*, 2024. 2, 14
- [50] Jishnu Mukhoti, Yarin Gal, Philip HS Torr, and Puneet K Dokania. Fine-tuning can cripple your foundation model; preserving features may be the solution. *arXiv preprint arXiv:2308.13320*, 2023. 1, 17
- [51] Behnam Neyshabur, Ryota Tomioka, and Nathan Srebro. Norm-based capacity control in neural networks. In *Conference on Learning Theory*, pages 1376–1401, 2015. 8
- [52] OpenAI. Gpt-4 technical report. *arXiv*, 2023. 1
- [53] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022. 1
- [54] Sheng Y Peng, Pin-Yu Chen, Matthew Hull, and Duen H Chau. Navigating the safety landscape: Measuring risks in finetuning large language models. *Advances in Neural Information Processing Systems*, 37:95692–95715, 2024. 1, 3, 4, 14
- [55] Ethan Perez, Saffron Huang, Francis Song, Trevor Cai, Roman Ring, John Aslanides, Amelia Glaese, Nat McAleese, and Geoffrey Irving. Red teaming language models with language models. In *EMNLP*, 2022. 2
- [56] Xiangyu Qi, Kaixuan Huang, Ashwinee Panda, Peter Henderson, Mengdi Wang, and Prateek Mittal. Visual adversarial examples jailbreak aligned large language models. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pages 21527–21536, 2024. 3
- [57] Xiangyu Qi, Ashwinee Panda, Kaifeng Lyu, Xiao Ma, Subhrajit Roy, Ahmad Beirami, Prateek Mittal, and Peter Henderson. Safety alignment should be made more than just a few tokens deep. In *The Thirteenth International Conference on Learning Representations*, 2024. 1, 2, 5
- [58] Xiangyu Qi, Yi Zeng, Tinghao Xie, Pin-Yu Chen, Ruoxi Jia, Prateek Mittal, and Peter Henderson. Fine-tuning aligned language models compromises safety, even when users do not intend to! In *The Twelfth International Conference on Learning Representations*, 2023. 1, 2, 3, 5
- [59] Domenic Rosati, Jan Wehner, Kai Williams, Łukasz Bartoszcze, David Atanasov, Robie Gonzales, Subhabrata Majumdar, Carsten Maple, Hassan Sajjad, and Frank Rudzicz. Representation noising effectively prevents harmful fine-tuning on llms. *arXiv e-prints*, pages arXiv–2405, 2024. 1
- [60] Hadi Salman, Jerry Li, Ilya Razenshteyn, Pengchuan Zhang, Huan Zhang, Sebastien Bubeck, and Greg Yang. Provably robust deep learning via adversarially trained smoothed classifiers. *Advances in Neural Information Processing Systems*, 32, 2019. 6, 15
- [61] Rico Sennrich. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*, 2015. 8
- [62] Leslie N Smith and Nicholay Topin. Exploring loss function topology with cyclical learning rates. *arXiv preprint arXiv:1702.04283*, 2017. 3
- [63] Jacob Mitchell Springer, Sachin Goyal, Kaiyue Wen, Tanishq Kumar, Xiang Yue, Sadhika Malladi, Graham Neubig, and Aditi Raghunathan. Overtrained language models are harder to fine-tune. *arXiv preprint arXiv:2503.19206*, 2025. 14
- [64] Mingjie Sun, Zhuang Liu, Anna Bair, and J Zico Kolter. A simple and effective pruning approach for large language models. *arXiv preprint arXiv:2306.11695*, 2023. 2, 14
- [65] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *International Conference on Learning Representations*, 2014. 2, 5
- [66] Martin J Wainwright. *High-dimensional statistics: A non-asymptotic viewpoint*, volume 48. Cambridge university press, 2019. 7
- [67] Jiong Xiao Wang, Jiazhao Li, Yiquan Li, Xiangyu Qi, Muhao Chen, Junjie Hu, Yixuan Li, Bo Li, and Chaowei Xiao. Mitigating fine-tuning jailbreak attack with backdoor enhanced alignment. *arXiv e-prints*, pages arXiv–2402, 2024. 1

