CeRA: Breaking the Linear Ceiling of Low-Rank Adaptation with Non-linearity Retained at Inference

Hung-Hsuan Chen
Computer Science and Information Engineering
National Central University
Taoyuan, Taiwan
hhchen1105@acm.org

Abstract

Low-Rank Adaptation (LoRA) dominates parameter-efficient fine-tuning (PEFT). However, it faces a "linear ceiling": increasing the rank yields diminishing returns in expressive capacity due to linear constraints. We introduce CeRA (Capacity-enhanced Rank Adaptation), a weight-level parallel adapter that injects SiLU gating and dropout to induce non-linearity during inference, thereby placing it in a different function class from adapters whose non-linearity exists during training and collapses to an affine map at inference time. On both the basic arithmetic (GSM8K) and the complex MATH benchmark, CeRA is markedly more parameter-efficient. Across a full rank $\times$ learning rate sweep, CeRA at rank 64 achieves the highest MATH pass@1 of any configuration in the grid (23.6%), matching or exceeding both a rank-512 LoRA (22.4%) and DoRA (19.8%) while using only 1/8 of the parameter budget. With the rank and learning rate fixed, CeRA equals or outperforms LoRA in 10 of 12 matched settings. Spectrally, CeRA's learned updates utilize the singular-value spectrum more broadly than linear adapters, which exhibit rank collapse at high rank, although a scale-matched control shows that this difference stems mostly from output scale and partially from non-linearity. Additionally, dropout appears to contribute to regularization rather than rank expansion. We release the code for reproducibility.[1]

Keywords PEFT · LoRA · LLM

1 Introduction

Parameter-Efficient Fine-Tuning (PEFT) is the de facto standard for fine-tuning Large Language Models (LLMs). Among various techniques, Low-Rank Adaptation (LoRA) [Hu et al., 2022] is widely used. Its design relies on the mergeability assumption: weight updates must be linear ($\Delta W = BA$) to allow merging with the base model for zero-latency inference.

We challenge this linear constraint. Variants that attempt to refine LoRA, such as weight decomposition in DoRA [Liu et al., 2024] or adaptive rank allocation [Zhang et al., 2023], primarily focus on optimizing the learning dynamics of the linear subspace but leave the hypothesis space unchanged. These methods remain bounded by the expressivity limits of linear transformations, creating a ceiling for reasoning-intensive tasks. In our experiments, a high-rank LoRA ($r = 512$) performs similarly to its low-rank counterpart ($r = 64$); since overparameterization typically eases optimization even in linear models [Chen and Chen, 2020], this suggests that the bottleneck may be the structural rigidity of linearity itself rather than the parameter count or optimization difficulty.

To overcome this ceiling, we introduce CeRA (Capacity-enhanced Rank Adaptation). CeRA shifts from linear space optimization to nonlinear. By injecting SiLU gating and dropout into a *weight*-level parallel adapter, CeRA provides the high-dimensional expressivity required for complex reasoning. Furthermore, in multi-tenant serving where many adapters share a single frozen base model and are kept unmerged—in the cloud (e.g., S-LoRA [Sheng et al., 2024],

[1] https://github.com/hhchen1105/cera

Punica [Chen et al., 2024]) and on edge devices (e.g., EdgeLoRA [Shen et al., 2025])—the latency cost of non-linearity is modest rather than prohibitive (details in Section 4.4).

Our contributions are threefold:

- We propose CeRA, a fine-grained, weight-level parallel adapter that integrates nonlinear gating to capture complex functional updates beyond linear approximations.
- We evaluate downstream exact-match accuracy and highlight the role of task complexity. On the challenging MATH dataset, CeRA ($r = 64$) matches or exceeds high-rank LoRA ($r = 512$) and DoRA, showing that capacity expansion improves reasoning without inflating the parameter budget.
- Using Singular Value Decomposition (SVD), we show that CeRA's learned updates spread energy into the tail of the singular value spectrum where linear methods collapse, and—via a scale-matched control experiment (Appendix D)—that much of this difference stems from output scale, so we treat spectral expansion as a descriptive diagnostic rather than the mechanism behind the accuracy gains.

2 CeRA: Capacity-enhanced Rank Adaptation

2.1 Preliminaries: The Linear Confinement

For a pre-trained weight matrix $W_0 \in \mathbb{R}^{d\times k}$, LoRA constrains the update ΔW to a low-rank decomposition BA, where $B \in \mathbb{R}^{d\times r}$, $A \in \mathbb{R}^{r\times k}$, and $r \ll \min(d, k)$. The forward pass is:

$$h = W_0 x + \frac{\alpha}{r} BAx, \tag{1}$$

where α is a hyperparameter.

This formulation allows ΔW to merge into W_0. However, we hypothesize that this constraint contributes to the rank under-utilization observed in complex tasks.

2.2 The CeRA Architecture

CeRA relaxes this linear constraint while retaining the parallel bottleneck structure to maintain parameter efficiency. Formally, CeRA defines the forward pass as:

$$h = W_0 x + s \cdot B\,\mathcal{D}(\sigma(Ax)), \tag{2}$$

where $A \in \mathbb{R}^{r\times k}$ projects the input to a latent dimension r, $\sigma(\cdot)$ is SiLU, $\mathcal{D}(\cdot)$ is dropout, $B \in \mathbb{R}^{d\times r}$ projects back to the output dimension, and s is a scaling scalar.

Weight-Level Granularity and Non-Linearity. Unlike module-level parallel adapters [He et al., 2022, Zhu et al., 2021] that process the aggregate output of an entire attention block, CeRA operates at the weight level. By injecting updates into the internal query (W_q) and value (W_v) projections, CeRA alters the attention mechanism's internal feature dynamics. The SiLU activation enables the adapter to suppress noise or amplify feature directions. Empirically, it is the component that broadens the adapter's effective rank (Section 4.2). Dropout $\mathcal{D}$ regularizes the bottleneck latent space and improves predictive performance, consistent with prior findings that structured edge-dropping regularization improves robustness [Yang and Chen, 2025].

3 Experiments

We use Llama-3.1-8B as the frozen backbone for the main experiments and vary only the adapter architecture. We train the models in `bfloat16` precision with AdamW optimizer.[2] For LoRA and DoRA, we use a fixed scaling hyperparameter $\alpha = 32$ (the common PEFT default). CeRA uses a fixed scale $s = 1$ throughout. For each method and rank, we select the learning rate based on MATH pass@1 (the full sweep is in Appendix A), which partially mitigates the coupling between the adapter scale and the learning rate.

We run two *separate* fine-tuning experiments, one per training dataset, to investigate capacity scaling and domain robustness: SlimOrca [Lian et al., 2023] ($\sim$300k GPT-4-augmented instruction pairs emphasizing Chain-of-Thought) and MathInstruct [Yue et al., 2024] (a composite of $\sim$100k mathematical problems drawn from sources including

[2]We verify the effectiveness of CeRA across model scales (1B/3B/8B) in Appendix B.

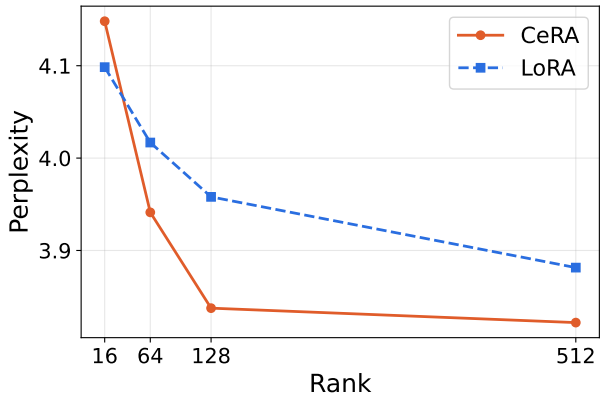

Figure 1: The capacity scaling behavior on SlimOrca. LoRA's performance plateaus rapidly, while CeRA (dropout $=$ 0.1) continues to improve (lower PPL is better).

