How to escape the alignment resistance dilemma of LLM: An extended elasticity theory from data dimension perspective

Anonymous authors
Paper under double-blind review

Abstract

Alignment in large language models can be rapidly undermined by subsequent fine-tuning, a phenomenon known as alignment resistance. Existing elasticity theory explains, from the perspective of data compression, why models tend to drift toward the pretraining distribution, and establishes the relationship between the scale gap separating pretraining data and alignment data and the rate of drift. However, even when the sizes of the pretraining data and the alignment dataset are held constant, the strength of resistance to rebound still varies across. To investigate this problem, we propose *Alleviation Elasticity Theory*. This theory inherits the "prediction is compression" perspective from the original elasticity theory, and further employs mathematical expressions to characterize the amount of abstract information contained in datasets as well as the amount of data information absorbed by the model after alignment, which we term the alignment data dimension and the internal model representation dimension, respectively. Subsequently, using entropy as a bridge, we incorporate both into the weights of the joint data distribution, thereby obtaining a more precise characterization of the rate at which the model drifts toward the pretraining data distribution after being perturbed by fine-tuning. This theory provides a theoretical basis for identifying methods to mitigate alignment elasticity, namely that reducing the dimension of the alignment data or that of the internal model representation can improve alignment stability. While holding the sizes of the pretraining data and the alignment dataset constant, our experiments compare datasets of different dimensions and under different alignment methods, not only validating the theory but also demonstrating the effectiveness, under the guidance of this theory, of enhancing large models' resistance to rebound. Code at `https://anonymous.4open.science/r/Alleviation-Elasticity-Theory-278F`.

1 Introduction

Large language models typically acquire instruction-following behavior, adherence to user preferences, and safety constraints through alignment training. Deployment also requires further fine-tuning to accommodate domain-specific data and new tasks, raising a central question: can a model retain the alignment established in an earlier stage after being fine-tuned on a new task? Qi et al. (2024) show that even a small amount of subsequent fine-tuning can weaken certain aligned behaviors, and that benign task data can cause similar degradation. This fragility limits further adaptation: strong alignment at one stage does not guarantee that alignment will remain stable under subsequent updates. We refer to this problem as elastic resistance to alignment in large language models, or alignment rebound.

Elastic resistance to alignment is closely related to catastrophic forgetting in machine learning, but alignment rebound in large language models is not merely forgetting. Catastrophic forgetting concerns declining performance on old tasks as a model learns new ones. Elastic weight consolidation protects previously acquired knowledge by constraining changes to important parameters (Kirkpatrick et al., 2017), while gradient episodic memory constrains parameter updates using stored examples from old tasks to reduce interference with existing capabilities (Lopez-Paz & Ranzato, 2017). These approaches offer important insights into knowledge retention, but a decline in old-task perfor-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

mance alone does not identify the direction in which the model distribution changes. Alignment rebound additionally concerns a tendency for the model distribution to approach the pretraining distribution. Its analysis therefore requires not only an account of interference between old and new tasks, but also an explanation of how pretraining and alignment shape subsequent updates.

Research on shallow safety alignment has sought to improve robustness to subsequent fine-tuning by deepening alignment or constraining updates at initial output positions (Qi et al., 2025). Ji et al. (2025) further view language models as data compressors and, under specific distributional assumptions, establish a relationship between pretraining and alignment dataset sizes and the strength of alignment elasticity. Their analysis explains why alignment distributions are more readily disrupted than pretraining distributions. However, the data-size ratio characterized by this relationship is not strictly proportional to elasticity strength. Ji et al. (2025) also identify the detailed characterization of real-world data as an important obstacle to translating theory into practical algorithms. Further investigation is therefore needed to move from explaining rebound to mitigating it.

Our preliminary experiments show that large language models with the same architecture and pretraining scale can exhibit different degrees of alignment rebound after alignment on equally sized datasets. Under an identical subsequent fine-tuning protocol, replacing the alignment corpus with another of equal size or changing the alignment method produces different alignment-score trajectories. This observation suggests that corpus selection and alignment methods merit investigation beyond dataset size alone. Intrinsic dimension provides a mathematical tool for describing such differences: it characterizes the effective degrees of freedom and internal geometry of data in a chosen representation space and at a specified measurement scale. We use *dataset dimension* to denote the dimension estimated by GRIDE after a fixed reference encoder maps text into vectors, and *model-perceived dimension* to denote the intrinsic dimension of representations within the model's Transformer layers (Cheng et al., 2025). The former compares corpus structure; the latter captures differences in internal representations of the same corpus across model states. The study of a high-dimensional abstraction phase in language Transformers by Cheng et al. (2025) shows that layerwise intrinsic dimension can reflect changes in representational structure and function, providing a basis for this analysis. This motivates our central question: can dataset dimension and model-perceived dimension explain rebound differences beyond data scale and suggest testable interventions for mitigating alignment elasticity?

To help large language models overcome resistance to alignment, we extend existing elasticity theory by incorporating dataset dimension and model-perceived dimension into the elasticity coefficient k. Entropy connects the mathematical characterization of dataset dimension to that of data properties, allowing data scale, dataset dimension, and model-perceived dimension to jointly influence elasticity in the resulting theorem.

Our contributions are as follows: **Theory:** To alleviate model resistance to alignment, we extend the existing elasticity theory to *Alleviation Elasticity Theory*: this theorem uses information entropy as a tool to establish a connection between the abstract information contained in the dataset and model compression, characterizing the ratio of the model's tendency to rebound towards the pre-training data versus the aligned data after fine-tuning. **Method:** Based on the Resistance Alignment Theorem, we present effective method to enhance robustness of alignment: either shrink the dataset' s own dimensions or reduce how the model represents the dataset. This will help guide how we choose training dataset, and training strategy for the particular alignment task. **Experiments:** Through experimental designs, such as controlled alignment datasets or methods similar to those in elasticity theory, we demonstrate the validity of the theoretical conclusions; we also prove that the proposed methods for alignment selection can effectively enhance the robustness of the alignment.

2 Related Work

2.1 Forgetting and Continual Learning in Machine Learning

A decline in old-task performance when a model learns a new task is a central problem in continual learning, commonly known as catastrophic forgetting. Replay mechanisms, parameter constraints, and restrictions on update directions protect existing capabilities at the levels of data, parameters, and optimization, respectively. Elastic weight consolidation uses parameter importance to reduce drift in critical parameters (Kirkpatrick et al., 2017), while gradient episodic memory constrains the damage

that new-task updates can cause to old tasks (Lopez-Paz & Ranzato, 2017). These studies indicate that capability retention depends both on previously acquired knowledge and on how subsequent training acts on it. Alignment rebound is related to catastrophic forgetting but emphasizes a different perspective: aligned behavior may move toward an earlier pretrained state, rather than merely exhibit a decline in an arbitrary old-task metric.