- [68] Zhichao Wang, Bin Bi, Shiva Kumar Pentyala, Kiran Ramnath, Sougata Chaudhuri, Shubham Mehrotra, Xiang-Bo Mao, Sitaram Asur, et al. A comprehensive survey of llm alignment techniques: Rlhf, rlaif, ppo, dpo and more. *arXiv preprint arXiv:2407.16216*, 2024. [2](#)
- [69] Alexander Wei, Nika Haghtalab, and Jacob Steinhardt. Jailbroken: How does llm safety training fail? *Advances in Neural Information Processing Systems*, 36, 2023. [2](#)
- [70] Zeming Wei, Yifei Wang, and Yisen Wang. Jailbreak and guard aligned language models with only few in-context demonstrations. *arXiv preprint arXiv:2310.06387*, 2023. [3](#), [7](#)
- [71] An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024. [3](#), [5](#), [7](#), [8](#)
- [72] Xianjun Yang, Xiao Wang, Qi Zhang, Linda Petzold, William Yang Wang, Xun Zhao, and Dahua Lin. Shadow alignment: The ease of subverting safely-aligned language models. *arXiv preprint arXiv:2310.02949*, 2023. [2](#)
- [73] Zonghao Ying, Aishan Liu, Tianyuan Zhang, Zhengmin Yu, Siyuan Liang, Xianglong Liu, and Dacheng Tao. Jailbreak vision language models via bi-modal adversarial prompt. *arXiv preprint arXiv:2406.04031*, 2024. [3](#)
- [74] Yihao Zhang, Hangzhou He, Jingyu Zhu, Huanran Chen, Yifei Wang, and Zeming Wei. On the duality between sharpness-aware minimization and adversarial training. *arXiv preprint arXiv:2402.15152*, 2024. [2](#), [7](#)
- [75] Yihao Zhang, Zeming Wei, Jun Sun, and Meng Sun. Adversarial representation engineering: A general model editing framework for large language models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. [2](#)
- [76] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36, 2024. [5](#)
- [77] Andy Zou, Zifan Wang, Nicholas Carlini, Milad Nasr, J Zico Kolter, and Matt Fredrikson. Universal and transferable adversarial attacks on aligned language models. *arXiv preprint arXiv:2307.15043*, 2023. [1](#), [2](#), [3](#), [4](#), [7](#), [8](#)

A Notations

d	Number of parameters in a language model.
\mathcal{D}	Dataset for a benchmark.
$J_{\mathcal{D}}$	Benchmark that maps from \mathbb{R}^d to \mathbb{R} .
α	Perturbation scale.
b	Auxiliary variable that only for finding worst-case direction.
Φ	CDF of the standard Gaussian distribution.
Φ^{-1}	CDF of the standard Gaussian distribution.
θ	Model’s parameters.
θ_0	Model’s parameters before SFT.
θ_{sft}	Model’s parameters after SFT.
σ	Basin size.
e_i	One-hot vector for token i .
W	The embedding matrix

B Related Work

This work is greatly inspired by [54], which argues that the safety loss landscape resembles a basin, within which the model is safe and outside of which it is not. Our study extends [54] by investigating the loss landscape across additional capabilities and providing a deeper analysis of the relationship between loss landscapes and catastrophic forgetting during fine-tuning.

Concurrent work [63] suggests that over-pre-trained large language models are harder to fine-tune because they lack robustness to parameter perturbations. Our work complements this study by explaining how robustness to Gaussian parameter perturbations provides a lower bound on performance degradation during fine-tuning.

C Why Do the Loss Landscapes of LLMs Resemble Basins?

This paper presents a seemingly different conclusion from [54], where we argue that the loss landscape of large language models resembles a basin, whereas they suggest that LLM performance degrades gradually as the perturbation budget increases. The key reason is that [54] uses benchmarks that evaluate models by log-likelihood, where the loss landscape is expected to be continuous and smooth. In contrast, we use benchmarks that evaluate models based on their generation results.

This raises the question of why large language models perform nearly identically within a certain range of Gaussian perturbations to their parameters but rapidly lose all capabilities when perturbations exceed this range. This phenomenon aligns with recent findings that large language models can resist common noise perturbations [49, 29, 64], which show that removing certain modules in LLMs leaves their capabilities largely intact, but removing more causes a swift loss of conversational ability.

In this work, we do not have a definitive answer to this question. However, we hypothesize that this may stem from the over-parameterization property of LLMs. Mode connectivity is an important property of over-parameterized models, stating that the modes of an over-parameterized network are connected to one another [26, 25, 45]. For large language models, beyond permutation invariance in feed-forward networks [1], there is also rotational invariance in key and query matrices. Consequently, these equivalent modes may be closer to each other, forming a subspace with dimensions nearly equivalent to the original space. We leave this exploration to future work.