GSM8K [Cobbe et al., 2021] and MATH [Hendrycks et al., 2021]); we do not mix the two datasets in a single run. The SlimOrca-trained models are used for the capacity-scaling and training-dynamics analyses of Sections 3.1–3.2, while the MathInstruct-trained models are used for the downstream accuracy evaluation of Section 3.3; the spectral analysis in Section 4 reports both. For evaluation, we use 500 competition-level MATH problems[3] and the full GSM8K test set (1,319 problems). MATH and GSM8K pass@1 are computed with greedy decoding, while MATH pass@10 uses nucleus sampling ($T = 0.8$, top-$p = 0.95$) with 10 samples per problem and is reported with the unbiased pass@k estimator of Chen et al. [2021]. The exact match is computed by extracting the last `\boxed{}` expression from the generation (falling back to regex patterns when absent) and comparing it to the gold answer after LaTeX normalization—stripping `\text`/`\textbf`-style wrappers, canonicalizing `\frac`, and removing whitespace. All methods are evaluated under identical settings.

We compare CeRA with LoRA and the SOTA linear variant DoRA [Liu et al., 2024]. Evaluation metrics include Perplexity (PPL) for predictive performance, exact match (pass@k) for downstream problem-solving accuracy, and Effective Rank (ER) for spectral utilization.

3.1 The Capacity Scaling Behavior

We first evaluate whether non-linearity mitigates the rank saturation observed with linear adapters on the SlimOrca dataset with respect to predictive performance.

Perplexity Saturation. Figure 1 illustrates the rapid diminishing returns for LoRA. Increasing the parameter budget by $32\times$ (from $r = 16$ to $r = 512$) yields a performance plateau around a perplexity of 3.90. This is consistent with the linear ceiling: additional ranks yield minimal gains in expressivity. In contrast, CeRA reaches a perplexity of 3.84 at rank 128, outperforming LoRA at rank 512 (≈ 3.88) with $4\times$ fewer parameters; even at rank 64 (3.94), CeRA already matches LoRA at rank 128. This indicates that the plateau is a structural bottleneck rather than a parameter-capacity issue.

3.2 Optimization Robustness to Suboptimal Hyperparameters

A critical yet often overlooked bottleneck in deploying PEFT is optimization fragility. Linear adapters like LoRA require exhaustive hyperparameter searches because their performance is sensitive to the learning rate. To evaluate whether CeRA's capacity expansion mitigates this fragility, we analyze training dynamics under different learning rate regimes.

Figure 2 shows the validation perplexity trajectories on SlimOrca for CeRA at dropout $D \in \{0.0, 0.1, 0.2, 0.3\}$ and for LoRA (which uses no dropout), at a lower (1×10^{-4}) and a higher (5×10^{-4}) learning rate.

[3]We take the first 500 problems of the `MATH-lighteval` test split in dataset order. Because that split is grouped by subject, these 500 problems all fall under the *Algebra* category, spanning difficulty Levels 1–5. Our MATH pass@1, therefore, measures competition-level *algebraic* reasoning specifically and should not be confused with the subject-stratified MATH-500 subset of Lightman et al. [2024]. We use the same fixed 500-problem slice for every method, so all reported comparisons remain matched.

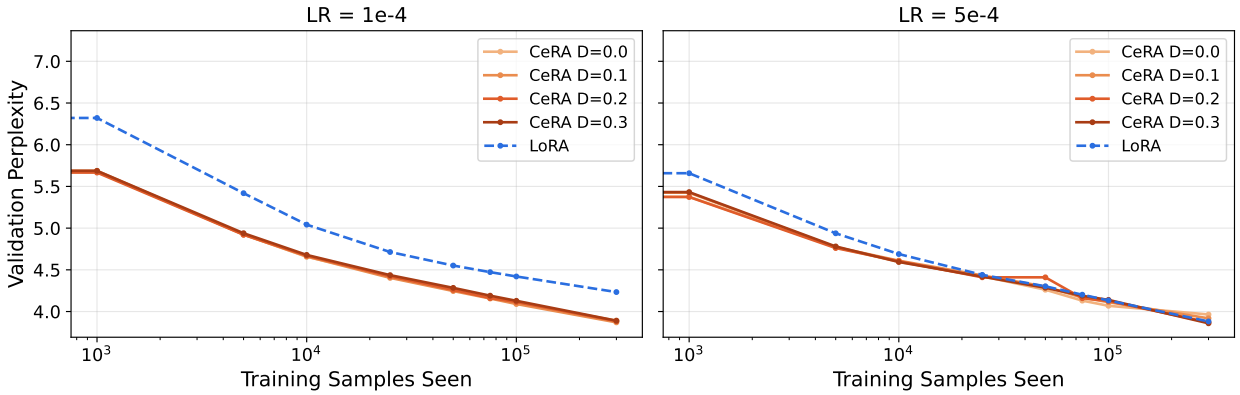

Figure 2: Validation perplexity during training on SlimOrca at a low (LR=1e-4) and a higher (LR=5e-4) learning rate, for CeRA at dropout $D \in \{0.0, 0.1, 0.2, 0.3\}$ and LoRA without dropout. At the low learning rate, LoRA's descent slows and a perplexity gap persists, whereas CeRA descends smoothly to a competitive final perplexity; at the higher learning rate the methods converge comparably.

Table 1: Downstream task accuracy (exact match). CeRA ($r = 64$) attains the highest MATH pass@1, MATH pass@10, and GSM8K pass@1 of all configurations, exceeding or matching DoRA ($r = 64$, $r = 128$) and high-rank LoRA ($r = 512$) at one-eighth of its parameter budget.

Method	Rank	Params	MATH (Complex) pass@1	MATH (Complex) pass@10	GSM8K (Basic) pass@1
LoRA	64	27.3M	21.6%	54.8%	54.44%
	128	54.5M	20.6%	51.2%	55.57%
	512	218.1M	<u>22.4%</u>	<u>55.4%</u>	<u>57.16%</u>
DoRA	64	27.3M	19.8%	54.6%	55.12%
	128	54.5M	21.2%	51.4%	51.63%
CeRA (Ours)	64	27.3M	**23.6%**	**56.6%**	**58.76%**

Comparable convergence at the higher learning rate. In the right panel (5×10^{-4}), a learning rate that suits the linear baseline, both LoRA and CeRA variants converge aggressively and reach comparable early-stage plateaus. Adding nonlinear structural expansion therefore does not slow convergence or harm stability in this regime.

Robustness to a low learning rate. In the left panel (1×10^{-4}), LoRA's descent slows and a perplexity gap persists throughout training, whereas every CeRA configuration continues to descend smoothly to a lower final perplexity. CeRA thus tolerates an under-tuned (low) learning rate better than LoRA on this objective. We do not claim uniform learning rate robustness: at the opposite extreme—the largest rank under the highest learning rate—CeRA itself degrades, as the complete learning rate sweep in Appendix A shows.

3.3 Downstream Task Accuracy and the Role of Task Complexity

Although PPL measures the predictive distribution, it is crucial to verify whether CeRA yields correct final answers. We evaluate exact-match (pass@k) accuracy on two datasets of contrasting difficulty: GSM8K (fundamental arithmetic) and MATH (complex mathematics).

High Parameter Efficiency. Table 1 shows that CeRA ($r = 64$) reaches a pass@1 of 23.6% on MATH, the highest of all configurations, exceeding LoRA of the same rank 64 (21.6%) and matching the accuracy of an $8\times$ larger LoRA at $r = 512$ (22.4%). By introducing non-linearity, CeRA overcomes the linear ceiling without inflating the parameter budget.

On the high-complexity MATH dataset, DoRA at rank 128 (21.2%) outperforms LoRA at the same rank (20.6%) but remains below CeRA at rank 64 (23.6%). This indicates that altering the adapter's functional class via inference-time

Table 2: A case study on iterative reasoning. LoRA with a larger rank of 512 suffers from state collapse. In contrast, CeRA, despite its smaller rank, successfully models nonlinear recursive updates, generating a dynamic trajectory with far fewer parameters.

Q: Consider the logistic map $x_{n+1} = 3.5x_n(1 - x_n)$ with $x_0 = 0.4$. Calculate the first few iterations.
LoRA (Rank 512) – State Collapse Step 1: $x_1 = 3.5(0.4)(0.6) = 0.84$. Step 2: $x_2 = 3.5(0.84)(0.16) = 0.4704$. Step 3: $x_3 = 3.5(0.4704)(0.5296) \approx 0.8719$. Step 4: $x_4 = 0.8719$. Step 5: $x_5 = 0.8719$. ... (Model repeats 0.8719 indefinitely)
CeRA (Rank 128) – Dynamic Tracking Step 1: $x_1 = 3.5(0.4)(0.6) = 0.84$. Step 2: $x_2 = 3.5(0.84)(0.16) = 0.4704$. Step 3: $x_3 = 3.5(0.4704)(0.5296) \approx 0.8719$. Step 4: $x_4 = 3.5(0.8719)(0.1281) \approx 0.3909$. Step 5: $x_5 = 3.5(0.3909)(0.6091) \approx 0.8333$. (Model continues to update values dynamically)

non-linearity is, on MATH in our setup, a more effective strategy than optimizing the gradient dynamics within a linear subspace.

