On the Editability of Delta Parameters in Post-Trained Models

Qiaoyu Tang[1,2], **Le Yu**[3], **Bowen Yu**[3] *, **Hongyu Lin**[1]*,
Keming Lu[3], **Yaojie Lu**[1], **Xianpei Han**[1], **Le Sun**[1]
[1]Chinese Information Processing Laboratory, Institute of Software,
Chinese Academy of Sciences [2]University of Chinese Academy of Sciences
[3]Qwen Team, Alibaba Group
{tangqiaoyu2020,hongyu,luyaojie,xianpei,sunle}@iscas.ac.cn
{chuanyi.yl,yubowen.ybw,lukeming.lkm}@alibaba-inc.com

Abstract

Post-training has emerged as a crucial paradigm for adapting large-scale pre-trained models to various tasks, whose effects are fully reflected by delta parameters (i.e., the disparity between post-trained and pre-trained parameters). While numerous studies have explored delta parameter properties via operations like pruning, quantization, low-rank approximation, and extrapolation, a fundamental question remains: what properties of delta parameters are essential for maintaining performance? In this work, we investigate delta parameter properties along two dimensions: magnitude and sign. Through experiments on instruct language models, reasoning language models, and vision models, we find that delta parameters exhibit considerable *editability*: individual values, distribution shape, relative relationships, and even signs can be substantially modified while maintaining post-trained model's performance. To understand these phenomena, we propose a loss-based local surrogate analysis that examines editing effects through a second-order Taylor expansion. Our analysis introduces the concept of editing intensity, which helps explain the stability boundaries of different editing operations.[1]

1 Introduction

Post-training has become a critical step in developing large-scale models (Han et al., 2024; Xin et al., 2024; Dodge et al., 2020; Zhao et al., 2023). Through supervised fine-tuning and reinforcement learning, post-training endows pre-trained models with diverse capabilities such as instruction following (Rafailov et al., 2023; Ethayarajh et al., 2024), mathematical reasoning (Luo et al., 2023; Tong et al., 2024), code generation (Wang et al., 2025), and visual recognition (Chen et al., 2022; Sandler et al., 2022). The effect of post-training is fully reflected in the *delta parameters*, which are defined as the difference between post-trained and pre-trained parameters (Ilharco et al., 2023; Yu et al., 2024). Understanding the properties of delta parameters is therefore crucial for understanding post-training itself.

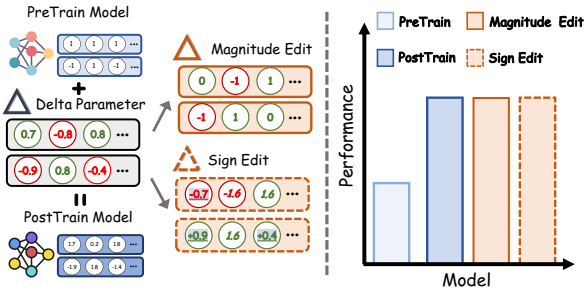


Figure 1: Delta parameters exhibit editability in both magnitude and sign. We investigate editing operations that modify magnitude (e.g., dropping and rescaling) or flip signs (with rescaling). Despite substantial modifications to delta parameters, the edited models can largely preserve the post-trained model's performance.

Recent years have witnessed various methods that edit delta parameters for different benefits. For instance, DARE (Yu et al., 2024) and DELLA-Merging (Deep et al., 2024) showed that models can achieve comparable performance with only a small fraction of delta parameters. BitDelta (Liu et al., 2024) demonstrated that delta parameters can be quantized to 1 bit with modest performance degradation. EXPO (Zheng et al., 2024) observed that extrapolating delta parameters with a suitable scaling factor can even enhance alignment performance. These works demonstrate that editing delta parameters can yield benefits ranging from efficient storage to improved alignment. However, they focus on different operations with different objectives, leading to scattered findings. A fundamental question remains unanswered: *What properties of delta parameters are essential for maintaining perfor-*

* Corresponding authors.
[1]Code is available at https://github.com/icip-cas/DeltaParameterEdit.

Findings of the Association for Computational Linguistics: ACL 2026, pages 36809–36824

mance, and what can be freely manipulated?

In this work, we systematically investigate delta parameter properties along two dimensions: *magnitude* (the absolute value) and *sign* (the direction of change). We conduct experiments across instruct language models (LLaMA-3-8B-Instruct (Dubey et al., 2024), Mistral-7B-Instruct-v0.3 (Jiang et al., 2023), Qwen2-7B-Instruct (Yang et al., 2024)), reasoning language models (Qwen3-1.7B (Team, 2025)), and vision models (ViT-B-32 (Radford et al., 2021)), covering post-training techniques including SFT, RLHF (Qwen et al., 2025), and RLVR (DeepSeek-AI et al., 2025). As shown in Figure 1, we find that delta parameters exhibit considerable *editability*: individual values, distribution shape, relative relationships, and even signs can be substantially modified while maintaining post-trained model's performance. In the magnitude dimension, we find that within a reasonable editing range, what matters more is the overall statistical properties such as the mean of the magnitude, rather than individual parameter values. More surprisingly, in the sign dimension, we discover that a substantial proportion of signs can be flipped while still maintaining comparable performance to the post-trained model. This finding suggests that the direction of parameter updates, often assumed to be important in prior work (Yadav et al., 2023; Liu et al., 2024), also exhibits editability.

To understand these phenomena, we propose a loss-based local surrogate analysis. We analyze the effect of delta parameter editing through a second-order Taylor expansion of the loss function. This analysis reveals that the stability of editing operations is related to an *editing intensity* term, which explains why high drop rates and sign-flip operations are more prone to performance degradation.

2 Preliminaries

2.1 Notation

Let $W_{pre} \in \mathbb{R}^{d\times k}$ denote the parameters of a pre-trained model, where d and k represent the output and input dimensions. A post-trained model with parameters $W_{post} \in \mathbb{R}^{d\times k}$ can be derived from the pre-trained backbone through supervised fine-tuning or reinforcement learning. The delta parameters are defined as the difference between post-trained and pre-trained parameters: $\Delta W = W_{post} - W_{pre} \in \mathbb{R}^{d\times k}$. Since delta parameters reflect the complete effect of post-training, understanding their properties is crucial for understanding post-training itself.

Delta parameter editing refers to applying a transformation $\mathcal{F}$ to the original delta parameters, yielding edited delta parameters $\Delta\widetilde{W}_{edit} = \mathcal{F}(\Delta W)$. The final edited model is then obtained as $W_{edit} = W_{pre} + \Delta\widetilde{W}_{edit}$. Various editing operations have been explored in prior work, including pruning, quantization, and extrapolation.

2.2 Representative Methods

DARE (Yu et al., 2024) is a representative delta parameter editing method designed to reduce parameter redundancy and further mitigate conflicts in model merging. Specifically, DARE first drops delta parameters with probability p, then rescales the remaining parameters by $1/(1-p)$:

$$\Delta\widetilde{W}_{DARE} = \frac{1-M}{1-p} \odot \Delta W, \tag{1}$$

where $M \sim \text{Bernoulli}(p)$ is a random binary mask and $\odot$ denotes element-wise multiplication. With this operation, DARE can drop up to 90% of delta parameters while maintaining model performance.

BitDelta (Liu et al., 2024) proposes a quantization method for delta parameters. It preserves only the sign $\text{sign}(\Delta W)$ and replaces all magnitudes with the average magnitude $\text{AVG}(|\Delta W|)$:

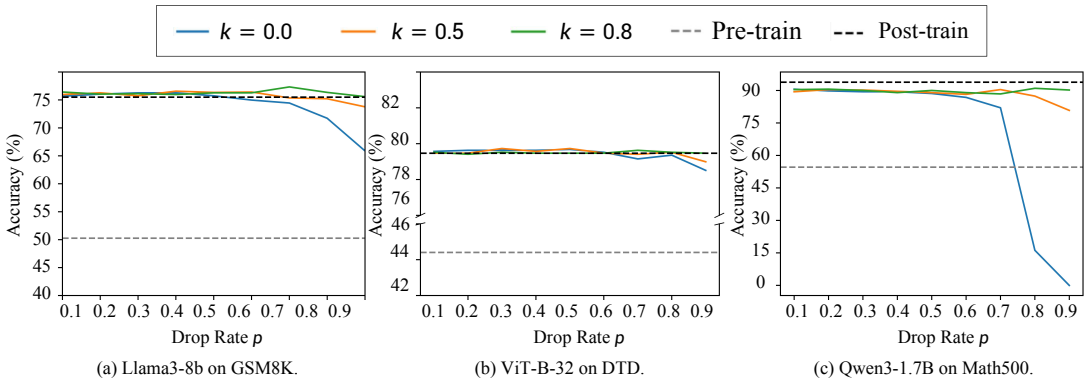
$$\Delta\widetilde{W}_{BitDelta} = \text{AVG}(|\Delta W|) \cdot \text{sign}(\Delta W). \tag{2}$$

In this way, BitDelta quantizes delta parameters to 1-bit while maintaining most of the model performance with slight degradation.