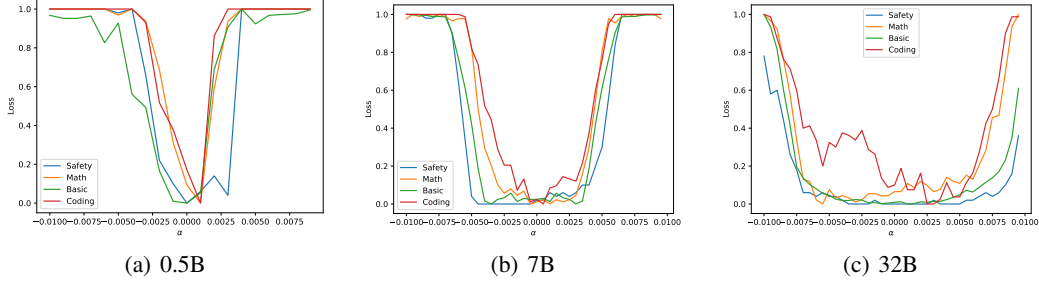


Figure 6: The most-case loss landscape of three Qwen2.5 models with different sizes. The larger the model, the larger the basins.

D More Experiments

D.1 More Experimental Details

Clarification on All Models Used in This Paper. In this paper, we exclusively use the instructed or chat versions of models. We do not use any base models, as they are completely unable to engage in conversation and thus cannot be evaluated on any benchmark. Due to page limitations, we omit “instruct” or “chat” in the main text. For example, Qwen2.5-7B in the main text refers to Qwen2.5-7B-Instruct, not the pre-trained base model.

Normalizing the Loss Landscape. As clarified in Sec. 3.1, since the values of different benchmarks are not directly comparable, we normalize each landscape to the interval $[0, 1]$ and invert benchmarks where higher values indicate better performance, unifying them such that lower values indicate better performance for consistent visualization. This normalization involves three steps. First, we subtract the minimum value. Then, we divide by the maximum value. Finally, we invert the value.

D.2 Loss Landscapes of Large Models

We also visualize the loss landscapes of larger models, as shown in Fig. 6. We draw the following conclusions:

Larger models tend to have larger basins. As shown, for each capability, including basic, math, safety, and coding, the basic capability basin of Qwen2.5-0.5B is small, while that of Qwen2.5-7B is larger. Qwen2.5-32B has the largest basin, nearly twice the size of Qwen2.5-7B.

Larger models have greater expressive power within their basins. As analyzed in Sec. 4.3, the larger the model and its basin, the greater the expressive power. Since larger models have both more parameters and larger basin sizes, they exhibit significantly greater expressive power than smaller models.

Larger models are more robust to fine-tuning and jailbreaking. As analyzed in Sec. 4.1, the larger the basin size, the greater the robustness against fine-tuning and jailbreaking attacks. Therefore, we conclude that larger models are more robust to fine-tuning, less likely to compromise capabilities, and more resistant to jailbreaking attacks. This may be one of the benefits of scaling up models.

D.3 Hypothesis Testing

To determine the basin size for a given model, we need to test Definition 4.1. However, this requires calculating an expectation $\mathbb{E}_{\epsilon \sim \mathcal{N}(0, \sigma^2 I)} [J_{\mathcal{D}}(f_{\theta+\epsilon})]$ over a high-dimensional space d . Following [18, 60, 14], we use hypothesis testing to control the type-I error in Monte Carlo sampling for estimating this expectation. Specifically, we apply the Clopper-Pearson bound to obtain a lower bound for $\mathbb{E}_{\epsilon \sim \mathcal{N}(0, \sigma^2 I)} [J_{\mathcal{D}}(f_{\theta+\epsilon})]$ with a type-I error of 0.01.

Table 1: The distance of different tuning configurations and whether they are located in basin and guaranteed regions. As shown, fine-tuning configurations are always within the basin but not the guaranteed region.

Model	Tuning Configs	Basin Size	Tuning Distance	Guaranteed Region
Qwen2.5-7B	Qwen2.5-7B-1M	$0.003 \cdot \sqrt{7B}$	152.06	0.1
Qwen2.5-7B	Qwen2.5-7B-Base	$0.003 \cdot \sqrt{7B}$	25.98	0.1
Qwen2.5-Math-7B	DeepSeek-Distill	$0.003 \cdot \sqrt{7B}$	358.99	0.1
Qwen2.5-7B	Qwen2.5-Math-7B	$0.003 \cdot \sqrt{7B}$	1820.83	0.1
Qwen2.5-7B	Alpaca lr=5e-5	$0.003 \cdot \sqrt{7B}$	100.87	0.1
Qwen2.5-7B	Alpaca lr=2e-5	$0.003 \cdot \sqrt{7B}$	37.87	0.1
Qwen2.5-7B	AdvBench lr=5e-5	$0.003 \cdot \sqrt{7B}$	26.91	0.1

D.4 About Norm Constraints in Landscape Visualization

One might argue that the norm constraint should be $\|\delta\|_2 = \mathbb{E}[\|\mathcal{N}(\mathbf{0}, \mathbf{I})\|_2]$ instead of $\|\delta\|_2^2 = \mathbb{E}[\|\mathcal{N}(\mathbf{0}, \mathbf{I})\|_2^2]$. By Jensen’s inequality, $\mathbb{E}[\|\mathcal{N}(\mathbf{0}, \mathbf{I})\|_2^2] \geq \mathbb{E}[\|\mathcal{N}(\mathbf{0}, \mathbf{I})\|_2]^2$. However, these two constraints become equivalent as $d \rightarrow \infty$. For the high-dimensional spaces typical of large language models, these constraints are nearly identical.