For each method and rank, we select the learning rate on MATH pass@1 and report the corresponding test accuracy; the full sweep is shown in Table 5 in Appendix A.

Capacity Expansion vs. Linear Re-parameterization. We compare CeRA against DoRA, a SOTA method that decomposes weight updates into magnitude and direction. On the high-complexity MATH dataset, DoRA improves over LoRA at the matched rank 128 but remains bound by its linear formulation. CeRA at rank 64 surpasses DoRA at the same rank, and, crucially, CeRA achieves this with half of DoRA's rank-128 budget, so the gain reflects the expanded functional class rather than added parameters. The case study below traces how the gap between linear and nonlinear adapters surfaces on a single reasoning problem, using LoRA as the linear representative.

Task Complexity Comparison. Comparing results across datasets reveals a relationship between adapter expressivity and task complexity. On the basic GSM8K dataset, the intrinsic dimensionality is low; LoRA's linear subspace suffices. DoRA's lower GSM8K accuracy at rank 128 in our runs is consistent with the observation that LoRA variants favor different learning rate ranges, so a learning rate selected for one objective need not be optimal for another [Lee et al., 2026]; we report each method at its MATH-pass@1-selected learning rate (Appendix A).

3.4 Qualitative Analysis: Escaping the Linear Trap

Although quantitative metrics provide solid evidence of CeRA's superiority, it is crucial to examine how capacity expansion translates into actual reasoning capabilities. We conduct qualitative case studies. This section shows an iterative reasoning task. Iterative problems are challenging because they require the model to maintain and update a dynamic hidden state across multiple time steps without incurring error accumulation or state collapse.

Table 2 presents a representative example. LoRA ($r = 512$) exhibits state collapse. After correctly calculating the first three steps, the model fails to update its internal representation for the next iteration. It degenerates into a loop and outputs the same value at every following step, suggesting that LoRA's linear space is perhaps too rigid to capture the hidden dynamics of the recursive function. In contrast, CeRA, even with a smaller rank, recognizes that the value must change at each step. Although CeRA uses a smaller rank budget, its nonlinear components enable the model to project hidden states into a broader representation space, thereby escaping the linear trap.

Additional cases in both directions—including problems where LoRA's extra capacity does help, such as complex-number arithmetic—are collected in Appendix E.

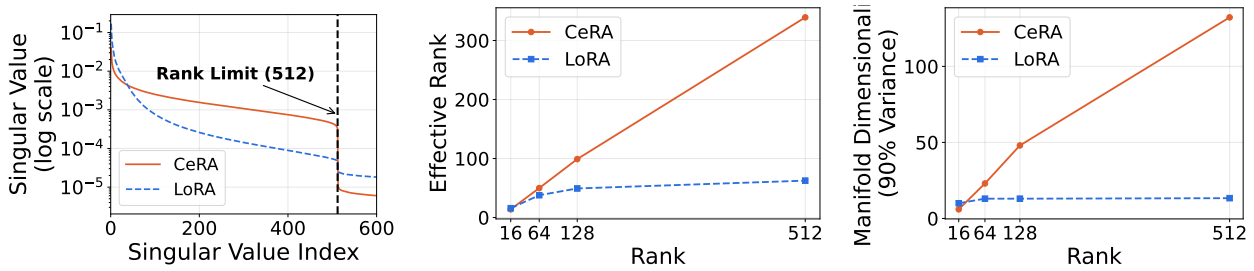

Figure 3: Spectral Analysis on SlimOrca. Left: Spectral Signature. LoRA exhibits rank collapse, whereas CeRA maintains a heavy tail. Middle: Effective Rank scales efficiently for CeRA but saturates for LoRA. Right: manifold dimensionality (90% variance). LoRA plateaus early, whereas CeRA's spectral dimensionality keeps growing with rank.

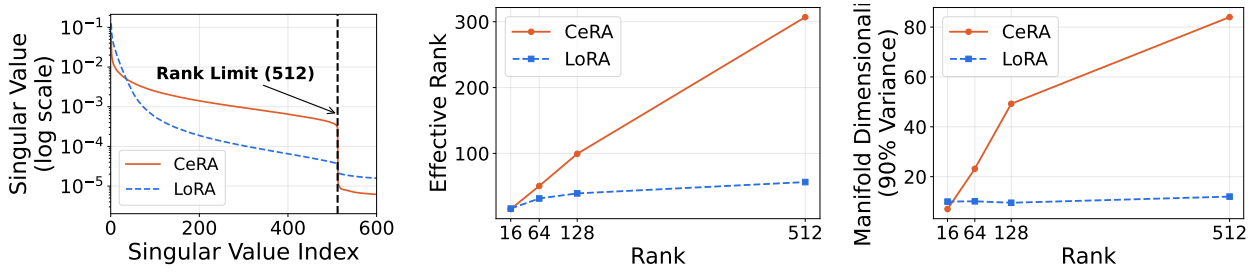

Figure 4: Spectral Analysis on MathInstruct. Left: Spectral Signature. LoRA exhibits a sharp drop. Middle: Effective Rank. LoRA's capacity saturates early. Right: Manifold Dimensionality (90% variance). CeRA requires more singular components to reach 90%.

4 Mechanism & Design Analysis

4.1 Spectral Signature, Effective Rank, & Manifold Dimensionality

We analyze the singular-value spectrum of the learned adapters and quantify the structural expansion using the effective rank (ER) metric [Roy and Vetterli, 2007]. The same signature appears in both SlimOrca and MathInstruct, the latter providing a mechanistic correlate of CeRA's MATH accuracy (Section 3.3).

Effective Rank. The effective rank measures the actual dimensionality of the information encoded in the activation space. For a matrix of adapter activations H with normalized singular values p_i, the effective rank is the exponential of the Shannon entropy. We mean-center the activation matrix H (column-wise) before computing its singular values.

$$\mathrm{ER}(H) = \exp\left(-\sum_{i=1}^{k} p_i \ln p_i\right) \tag{3}$$

A higher ER implies a more uniform distribution of energy across dimensions, indicating a broader representation space.

Manifold dimensionality. We also report the manifold dimensionality d_{90}, the smallest number of leading singular directions whose cumulative variance reaches 90% of the total:

$$d_{90}(H) = \min\left\{k : \frac{\sum_{i=1}^{k} \sigma_i^2}{\sum_j \sigma_j^2} \geq 0.9\right\}, \tag{4}$$

where $\sigma_1 \geq \sigma_2 \geq \cdots$ are the singular values of the (mean-centered) activation matrix H. ER weights every direction by its entropy; d_{90} is a hard threshold for explained variance; the two are complementary measures of how broadly variance is spread across the spectrum.

Spectral signature and effective rank across datasets. On both datasets, LoRA exhibits rank collapse: its singular values decay sharply after the leading directions (Figures 3 and 4, left), so increasing the rank budget contributes little to the expressivity. CeRA instead retains energy in the tail of the spectrum and expands its effective rank with the budget

Table 3: Deconstructing CeRA (rank 512). We report ER and test PPL on SlimOrca. ER here is computed per adapted module and then averaged; its absolute values are therefore not directly comparable to the pooled-spectrum ER of Table 12 in Appendix D, and comparisons are meaningful only within this table.

Model Variant	Component Change	ER ↑	Test PPL ↓
CeRA	Full Architecture (SiLU, Dropout 0.3)	**376.90**	3.91
(a) Granularity	Module-level (Parallel)	354.52	3.94
(b) Activation	Identity (Linear)	354.11	4.24
	ReLU	339.70	**3.69**
(c) Dropout	No Dropout	375.23	3.99

(Figure 3, middle). The same contrast holds on MathInstruct (Figure 4, middle): LoRA's effective rank saturates early while CeRA's keeps growing; CeRA's learned update does not collapse into a narrow low-rank subspace.

Manifold dimensionality and its link to reasoning. The manifold dimensionality—the number of singular directions needed to capture 90% of the variance—tells the same story (Figures 3 and 4, right): LoRA plateaus early, whereas CeRA's dimensionality grows with rank: the optimizer recruits the tail dimensions rather than concentrating variance in a few dominant directions. On MathInstruct, this broader representation coincides with CeRA's higher MATH pass@1 accuracy (Section 3.3).