These two methods demonstrate that aggressive modifications to delta parameters do not necessarily cause severe performance degradation. This raises a natural question: what properties of delta parameters are essential for maintaining post-trained model performance? In the following sections, we investigate this question along two dimensions: magnitude and sign.

3 Editability of Delta Parameters

In this section, we examine the properties of delta parameters through experiments. We conduct experiments on instruct language models (LLaMA-3-8B-Instruct (Dubey et al., 2024), Mistral-7B-Instruct-v0.3 (Jiang et al., 2023), Qwen2-7B-Instruct (Yang et al., 2024)), reasoning language models (Qwen3-1.7B (Team, 2025)), and vision models (ViT-B-32 (Radford et al., 2021)). These



(a) Llama3-8b on GSM8K. (b) ViT-B-32 on DTD. (c) Qwen3-1.7B on Math500.

Figure 2: Performance under magnitude editing with varying drop rate p and scaling coefficient k.

models cover most post-training techniques, including SFT, RFT, RLHF, and RLVR. We select appropriate evaluation tasks for each category of models. For instruct language models, we evaluate on 8 tasks: ARC Challenge (Clark et al., 2018), GSM8K (Cobbe et al., 2021), HellaSwag (Zellers et al., 2019), HumanEval (Chen et al., 2021), IFEval (Zhou et al., 2023), MMLU (Hendrycks et al., 2020), TruthfulQA (Lin et al., 2021), and Winogrande (Sakaguchi et al., 2021). For reasoning language models, we evaluate on MATH-500 (Lightman et al., 2023), AIME 2025 (American Invitational Mathematics Examination problems), GPQA Diamond (Rein et al., 2024), and LiveCodeBench (Jain et al., 2024). For vision models, we evaluate on 8 image classification tasks: Cars (Krause et al., 2013), DTD (Cimpoi et al., 2014), EuroSAT (Helber et al., 2019), GTSRB (Stallkamp et al., 2011), MNIST (LeCun et al., 2010), RESISC45 (Cheng et al., 2017), SUN397 (Xiao et al., 2016), and SVHN (Netzer et al., 2011).

3.1 Editability in Magnitude

To investigate the editability of delta parameters in magnitude, we begin with DARE, a representative delta parameter editing method. DARE randomly selects a proportion p of delta parameters and sets them to zero, then rescales the remaining parameters by $1/(1-p)$. With this operation, DARE can drop up to 90% of delta parameters while maintaining model performance. The original paper explains this phenomenon through the lens of expected embeddings. Consider a linear transformation $h = Wx + b$, which is the basic operation in neural networks. Let ΔW and Δb denote the delta parameters. After applying DARE with rescale factor γ, the expectation of the output becomes:

$$\mathbb{E}[\hat{h}] = W_{pre}x + b_{pre} + (1-p)\cdot\gamma\cdot(\Delta W x + \Delta b).$$

By setting $\gamma = 1/(1-p)$, we have $\mathbb{E}[\hat{h}] = h$, i.e., the expected output is preserved. This preservation of expected output is argued to be the key to maintaining model performance.

DARE sets the selected parameters to zero (i.e., multiplies them by 0). A natural question arises: can we multiply by coefficients other than zero, scaling up or down some parameters while rescaling the rest, and still recover model performance? In other words, if we randomly select delta parameters with probability p and multiply them by a coefficient k, what should the rescale factor γ be for the remaining parameters? Based on DARE's theoretical framework, the rescale factor for the remaining parameters should be $(1-kp)/(1-p)$ to preserve the expected output (detailed derivation in Appendix A). This yields a generalized formulation:

$$\Delta\widetilde{W} = k\cdot M\odot\Delta W + \frac{1-kp}{1-p}\cdot(1-M)\odot\Delta W, \quad (3)$$

where $M \sim \text{Bernoulli}(p)$ is a random binary mask. When $k = 0$, this reduces to the original DARE. When $k = 1$, no editing is performed.

To verify this hypothesis, we conduct experiments with different values of k and drop rates p. Figure 2 shows the results on representative models and datasets. When the drop rate p is relatively small, model performance remains nearly identical to the original post-trained model across a wide range of k values. When p is larger, performance slightly decreases but remains comparable to the original DARE setting ($k = 0$). Similar patterns are observed on other settings (see Appendix B). These results indicate that scaling a subset of delta

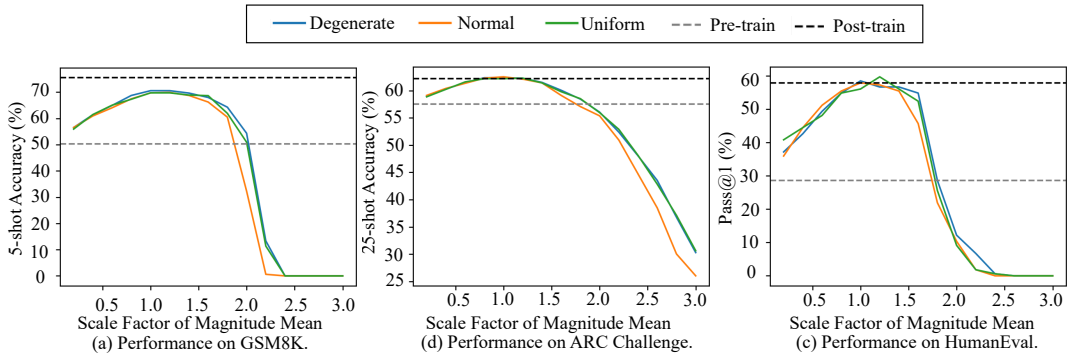


Figure 3: Performance under varying magnitude mean and distribution shape. The x-axis represents the scaling factor of the mean (1.0 is the original mean). Different curves correspond to different distribution shapes: uniform, normal, and degenerate.

parameters by various coefficients, while appropriately rescaling the remaining parameters, can maintain nearly the same model performance.

The above findings demonstrate that delta parameters exhibit considerable editability in magnitude: we can scale a subset of parameters by different coefficients while rescaling the rest to preserve performance. This raises a further question: what properties of magnitude are truly essential for maintaining model performance? We consider three levels of properties, from fine-grained to coarse-grained: (1) the specific value of each individual parameter, (2) the relative relationships among parameters (e.g., ordering by magnitude), and (3) the global statistical properties (e.g., mean of the magnitude, distribution shape). We design a series of experiments to keep the sign and investigate the importance of each property.

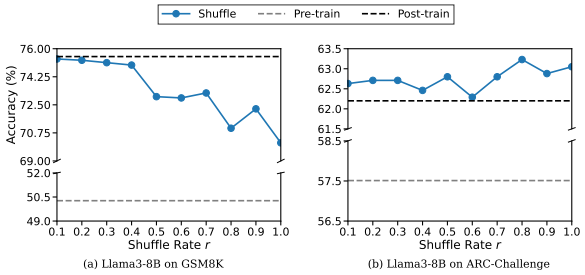
Specific Values. To investigate whether specific values are essential, we design an experiment that changes specific values while preserving relative relationships and global statistics. Specifically, we apply a power transformation to all magnitude values: raising each magnitude to the power of α (we test $\alpha = 0.5$ and $\alpha = 1.5$), which alters each individual value but maintains the relative ordering among parameters. We then rescale the transformed magnitudes to restore the original mean. Table 1 shows the results. We find that under both transformations, performance remains nearly unchanged across all datasets. This suggests that specific magnitude values have limited impact on the capabilities learned through post-training.