Among specific approaches, Learning without Forgetting distills the old model's outputs on new-task data, constraining changes in old-task predictions without accessing the old training data (Li & Hoiem, 2016). iCaRL combines representative exemplar storage with distillation for class-incremental learning (Rebuffi et al., 2017). Deep generative replay trains jointly on generated old-task examples and new-task data, reducing reliance on a complete historical dataset (Shin et al., 2017). Unlike these data- or output-level retention mechanisms, Synaptic Intelligence accumulates parameter contributions along the training trajectory and uses them to constrain subsequent changes to important parameters (Zenke et al., 2017). These methods offer approaches based on compensating for changes in data distributions, preserving outputs, and protecting parameters, providing useful comparisons for the study of alignment elasticity.

2.2 Alignment Rebound and Representational Structure in Language Models

Studies of alignment fragility show that subsequent fine-tuning can alter established safety behaviors and analyze this vulnerability through sequence structure and training positions (Qi et al., 2024; 2025). Ji et al. (2025) investigate the phenomenon from a compression perspective and develop a theory of elastic resistance to alignment. Other fine-tuning studies provide further evidence of this risk. Shadow Alignment shows that fine-tuning on a small number of harmful examples can weaken safety behaviors in multiple aligned models while preserving their ability to answer ordinary questions (Yang et al., 2023). To address degradation under subsequent fine-tuning, Vaccine introduces constructed perturbations into hidden representations during alignment to improve robustness to later harmful fine-tuning (Huang et al., 2024). Safe LoRA uses the weight difference between base and aligned models to construct an alignment subspace, then projects LoRA updates in selected layers after fine-tuning to reduce safety degradation (Hsu et al., 2024).

The connection between probabilistic prediction and code length makes compression a useful framework for analyzing language models (Shannon, 1948; Delétang et al., 2024). The relationship between quantization entropy and information dimension provides a basis for incorporating geometric structure (Rényi, 1959). GRIDE estimates intrinsic dimension from ratios of nearest-neighbor distances at different neighbor orders (Denti et al., 2022), and layerwise representation analysis applies this tool to language Transformers (Cheng et al., 2025). Theorem 4.2 of Ji et al. (2025) establishes an important relationship between model elasticity and the ratio of pretraining to alignment dataset sizes. Unlike other studies on model alignment, we approach from a perspective closer to LLM training methods, and for the first time, we investigate at the levels of dataset dimension and model perception dimension how these factors affect model alignment during fine-tuning. These studies propose interventions based on representational robustness during alignment and constraints on subsequent updates, directly informing our investigation of the relationship between training methods and alignment elasticity. We use low-rank adaptation and bias-only fine-tuning as interventions that change model-perceived dimension (Hu et al., 2022; Ben Zaken et al., 2022). For intrinsic-dimension estimation, TWO-NN uses the ratio between the second-nearest and nearest neighbor distances to estimate data-manifold dimension under local density conditions and discusses the scale dependence of the estimates (Facco et al., 2017).

3 Theoretical Analysis of Alignment Elasticity

This section will first introduce the main core of this paper and the *Alleviation Elasticity Theory*; then present two important steps for proving the elasticity theorem; finally, it ends with specific methods for enhancing alignment robustness based on this theorem.

3.1 Alleviation Elasticity Theory.

Alleviation Elasticity Theory describes the model's behavior during fine-tuning (perturbation). After pretraining and alignment, the theorem characterizes the speed at which the model approaches the pretraining and alignment data distributions. The speed is called the elasticity coefficient.

Theorem 1. ***Alleviation Elasticity Theory.*** *Let $D_p = D_1 \cup D_2 \cup D_3$, where the three subsets have equal sizes. Alignment data D_a follow the same distribution as D_2, and perturbation data D_t follow the same distribution as D_3. Model p_θ is trained on $D = D_p \cup D_a \cup D_t$, and the leaf masses of the pruned trees for the three components follow Pareto distributions. The elasticity coefficient is κ. The variation in the amount of perturbation data gives*

$$\begin{aligned} \frac{d\gamma_{p_\theta}^{D_a/D}}{d\lambda} &= \Theta\left(-\kappa \frac{d\gamma_{p_\theta}^{D_p/D}}{d\lambda}\right), \\ \frac{d\gamma_{p_\theta}^{D_p/D}}{d\lambda} < 0, \quad \frac{d\gamma_{p_\theta}^{D_a/D}}{d\lambda} &> 0, \\ \lambda = |D_t| H_t G(\mu_t) / |D_a| H_a G(\mu_a) &\ll 1, \\ \kappa = |D_p| H_p G(\mu_p) / |D_a| H_a G(\mu_a) &\gg 1. \end{aligned} \tag{1}$$

Here, D_p, D_a, D_t denote the pretraining, alignment, and subsequent fine-tuning datasets, respectively; $|D_i|$ denotes dataset size; H_i is the leaf-node entropy of each dataset (will be illustrated below); μ_i is the model-perceived dimension of the dataset; and G maps model-perceived dimension to its contribution to the weight. The model's normalized compression rate for a dataset is $\gamma_{p_\theta}^{D_i/D} = \gamma_{p_\theta}^{D_i} - \log M$, where M is the number of leaves in the corresponding pruned tree. A smaller normalized compression rate indicates better compression of the dataset, or equivalently, more accurate prediction on that dataset.

Definition 3.1 Token Tree $\mathcal{T}$. For a dataset $\mathcal{D} = \{z_i \in \{0|1\}^\infty \mid i = 1, 2, \cdots\}$, the token tree of $\mathcal{D}$, denoted as $\mathcal{T}_\mathcal{D}$, is defined as follows: each node has child nodes labeled 0 or 1, along with an end-of-sequence (EOS) leaf node. The path from the root to a leaf node defines each response z_i, with the corresponding EOS node weight representing the response's probability while the weight of non-leaf nodes is the sum of the weights of their child nodes.

Definition 3.1 describes the token-tree representation of a dataset used in elasticity theory (Ji et al., 2025). In this representation, model training is conceptualized as learning the weights of token-tree nodes. Because practical models have finitely many parameters, they cannot accurately capture node weights at arbitrary depths. The depth d of the portion of the token tree that a model can represent perfectly is therefore assumed to increase monotonically with model size. The leaf-node entropy of the token-tree representation of dataset D is defined as

$$H(D) = -\sum_{i=1}^{n} P_i \log P_i. \tag{2}$$

Here, P_i denotes the probability of a token-tree leaf, and the sum runs over all leaves. Leaf-node entropy supplements the amount of data with how much distinguishable structural information the data contains at a specified scale.