Interpretation of the spectral results. These spectral contrasts are descriptive, not causal. The linear baselines and CeRA train under different effective output scales (α/r vs. $s = 1$), and a control experiment that matches the scale (Appendix D) shows that output scale, not non-linearity, accounts for most of the ER difference between LoRA and CeRA; moreover, ER does not track task performance (Section 4.2). We therefore use ER and d_{90} as summaries of the learned update's spectrum, and ground CeRA's justification in downstream accuracy and perplexity.

4.2 Ablation Study

We isolate CeRA's core components on SlimOrca. We run the ablation at rank 512 rather than a smaller rank because, at low rank, the ER of all variants saturates quickly and their differences compress, whereas at rank 512, the spectrum has room to separate them (Table 3).

Granularity. Replacing the weight-level with a module-level parallel adapter degrades PPL and lowers ER, confirming that intervening inside the W_q/W_v projections matters.

Activation. Here, PPL and ER diverge. Identity, which removes the activation entirely, is the most damaging change to predictive performance (PPL 4.24). Both non-linearities we test lower PPL relative to this linear variant, but they behave oppositely on the spectrum: ReLU attains the lowest PPL (3.69) yet collapses ER to 339.70—below even the linear baseline (354.11)—whereas SiLU expands it (to 376.90). The spectral expansion is thus specific to SiLU, not a generic consequence of adding non-linearity. Because our downstream MATH gains are not predicted by SlimOrca PPL alone, we justify the SiLU activation on downstream accuracy rather than validation perplexity.

Dropout. Removing structural dropout raises PPL but leaves ER essentially unchanged (375.23 vs. 376.90). Dropout therefore contributes mainly as a regularizer of the bottleneck, not as a driver of spectral expansion; the spectral expansion is attributable to SiLU.

4.3 Hyperparameter Robustness and Optimization Stability

A key factor for the adoption of a PEFT method is its robustness to hyperparameter choices. Recent variants, while effective in certain regimes, often introduce fragility in optimization. We evaluate the learning rate sensitivity of CeRA relative to the weight-decomposed DoRA.

Optimization Fragility in Linear Re-parameterization. DoRA decouples magnitude and direction to enhance flexibility. More generally, Lee et al. [2026] show that a LoRA variant's forward design and update rule can reshape the loss Hessian over training, and that the usable learning rate range scales inversely with sharpness (the largest Hessian

eigenvalue). We read DoRA's rank-dependent learning rate window in this light, consistent with the oscillations we observe in its high learning rate region.

Plug-and-Play Stability of CeRA. Because CeRA expands capacity through non-linearity rather than re-parameterizing the base-weight gradient pathways, its optimization landscape is generally stable across the learning-rate range we test. In our experiments, CeRA reaches competitive or better convergence under the same hyperparameter suite as LoRA at small and medium ranks; the one exception is the largest rank under aggressive learning rates ($r = 512$; see Section 3.2 and Appendix A), where CeRA degrades.

4.4 Computational and Memory Complexity

Since CeRA introduces non-linearity, the adapter weights cannot be merged into the base weights ($W_0 + \Delta W$) for zero-latency inference. We provide an analysis of the computational and memory overhead incurred by this unmerged paradigm to demonstrate its viability.

FLOPs Overhead Analysis. Let a linear projection layer with input dimension d, output dimension k, and adapter rank r. An unmerged linear adapter requires an additional $O(d \times r + r \times k)$ FLOPs. For CeRA, the forward pass requires the same matrix multiplications as the unmerged linear adapter, plus the element-wise SiLU activation, which takes $O(r)$ operations. Since $r \ll \min(d, k)$, the term $O(r)$ is negligible. Thus, the theoretical FLOPs overhead of CeRA is nearly identical to that of an unmerged LoRA.

Empirical Throughput in Multi-Tenant Serving. In multi-tenant serving infrastructures (e.g., S-LoRA [Sheng et al., 2024], Punica [Chen et al., 2024]), dynamic adapter switching is required to serve diverse user requests. In these systems, all adapters must be evaluated in an unmerged state. Under this paradigm, our benchmarks with Llama-3.1-8B indicate that CeRA incurs only a 6% latency overhead compared to the unmerged linear baseline. The throughput remains consistently stable across varying ranks (≈ 51 tokens/second for both $r = 64$ and $r = 128$). The marginal latency gap is dominated primarily by memory-bound kernel-launching costs for the SiLU operation, rather than by arithmetic bottlenecks.

Memory Utilization. During training, CeRA's memory footprint is highly efficient. Because a low-rank CeRA ($r = 64$) can achieve the performance of a high-rank LoRA ($r = 512$), the optimizer states and gradient buffers from the trainable parameters are reduced by a factor of 8. This reduction in VRAM outweighs the extra activation memory to store the intermediate SiLU states. Since these nonlinear activations occur only within the heavily compressed r-dimensional latent space ($r \ll d$), their spatial overhead is marginal.

5 Related Work

LoRA [Hu et al., 2022] established the standard for PEFT, assuming that weight updates reside in a low-rank linear subspace. Subsequent methods optimize this paradigm through dynamic rank allocation (AdaLoRA [Zhang et al., 2023]), weight decomposition (DoRA [Liu et al., 2024]), or weight quantization (QLoRA [Dettmers et al., 2023]). Although these methods improve learning dynamics, they remain bound by the linear formulation. CeRA is orthogonal to them; it alters the functional form of the update, replacing the linear subspace with a nonlinear capacity expansion. We further distinguish methods by *when* their non-linearity acts: Type-1 methods are nonlinear only during training and reduce to a mergeable affine map at inference, whereas CeRA (Type-2) retains its non-linearity at inference and stays non-mergeable. Among nonlinear low-rank methods, the closest are LoRAN [Li et al., 2024], which is also Type-2, and AuroRA [Dong et al., 2025], a Type-1 method whose non-linearity collapses to an affine map at inference. Concurrently, Lee et al. [2026] report that mergeable linear LoRA variants rarely beat vanilla LoRA once learning rates are tuned; CeRA differs in kind by retaining a non-mergeable inference-time non-linearity, and our experiments accordingly compare all methods within matched (rank, learning-rate) cells (Appendices A and B). A detailed methodological comparison with nonlinear low-rank adapters is given in Appendix C.

Furthermore, unlike early sequential adapters [Houlsby et al., 2019] or traditional parallel adapters [He et al., 2022, Zhu et al., 2021] that operate coarsely at the module level (processing the aggregate output of an attention block), CeRA operates at the fine-grained weight level. As summarized in Table 4, CeRA modifies the attention mechanism's internal feature dynamics by injecting non-linearity into the internal query and value projections.

Table 4: Feature Comparison. Unlike LoRA and its variants, CeRA introduces nonlinearity. Unlike traditional adapters—both sequential (Houlsby) and parallel—which operate at the module level, CeRA operates at the fine-grained weight level, enabling precise capacity expansion within attention projections.

Feature	LoRA	Houlsby Adapter	Parallel Adapter	CeRA (Ours)
Insertion Point (Granularity)	Weight-level (W_q, W_v internal)	Module-level (After Attn/FFN)	Module-level (Parallel to Attn/FFN)	Weight-level (W_q, W_v internal)
Structure	Linear Update	Nonlinear MLP	Nonlinear MLP	Nonlinear bottleneck MLP (SiLU)
Non-Linearity	No	Yes	Yes	Yes
Mergeable?	Yes	No	No	No
Expressivity	Low	High	High	High

6 Conclusion and Limitations

We revisit the linear assumption in Parameter-Efficient Fine-Tuning. For reasoning-intensive tasks, we observe that increasing LoRA's rank yields diminishing returns, raising the question of whether linear adapters face a capacity ceiling. We introduce CeRA, a weight-level architecture that uses SiLU gating and structural dropout to induce nonlinear capacity expansion. Under a fixed parameter budget, CeRA is competitive with much larger linear adapters: at rank 64 it matches or exceeds LoRA and DoRA of the same rank and a rank-512 LoRA in exact-match accuracy on MATH, using one-eighth of the trainable parameters. On MATH in our setting, altering the adapter's structural capacity is thus a more effective use of budget than optimizing gradient dynamics within a linear subspace.