Relative Relationships. DARE's zero-out operation already suggests that relative relationships can be partially disrupted without severe performance degradation. To thoroughly investigate this factor, we design a shuffle experiment: we randomly shuffle a proportion r of delta parameter magnitudes across positions, varying r from 10% to 100%. This operation progressively destroys relative relationships while preserving the global distribution and the specific values. Figure 4 shows the results. When the shuffle rate is low, we observe limited performance degradation. As the shuffle rate increases, performance gradually decreases on some datasets such as GSM8K, but remains reasonable even at 100% shuffle rate. On other datasets such as ARC-Challenge, performance is almost unaffected even at high shuffle rates. Full results across six tasks are provided in Appendix E.2. We find that reasoning-intensive tasks like GSM8K are more sensitive to shuffling, while knowledge-oriented tasks largely rely on aggregate statistics and are less affected. This suggests that relative relationships contribute to performance to some extent, particularly when severely disrupted.

Task	Original	Power 0.5	Power 1.5
ARC Challenge	62.20	62.46	61.95
GSM8K	75.51	74.67	75.36
HellaSwag	78.84	79.14	78.22
IFEval	47.12	47.28	47.07
MMLU	65.82	65.53	64.89
TruthfulQA	51.65	51.41	52.21
Winogrande	75.77	76.09	75.45

Table 1: Performance comparison between post-trained model and power & rescale model on LLaMA-3-8B-Instruct.

Global Statistical Properties. We investigate two aspects of global statistics: the distribution



(a) Llama3-8B on GSM8K (b) Llama3-8B on ARC-Challenge

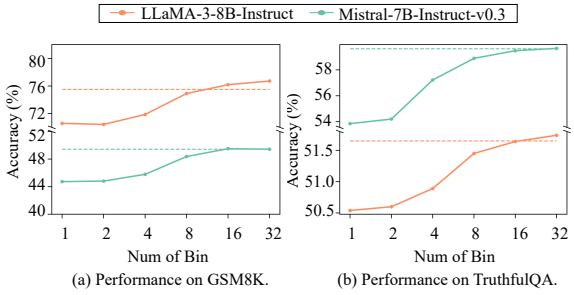
Figure 4: Performance of LLaMA-3-8B-Instruct on GSM8K and ARC Challenge with different shuffle rates.

shape and the mean of the magnitude ($|\Delta W|$). We design an experiment that jointly varies both factors. For the distribution shape, we consider three options: (1) a uniform distribution, (2) a normal distribution, and (3) a degenerate distribution (i.e., all magnitudes set to the same value). For the mean of the magnitude, we scale all magnitudes by a constant factor α ranging from 0.1 to 3.0. Figure 3 shows the results on three representative tasks. We observe two clear patterns. First, at the same mean value, the three distributions achieve nearly identical performance across all tasks. This indicates that the specific distribution shape has limited impact on performance. Second, when the mean deviates from the original mean of magnitude, performance degrades noticeably. These results indicate that the mean of magnitude is a relatively important low-dimensional indicator, while the distribution shape has minimal impact.

The above experiments suggest a hierarchy of importance among magnitude properties within our experimental setting. The mean of the magnitude appears to be the most sensitive factor. The relative relationships among parameters have some impact on performance, particularly when severely disrupted. Within a reasonable editing range, the specific value of each individual parameter and the distribution shape show less sensitivity.

This understanding is consistent with the phenomena observed in existing delta parameter editing methods. For DARE, although individual values are perturbed through random dropping and rescaling, the mean of magnitude is preserved by the rescale operation, and relative relationships are partially maintained among the non-dropped parameters. This may explain why DARE can maintain performance. For BitDelta, all magnitudes are replaced with the mean value, which preserves the mean but completely destroys relative relationships. According to our analysis, this would be expected to cause some performance degradation, which is consistent with the empirical observations in the original paper.

Based on this understanding, we hypothesize that partially restoring relative relationships could improve BitDelta's performance. To verify this, we propose a simple modification: instead of replacing all magnitudes with a single value, we partition parameters into K bins based on their original magnitude ranking. Parameters within each bin are then assigned the mean magnitude of that bin. When $K = 1$, this reduces to the original BitDelta. As K increases, more relative relationship information is preserved. Figure 5 shows the results. Performance improves consistently as K increases. When $K = 16$, performance approaches that of the original post-trained model. This supports our analysis: while the mean of the magnitude is important, partially preserving relative relationships provides additional benefits.



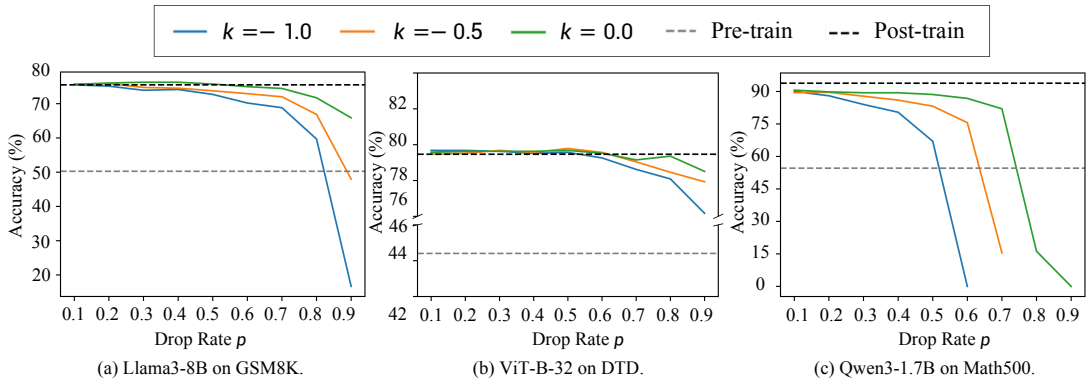
(a) Performance on GSM8K. (b) Performance on TruthfulQA.

Figure 5: Effectiveness of increasing the number of bins in BitDelta. The left subplot shows the performance of LLaMA3-8B-Instruct and Mistral-7B-Instruct-v0.3 on the GSM8K dataset. The right subplot shows the performance on the TruthfulQA dataset. In each subplot, we use the dashed line to represent the performance of the original post-trained model.

3.2 Editability in Sign

The previous subsection demonstrates that delta parameters exhibit considerable editability in magnitude. In this section, we investigate whether delta parameters also exhibit editability in the sign dimension. Intuitively, the sign represents the direction of parameter adjustment during post-training, indicating whether a parameter should increase or decrease relative to the pre-trained value. This directional information is often assumed to be important (Yadav et al., 2023; Liu et al., 2024).

To investigate whether signs can be modified, we extend the generalized formulation in Equation 3 to negative k values. When $k < 0$, the selected



(a) Llama3-8B on GSM8K. (b) ViT-B-32 on DTD. (c) Qwen3-1.7B on Math500.

Figure 6: Performance under sign editing with varying flip rate p and scaling coefficient $k < 0$.

parameters are multiplied by a negative coefficient, which flips their signs and scales their magnitudes. Then we rescale the remaining parameters by $(1 - kp)/(1 - p)$, which is expected to preserve the expected output.

To verify this, we conduct sign-flip experiments across multiple models. Specifically, we consider two representative settings: $k = -0.5$ (flipping signs while reducing magnitude by half) and $k = -1.0$ (fully flipping signs without magnitude change). For each setting, we vary the flip proportion p and observe the resulting performance. Figure 6 shows the representative results (full results are shown in Appendix B). We observe interesting patterns across different ranges of p. When the flip proportion is small, almost all models across all tasks can tolerate sign flipping with only minor performance degradation. This is a notable finding given the common assumption that signs encode essential directional information. When p becomes larger, different models and tasks exhibit varying degrees of robustness. For instance, LLaMA-3-8B-Instruct on GSM8K can tolerate up to 60% sign flipping while maintaining reasonable performance. More strikingly, ViT-B-32 on several vision tasks can tolerate up to 90% complete sign flipping with almost no performance degradation after rescaling. These results suggest that the editability of signs varies across models and tasks, but a substantial degree of sign modification is generally tolerable.

Combined with our findings on magnitude, delta parameters show editability in both magnitude and sign: within a reasonable editing range, individual values, distribution shape, relative relationships, and even signs can be substantially modified while maintaining model performance. We further verify that these findings generalize to larger-scale models (LLaMA-3-70B (Dubey et al., 2024)) and MoE architectures (Mixtral-8x7B (Jiang et al., 2024)) in Appendix E.1, and provide a detailed failure analysis examining how performance degrades beyond stability boundaries in Appendix E.3.

4 Understanding the Editability of Delta Parameters

In the previous section, we observed that delta parameters exhibit considerable editability in both magnitude and sign. At the same time, we also observed some stability boundaries: performance degrades sharply when the drop rate is too high, and sign-flip becomes unstable earlier than magnitude-only editing at comparable modification rates. DARE's theoretical framework provides a useful intuition by approximately preserving expected outputs, but it does not readily explain these phenomena. In this section, we enrich this understanding by analyzing the loss change induced by editing perturbations using a second-order surrogate, which allows us to better understand the editability of delta parameters.