The elasticity coefficient κ in Equation (1) is lasticity coefficient. From the perspective of data compression, it represents the ratio of weights between the pretraining dataset and the aligned dataset, where the weights include dataset size, leaf node entropy, and the model's perceived dimensions. In the section 3.2, we explain why these weights are included in the elasticity coefficient κ. In the section 3.3, we show that leaf node entropy is equivalent to the dataset dimensions.

3.2 Characterization of the Elastic Coefficient

The elasticity coefficient is essentially the ratio of the weights after jointly compressing the pretraining dataset and the alignment dataset. In this section, we first illustrate these weight is characterized by the dataset's weights, as well as the internal geometric structure characterized by the entropy of its leaf nodes. We then present the certain conditions must be met for the theorem to hold.

3.2.1 Constructing the Effective Volume of the Joint Distribution

When each leaf index is encoded with an ideal marginal code, the expected structural code length for n_i samples is $n_i H_i(s)$. The two factors have distinct origins: n_i counts samples, while H_i measures the average structural code length required per sample.

The same input structure need not impose the same effective constraints under different training methods. To preserve this distinction, we introduce a factorization assumption that requires empirical calibration: the effective contribution of a unit of structural code length to the current model can be approximated by a response function G of internal dimension. For a fixed architecture, scale, and comparison range, define

$$V_i(s,\theta) = n_i H_i(s) G(\mu_i(\theta)), \qquad \kappa = \frac{V_p}{V_a} = \frac{n_p}{n_a}\frac{H_p G(\mu_p)}{H_a G(\mu_a)}. \tag{3}$$

Here, $G > 0$ maps model-perceived dimension to its contribution to effective volume. Equation (3) separates sample count, data structure, and model representation: the first factor reflects quantity, the second derives from data dimension, and the third maps representation to effective contribution. A simple choice is $G(\mu) = \mu$.

3.2.2 Conditions for Incorporating the Effective Contribution Ratio into the Theorem

Fix H_i and $G(\mu_i)$ at the reference state after alignment, and write $\lambda = V_t/V_a$. Let X_i denote the leaf-mass variable in the compression model. The mixture weighted by effective contributions can then be written as

$$X_D = \frac{V_p X_p + V_a X_a + V_t X_t}{V_p + V_a + V_t} = \frac{\kappa X_p + X_a + \lambda X_t}{\kappa + 1 + \lambda}. \tag{4}$$

The second equality follows by dividing both the numerator and denominator by V_a, yielding κ and λ. If $\kappa \gg 1, \lambda \ll 1$ and the reweighted mass variables and pretraining decomposition still satisfy the original assumptions and associated conditions, κ, λ can replace k, l, giving the following conditional relationship in this surrogate compression model:

$$\frac{\partial \gamma_a}{\partial \lambda} = \Theta\left(-\kappa \frac{\partial \gamma_p}{\partial \lambda}\right), \qquad \lambda = \frac{H_t G(\mu_t)}{H_a G(\mu_a)}\, l. \tag{5}$$

γ_i retains its meaning as a normalized compression rate. When the structural factors are locally fixed, the chain rule for changing from λ back to l introduces the same coefficient on both sides; the additional constant derivative factor does not invalidate the theorem. A mixture expression of the same form must also preserve the original Pareto parameter distributions and independence conditions.

When $H_i G(\mu_i)$ is the same across all data sources, V_i is proportional to sample count, and hence $\kappa = k$ and $\lambda = l$. When these factors differ, equally sized pretraining and alignment datasets no longer imply equal effective contributions to the joint compression distribution.

3.3 Leaf-Node Entropy and Dataset Dimension

This section primarily formalizes the relationship between leaf node entropy and the dimensionality of the dataset. The proof approach connects the two by linking the dataset to the covering number, the covering number to the quantized entropy, and the quantized entropy to the leaf node entropy. Detailed proofs appear in Appendix B.

3.3.1 From Dataset Dimension to Covering Numbers

Fix a reference encoder E, a model that maps text into vectors, and let $Z_i = E(X_i)$, where $X_i \sim P_i$ and Z_i is the vector representation of the text. Assume that the support $\mathcal{M}_i$ of $Z_i = E(X_i)$ is a compact δ_i-dimensional manifold with finite positive volume. Its volume measure ν_i satisfies uniform small-ball bounds: $\nu_i(B_i(z,r))$ is comparable to r^{δ_i}, with constants independent of the center and of sufficiently small radii. Let $N_i(\epsilon_s)$ be the minimum number of radius-ϵ_s balls needed to cover $\mathcal{M}_i$. Then

$$N_i(\epsilon_s) = \Theta(\epsilon_s^{-\delta_i}), \qquad \log_2 N_i(\epsilon_s) = \delta_i \log_2(1/\epsilon_s) + O(1). \tag{6}$$

This characterizes the number of distinguishable geometric regions at a given resolution, without yet considering their probability masses. See Appendix B.1 for the proof.

3.3.2 From Covering Numbers to Quantization Entropy

Further assume that the distribution of Z_i has a density f_i with respect to ν_i, with resolution-independent bounds $0 < \underline{f}_i \leq f_i \leq \overline{f}_i < \infty$. Construct nearest-center quantization cells from a maximal separated set, and denote the cell index by $U_{i,s}$ and its probabilities by $q_{i,s}(u)$. The geometric and density conditions give $c_i \epsilon_s^{\delta_i} \leq q_{i,s}(u) \leq C_i \epsilon_s^{\delta_i}$, where c_i, C_i are positive constants independent of resolution. The quantization entropy therefore satisfies

$$\begin{aligned} H_i(s) := H(U_{i,s}) &= -\sum_u q_{i,s}(u) \log_2 q_{i,s}(u) \\ &= \delta_i \log_2(1/\epsilon_s) + O(1). \end{aligned} \tag{7}$$

Cell probabilities thus connect geometric covering complexity to average code length, rather than directly equating entropy with the logarithm of the covering number. See Appendix B.2.

3.3.3 From Quantization Entropy to Leaf-Node Entropy

Using the same sampling unit and distribution, let $U = U_{i,s}$ denote the quantization-cell index and $J = J_{i,d(s)}$ the token-tree leaf index at pruning depth $d(s)$. Assume additionally that finite, scale-independent positive integers K_1, K_2 exist: each leaf corresponds to at most K_1 quantization cells, and each cell to at most K_2 leaves, counting only correspondences with positive joint probability. Conditional-entropy bounds and the entropy chain rule give

$$-\log_2 K_1 \leq H(J) - H(U) \leq \log_2 K_2. \tag{8}$$

Combining this with Eq. (7) yields

$$H_i^{\text{tok}}(d(s)) := H(J_{i,d(s)}) = \delta_i \log_2(1/\epsilon_s) + O(1). \tag{9}$$

The remainders are uniformly bounded with respect to resolution, but may depend on the data distribution, geometric conditions, and partition correspondence constants; see Appendix B.3 for the full proof. Transferring the dimension dependence to leaf-node entropy thus relies on these explicit conditions and provides a conditional basis for the effective compression-volume analysis that follows.