This is because $\mathbb{E}[\|\mathcal{N}(\mathbf{0}, \mathbf{I})\|_2^2] - \mathbb{E}[\|\mathcal{N}(\mathbf{0}, \mathbf{I})\|_2]^2 = \text{Var}(\|\mathcal{N}(\mathbf{0}, \mathbf{I})\|_2) = \frac{1}{2} + O(\frac{1}{d})$. Thus, the relative error $\frac{\text{Var}(\|\mathcal{N}(\mathbf{0}, \mathbf{I})\|_2)}{\mathbb{E}[\|\mathcal{N}(\mathbf{0}, \mathbf{I})\|_2]^2}$ approaches zero at a rate of $O(1/d)$.

E Norm of Common Fine-tuning

We also test the ℓ_2 norm of common post-training techniques to determine whether they are located within basins or guaranteed regions. As shown in Table 1, we draw the following conclusions:

Common fine-tuning configurations are all located within the original basin. As shown, the ℓ_2 norm of the basin size is approximately 250.99. All common fine-tuning configurations are located within basins. For reference, the ℓ_2 norm between two distinct pre-trained models, Qwen2.5-Math-7B and Qwen2.5-7B, is much larger than this distance. This verifies our claim that benign fine-tuning within the basin does not compromise capabilities.

Larger learning rates lead to greater fine-tuning distances and a higher likelihood of catastrophic forgetting. As shown, the larger the learning rate, the more likely the optimizer is to favor solutions with greater distances. Consequently, such configurations are more likely to compromise capabilities.

Common fine-tuning configurations are not located within the guaranteed region. As shown, the theoretical guaranteed region is small enough to prevent capability compromise. This is because the theoretical guaranteed region serves only as a lower bound, i.e., only when fine-tuning along the worst-case direction does moving outside the guaranteed region compromise capabilities. Since normal fine-tuning does not align with worst-case directions, it typically has a much larger fine-tuning distance budget than these guaranteed lower bounds.

F Comparison of Theorem 4.2 and Theorem 4.3

Beyond the fact that Theorem 4.3 is strictly stronger than Theorem 4.2 (i.e., the Lipschitz constant of Theorem 4.3 at each point is strictly less than $\frac{1}{\sqrt{2\pi}\sigma}$), there is also a significant difference. Theorem 4.3 indicates that the higher the clean accuracy p_A , the smaller the Lipschitz constant, and thus the greater the robustness; when $p_A = 1$, the Lipschitz constant becomes zero, making forgetting extremely unlikely. Conversely, the lower the clean accuracy p_A , the larger the Lipschitz constant, and the more likely performance degradation occurs. This eliminates the forgetting-learning tradeoff even along

the worst-case direction, i.e., as long as the performance on old tasks exceeds that on new tasks, the smoothness constraints make it more likely to learn new tasks than to forget old tasks, and the performance degradation on old tasks is always less than the maximum performance gain on new tasks.

G Limitations and Future Work

Preference for Pre-training Over Fine-tuning. The primary limitation of this work is our inability to apply the Gaussian-augmented optimizer during pre-training. When using the Gaussian-augmented optimizer only during fine-tuning, although the basin of basic capabilities can be widened, the points within the basin perform similarly to a randomly initialized model trained directly on the fine-tuning dataset without pre-training. Consequently, its other capabilities lag behind those of the original model. We hypothesize that this could be addressed by applying the Gaussian-augmented optimizer during pre-training rather than fine-tuning. Recent studies suggest that the pre-training phase is when the model acquires its capabilities, while fine-tuning merely activates these capabilities [50]. Using the Gaussian-augmented optimizer during pre-training may identify basins where most points already possess these capabilities.

Preference for Larger Models Over Smaller Models. As analyzed in Appendix D.2, larger models exhibit greater expressive power and more readily form larger basins. The primary obstacle preventing us from applying the Gaussian-augmented optimizer to both pre-train and larger models is their high memory usage. For a 7B model, a single 80GB GPU is insufficient to train with a batch size of 1, requiring the use of current training frameworks, such as DeepSpeed or Megatron, with optimizer offload modes. However, these frameworks support only a limited set of optimizers, such as Adam or SGD, and do not accommodate custom optimizers.