4.1 A Loss-Based Local Surrogate Analysis

In this section, we view the delta parameter editing operation as a perturbation to the post-trained model. We denote the edited model as $W_{edit} = W_{pre} + \Delta\widetilde{W}_{edit}$, where $\Delta\widetilde{W}_{edit}$ is the edited delta parameters. We define the editing perturbation as $e \triangleq \Delta\widetilde{W}_{edit} - \Delta W$, which describes the deviation of the edited delta parameters from the original ones. We focus on the loss change caused by editing:

$$\Delta\mathcal{L} \triangleq \mathcal{L}(W_{edit}) - \mathcal{L}(W_{post}). \tag{4}$$

The goal of editing is to control $|\Delta\mathcal{L}|$ and avoid significant performance degradation.

To analyze how editing affects $\Delta\mathcal{L}$, we apply a second-order Taylor expansion:

$$\Delta\mathcal{L} \approx g^\top e + \frac{1}{2} e^\top C e, \tag{5}$$

where $g = \nabla\mathcal{L}(W_{post})$ is the gradient and $C \succeq 0$ is a positive semi-definite curvature proxy (e.g., Gauss-Newton or Fisher information). We use a PSD proxy instead of the exact Hessian, which may be indefinite in deep networks, so that the quadratic term $\frac{1}{2} e^\top C e$ behaves as a non-negative curvature cost. Our subsequent derivations do not require C to exactly equal the true Hessian—they only rely on C being a reasonable PSD curvature surrogate. This expansion decomposes $\Delta\mathcal{L}$ into a first-order term $g^\top e$ and a second-order term $\frac{1}{2} e^\top C e$, which we analyze in the following subsections.

4.2 Editing Intensity

In this part, we focus on the generalized editing formulation defined in Equation 3. We analyze how the choice of (p, k) affects the loss change $\Delta\mathcal{L}$. The editing perturbation e can be written as:

$$e_i = \begin{cases} (k-1)\Delta w_i & \text{with probability } p \\ \frac{p(1-k)}{1-p}\Delta w_i & \text{with probability } 1-p \end{cases}.$$

For the first-order term $g^\top e = \sum_i g_i e_i$, we can compute its expectation and variance (detailed derivation in Appendix C):

$$\mathbb{E}[g^\top e] = 0$$
$$\mathrm{Var}(g^\top e) = \frac{p}{1-p}(1-k)^2 \cdot \sum_i (g_i \Delta w_i)^2.$$

The expectation being zero indicates that the rescale operation centers the first-order contribution in expectation. This is the foundation for why this type of editing can largely preserve performance. The variance is non-zero and scales with (p, k), which means that as the editing becomes more aggressive, the model after a single editing realization may have larger loss deviation.

For the second-order term $\frac{1}{2} e^\top C e$, we note that under our randomized editing (Equation 3), the i.i.d. Bernoulli mask yields $\mathbb{E}[e_i] = 0$ and, by independence, $\mathbb{E}[e_i e_j] = 0$ for $i \neq j$. Therefore, off-diagonal interactions in C average out in expectation, and for any symmetric C, the expected quadratic cost depends only on its diagonal elements: $\mathbb{E}[e^\top C e] = \sum_i C_{ii}\,\mathbb{E}[e_i^2]$. Denoting $s_i = C_{ii} \geq 0$, the expectation is:

$$\mathbb{E}\left[\frac{1}{2}\sum_i s_i e_i^2\right] = \frac{1}{2} \cdot \frac{p}{1-p}(1-k)^2 \cdot \sum_i s_i (\Delta w_i)^2.$$

Since $s_i \geq 0$, this term is always non-negative, representing a curvature cost that accumulates with the perturbation magnitude. The expectation scales with (p, k) through the factor $\frac{p}{1-p}(1-k)^2$, and with the model/task through $\sum_i s_i (\Delta w_i)^2$.

Both the variance of the first-order term and the expectation of the second-order term share a common factor that depends on (p, k). This factor directly controls the magnitude of loss change: larger values lead to larger variance in the first-order term and larger expected cost in the second-order term. We define the **editing intensity**:

$$\mathcal{I}(p, k) \triangleq \frac{p}{1-p}(1-k)^2. \tag{6}$$

For a fixed model and task, $\mathrm{Var}(g^\top e) \propto \mathcal{I}$ and $\mathbb{E}[e^\top C e] \propto \mathcal{I}$. Thus, larger $\mathcal{I}$ leads to larger loss fluctuations and larger expected curvature cost, making $\Delta\mathcal{L}$ more likely to increase, and consequently causing the edited model to deviate further from the post-trained model.

The editing intensity explains the boundary phenomena observed in Section 3. First, when $p \to 1$, $\mathcal{I} \to \infty$ due to the $\frac{p}{1-p}$ factor, which explains why high drop rates lead to instability regardless of the value of k. Second, when $k < 0$ (sign-flip), $(1-k)^2 > 1$. For example, when $k = -1$, $(1-k)^2 = 4$, which is four times larger than when $k = 0$ (original DARE). This means that at the same drop rate p, sign-flip has significantly larger editing intensity than magnitude-only editing, which explains why sign-flip enters the unstable region earlier and can only tolerate smaller values of p.

To validate the proposed editing intensity, we evaluate it on LLaMA-3-8B-Instruct using the GSM8K benchmark. Specifically, we sweep over a wide range of (p, k) to generate a large collection of edited models and measure their downstream performance as a function of the corresponding editing intensity. As shown in Figure 7, editing intensity exhibits a clear negative correlation with performance: as $\mathcal{I}$ increases, performance consistently decreases. Moreover, when $\mathcal{I}$ remains small, the edited models stay close to the post-trained model in terms of performance. We further show the relationship on a log-scale in Figure 14, where

we observe that under this setting, when $\mathcal{I} \leq 2$, performance shows nearly no degradation. This suggests $\mathcal{I}$ can serve as a practical diagnostic for anticipating whether a given editing configuration remains within the safe regime.

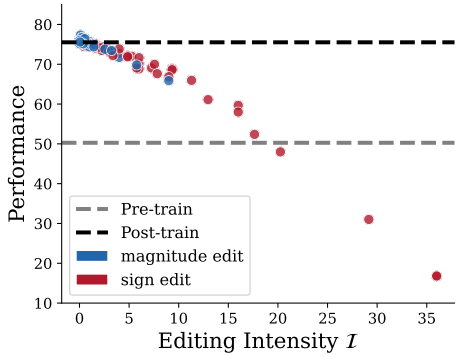


Figure 7: The relationship between editing intensity and performance on LLaMA-3-8B-Instruct using GSM8K benchmark.

5 Related Work

Post-training of Large-Scale Models Post-training is widely adopted to achieve a pre-trained backbone toward downstream capability and alignment objectives (Dodge et al., 2020; Zhao et al., 2023; Team, 2025). Concretely, the training signal may come from supervised demonstrations (Zhao et al., 2024; Lambert et al., 2025; Moshkov et al., 2025), preference-based optimization, e.g., PPO or DPO, (Ouyang et al., 2022; DeepSeek-AI et al., 2024; Xu et al., 2024; Wang et al., 2024), or verifiable feedback produced by rule-based or model-based verifiers, (Shao et al., 2024; Yu et al., 2025). The effectiveness of post-training can be denoted by the delta parameters, which represent the difference between post-trained and pre-trained parameters (Ilharco et al., 2023; Yu et al., 2024). Given the close correlations between delta parameters and the post-training process, investigating the properties of delta parameters becomes particularly important. In this paper, we discovered the editability of delta parameters, suggesting that the effects of post-training can be approximately preserved under diverse parameter configurations.

Delta Parameter Editing Delta parameter editing has been explored for various purposes in recent years. One line of work focuses on model merging, which aims to combine multiple post-trained models into a single model. DARE (Yu et al., 2024) reduces parameter conflicts by randomly dropping delta parameters and rescaling the rest. DELLA-Merging (Deep et al., 2024) extends DARE with magnitude-aware dropping. TIES-Merging (Yadav et al., 2023) resolves sign conflicts and retains only large-magnitude parameters. Twin-Merging (Lu et al., 2024) applies singular value decomposition to extract task-specific knowledge. Another line of work focuses on *model compression*. BitDelta (Liu et al., 2024) quantizes delta parameters to 1-bit by preserving only signs and a shared magnitude scalar. A third line of work focuses on *model enhancement*. EXPO (Zheng et al., 2024) extrapolates delta parameters with a scaling factor to improve alignment performance. While these methods achieve their respective goals through different operations, there remains limited understanding of what properties of delta parameters are essential for maintaining model performance. Our work aims to investigate this question through systematic experiments and provide insights that complement existing methods.