Although CeRA offers clear advantages, it has limitations. First, because CeRA retains its non-linearity at inference, its update cannot be merged into the base weights ($W_0 + \Delta W$) for zero-latency inference. In multi-tenant serving, this is not a drawback: many adapters share a single frozen base and are kept unmerged in any case—in the cloud (S-LoRA [Sheng et al., 2024], Punica [Chen et al., 2024]) and on edge devices (EdgeLoRA [Shen et al., 2025])—so the additional cost is modest. However, for single-adapter deployment under strict latency or memory-bandwidth budgets, CeRA's non-mergeability is a genuine cost, and an affine-at-inference, mergeable method (LoRA, or AuroRA's static form [Dong et al., 2025]) is the appropriate choice. Second, although we evaluate across model scales within the Llama family (1B, 3B, 8B; Appendix B), our analysis is confined to the Llama-3 family. Validating the capacity-expansion hypothesis on other architectures (e.g., Mixture-of-Experts) and exploring hybrid designs that pair DoRA's stable optimization with CeRA's expressivity remain future work. Third, our linear and nonlinear adapters do not share a matched effective gain: the linear baselines use a fixed $\alpha = 32$, so their update scale α/r *decreases* with rank, whereas CeRA uses unit scaling. This regime may suppress high-rank adapters [Kalajdzievski, 2023], so the rank saturation we interpret as a linear ceiling may be partly due to scaling.

Acknowledgments and GenAI Usage Disclosure

We acknowledge support from the National Science and Technology Council of Taiwan under grant number 113-2221-E-008-100-MY3. The authors used Gemini and Claude to improve language and readability. The authors used Claude Code to help with the coding and experimentation. The authors reviewed and edited the content and code as needed.

References

Lequn Chen, Zihao Ye, Yongji Wu, Danyang Zhuo, Luis Ceze, and Arvind Krishnamurthy. Punica: Multi-tenant LoRA serving. *Proceedings of Machine Learning and Systems*, 6:1–13, 2024.

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

Pu Chen and Hung-Hsuan Chen. Accelerating matrix factorization by overparameterization. In *International Conference on Deep Learning Theory and Applications*, 2020.

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

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. QLoRA: Efficient finetuning of quantized LLMs. *Advances in Neural Information Processing Systems*, 36:10088–10115, 2023.

Haonan Dong, Wenhao Zhu, Guojie Song, and Liang Wang. AuroRA: Breaking low-rank bottleneck of LoRA with nonlinear mapping. *arXiv preprint arXiv:2505.18738*, 2025.

Junxian He, Chunting Zhou, Xuezhe Ma, Taylor Berg-Kirkpatrick, and Graham Neubig. Towards a unified view of parameter-efficient transfer learning. In *International Conference on Learning Representations*, 2022.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the MATH dataset. *Advances in Neural Information Processing Systems (Datasets and Benchmarks Track)*, 2021.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for NLP. In *International Conference on Machine Learning*, pages 2790–2799. PMLR, 2019.

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.

Damjan Kalajdzievski. A rank stabilization scaling factor for fine-tuning with LoRA. *arXiv preprint arXiv:2312.03732*, 2023.

Yu-Ang Lee, Ching-Yun Ko, Pin-Yu Chen, and Mi-Yen Yeh. Learning rate matters: Vanilla LoRA may suffice for LLM fine-tuning. *arXiv preprint arXiv:2602.04998*, 2026.

Yinqiao Li, Linqi Song, and Hanxu Hou. LoRAN: Improved low-rank adaptation by a non-linear transformation. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 3134–3143, 2024.

Wing Lian, Guan Wang, Bleys Goodson, Eugene Pentland, Austin Cook, Chanvichet Vong, and "Teknium". SlimOrca: An open dataset of GPT-4 augmented FLAN reasoning traces, with verification, 2023. URL `https://huggingface.co/Open-Orca/SlimOrca`.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *International Conference on Learning Representations*, 2024.

Shih-Yang Liu, Chien-Yi Wang, Hongxu Yin, Pavlo Molchanov, Yu-Chiang Frank Wang, Kwang-Ting Cheng, and Min-Hung Chen. DoRA: Weight-decomposed low-rank adaptation. In *Forty-first International Conference on Machine Learning*, 2024.

Ziyue Liu, Ruijie Zhang, Zhengyang Wang, Mingsong Yan, Zi Yang, Paul D. Hovland, Bogdan Nicolae, Franck Cappello, Sui Tang, and Zheng Zhang. CoLA: Compute-efficient pre-training of LLMs via low-rank activation. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 4627–4645, 2025.

Olivier Roy and Martin Vetterli. The effective rank: A measure of effective dimensionality. In *2007 15th European Signal Processing Conference*, pages 606–610. IEEE, 2007.

Zheyu Shen, Yexiao He, Ziyao Wang, Yuning Zhang, Guoheng Sun, Wanghao Ye, and Ang Li. EdgeLoRA: An efficient multi-tenant LLM serving system on edge devices. In *Proceedings of the 23rd Annual International Conference on Mobile Systems, Applications and Services*, pages 138–153, 2025.

Ying Sheng, Shiyi Cao, Dacheng Li, Coleman Hooper, Nicholas Lee, Shuo Yang, Christopher Chou, Banghua Zhu, Lianmin Zheng, Kurt Keutzer, et al. S-LoRA: Serving thousands of concurrent LoRA adapters. *Proceedings of Machine Learning and Systems*, 6, 2024.

Yuan-Chih Yang and Hung-Hsuan Chen. Dynamic DropConnect: Enhancing neural network robustness through adaptive edge dropping strategies. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, 2025.

Xiang Yue, Xingwei Qu, Ge Zhang, Yao Fu, Wenhao Huang, Huan Sun, Yu Su, and Wenhu Chen. MAmmoTH: Building math generalist models through hybrid instruction tuning. In *International Conference on Learning Representations*, 2024.

Qingru Zhang, Minshuo Chen, Alexander Bukharin, Nikos Karampatziakis, Pengcheng He, Yu Cheng, Weizhu Chen, and Tuo Zhao. AdaLoRA: Adaptive budget allocation for parameter-efficient fine-tuning. In *International Conference on Learning Representations*, 2023.

Yibo Zhong, Haoxiang Jiang, Lincan Li, Ryumei Nakada, Tianci Liu, Linjun Zhang, Huaxiu Yao, and Haoyu Wang. PEANuT: Parameter-efficient adaptation with weight-aware neural tweakers. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 1*, pages 2054–2065, 2026.

Yaoming Zhu, Jiangtao Feng, Chengqi Zhao, Mingxuan Wang, and Lei Li. Counter-interference adapter for multilingual machine translation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2812–2823, 2021.

A Hyperparameter Choice and Complete Learning Rate Sweep

To ensure a fair comparison, we conducted an extensive learning rate sweep for all evaluated methods (LoRA, DoRA, and CeRA) under identical training budgets. Rather than reporting a single tuned configuration per method, we present the full sweep so that comparisons can be made within identical (rank, learning rate) settings. The dropout rate was set to $p = 0.1$ for all CeRA main results, while the LoRA and DoRA baselines use no dropout. The higher $p = 0.3$ appears only in the SlimOrca ablation (Table 3).

Optimization Setup. All models were fine-tuned with the AdamW optimizer, using bfloat16 precision and gradient clipping to a max norm of 1.0. Batch size (4), gradient accumulation steps (16, i.e., an effective batch size of 64), and sequence length (512) were kept constant across all methods to control for batch-level optimization effects.

Sweep Protocol. For each method, we sweep the learning rate over $\{1 \times 10^{-4}, 3 \times 10^{-4}, 5 \times 10^{-4}, 1 \times 10^{-3}\}$ at ranks $\{64, 128, 512\}$, with $\alpha = 32$. The full grid is reported in Table 5. The best learning rate per (method, rank) is selected on MATH pass@1; this selection is applied identically to all methods and yields the values in Table 1. We use this complete grid, rather than only the per-method best configurations, for the matched-cell analysis below.

Table 5: Complete learning rate sweep on MATH pass@1 (%) for Llama-3.1-8B. Each row is a matched (rank, learning rate) setting; **bold** marks the best learning rate per (method, rank), i.e. the values reported in Table 1. ✓ marks cells where CeRA matches or exceeds the baseline at that identical setting: CeRA $\geq$ LoRA in 10 of 12 cells and CeRA $\geq$ DoRA in 10 of 12. The $r = 512$/5e-4 cell ties with DoRA; all CeRA shortfalls relative to LoRA occur at $r = 512$ under the highest learning rates.