3.4 Inspiration from the Alleviation Elasticity Theory

By the definition of κ, aligning a model on a lower-dimensional dataset decreases κ. If the change in KL divergence between the model and pretraining distributions remains nearly unchanged, the change in KL divergence between the model and alignment distributions becomes smaller. The rate of change in the model's compression rate on the alignment dataset therefore decreases, weakening elasticity. Similarly, when different alignment methods are used, a method that produces a lower model-perceived dimension reduces the rate of change in the alignment-dataset compression rate, thereby mitigating elasticity.

4 Experiments

4.1 Experimental Setup

All three experiments use DistilGPT2 as the trainable base model and IMDb as the sentiment data source (Maas et al., 2011). A frozen SST-2 DistilBERT classifier evaluates the model's generated continuations. The alignment stage uses supervised fine-tuning on positive reviews, followed by subsequent fine-tuning on negative reviews.

We generate one continuation for each fixed prompt and provide only the continuation to the scorer. If m of M continuations are classified as positive, the model's positive rate is $S = m/M$. We use this proportion to measure alignment at each evaluation point. Let $S(0)$ denote the score after alignment but before subsequent fine-tuning. To compare models with different initial scores, we also use

$$P(t) = \frac{S(t)}{S(0)}, \qquad R(t) = \frac{S(t) - B}{S(0) - B} \quad (S(0) > B). \tag{10}$$

P expresses the raw score as a fraction of its starting value; R uses the experiment's own base-model score B to measure alignment-gain retention. Dataset dimension is computed using GRIDE after the same frozen model maps text semantics into vectors. Model-perceived dimension is measured in the aligned model as the largest dimension across its Transformer layers.

4.2 Experiment 1: Alignment Dataset Dimension and Initial Alignment

This experiment examines initial alignment scores after alignment on datasets of different dimensions, without subsequent negative-sample fine-tuning. We construct candidate subsets of different dimensions from the same pool of 6,000 positive examples and select 12 alignment datasets with dimensions distributed evenly from 16.53 to 24.32. Each group contains 2,000 examples. All groups start from identical weights and undergo 125 full-parameter SFT updates with the same learning rate of 2×10^{-4} and batch size of 16, visiting all 2,000 example positions in each group.

Evaluation uses 48 positive reviews and 48 negative reviews, and greedy decoding. Figure 1 pairs each of the 12 groups' dimensions with its initial alignment score. As dimension increases from 16.53 to 24.32, the positive rate exhibits a downward trend, suggesting that within this dimension range, lower-dimensional alignment datasets yield better initial alignment.

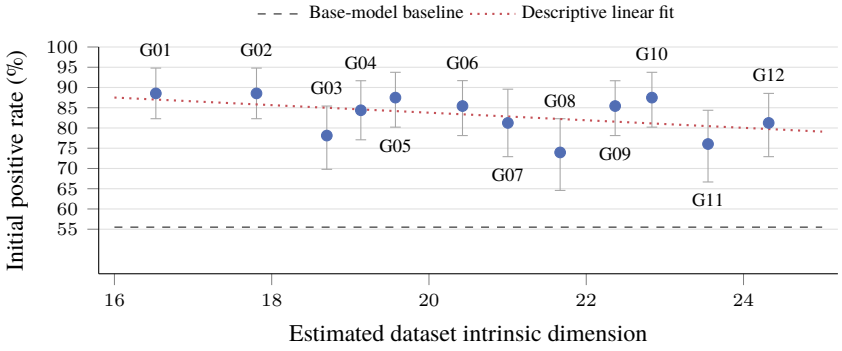

Figure 1: Initial alignment results for 12 groups with equal sample counts. Each point represents a group of 2,000 examples. The vertical axis shows the proportion of model outputs classified as positive. Error bars are 95% prompt-bootstrap intervals for a fixed model.

Table 1: The 12 equal-sized groups in Experiment 1. Each group contains 2,000 examples.

Group	Est. dimension	Initial (%)	Group	Est. dimension	Initial (%)
G01	16.53	88.54	G07	21.00	81.25
G02	17.81	88.54	G08	21.67	73.96
G03	18.70	78.12	G09	22.37	85.42
G04	19.13	84.38	G10	22.84	87.50
G05	19.57	87.50	G11	23.55	76.04
G06	20.43	85.42	G12	24.32	81.25

The Pearson correlation is -0.448 and the Spearman correlation is -0.486. Treating the points provisionally as independent observations yields corresponding p-values of 0.144 and 0.109. Resampling the shared evaluation prompts gives a Spearman interval of $[-0.696, -0.035]$. These results suggest a negative association between alignment dataset dimension and initial alignment performance.

4.3 Experiment 2: Lower-Dimensional Datasets Weaken Alignment Rebound

The second experiment examines how alignment dataset dimension affects elasticity during subsequent fine-tuning. We use three equal-sized alignment groups, each containing 2,000 positive examples: a low-dimensional group, an equal-sized random group, and a high-dimensional group, with estimated dimensions of 16.53, 19.16, and 24.32, respectively. The random group is not a larger full

dataset, but an equal-sized reference sampled from the same candidate pool. All three groups undergo 100 positive SFT updates, followed by fine-tuning on the same negative candidate pool, with evaluation at steps 0, 10, 30, 60, and 100.

Figure 2 contrasts initial alignment with subsequent retention. The high-dimensional group begins at a positive rate of 88.75%, slightly above the low-dimensional group's 87.50%. After the first 10 steps, however, the high-dimensional group falls to 54.17%, while the low-dimensional group remains at 75.00%, corresponding to drops of 34.58 and 12.50 percentage points, respectively. Thus, **although the low-dimensional group does not achieve the highest initial score in this run, it exhibits weaker alignment rebound.**

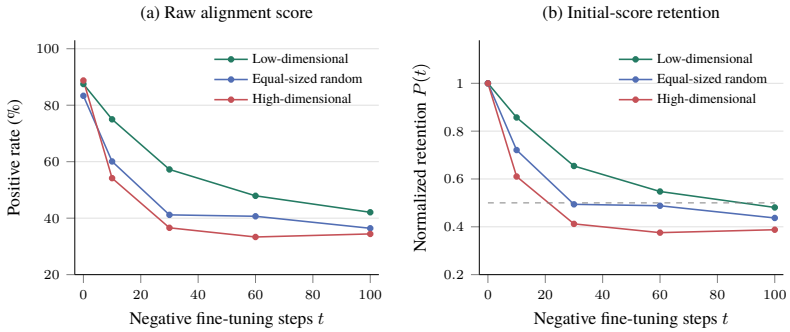

Figure 2: Subsequent negative-sample fine-tuning trajectories for groups with different dimensions and equal candidate sample counts. The low-dimensional group retains more positive alignment at every observed nonzero checkpoint. The right panel is normalized by each group's initial score; the dashed line indicates half the initial raw score, not the base-model baseline.