6 Conclusion

In this work, we investigated the properties of delta parameters in post-trained models along magnitude and sign. Through experiments across instruct language models, reasoning language models, and vision models, we find that delta parameters exhibit considerable editability. In the magnitude dimension, we observe that within a reasonable editing range, the mean of magnitude is the most sensitive factor, while individual values and distribution shape show less impact. In the sign dimension, we find that a substantial proportion of signs can be flipped while maintaining reasonable performance. To understand these phenomena, we proposed a loss-based local surrogate analysis using second-order Taylor expansion. This analysis introduces the concept of editing intensity, which helps explain the stability boundaries of different editing operations. Our findings provide insights for understanding and designing delta parameter editing methods. These findings suggest practical directions for future editing method design: practitioners may prioritize preserving the mean of magnitude and signs first, followed by relative relationships, as these are the most performance-sensitive factors. Moreover, post-training procedures could be explored to produce edit-friendly delta parameters whose performance relies more on aggregate statistics than on individual values.

Limitations

In this work, we investigated the editability of delta parameters across many models. While our analysis in Section 4 provides a loss-based local surrogate through a second-order Taylor expansion around W_{post}, it should be interpreted as a qualitative explanatory lens rather than a tight predictor of $\Delta\mathcal{L}$, particularly in the high editing intensity regime where higher-order terms and off-diagonal curvature interactions may become non-negligible. A more rigorous theoretical characterization would further strengthen our findings.

Acknowledgments

We sincerely thank the reviewers for their insightful comments and valuable suggestions. We are grateful to Liyuan Mao for insightful discussions. This work was supported by Beijing Natural Science Foundation (L243006), the Natural Science Foundation of China (No. 62476265, 62306303).

References

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, and 1 others. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.

Shoufa Chen, Chongjian Ge, Zhan Tong, Jiangliu Wang, Yibing Song, Jue Wang, and Ping Luo. 2022. Adaptformer: Adapting vision transformers for scalable visual recognition. In *Advances in Neural Information Processing Systems 35*.

Gong Cheng, Junwei Han, and Xiaoqiang Lu. 2017. Remote sensing image scene classification: Benchmark and state of the art. *Proceedings of the IEEE*, 105(10):1865–1883.

Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. 2014. Describing textures in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3606–3613.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, and 1 others. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Pala Tej Deep, Rishabh Bhardwaj, and Soujanya Poria. 2024. Della-merging: Reducing interference in model merging through magnitude-based sampling. *CoRR*, abs/2406.11617.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, and 181 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.

DeepSeek-AI, Aixin Liu, Bei Feng, Bin Wang, Bingxuan Wang, Bo Liu, Chenggang Zhao, Chengqi Dengr, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fuli Luo, Guangbo Hao, Guanting Chen, and 138 others. 2024. Deepseek-v2: A strong, economical, and efficient mixture-of-experts language model. *Preprint*, arXiv:2405.04434.

Mucong Ding, Chenghao Deng, Jocelyn Choo, Zichu Wu, Aakriti Agrawal, Avi Schwarzschild, Tianyi Zhou, Tom Goldstein, John Langford, Anima Anandkumar, and 1 others. 2024. Easy2hard-bench: Standardized difficulty labels for profiling llm performance and generalization. *Advances in Neural Information Processing Systems*, 37:44323–44365.

Jesse Dodge, Gabriel Ilharco, Roy Schwartz, Ali Farhadi, Hannaneh Hajishirzi, and Noah A. Smith. 2020. Fine-tuning pretrained language models: Weight initializations, data orders, and early stopping. *CoRR*, abs/2002.06305.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, and 82 others. 2024. The llama 3 herd of models. *CoRR*, abs/2407.21783.

Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. 2024. KTO: model alignment as prospect theoretic optimization. In *International Conference on Machine Learning*. PMLR.

Zeyu Han, Chao Gao, Jinyang Liu, Jeff Zhang, and Sai Qian Zhang. 2024. Parameter-efficient fine-tuning for large models: A comprehensive survey. *CoRR*, abs/2403.14608.

Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. 2019. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 12(7):2217–2226.

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 Túlio Ribeiro, Mitchell Wortsman, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. 2023. Editing models with task arithmetic. In *The Eleventh International Conference on Learning Representations*. OpenReview.net.

Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. 2024. Livecodebench: Holistic and contamination free evaluation of large language models for code. *arXiv preprint arXiv:2403.07974*.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *CoRR*, abs/2310.06825.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, and 7 others. 2024. Mixtral of experts. *Preprint*, arXiv:2401.04088.

Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 2013. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 554–561.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V. Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, Yuling Gu, Saumya Malik, Victoria Graf, Jena D. Hwang, Jiangjiang Yang, Ronan Le Bras, Oyvind Tafjord, Chris Wilhelm, Luca Soldaini, and 4 others. 2025. Tulu 3: Pushing frontiers in open language model post-training. *Preprint*, arXiv:2411.15124.

Yann LeCun, Corinna Cortes, and CJ Burges. 2010. Mnist handwritten digit database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, 2.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2023. Let's verify step by step. *Preprint*, arXiv:2305.20050.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2021. Truthfulqa: Measuring how models mimic human falsehoods. *arXiv preprint arXiv:2109.07958*.

James Liu, Guangxuan Xiao, Kai Li, Jason D. Lee, Song Han, Tri Dao, and Tianle Cai. 2024. Bitdelta: Your fine-tune may only be worth one bit. *CoRR*, abs/2402.10193.

Zhenyi Lu, Chenghao Fan, Wei Wei, Xiaoye Qu, Dangyang Chen, and Yu Cheng. 2024. Twin-merging: Dynamic integration of modular expertise in model merging. *CoRR*, abs/2406.15479.

Haipeng Luo, Qingfeng Sun, Can Xu, Pu Zhao, Jianguang Lou, Chongyang Tao, Xiubo Geng, Qingwei Lin, Shifeng Chen, and Dongmei Zhang. 2023. Wizardmath: Empowering mathematical reasoning for large language models via reinforced evol-instruct. *CoRR*, abs/2308.09583.

Ivan Moshkov, Darragh Hanley, Ivan Sorokin, Shubham Toshniwal, Christof Henkel, Benedikt Schifferer, Wei Du, and Igor Gitman. 2025. Aimo-2 winning solution: Building state-of-the-art mathematical reasoning models with openmathreasoning dataset. *Preprint*, arXiv:2504.16891.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Baolin Wu, Andrew Y Ng, and 1 others. 2011. Reading digits in natural images with unsupervised feature learning. In *NIPS workshop on deep learning and unsupervised feature learning*, volume 2011, page 4. Granada.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. *Preprint*, arXiv:2203.02155.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, and 25 others. 2025. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, and 1 others. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D. Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems 36*.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. 2024. Gpqa: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2021. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106.

Mark Sandler, Andrey Zhmoginov, Max Vladymyrov, and Andrew Jackson. 2022. Fine-tuning image transformers using learnable memory. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 12145–12154. IEEE.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *Preprint*, arXiv:2402.03300.

Johannes Stallkamp, Marc Schlipsing, Jan Salmen, and Christian Igel. 2011. The german traffic sign recognition benchmark: a multi-class classification competition. In *The 2011 international joint conference on neural networks*, pages 1453–1460. IEEE.

Qwen Team. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Yuxuan Tong, Xiwen Zhang, Rui Wang, Ruidong Wu, and Junxian He. 2024. Dart-math: Difficulty-aware rejection tuning for mathematical problem-solving. *CoRR*, abs/2407.13690.

Junqiao Wang, Zeng Zhang, Yangfan He, Zihao Zhang, Xinyuan Song, Yuyang Song, Tianyu Shi, Yuchen Li, Hengyuan Xu, Kunyu Wu, Xin Yi, Zhongwei Wan, Xinhang Yuan, Zijun Wang, Kuan Lu, Menghao Huo, Tang Jingqun, Guangwu Qian, Keqin Li, and 2 others. 2025. Enhancing code llms with reinforcement learning in code generation: A survey. *Preprint*, arXiv:2412.20367.