Rank	LR	LoRA	DoRA	CeRA	CeRA $\geq$ LoRA	CeRA $\geq$ DoRA
64	1×10^{-4}	12.6	9.8	14.6	✓	✓
	3×10^{-4}	**21.6**	**19.8**	**23.6**	✓	✓
	5×10^{-4}	18.2	18.6	22.0	✓	✓
	1×10^{-3}	15.0	7.2	21.8	✓	✓
128	1×10^{-4}	13.4	11.2	15.6	✓	✓
	3×10^{-4}	**20.6**	15.8	21.6	✓	✓
	5×10^{-4}	17.0	**21.2**	18.0	✓	
	1×10^{-3}	13.4	18.2	**22.0**	✓	✓
512	1×10^{-4}	14.4	12.6	19.6	✓	✓
	3×10^{-4}	**22.4**	16.6	**23.0**	✓	✓
	5×10^{-4}	18.4	17.0	17.0		✓
	1×10^{-3}	18.0	**17.6**	10.6		

Matched-cell Comparison (CeRA vs. LoRA and CeRA vs. DoRA). Because cross-run comparisons at mismatched learning rates can be misleading, we compare CeRA vs. LoRA (and CeRA vs. DoRA) only within identical (rank, learning rate) settings. As shown in Table 5, CeRA matches or exceeds LoRA in 10 of the 12 matched cells; under a one-sided sign test, this consistency is significant at $p \approx 0.019$. The two exceptions both occur at the largest rank ($r = 512$) under the two highest learning rates, where CeRA's accuracy drops. This matched-cell consistency, rather than any single headline gap, is the basis for our claim that inference-time non-linearity—not parameter count—is what lifts the linear ceiling; the same holds against DoRA (CeRA matches or wins in 10 of 12). We emphasize this because Lee et al. [2026] show that mergeable, linear LoRA variants (e.g., PiSSA, MiLoRA, DoRA) do not consistently beat vanilla LoRA once their learning rates are tuned, with peak accuracies typically within 1–2%. CeRA lies outside the class they study: it retains an input-dependent non-linearity at inference (Type-2, non-mergeable). Our evidence is correspondingly not a single tuned gap but (i) the matched-cell consistency above (Table 5) and (ii) the rank-scaling perplexity margin across 1B/3B/8B (Appendix B), both controlled exactly for learning rate.

Table 6: Llama-3.2-1B, MathInstruct held-out perplexity (lower is better), from the same runs as Table 9. **Bold** denotes the best learning rate for each (method, rank). ✓ marks cells where CeRA $\leq$ the baseline; CeRA is lower than both LoRA and DoRA in all 12 matched cells, and the margin widens with rank.

Rank	LR	LoRA	DoRA	CeRA	CeRA $\leq$ LoRA	CeRA $\leq$ DoRA
64	1×10^{-4}	2.62	2.62	2.60	✓	✓
	3×10^{-4}	2.48	2.48	2.46	✓	✓
	5×10^{-4}	2.45	**2.43**	**2.41**	✓	✓
	1×10^{-3}	**2.44**	**2.43**	**2.41**	✓	✓
128	1×10^{-4}	2.62	2.62	2.54	✓	✓
	3×10^{-4}	2.47	2.46	2.38	✓	✓
	5×10^{-4}	2.40	**2.40**	**2.34**	✓	✓
	1×10^{-3}	**2.39**	**2.40**	2.36	✓	✓
512	1×10^{-4}	2.62	2.62	2.39	✓	✓
	3×10^{-4}	2.46	2.46	**2.24**	✓	✓
	5×10^{-4}	2.40	2.39	**2.24**	✓	✓
	1×10^{-3}	**2.36**	**2.35**	2.29	✓	✓

LoRA. Vanilla LoRA attains its best MATH pass@1 at 3×10^{-4} across all ranks, which we therefore use as its reference configuration. Its accuracy degrades steadily as the learning rate exceeds this threshold, reflecting the learning rate sensitivity commonly reported for linear adapters.

DoRA. DoRA is the most sensitive to the learning rate of the three methods. It underfits at the lowest learning rate (1×10^{-4}: 9.8/11.2/12.6 across ranks) and becomes unstable at the highest (1×10^{-3}: 7.2 at $r = 64$). Its best learning rate is not uniform but varies with rank (3×10^{-4} at $r = 64$, 5×10^{-4} at $r = 128$). This rank-dependent usable window is consistent with the general sharpness–learning-rate relationship of Lee et al. [2026]. We also empirically observe DoRA's relative weakness on GSM8K (Table 1).

CeRA. CeRA reaches its best or near-best accuracy at 3×10^{-4}—the same learning rate that is optimal for LoRA—so no separate tuning was required to switch from LoRA to CeRA. CeRA is notably robust on the high learning rate side at small and medium rank: at $r = 64$ it stays within ~ 2 points across $\{3\text{e-}4, 5\text{e-}4, 1\text{e-}3\}$ (23.6/22.0/21.8), and at $r = 128$ it does not collapse at 1×10^{-3} (22.0). This robustness has two boundaries we state explicitly: like all methods, CeRA underfits at the lowest learning rate (1×10^{-4}), and at the largest rank ($r = 512$), its accuracy drops under aggressive learning rates (10.6 at 1×10^{-3}).

B Model Scaling Test

To test whether CeRA's behavior is specific to the 8B backbone, we repeat the rank and learning rate sweep on two smaller models in the same family, Llama-3.2-1B and Llama-3.2-3B, fine-tuned on MathInstruct with AdamW at a fixed learning rate. For each (rank, learning rate) cell, we report greedy pass@1 on our 500-problem algebra slice and MathInstruct held-out perplexity, computed from the same run. The 8B accuracy sweep appears in Appendix A (Table 5); its perplexity is in Table 8.

The fitting advantage is consistent and scales with rank. CeRA attains a lower MathInstruct held-out perplexity than *both* LoRA and DoRA in 35 of the 36 matched (rank, learning rate) cells across 1B, 3B, and 8B (Tables 6, 7, 8). The sole exception is the "high rank/high learning rate" cell (r=512, 1×10^{-3}) at 8B, where CeRA's training destabilizes—the same cell in which its MATH pass@1 collapses (Table 5), so the instability appears consistently across both metrics. Elsewhere, the margin widens with rank on every scale: the best learning rate gap over LoRA grows from $\approx$0.03 at $r = 64$ to $\approx$0.10 – 0.12 at $r = 512$, mirroring the capacity-scaling trend observed in SlimOrca in Figure 1. CeRA continues to use an additional rank where the linear baselines plateau.

Downstream accuracy is gated by backbone capability. The exact-match picture is scale-dependent (Tables 9, 10). Despite CeRA's near-uniform perplexity advantage, the gain appears only as exact-match accuracy, where MATH is actually learnable. At 8B, MATH is the discriminating task and CeRA $\geq$ LoRA in 10 of 12 matched cells (Table 5). At 1B, MATH sits near the noise floor for all methods ($\leq 5\%$ pass@1) and is uninformative as an accuracy benchmark.

Table 7: Llama-3.2-3B, MathInstruct held-out perplexity, from the same runs as Table 10. Conventions as in Table 6; CeRA is lower than both baselines in all 12 matched cells.

Rank	LR	LoRA	DoRA	CeRA	CeRA $\leq$ LoRA	CeRA $\leq$ DoRA
64	1×10^{-4}	2.34	2.34	2.32	✓	✓
	3×10^{-4}	2.20	2.20	2.17	✓	✓
	5×10^{-4}	**2.14**	2.15	**2.11**	✓	✓
	1×10^{-3}	**2.14**	**2.11**	**2.11**	✓	✓
128	1×10^{-4}	2.33	2.34	2.26	✓	✓
	3×10^{-4}	2.19	2.18	2.08	✓	✓
	5×10^{-4}	2.12	2.11	**2.02**	✓	✓
	1×10^{-3}	**2.09**	**2.09**	2.06	✓	✓
512	1×10^{-4}	2.34	2.34	2.09	✓	✓
	3×10^{-4}	2.18	2.18	**1.93**	✓	✓
	5×10^{-4}	2.09	2.10	1.94	✓	✓
	1×10^{-3}	**2.03**	**2.06**	2.00	✓	✓

Table 8: Llama-3.1-8B, MathInstruct held-out perplexity, from the same runs as Table 5. Conventions as in Table 6. CeRA is lower than both LoRA and DoRA in 11 of 12 cells; the sole exception is the "high rank/high learning rate" collapse at $r = 512/1 \times 10^{-3}$ (the same cell where MATH pass@1 collapses).