This advantage of the low-dimensional group persists throughout fine-tuning. At the first checkpoint, it retains 85.71% of its initial score, compared with 61.03% for the high-dimensional group. At step 100, the raw positive rates are 42.08% and 34.42%, respectively, with corresponding relative retention rates of 48.09% and 38.78%. The mean areas under the normalized curves, computed on the same grid, are 0.661 and 0.442, respectively. The low-dimensional group exceeds both other groups at every observed nonzero checkpoint, indicating that lower-dimensional datasets more effectively mitigate alignment rebound.

Table 2: Dataset-group experiment. The drop is the initial score minus the step-10 score, in percentage points. Mean area is the trapezoidal average of $P(t)$ over steps 0–100.

Group	Est. dimension	Initial (%)	Step-10 drop	Final (%)	Mean area
Low-dimensional	16.53	87.50	12.50	42.08	0.661
Equal-sized random	19.16	83.33	23.25	36.42	0.531
High-dimensional	24.32	88.75	34.58	34.42	0.442

Using half the initial raw score as the threshold, the high-dimensional and random groups are first observed to cross it at step 30, while the low-dimensional group does not cross it. This run supports the view that lower-dimensional alignment data can weaken alignment elasticity: **under the dimension grouping and training configuration used here, the low-dimensional group exhibits weaker rebound than the high-dimensional group.**

4.4 Experiment 3: Effects of Alignment Methods on Elasticity

This experiment holds the examples fixed and compares full-parameter SFT, LoRA, and BitFit. All three methods use exactly the same 2,000 positive reviews and batch order, with a shared dataset-dimension estimate of 19.44, a learning rate of 2×10^{-4}, a batch size of 16, and 125 updates during positive alignment. LoRA uses rank 8 and a scaling coefficient of 16 on attention projections; BitFit

updates only biases. After positive alignment, the LoRA weights are merged, and all three branches undergo full-parameter negative-sample fine-tuning using the same 1,600 examples in the same order.

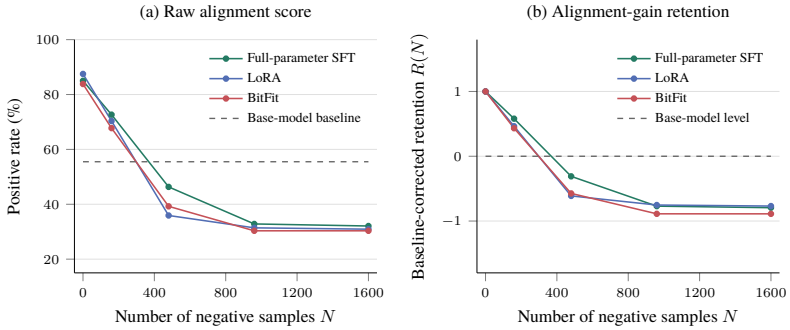

Figure 3: Different positive-alignment parameter-update methods applied to the same 2,000 alignment examples, followed by identical full-parameter negative-sample fine-tuning. Full-parameter SFT retains alignment best at the first checkpoint and also achieves higher scores later. The right panel corrects for this experiment's base-model score of 55.50%; negative values indicate performance below baseline.

Table 3: Different methods on identical data. $R(160)$ is baseline-corrected retention after 160 unique negative examples; final scores correspond to 1,600 examples.

Method	Peak dimension	Initial (%)	First$N(160)$	Final (%)
Full-parameter SFT	10.98	85.00	72.66	32.08
LoRA	12.13	87.50	70.33	30.92
BitFit	11.53	83.83	67.75	30.33

Early stage. Full-parameter SFT, which has the lowest peak dimension, achieves the best retention at the first checkpoint: its positive rate falls from 85.00% to 72.67%. LoRA falls from 87.50% to 70.33%, a drop of 17.17 percentage points, while BitFit falls from 83.83% to 67.75%, a drop of 16.08 percentage points. Clearly, full-parameter SFT retains more alignment benefits. This echoes the early retention advantage of lower dimension in the dataset-group experiment. **Later stage.** By the final checkpoint, the forward proportion of full-parameter SFT was 72.66%, higher than BitFit's 67.75% and LoRA's 70.33%. The average curve area for alignment effect and alignment retention was also the largest for full-parameter SFT. Therefore, alignment methods with lower model-aware dimensions indeed have an advantage in maintaining alignment effectiveness.

In multiple rounds of experiments, we found that alignment methods with lower model-perceived dimensions do not exhibit as pronounced an advantage in reduced robustness as observed in low-dimensional datasets. In some cases, alignment methods with higher model-perceived dimensions even better maintain alignment effects. This situation arises because, when using different alignment methods, there are differences beyond model-perceived dimensions, making it impossible to control a single variable. Nevertheless, overall, the experiments still demonstrate the general pattern that alignment methods with lower model-perceived dimensions tend to have weaker robustness.

5 Conclusion

This paper investigates the problem of alignment resistance in large models under fine-tuning or perturbation after pretraining and alignment training. To enhance alignment robustness, this paper develops the existing elasticity theory, and proposes *Alleviation Elasticity Theory*. The theory establishes a qualitative relationship between the resistance speed and the dataset's intrinsic dimension and model's perceived dimension. It can effectively guide the selection of datasets and alignment methods for the same alignment task. This paper provides a detailed proof of the theory, experimental verification, and evidence of the theory's effectiveness in guiding practice.

AI Use Statement

Generative AI assisted with coding, manuscript restructuring and editing, literature cross-checking, LaTeX tables and schematic figures.

Reproducibility Statement

To facilitate reproducibility, we provide the code used for the experiments at `https://anonymous.4open.science/r/Alleviation-Elasticity-Theory-278F`.

REFERENCES

Elad Ben Zaken, Yoav Goldberg, and Shauli Ravfogel. BitFit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 1–9, 2022. doi: 10.18653/v1/2022.acl-short.1. URL https://aclanthology.org/2022.acl-short.1/.

Emily Cheng, Diego Doimo, Corentin Kervadec, Iuri Macocco, Jade Yu, Alessandro Laio, and Marco Baroni. Emergence of a high-dimensional abstraction phase in language transformers. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=0fD3iIBhlV.

Grégoire Delétang, Anian Ruoss, Paul-Ambroise Duquenne, Elliot Catt, Tim Genewein, Christopher Mattern, Jordi Grau-Moya, Li Kevin Wenliang, Matthew Aitchison, Laurent Orseau, Marcus Hutter, and Joel Veness. Language modeling is compression. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=jznbgiynus.