Zhichao Wang, Bin Bi, Shiva Kumar Pentyala, Kiran Ramnath, Sougata Chaudhuri, Shubham Mehrotra, Zixu, Zhu, Xiang-Bo Mao, Sitaram Asur, Na, and Cheng. 2024. A comprehensive survey of llm alignment techniques: Rlhf, rlaif, ppo, dpo and more. *Preprint*, arXiv:2407.16216.

Jianxiong Xiao, Krista A Ehinger, James Hays, Antonio Torralba, and Aude Oliva. 2016. Sun database: Exploring a large collection of scene categories. *International Journal of Computer Vision*, 119:3–22.

Yi Xin, Siqi Luo, Haodi Zhou, Junlong Du, Xiaohong Liu, Yue Fan, Qing Li, and Yuntao Du. 2024. Parameter-efficient fine-tuning for pre-trained vision models: A survey. *CoRR*, abs/2402.02242.

Shusheng Xu, Wei Fu, Jiaxuan Gao, Wenjie Ye, Weilin Liu, Zhiyu Mei, Guangju Wang, Chao Yu, and Yi Wu. 2024. Is DPO superior to PPO for LLM alignment? a comprehensive study. In *Forty-first International Conference on Machine Learning*.

Prateek Yadav, Derek Tam, Leshem Choshen, Colin A. Raffel, and Mohit Bansal. 2023. Ties-merging: Resolving interference when merging models. In *Advances in Neural Information Processing Systems 36*.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, and 43 others. 2024. Qwen2 technical report. *Preprint*, arXiv:2407.10671.

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 *International Conference on Machine Learning*. PMLR.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Wang Zhang, and 16 others. 2025. Dapo: An open-source llm reinforcement learning system at scale. *Preprint*, arXiv:2503.14476.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? *arXiv preprint arXiv:1905.07830*.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, and 3 others. 2023. A survey of large language models. *CoRR*, abs/2303.18223.

Wenting Zhao, Xiang Ren, Jack Hessel, Claire Cardie, Yejin Choi, and Yuntian Deng. 2024. Wildchat: 1m chatGPT interaction logs in the wild. In *The Twelfth International Conference on Learning Representations*.

Chujie Zheng, Ziqi Wang, Heng Ji, Minlie Huang, and Nanyun Peng. 2024. Weak-to-strong extrapolation expedites alignment. *CoRR*, abs/2404.16792.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. 2023. Instruction-following evaluation for large language models. *Preprint*, arXiv:2311.07911.

A Deriving the Rescale Factor in Equation 3

We derive the rescale factor used in Equation 3 from the same expected-output preservation intuition as in DARE. Consider a linear transformation $h = Wx + b$ with delta parameters $(\Delta W, \Delta b)$. We apply a coordinate-wise random scaling to the delta parameters: with probability p we multiply by k, and with probability $1 - p$ we multiply by an unknown factor γ. Let $M \sim \text{Bernoulli}(p)$ be the binary mask (element-wise) indicating which coordinates take the factor k.

After editing, the output becomes

$$\hat{h} = \big(W_{\text{pre}} + \Delta\widetilde{W}\big)x + \big(b_{\text{pre}} + \Delta\widetilde{b}\big),$$

where,

$$\Delta\widetilde{W} = k \cdot M \odot \Delta W \ + \ \gamma \cdot (1 - M) \odot \Delta W,$$
$$\Delta\widetilde{b} = k \cdot M \odot \Delta b \ + \ \gamma \cdot (1 - M) \odot \Delta b.$$

Taking expectation over the editing randomness (i.e., over M), we have $\mathbb{E}[M] = p$ and $\mathbb{E}[1 - M] = 1 - p$. Thus

$$\mathbb{E}[\Delta\widetilde{W}] = \Big(pk + (1 - p)\gamma\Big)\Delta W,$$
$$\mathbb{E}[\Delta\widetilde{b}] = \Big(pk + (1 - p)\gamma\Big)\Delta b.$$

DARE-style expected-output preservation requires the expected delta contribution to match the original delta contribution, i.e., $\mathbb{E}[\Delta\widetilde{W}] = \Delta W$ and $\mathbb{E}[\Delta\widetilde{b}] = \Delta b$. This yields a single scalar condition:

$$pk + (1 - p)\gamma = 1.$$

Solving for γ gives the rescale factor used in Equation 3:

$$\gamma = \frac{1 - kp}{1 - p}.$$

B Full Experimental Results

We conduct a thorough experimental validation on the editability of delta parameters. The results of LLaMA3-8B-Instruct, Mistral-7B-Instruct-v0.3, ViT-B-32 and Qwen3-1.7B across eight benchmarks are presented in Figure 8, Figure 9, Figure 10, and Figure 11, Figure 12, respectively.

The full results of LLaMA-3-8B-Instruct on all datasets for experiments with varying magnitude mean and distribution shape are shown in Figure 13.

C Derivations for Editing Intensity

This appendix provides derivations for the results in Section 4. We start from the two-point distribution of the editing perturbation already given in the main text:

$$e_i = \begin{cases} (k-1)\Delta w_i & \text{with probability } p, \\ \frac{p(1-k)}{1-p}\Delta w_i & \text{with probability } 1 - p. \end{cases}$$

We take expectation/variance over the editing randomness, treating $(g, \Delta W)$ as fixed for the local surrogate.

C.1 First-order term: $\mathbb{E}[g^\top e] = 0$

We compute $\mathbb{E}[e_i]$ directly:

$$\begin{aligned}\mathbb{E}[e_i] &= p(k-1)\Delta w_i + (1-p)\frac{p(1-k)}{1-p}\Delta w_i \\ &= p(k-1)\Delta w_i + p(1-k)\Delta w_i \\ &= 0.\end{aligned}$$

Therefore,

$$\mathbb{E}[g^\top e] = \mathbb{E}\left[\sum_i g_i e_i\right] = \sum_i g_i\,\mathbb{E}[e_i] = 0.$$

C.2 Variance of the first-order term

We assume the coordinate-wise editing randomness is independent across i (e.g., induced by an i.i.d. Bernoulli mask). Since $\mathbb{E}[e_i] = 0$, we have $\text{Var}(e_i) = \mathbb{E}[e_i^2]$. First compute $\mathbb{E}[e_i^2]$:

$$\begin{aligned}\mathbb{E}[e_i^2] &= p(k-1)^2(\Delta w_i)^2 + (1-p)\left(\frac{p(1-k)}{1-p}\right)^2(\Delta w_i)^2 \\ &= (1-k)^2(\Delta w_i)^2\left(p + \frac{p^2}{1-p}\right) \\ &= (1-k)^2(\Delta w_i)^2 \cdot \frac{p}{1-p}.\end{aligned}$$

Now let $X_i \triangleq g_i e_i$. Under independence across i, $\text{Var}(\sum_i X_i) = \sum_i \text{Var}(X_i)$. Thus:

$$\begin{aligned}\text{Var}(g^\top e) &= \text{Var}\left(\sum_i g_i e_i\right) = \sum_i \text{Var}(g_i e_i) \\ &= \sum_i g_i^2\,\text{Var}(e_i) = \sum_i g_i^2\,\mathbb{E}[e_i^2] \\ &= \frac{p}{1-p}(1-k)^2\sum_i (g_i \Delta w_i)^2.\end{aligned}$$

C.3 Expected second-order term under randomized masking

As shown in Section 4.2, under i.i.d. Bernoulli masking, $\mathbb{E}[e_i] = 0$ and $\mathbb{E}[e_i e_j] = 0$ for $i \neq j$. Therefore, for any symmetric curvature proxy

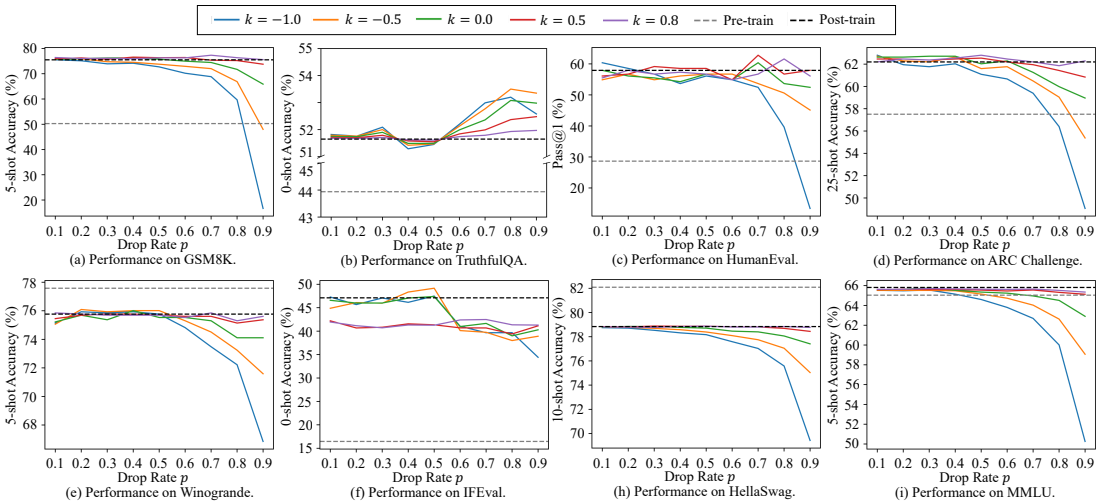


Figure 8: The performance of LLaMA3-8B-Instruct on the all benchmarks under varying p and k.