Rank	LR	LoRA	DoRA	CeRA	CeRA $\leq$ LoRA	CeRA $\leq$ DoRA
64	1×10^{-4}	2.014	2.014	2.003	✓	✓
	3×10^{-4}	1.940	1.939	1.906	✓	✓
	5×10^{-4}	**1.919**	**1.916**	**1.885**	✓	✓
	1×10^{-3}	1.956	1.952	1.899	✓	✓
128	1×10^{-4}	2.013	2.016	1.963	✓	✓
	3×10^{-4}	1.929	1.927	1.862	✓	✓
	5×10^{-4}	**1.916**	**1.904**	**1.828**	✓	✓
	1×10^{-3}	1.936	1.921	1.881	✓	✓
512	1×10^{-4}	2.013	2.014	1.852	✓	✓
	3×10^{-4}	1.910	1.915	**1.790**	✓	✓
	5×10^{-4}	1.910	**1.893**	1.837	✓	✓
	1×10^{-3}	**1.897**	**1.893**	2.417		

However, CeRA's direction is consistent (CeRA $\geq$ LoRA in all 12 cells, at very small absolute values). At 3B, MATH is only weakly learnable, and CeRA and LoRA are statistically comparable on it (CeRA $\geq$ LoRA in 6 of 12 cells), although CeRA's perplexity is lower (better) in all 12. Better distribution fitting, therefore, does not automatically convert into higher exact-match accuracy when the task sits at the backbone's capability frontier.

Interpretation. Read together, these results indicate that CeRA's nonlinear capacity yields a consistent, rank-scaling improvement in the adapter's fit to the target distribution across all scales. Meanwhile, its *downstream* advantage emerges only when the backbone is sufficiently capable to represent the harder task. This refines, rather than contradicts, the task-complexity relationship of Section 3.3: "complexity" is relative to backbone capability, not absolute.

Robustness. CeRA's high learning rate stability persists at smaller scales: at $r \leq 128$ it shows no collapse at 1×10^{-3} on either backbone, whereas LoRA degrades at high learning rate (e.g., 3B MATH pass@1 $10.0 \rightarrow 6.6$ from 5×10^{-4} to 1×10^{-3}). The high-rank collapse observed for CeRA appears only at 8B and $r = 512$ under the highest learning rate.

Table 9: Llama-3.2-1B, complete learning rate sweep (MATH pass@1, %). Each row is a matched (rank, learning rate) setting; **bold** denotes the best learning rate for each (method, rank). ✓ marks cells where CeRA $\geq$ the baseline at that identical setting: CeRA $\geq$ LoRA in all 12 cells and $\geq$ DoRA in 9 of 12. Absolute MATH accuracy is near the noise floor at this scale ($\leq 5\%$), so these consistencies are directional only.

Rank	LR	LoRA	DoRA	CeRA	CeRA $\geq$ LoRA	CeRA $\geq$ DoRA
64	1×10^{-4}	0.8	1.6	1.6	✓	✓
	3×10^{-4}	**1.8**	1.6	**3.2**	✓	✓
	5×10^{-4}	1.6	2.4	1.6	✓	
	1×10^{-3}	1.2	**3.2**	2.2	✓	
128	1×10^{-4}	0.4	1.6	1.0	✓	
	3×10^{-4}	2.2	1.2	**5.0**	✓	✓
	5×10^{-4}	**3.0**	2.0	3.2	✓	✓
	1×10^{-3}	2.2	**2.6**	2.8	✓	✓
512	1×10^{-4}	1.8	1.4	3.2	✓	✓
	3×10^{-4}	2.6	1.8	**4.6**	✓	✓
	5×10^{-4}	**3.0**	**2.2**	3.4	✓	✓
	1×10^{-3}	**3.0**	2.0	4.0	✓	✓

Table 10: Llama-3.2-3B, complete learning rate sweep (MATH pass@1, %). Conventions as in Table 9. CeRA $\geq$ LoRA in 6 of 12 matched cells and $\geq$ DoRA in 8 of 12; on MATH, the three methods are statistically comparable at this scale.

Rank	LR	LoRA	DoRA	CeRA	CeRA $\geq$ LoRA	CeRA $\geq$ DoRA
64	1×10^{-4}	7.4	5.4	7.0		✓
	3×10^{-4}	**11.0**	7.8	**10.8**		✓
	5×10^{-4}	8.0	8.2	10.4	✓	✓
	1×10^{-3}	9.6	**8.6**	9.8	✓	✓
128	1×10^{-4}	4.8	5.6	7.8	✓	✓
	3×10^{-4}	8.4	**11.0**	7.4		
	5×10^{-4}	**11.2**	**11.0**	**8.8**		
	1×10^{-3}	9.0	9.8	8.4		
512	1×10^{-4}	6.0	5.4	10.2	✓	✓
	3×10^{-4}	8.2	8.8	7.8		
	5×10^{-4}	**10.0**	**11.2**	**11.2**	✓	✓
	1×10^{-3}	6.6	8.0	10.4	✓	✓

C Extended Comparison with Nonlinear PEFT Methods

We classify nonlinear low-rank adapters by whether the adapter's forward map is nonlinear in the input x and, if so, whether that nonlinearity survives at inference (Table 11). The first group is effectively *linear*: LoRA, DoRA, and PEANuT (formerly NEAT) [Zhong et al., 2026] are linear in x both in training and inference and remain mergeable. Although PEANuT applies a nonlinear network to the *frozen* weights, $f(W_0)$, this yields a fixed update matrix, so the adapter stays linear in x. Its nonlinearity lies in the weight-space parameterization, not in the input–output map. *Type-1* methods are nonlinear in x during training but collapse to a fixed affine map at inference, so they remain mergeable: AuroRA [Dong et al., 2025] trains with an input-dependent update $B\sigma(Ax)$ but reverts at inference to a static update $\Delta W = B\sigma(A)$, which is affine in x. *Type-2* methods retain their input-dependent nonlinearity at inference and are therefore not mergeable. CeRA is Type-2. Its closest neighbor is LoRAN [Li et al., 2024], which inserts a sinusoidal activation after the low-rank projection. CeRA differs in that it applies SiLU gating with structural dropout at the weight level (within W_q and W_v). Finally, CoLA [Liu et al., 2025] is a from-scratch pre-training method that replaces dense projections with low-rank autoencoders, targeting a different setting from PEFT on a frozen base, and therefore lies outside this taxonomy.

Table 11: Nonlinear (NL) low-rank adaptation methods classified by whether the adapter's forward map is nonlinear *in the input* x at training and at inference. Only methods that remain nonlinear in x at inference (Type-2) are non-mergeable. CeRA is Type-2.

Method	Mechanism	NL in x (train)	NL in x (infer)	Mergeable	Class
LoRA	Linear BA	No	No	Yes	Linear
DoRA	Magnitude–direction reparam.	No	No	Yes	Linear
PEANuT (NEAT)	Weight-space $f(W_0)$	No	No	Yes	Linear
AuroRA	$\sigma(Ax)$ (train) $\rightarrow \sigma(A)x$ (infer)	Yes	No	Yes	Type-1
LoRAN	$f(Ax)$, Sinter (sine)	Yes	Yes	No	Type-2
CoLA[†]	Autoencoder (pre-training)	Yes	Yes	n/a	—
CeRA (Ours)	$\sigma(Ax)$, SiLU + dropout	Yes	Yes	No	Type-2

[†] CoLA pre-trains from scratch, not a PEFT adapter on a frozen base; mergeability axis does not apply.

D Effective Rank vs. Output Scale vs. Non-linearity

In the main paper, we report effective rank (ER) as a spectral diagnostic. We had initially intended ER to serve a stronger, mechanistic role—non-linearity expands the spectrum of the adapted representation—but a control experiment leads us to withdraw that causal reading. Holding the measurement pipeline, rank (512), learning rate (5×10^{-4}), and dropout (0) fixed, we vary only the adapter's output scale and its activation (Table 12).