Francesco Denti, Diego Doimo, Alessandro Laio, and Antonietta Mira. The generalized ratios intrinsic dimension estimator. *Scientific Reports*, 12:20005, 2022. doi: 10.1038/s41598-022-20991-1. URL https://www.nature.com/articles/s41598-022-20991-1.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://arxiv.org/abs/2106.09685.

Jiaming Ji, Kaile Wang, Tianyi Alex Qiu, Boyuan Chen, Jiayi Zhou, Changye Li, Hantao Lou, Josef Dai, Yunhuai Liu, and Yaodong Yang. Language models resist alignment: Evidence from data compression. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 23411–23432, 2025. doi: 10.18653/v1/2025.acl-long.1141. URL https://aclanthology.org/2025.acl-long.1141/.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, 2017. URL https://arxiv.org/abs/1612.00796.

David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. In *Advances in Neural Information Processing Systems*, 2017. URL https://arxiv.org/abs/1706.08840.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pp. 142–150, 2011. URL https://aclanthology.org/P11-1015/.

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 *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2310.03693.

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 *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=6Mxhg9PtDE.

Alfréd Rényi. On the dimension and entropy of probability distributions. *Acta Mathematica Academiae Scientiarum Hungaricae*, 10:193–215, 1959. Original article (Rényi Institute archive).

Claude E. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27(3):379–423, 1948. doi: 10.1002/j.1538-7305.1948.tb01338.x. URL https://onlinelibrary.wiley.com/doi/10.1002/j.1538-7305.1948.tb01338.x.

Elena Facco, Maria d'Errico, Alex Rodriguez, and Alessandro Laio. Estimating the intrinsic dimension of datasets by a minimal neighborhood information. *Scientific Reports*, 7:12140, 2017. URL https://doi.org/10.1038/s41598-017-11873-y.

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. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/77baa7c2a3a675823e89131698fd6e19-Abstract-Conference.html.

Tiansheng Huang, Sihao Hu, and Ling Liu. Vaccine: Perturbation-aware alignment for large language models against harmful fine-tuning attack. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/873c86d9a979ab80d8e2919510d4446b-Abstract-Conference.html.

Zhizhong Li and Derek Hoiem. Learning without forgetting. In *Computer Vision – ECCV 2016*, volume 9908 of *Lecture Notes in Computer Science*, pp. 614–629, 2016. URL https://doi.org/10.1007/978-3-319-46493-0_37.

Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H. Lampert. iCaRL: Incremental classifier and representation learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2001–2010, 2017. URL https://openaccess.thecvf.com/content_cvpr_2017/html/Rebuffi_iCaRL_Incremental_Classifier_CVPR_2017_paper.html.

Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. Continual learning with deep generative replay. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/hash/0efbe98067c6c73dba1250d2beaa81f9-Abstract.html.

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. URL https://arxiv.org/abs/2310.02949.

Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 3987–3995, 2017. URL https://proceedings.mlr.press/v70/zenke17a.html.

A Implementation and Experimental Records

A.1 The Relationship Between k And Elasticity

In Ji et al. (2025), $l(\lambda) = |D_t|/|D_a| \ll 1$ and $k(\kappa) = |D_p|/|D_a| \gg 1$. In the following discussion, the KL divergence between the model distribution and a dataset distribution is identified with the model's normalized compression rate for that dataset. When the alignment dataset grows, k decreases. Consider the case where the change in KL divergence between the model and alignment distributions is held approximately fixed. so that the left-hand side of Eq. (1) remains nearly unchanged. The corresponding change in KL divergence between the model and pretraining distributions then increases, as does the change in the model's compression rate on the pretraining dataset. Since the model is moving toward the pretraining distribution, the magnitude of the rate of change in this compression rate increases, while the signed rate decreases. The model therefore approaches the pretraining distribution more rapidly, exhibiting greater elasticity and losing alignment more quickly. The rate of change of the KL divergence between the model and the pretraining distribution increases as k decreases, that is, the rate of change of the model's compression rate on the pretraining dataset becomes larger, and the model will rebound to the pretraining distribution at a faster pace. Conversely, when the pretraining dataset grows, k increases. Holding the change in KL divergence between the model and pretraining distributions approximately fixed increases the magnitude of the right-hand side of Eq. (1), and thus the rate of change in the alignment-dataset compression rate on the left-hand side. The model moves away from the alignment distribution more rapidly. again exhibiting greater elasticity and losing alignment more quickly.

A.2 Dimension Measurement

Conceptually, dataset dimension is defined on frozen reference representations, while model-perceived dimension is defined on layerwise representations of the aligned model. Both describe the dimension of point-cloud distributions, not the number of model parameters or hidden-vector width. Reference encoding uses attention-mask-aware mean pooling of the final hidden states followed by L2 normalization. In the method-comparison experiment, layerwise hidden states at the last valid token are extracted for 256 fixed held-out texts, and the maximum layerwise estimate is used. The peak occurs at layer 5, with layer index 0 denoting the embedding output.

The historical implementation uses neighbor orders $k_1 = 3, k_2 = 8$ and computes $r_{j,k_2}/r_{j,k_1}$ using Euclidean distances, with estimator

$$\widetilde{d} = \frac{\log(k_2/k_1)}{m^{-1}\sum_{j=1}^{m}\log(r_{j,k_2}/r_{j,k_1})}. \tag{A.1}$$

The formal GRIDE method derives the distance-ratio distribution from a local point process and performs likelihood-based estimation (Denti et al., 2022).

A.3 Scope of Controls in the Three Experimental Records

The initial scan selects 12 groups from 48 candidate subsets based only on reference dimension. The training seed is 17, and every group visits all 2,000 example positions once, with a batch size of 16. All groups share 96 evaluation prompts, and greedy decoding generates at most 24 new tokens. The 95% error bars result from prompt resampling, not confidence intervals across training seeds. The table below gives all scatter-plot coordinates; no new training runs have been added.

Table 4: The 12 equal-sized groups in Experiment 1. Each group contains 2,000 examples.

Group	Est. dimension	Initial (%)	Group	Est. dimension	Initial (%)
G01	16.53	88.54	G07	21.00	81.25
G02	17.81	88.54	G08	21.67	73.96
G03	18.70	78.12	G09	22.37	85.42
G04	19.13	84.38	G10	22.84	87.50
G05	19.57	87.50	G11	23.55	76.04
G06	20.43	85.42	G12	24.32	81.25

The dataset-group rebound experiment also uses seed 17, but positive training lasts 100 steps and uses a training function that samples randomly at each step. The output files record each group's candidate sample count but not the batch size or the complete example-access history, so the 100 steps cannot be converted into a verified count of unique examples. The evaluation prompts come from an earlier procedure that concatenates positive and negative texts and then takes the first 48, making the actual prompts positive-source only. Its initial scores are not directly comparable to the balanced-prompt scores in Experiment 1, nor should the baseline from Experiment 1 be used to fill in the missing baseline for this run.