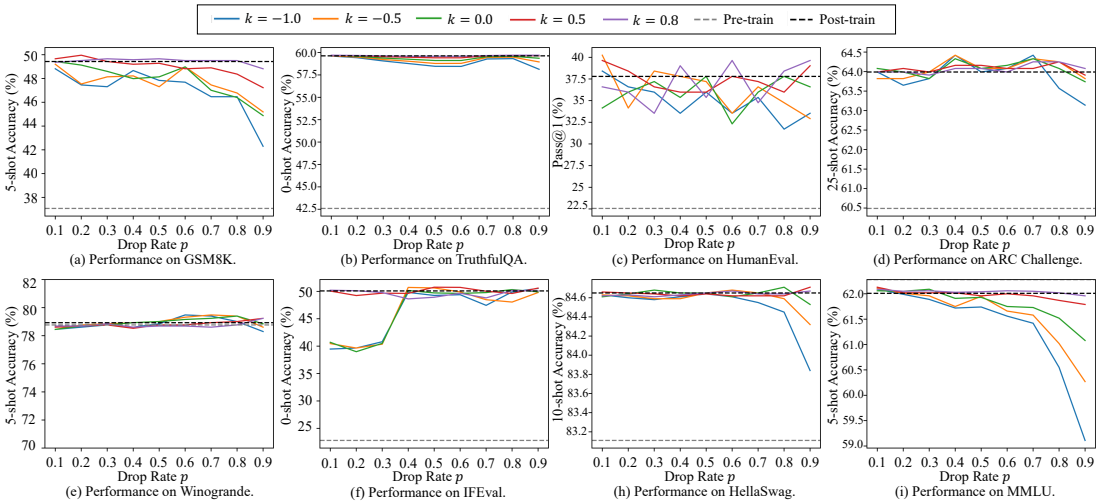


Figure 9: The performance of Mistral-7B-Instruct-v0.3 on the all benchmarks under varying p and k.

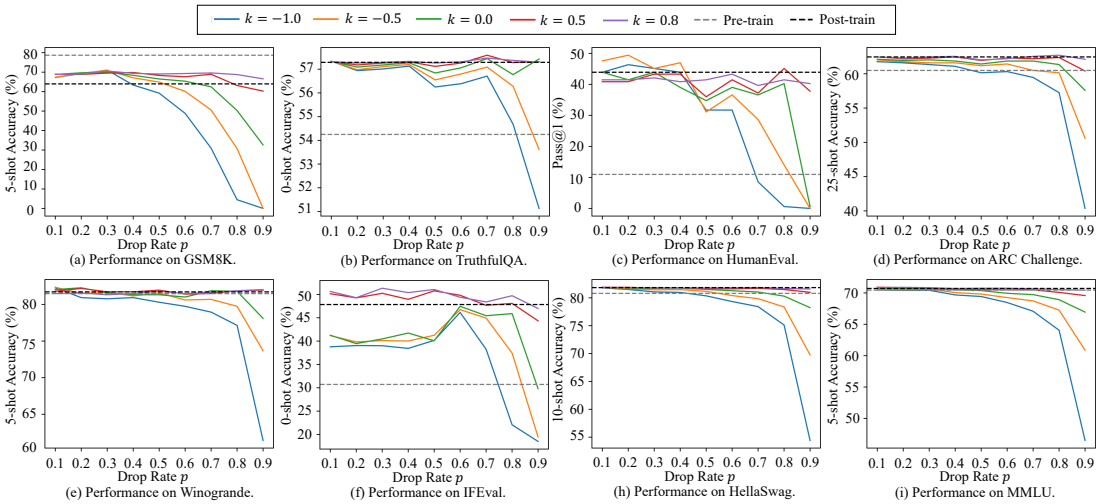


Figure 10: The performance of Qwen2-7B-Instruct on the all benchmarks under varying p and k.

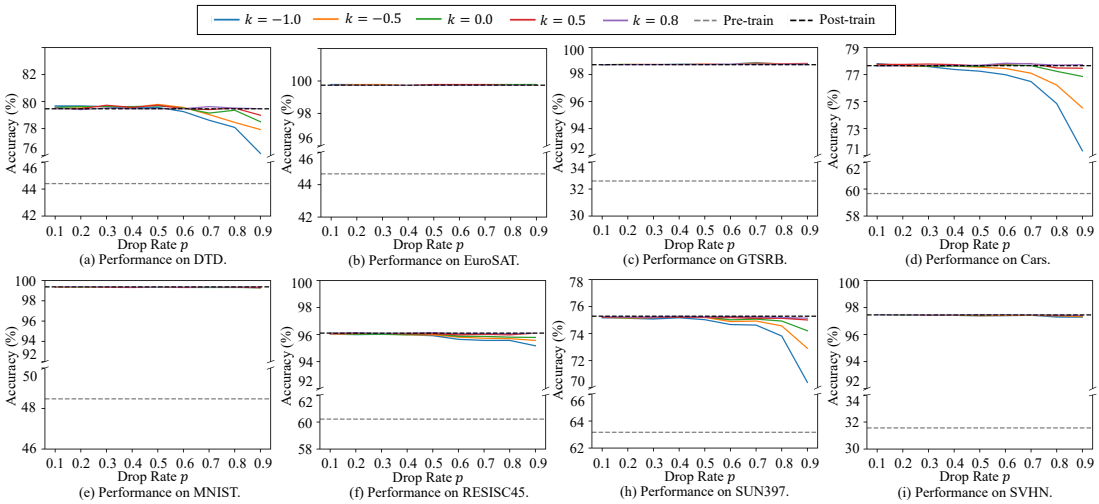


Figure 11: The performance of ViT-B-32 on the all benchmarks under varying p and k.

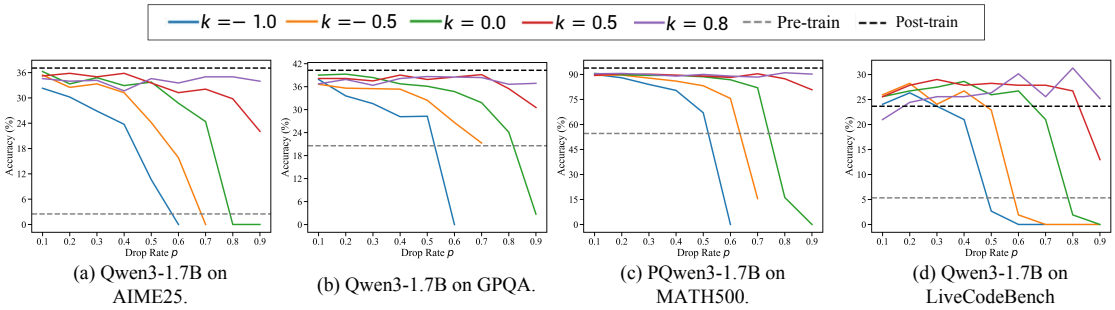


Figure 12: The performance of Qwen3-1.7B on the all benchmarks under varying p and k.