Table 12: Effective rank under a fixed measurement pipeline ($r = 512$, lr $= 5 \times 10^{-4}$, dropout $= 0$). ER is computed on the same activation matrix for all rows: a *single* SVD over the pooled spectrum of all adapted modules. Table 3 instead averages per-module ERs (and its identity variant uses dropout 0.3), so absolute ER values are not comparable across the two tables; only within-table comparisons are meaningful. Raising the linear adapter's output scaling *alone* accounts for most of the gap to CeRA; adding SiLU contributes only a small remainder.

Configuration	Scaling	ER
LoRA ($\alpha = 32$, default)	0.0625	100.11
LoRA ($\alpha = 512$)	1.0	299.60
CeRA, identity activation (linear)	1.0	348.62
CeRA, SiLU	1.0	387.56

Two observations follow. First, output scale, not non-linearity, drives most of the ER difference. Raising LoRA's scaling from its default $\alpha/r = 0.0625$ to 1.0 moves ER from 100.1 to 299.6 *without any architectural change*; a linear CeRA variant (identity activation, unit scaling) already reaches 348.6, and adding SiLU contributes only a further $\sim$11% (387.6). Of the total gap between default LoRA and CeRA-SiLU, roughly 70% is attributable to scaling and $\sim$14% to the SiLU non-linearity. Because ER is invariant to a global rescaling of the activation matrix by construction (rescaling all singular values by a constant leaves the normalized spectrum, and hence the entropy, unchanged), this shift is not a measurement artifact; it reflects the fact that the training-time gain alters the *learned* weights. The dependence on output scale also connects to a known property of the α/r convention, under which the effective update of a linear adapter is suppressed as rank grows [Kalajdzievski, 2023].

Second, ER does not track task performance. In our ablation (Table 3), the ReLU variant attains the *lowest* ER yet the *best* perplexity, while the linear variant has high ER but the *worst* perplexity. ER is therefore neither a clean function of architecture nor a reliable predictor of downstream quality. We consequently use ER and d_{90} as descriptive summaries of the adapted spectrum and ground CeRA's justification in downstream accuracy and perplexity rather than in spectral expansion. Isolating what genuinely distinguishes nonlinear from linear adapters once output scale is matched—for instance, via gain-matched spectral probes or conditional-linear-map analyses—remains open for future work.

E Case studies

To see how CeRA's capacity expansion plays out problem by problem, we compare CeRA ($r = 64$, 27.3M trainable parameters) with the strongest baseline from Table 1 (LoRA, $r = 512$, 218.1M, an $8\times$ larger budget). Both use greedy decoding, and both are evaluated on the same 500 MATH problems and the full GSM8K test set. We focus

Table 13: Representative bidirectional cases, CeRA ($r = 64$, 27.3M params) vs. LoRA ($r = 512$, 218.1M, an $8\times$ larger budget) on Llama-3.1-8B, greedy pass@1. Top block: CeRA correct, LoRA wrong—here LoRA's failures are typically *strategic or conceptual* (a wrong method or a misread structure). Bottom block: LoRA correct, CeRA wrong—here CeRA's failures are typically *low-level execution slips* (a dropped sign, factor, or term). These cases are selected to illustrate that tendency, not absolute.

ID	Problem (abbreviated)	Gold	CeRA	LoRA	Decisive difference (loser's error)
CeRA $r = 64$ correct / LoRA $r = 512$ wrong — LoRA errs strategically					
M14	Kite area from vertex coordinates ($\frac{1}{2}d_1d_2$)	75	75	65	LoRA invents a false identity $(AB)(CD) = (AC)(BD)$ instead of using the diagonals.
M28	Roots a, b of $x^2 - 5x + 9 = 0$; find $(a-1)(b-1)$	5	5	—	LoRA takes the quadratic formula into $\sqrt{-11}$ and never recovers; CeRA uses Vieta's formulas.
M87	-4 is a root of $x^2 + bx - 36 = 0$; find b	-5	-5	8	LoRA misapplies Vieta, assuming *both* roots equal -4.
M124	$b-a$ for integer solutions of $x^2-15 < 2x$	6	6	-2	Both factor $(x-5)(x+3)<0$; LoRA selects the *complementary* region ($x<-3$ or $x>5$ instead of $-3<x<5$), so its endpoints are wrong.
G24	\$19.50 after a 25% discount; original price	26	26	24.4	LoRA adds 25% of the *sale* price instead of dividing by 0.75.
G30	Ages 7 : 11, total 162; Allen's age in 10 yr	109	109	172	LoRA mis-applies the ratio, lands on Allen $= 162$, and reports $162 + 10 = 172$; CeRA solves $18x = 162$
G47	$2\times$ red vs. blue ties, red 50% pricier; total spend	800	800	600	LoRA ignores the 50% markup and misstructures the counts.
LoRA $r = 512$ correct / CeRA $r = 64$ wrong — CeRA errs by a slip					
M47	Simplify $(2 - 2i)(5 + 5i)$	20	$20i$	20	CeRA drops the sign on $-2i$ (expands as if $(2+2i)$): both cross terms become $+10i$ (so $+20i$, not 0) and $+10i^2$ replaces $-10i^2$, collapsing the real part to 0. Uses $i^2=-1$ correctly; result $20i$.
M59	Simplify $(3 - i)(6 + 2i)$	20	$16 + 6i$	20	CeRA mishandles the i^2 sign and keeps a stray $6i$.
M272	Compute $(34-10)+(20-9)+(55-10)$	80	100	80	CeRA regroups correctly to $34+20+26$ but miscomputes the final sum as 100 instead of 80.
G18	3-egg omelet daily for 4 weeks; dozens of eggs	7	2	7	CeRA drops the $\times 3$ eggs/omelet factor (28 vs. 84 eggs).
G58	\$40 bill, +25% fee, +\$3 delivery, +\$4 tip; total	57	47	57	CeRA loses the \$10 (25%) fee mid-calculation.
G60	25 oranges: 1 bad, 20% unripe, 2 sour; good ones	17	18	17	CeRA forgets to subtract the single bad orange.
G88	Marilyn sold $10\times$ Harald, 88,000 total; Harald's sales	8000	9777	8000	CeRA writes $10x = 88000 - x$ then combines to $9x = 88000$ (subtracts x instead of adding), giving $88000/9 \approx 9777$.

on *disagreements*—problems that exactly one of the two adapters answers correctly—since these isolate behavioral differences that the aggregate accuracies blur.

Disagreement statistics. On MATH, the two models diverge on 112 of 500 problems: CeRA alone is correct on 59 and LoRA on 53 (a further 59 are solved by both). On GSM8K they diverge on 353 of 1,319: CeRA alone is correct on 187, LoRA alone on 166. The near-symmetry of the raw counts is itself informative—the headline MATH gap (23.6 vs. 22.4) understates how differently the two adapters reason and shows that CeRA's edge does not come from dominating a fixed subset of problems but from a different error profile achieved with one-eighth of the parameters. Table 13 collects representative cases from each direction.

An asymmetry in error type. Reading the disagreements, the two error profiles differ in kind, not just in count. When LoRA loses, it tends to fail at the level of *strategy or structure*: it reaches for the quadratic formula and strands itself in complex roots where a Vieta's-formulas shortcut applies (M28), invents a false area identity rather than using the kite's diagonals (M14), selects the wrong region of a quadratic inequality (M124), or mishandles the structure of a percentage or ratio word problem (G24, G30, G47). When CeRA loses, by contrast, its high-level approach is usually sound, and the failure is a *low-level execution slip*: a dropped sign in complex-number multiplication (M47, M59), a dropped multiplicative factor or term (G18, G58), an off-by-one (G60), or a plain arithmetic mistake (M272, G88). Consistent with the capacity argument, CeRA's wins also skew toward the hardest items: among the Level-5 algebra problems on which the two disagree, CeRA wins 11 and LoRA 6.

Scope and caveats. This asymmetry is a tendency, not a law, and we explicitly state its limits. It is the dominant pattern we observe, but counterexamples exist in both directions: LoRA also makes arithmetic slips, and—more importantly for our claim—CeRA also does fail *conceptually*, for instance, by choosing a worse solution method (completing the square where factoring is clean) or setting up the wrong equation. The cases in Table 13 are selected to illustrate the prevailing trend, not to assert that CeRA never errs strategically. Therefore, we treat these traces as qualitative illustrations, with systematic evidence remaining from the matched-cell sweep (Appendix A, Table 5). A small number of raw disagreements are answer-extraction artifacts—a correct value rejected by the `\boxed{}` parser because of trailing formatting—which we excluded when selecting examples.