The method comparison uses training seed 41, a shared positive-example fingerprint, and a deterministic batch schedule, with 2,000 unique positive examples and 1,600 unique negative examples visited. Checkpoints at steps 0, 10, 30, 60, and 100 correspond to cumulative counts of 0, 160, 480, 960, and 1,600 negative examples. The numbers of trainable parameters during positive alignment are 81,912,576 for full-parameter SFT, 221,184 for LoRA, and 51,456 for BitFit. A shared learning rate does not imply that each method uses its individually optimal hyperparameters. Scoring uses 24 positive-source and 24 negative-source prompts, greedy continuations of at most 24 tokens, and a classifier that evaluates only the continuation.

B Detailed Proofs for Section 3.3

This appendix proves the covering-number, quantization-entropy, and leaf-node-entropy results in Section 3.3. We retain the representation distributions and notation of the main text. Unless otherwise stated, logarithms are base 2, and asymptotic relations in resolution are considered for a fixed data distribution and fixed geometric conditions.

B.1 Proof: From Dataset Dimension to Covering Numbers

1. Conditions and notation. Let P_i^Z denote the distribution of the representation random variable Z_i. Assume that its support, $\mathcal{M}_i = \mathrm{supp}(P_i^Z)$, is a compact δ_i-dimensional manifold, with $\delta_i > 0$. Let ν_i denote its manifold volume measure, and write $\mathcal{V}_i = \nu_i(\mathcal{M}_i) \in (0, \infty)$. Here, $\mathcal{V}_i$ denotes geometric volume.

Define the open ball within the manifold as $B_i(z, r) = \{y \in \mathcal{M}_i : \|y - z\|_2 < r\}$. Assume that constants $0 < a_i \le b_i < \infty$ and $\epsilon_{0,i} > 0$ exist such that, for every $z \in \mathcal{M}_i$ and $0 < r \le \epsilon_{0,i}$,

$$a_i r^{\delta_i} \le \nu_i(B_i(z, r)) \le b_i r^{\delta_i}. \tag{B.1}$$

These constants are independent of the center and radius. For $0 < \epsilon_s \le \epsilon_{0,i}$, let $N_i(\epsilon_s)$ denote the minimum number of open balls of radius ϵ_s, with centers on the manifold, required to cover all of $\mathcal{M}_i$.

2. Lower bound on the covering number. If n such balls cover the entire manifold, subadditivity of the measure gives

$$\mathcal{V}_i \le \sum_{j=1}^{n} \nu_i(B_i(z_j, \epsilon_s)) \le n b_i \epsilon_s^{\delta_i}. \tag{B.2}$$

Since this relation holds for any such cover,

$$N_i(\epsilon_s) \ge \frac{\mathcal{V}_i}{b_i} \epsilon_s^{-\delta_i}. \tag{B.3}$$

3. Upper bound on the covering number. Compactness ensures the existence of a finite maximal ϵ_s-separated set $\{z_1, \dots, z_m\} \subset \mathcal{M}_i$: distinct centers are at least ϵ_s apart, and no additional point can be added while preserving this separation. If a point were at distance at least ϵ_s from every center, it could be added to the set, contradicting maximality. The open balls of radius ϵ_s around these centers therefore cover the entire manifold, so $N_i(\epsilon_s) \le m$.

By the triangle inequality, the open balls of radius $\epsilon_s/2$ are pairwise disjoint. Applying the small-ball volume lower bound yields

$$m a_i (\epsilon_s/2)^{\delta_i} \le \sum_{j=1}^{m} \nu_i(B_i(z_j, \epsilon_s/2)) \le \mathcal{V}_i. \tag{B.4}$$

Hence, $N_i(\epsilon_s) \le 2^{\delta_i} \mathcal{V}_i a_i^{-1} \epsilon_s^{-\delta_i}$.

4. Combining the bounds. Together, these bounds give

$$\frac{\mathcal{V}_i}{b_i} \epsilon_s^{-\delta_i} \le N_i(\epsilon_s) \le \frac{2^{\delta_i} \mathcal{V}_i}{a_i} \epsilon_s^{-\delta_i}. \tag{B.5}$$

The multiplicative constants on both sides are independent of resolution. Thus, for a fixed manifold and fixed dimension, as $\epsilon_s \to 0$,

$$N_i(\epsilon_s) = \Theta(\epsilon_s^{-\delta_i}), \qquad \log_2 N_i(\epsilon_s) = \delta_i \log_2(1/\epsilon_s) + O(1). \tag{B.6}$$

The $O(1)$ term is uniformly bounded with respect to resolution, although its bound may depend on the dataset and its geometric properties.

B.2 Proof: From Covering Numbers to Quantization Entropy

1. Quantization cells and their probabilities. Using the separated set above as centers, construct a measurable nearest-center partition $\{C_{i,s,u}\}_{u=1}^{m}$, breaking distance ties in favor of the smaller index. Each vector is assigned to its closest center, and all vectors assigned to center u form the region $C_{i,s,u}$, called a quantization cell. Here, i indexes datasets, s indexes resolutions, and u indexes cells.

If $y \in B_i(z_u, \epsilon_s/2)$, then for any $v \neq u$, the triangle inequality and center-separation condition imply

$$\begin{aligned} \|y - z_v\|_2 &\geq \|z_v - z_u\|_2 - \|y - z_u\|_2 \\ &\geq \epsilon_s - \|y - z_u\|_2 > \epsilon_s/2 > \|y - z_u\|_2. \end{aligned} \tag{B.7}$$

Thus, z_u is the unique nearest center to y, and $y \in C_{i,s,u}$. Conversely, if $y \in C_{i,s,u}$, the open-ball covering property ensures that its distance to the nearest center z_u is strictly less than ϵ_s, so $y \in B_i(z_u, \epsilon_s)$. Consequently,

$$B_i(z_u, \epsilon_s/2) \subseteq C_{i,s,u} \subseteq B_i(z_u, \epsilon_s). \tag{B.8}$$

Further assume that P_i^Z has a density f_i with respect to ν_i, with $0 < \underline{f}_i \leq f_i(z) \leq \overline{f}_i < \infty$ holding ν_i-almost everywhere, where the density bounds are independent of resolution. Define the cell probability, the probability that a sample falls in cell u, as

$$q_{i,s}(u) = \Pr(Z_i \in C_{i,s,u}) = \int_{C_{i,s,u}} f_i(z)\, \mathrm{d}\nu_i(z), \qquad \sum_{u=1}^{m} q_{i,s}(u) = 1. \tag{B.9}$$

Then

$$\underline{f}_i\, \nu_i(C_{i,s,u}) \leq \int_{C_{i,s,u}} f_i(z)\, \mathrm{d}\nu_i(z) \leq \overline{f}_i\, \nu_i(C_{i,s,u}). \tag{B.10}$$

Applying Eq. (B.1) gives

$$\underline{f}_i a_i 2^{-\delta_i} \epsilon_s^{\delta_i} \leq q_{i,s}(u) \leq \overline{f}_i b_i \epsilon_s^{\delta_i}. \tag{B.11}$$

Let $c_i = \underline{f}_i a_i 2^{-\delta_i} > 0$ and $C_i = \overline{f}_i b_i < \infty$. Then

$$c_i \epsilon_s^{\delta_i} \leq q_{i,s}(u) \leq C_i \epsilon_s^{\delta_i}. \tag{B.12}$$

Equation (B.12) shows that dimension first affects cell volume and, through the density bounds, cell probability. Thus, almost all probability mass cannot concentrate in a single cell.