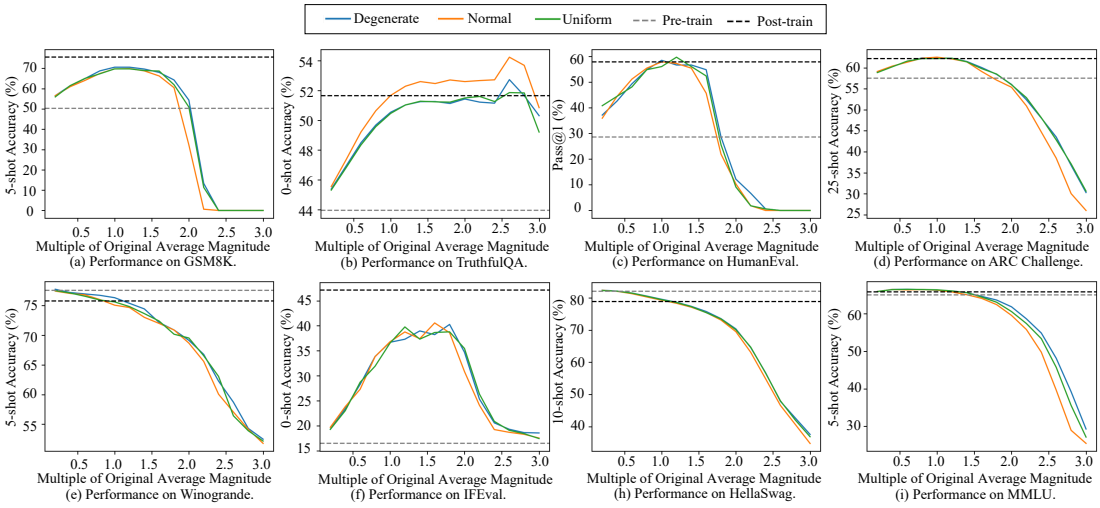


Figure 13: The full results on all datasets for experiments with varying magnitude mean and distribution shape.

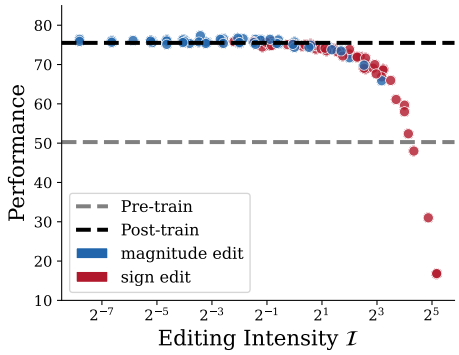


Figure 14: The relationship between editing intensity and performance in log-scale on LLaMA-3-8B-Instruct using GSM8K benchmark.

$C \succeq 0$, $\mathbb{E}[e^\top Ce] = \sum_i C_{ii}\,\mathbb{E}[e_i^2]$. Denoting $s_i = C_{ii} \geq 0$, we have:

$$\mathbb{E}\left[\frac{1}{2}\sum_i s_i e_i^2\right] = \frac{1}{2}\sum_i s_i\,\mathbb{E}[e_i^2] = \frac{1}{2}\cdot\frac{p}{1-p}(1-k)^2\sum_i s_i(\Delta w_i)^2.$$

This shows that both $\mathrm{Var}(g^\top e)$ and $\mathbb{E}\left[\sum_i s_i e_i^2\right]$ share the same (p,k)-dependent factor. We therefore define the editing intensity:

$$\mathcal{I}(p,k) \triangleq \frac{p}{1-p}(1-k)^2.$$

D The Use of Large Language Models

We utilized LLMs to aid and polish writing.

E Additional Experimental Results

E.1 Experiments on Larger-Scale and MoE Models

To verify whether our findings generalize to larger-scale and mixture-of-experts (MoE) architectures, we conduct additional experiments on LLaMA-3-70B-Instruct (Dubey et al., 2024) and Mixtral-8x7B-Instruct (Jiang et al., 2024) using the GSM8K benchmark. We test representative settings with $p \in \{0.3, 0.5, 0.7, 0.9\}$ under magnitude editing ($k = 0.5$) and sign editing ($k = -0.5$). Results are shown in Table 2.

The results are consistent with our findings on smaller models. For magnitude editing ($k = 0.5$), both models maintain performance close to the post-trained model even at $p = 0.9$. For sign editing ($k = -0.5$), performance remains stable at moderate p values but degrades at $p = 0.9$, consistent with the higher editing intensity of sign-flip operations. These results suggest that the editability of delta parameters extends to both larger-scale and MoE architectures.

Model	Post-train	k	p=0.3	p=0.5	p=0.7	p=0.9
LLaMA-3-70B	91.28	0.5	91.05	91.05	91.43	91.21
LLaMA-3-70B	91.28	−0.5	91.05	90.75	90.83	86.73
Mixtral-8x7B	64.29	0.5	65.35	64.06	64.06	63.53
Mixtral-8x7B	64.29	−0.5	64.59	64.82	63.84	59.29

Table 2: Performance (GSM8K accuracy) of larger-scale and MoE models under magnitude editing ($k = 0.5$) and sign editing ($k = -0.5$) with varying drop rates p.

E.2 Full Results on Relative Relationships

We present the complete shuffle experiment results across six tasks on LLaMA-3-8B-Instruct in Table 3. The shuffle rate r varies from 0.0 (no shuffle) to 1.0 (full shuffle).

r	ARC	GSM8K	HellaSwag	MMLU	TruthfulQA	Winogrande
0.0	62.20	75.51	78.84	65.82	51.65	75.77
0.1	62.63	75.36	79.01	65.85	51.67	75.93
0.3	62.71	75.13	79.13	65.93	50.92	76.40
0.5	62.80	73.01	79.42	66.09	50.70	75.69
0.7	62.80	73.24	79.40	66.09	50.99	76.32
1.0	63.05	70.13	79.41	65.98	50.65	75.85

Table 3: Performance of LLaMA-3-8B-Instruct across six tasks under varying shuffle rates.

GSM8K is the most sensitive task to shuffling, dropping approximately 5 points from $r = 0.0$ to $r = 1.0$, while tasks such as ARC-Challenge, HellaSwag, MMLU, and Winogrande remain largely unaffected even at 100% shuffle rate. This difference is likely related to task characteristics: reasoning-intensive tasks like GSM8K may depend more on fine-grained positional relationships among delta parameters, while knowledge-oriented tasks may rely more on aggregate statistics such as the mean magnitude.

E.3 Failure Analysis Beyond Stability Boundaries

We conduct two complementary analyses on LLaMA-3-8B-Instruct using GSM8K to examine how performance degrades as editing boundaries are exceeded.

Difficulty-Stratified Analysis. We partition GSM8K problems into three difficulty bins (Easy/Medium/Hard) following Ding et al. (2024), and measure accuracy under low (p=0.5, k=0.0),

medium (p=0.9, k=0.0), and high (p=0.9, k=$-$0.5) editing intensity. Results are shown in Table 4.

Difficulty	Post-train	Low Edit	Mid Edit	High Edit
Easy	88.84%	90.21% (↑1.5%)	85.42% (↓3.9%)	33.26% (↓62.6%)
Medium	78.41%	78.64% (↑0.3%)	66.14% (↓15.6%)	13.18% (↓83.2%)
Hard	58.64%	58.18% (↓0.8%)	46.59% (↓20.6%)	4.32% (↓92.6%)

Table 4: Difficulty-stratified accuracy under different editing intensities on GSM8K. Percentages in parentheses denote the relative change w.r.t. the post-trained model (↑ for improvement, ↓ for degradation).

Under low editing, all difficulty levels remain virtually unchanged. As editing intensity increases, harder problems degrade disproportionately faster—under high editing, hard problems lose 92.6% of their accuracy while easy problems retain about one-third. This suggests that complex reasoning capabilities are more fragile to delta parameter editing.

Error Type Analysis. We further sample 100 incorrect responses per editing level and classify errors into four categories to understand the qualitative shift in failure modes. Results are shown in Table 5.

Error Type	Low Edit	Mid Edit	High Edit
Arithmetic/Calculation Errors	25%	23%	14%
Localized Reasoning Errors	45%	26%	17%
Fundamental Understanding Errors	24%	41%	52%
Degenerate Generation	6%	10%	17%

Table 5: Distribution of error types across editing intensity levels on GSM8K. *Localized Reasoning Errors* refer to cases where the overall approach is correct but a specific step fails. *Fundamental Understanding Errors* refer to cases where the problem modeling itself is wrong.

As editing intensity increases, the dominant failure mode shifts from localized step-level mistakes (45%→17%) to fundamental misunderstanding of the problem (24%→52%), accompanied by the emergence of degenerate generation (6%→17%) where the model enters repetitive loops. This reveals a progressive degradation pattern—from *careless errors* to *conceptual failures* to *generation collapse*—suggesting that more complex capabilities are less robust to delta parameter editing and degrade earlier than simpler ones.