2. Bounds on quantization entropy. Since $-\log_2 x$ is monotonically decreasing, taking negative logarithms of the probability bounds gives

$$\delta_i \log_2(1/\epsilon_s) - \log_2 C_i \leq -\log_2 q_{i,s}(u) \leq \delta_i \log_2(1/\epsilon_s) - \log_2 c_i. \tag{B.13}$$

The quantization entropy is $H_i(s) = -\sum_{u=1}^{m} q_{i,s}(u) \log_2 q_{i,s}(u)$. Multiplying the preceding inequalities by $q_{i,s}(u)$ and summing, using the fact that these probabilities sum to 1, yields

$$\delta_i \log_2(1/\epsilon_s) - \log_2 C_i \leq H_i(s) \leq \delta_i \log_2(1/\epsilon_s) - \log_2 c_i. \tag{B.14}$$

Therefore, under the stated geometric and probabilistic conditions,

$$H_i(s) = \delta_i \log_2(1/\epsilon_s) + O(1). \tag{B.15}$$

The absolute value of the remainder $R_i(s)$ is bounded by $\max\{|\log_2 c_i|, |\log_2 C_i|\}$. This bound is independent of resolution but depends on the distribution, dimension, and geometric conditions, and may therefore differ across datasets. This is closely related to the use of quantization entropy to characterize information dimension (Rényi, 1959).

B.3 Proof: From Quantization Entropy to Leaf-Node Entropy

1. The two random variables being compared. Fix the distribution associated with a dataset and draw a single random sample X. Classify it in two ways:

- **Quantization:** Compute its representation vector $E(X)$ and identify the quantization cell containing it. Denote the cell index by U.
- **Token-tree classification:** Use the token sequence and pruning rule to identify its corresponding leaf. Denote the leaf index by J.

Thus, $H_{i,s}(U)$ is the entropy of the quantization-cell index, and $H_{i,d}(J)$ is the entropy of the token-tree leaf index.

2. Partition correspondence conditions. Allow the quantization scale ϵ_s to correspond to a pruning depth $d(s)$. Across all scales considered, assume that finite positive integers K_1, K_2, independent of the scale index s, exist such that:

- Samples belonging to any fixed token-tree leaf fall into at most K_1 quantization cells.
- Samples belonging to any fixed quantization cell originate from at most K_2 token-tree leaves.

Only correspondences with positive joint probability are counted. Formally, require

$$|\{u : \Pr(U = u, J = v) > 0\}| \le K_1, \qquad \Pr(J = v) > 0, \tag{B.16}$$

$$|\{v : \Pr(U = u, J = v) > 0\}| \le K_2, \qquad \Pr(U = u) > 0. \tag{B.17}$$

Intuitively, the two classifications need not be in one-to-one correspondence, but knowing one classification limits the possible values of the other.

3. Bounds on conditional entropy. $H(U \mid J)$ measures the average uncertainty remaining about the quantization-cell index after the token-tree leaf index is known; $H(J \mid U)$ measures the converse.

A discrete random variable with at most K positive-probability values has entropy at most $\log_2 K$, with equality when those K values are equiprobable. By Eq. (B.16), knowing $J = v$ leaves at most K_1 possible values for U, so

$$0 \le H(U \mid J = v) \le \log_2 K_1.$$

Averaging with respect to the probabilities of the different leaves gives

$$\begin{aligned} H(U \mid J) &= \sum_{v:\Pr(J=v)>0} \Pr(J = v) H(U \mid J = v) \\ &\le \sum_{v:\Pr(J=v)>0} \Pr(J = v) \log_2 K_1 = \log_2 K_1. \end{aligned} \tag{B.18}$$

Similarly, Eq. (B.17) implies

$$0 \le H(J \mid U) \le \log_2 K_2. \tag{B.19}$$

4. Bounding the entropy difference using conditional entropy. Consider the joint entropy $H(U, J)$ of the two indices. It can be expanded in either order:

$$H(U, J) = H(U) + H(J \mid U) = H(J) + H(U \mid J). \tag{B.20}$$

This is the chain rule for entropy: first specify one index, then describe the remaining information about the other. Mathematically, it follows from factorizing the joint probability, for example,

$$\Pr(U = u, J = v) = \Pr(U = u) \Pr(J = v \mid U = u) \tag{B.21}$$

Taking the negative logarithm and then the expectation yields the first expansion; exchanging U, J gives the second.

Rearranging Eq. (B.20) yields

$$H(J) - H(U) = H(J \mid U) - H(U \mid J). \tag{B.22}$$

The first term on the right lies in $[0, \log_2 K_2]$, and the second in $[0, \log_2 K_1]$. Therefore,

$$-\log_2 K_1 \le H(J) - H(U) \le \log_2 K_2. \tag{B.23}$$

Thus, the difference between quantization entropy and leaf-node entropy is controlled by constants that do not grow with scale.

5. Incorporating the dimension dependence of quantization entropy. For the corresponding dataset, scale, and pruning depth,

$$H_{i,s}(U) = \delta_i \log_2(1/\epsilon_s) + O(1), \tag{B.24}$$

If the two partitions satisfy the uniform correspondence conditions above, then

$$H_{i,d}(J) - H_{i,s}(U) = O(1). \tag{B.25}$$

The sum of two uniformly bounded remainders remains uniformly bounded. Hence, the leaf-node entropy satisfies

$$H_i^{\text{tok}}(d(s)) = \delta_i \log_2(1/\epsilon_s) + O(1). \tag{B.26}$$

where $H_i^{\text{tok}}(d(s)) := H_{i,d(s)}(J)$. More specifically, if the absolute value of the remainder in Eq. (B.24) is at most a constant B_i, the final remainder is bounded in absolute value by

$$B_i + \max\{\log_2 K_1, \log_2 K_2\}. \tag{B.27}$$

These bounds are independent of the scale index but depend on the fixed data distribution and geometric conditions.