MARS: Harmonizing Multimodal Convergence via Adaptive Rank Search

Minkyoung Cho[1] **Insu Jang**[1] **Shuowei Jin**[1] **Zesen Zhao**[1] **Adityan Jothi**[2]
Ethem F. Can[2] **Min-Hung Chen**[2,†] **Z. Morley Mao**[1,†]

[1]University of Michigan [2]NVIDIA

https://minkyoungcho.github.io/mars/

Abstract

Fine-tuning Vision Language Models (VLMs) with parameter-efficient methods like Low-Rank Adaptation (LoRA) is crucial for task adaptation. However, imbalanced training dynamics across modalities often lead to suboptimal accuracy due to negative interference, a challenge typically addressed with inefficient heuristic methods such as manually tuning separate learning rates. To overcome this, we introduce **MARS** (**M**ultimodal **A**daptive **R**ank **S**earch), an approach to discover optimal rank pairs that balance training dynamics while maximizing performance. Our key innovation, a proposed framework of dual scaling laws, enables this search: one law models module-specific convergence time to prune the search space to candidates with aligned dynamics, while the other predicts final task performance to select the optimal pair from the pruned set. By re-purposing the LoRA rank as a controller for modality-specific convergence speed, MARS outperforms baseline methods and provides a robust, automated strategy for optimizing VLM fine-tuning.

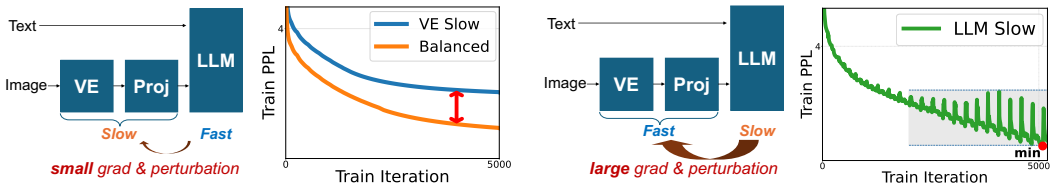
1 Introduction

Vision Language Models (VLMs) are built from three key components: a vision encoder (VE) that extracts visual representations, a projector that bridges visual and language spaces, and a large language model (LLM) backbone that reasons over the combined inputs. While early approaches kept the VE frozen and adapted only the projector or LLM, it has become increasingly clear that deeply integrated multimodal understanding requires fine-tuning all components jointly [1]. This has established comprehensive end-to-end fine-tuning as the dominant paradigm for state-of-the-art VLMs [2, 3, 4, 5]. However, given the immense scale of these models, full fine-tuning is computationally prohibitive, making parameter-efficient methods indispensable. Low-rank adaptation (LoRA) [6], which approximates weight updates with low-rank matrices, has emerged as the standard approach for enabling this comprehensive adaptation at tractable cost.

In practice, LoRA is typically applied uniformly across the VE and LLM, treating the two modules identically without differentiation [3, 4, 7]. However, because the VE and LLM originate from distinct pre-trained unimodal models with different parameter scales, they exhibit fundamentally different learning dynamics and converge at different rates under a shared training configuration. As illustrated in Figure 1, this imbalance leads to performance bottlenecks and training oscillations. A natural remedy is to assign differential learning rates to each module, but this is laborious in practice and depends on costly trial-and-error search [8, 9, 2]. Moreover, since the learning rate only scales gradient magnitudes, it only addresses the symptom – speed of convergence – rather than the root cause. A more fundamental lever is the LoRA rank itself, which directly controls each module's adaptation capacity and

[†]Equal advising. Correspondence: Minkyoung Cho (minkycho@umich.edu)

40th Conference on Neural Information Processing Systems (NeurIPS 2026).



(a) VE under-adapted causes a performance bottleneck. (b) LLM under-adapted causes training oscillations.

Figure 1: **Motivation: Imbalanced training dynamics lead to suboptimal performance.** Conceptual cases of module imbalance (left) and their corresponding empirical results (right).

acts as a regularizer [10]. Assigning differential ranks therefore offers a principled way to harmonize the training dynamics of the VE and LLM. However, identifying an optimal rank pair is challenging due to an inherent *two-fold disparity*: (1) a disparity in learning capacity, stemming from their differing parameter scales, and (2) a disparity in the required learning budget, as each module originates from a distinct pre-trained unimodal model with its own domain gap to the downstream task. Therefore, the core challenge lies in quantifying these disparities to align the convergence dynamics of all modules.

To address this, we introduce **MARS** (**M**ultimodal **A**daptive **R**ank **S**earch), an automated algorithm that assigns module-specific LoRA ranks to align the convergence dynamics of the VE and LLM. By searching for the rank pair that harmonizes each module's learning trajectory, MARS eliminates the need for heuristic tuning and systematically resolves the two-fold disparity. However, naively searching over all rank pairs is intractable; the combinatorial space is vast, and evaluating each candidate requires a full fine-tuning run. To make MARS efficient, we propose dual scaling laws tailored for VLM fine-tuning [11, 12, 13]: **Scaling Law-C** (Convergence) captures how each module's convergence rate scales with its LoRA rank, and **Scaling Law-P** (Performance) captures how final task performance varies with the rank pair. Their free parameters are estimated through a lightweight calibration phase on the target model. Armed with these laws, MARS efficiently identifies the optimal rank pair—retaining only candidates where both modules converge simultaneously via Scaling Law-C, and selecting the highest-performing one via Scaling Law-P—without running a single full fine-tuning job.

Our main contributions are:

- We identify and provide evidence that the imbalanced training dynamics in VLM fine-tuning, originating from a two-fold disparity, represent a key source of suboptimal performance. To overcome this, we propose MARS, an automated algorithm that systematically mitigates the imbalance by discovering optimal, modality-specific LoRA rank pairs.
- We are the first to propose and validate dual scaling laws for VLM fine-tuning that model module-specific convergence time (Scaling Law-C) and performance (Scaling Law-P), making the rank search feasible.
- MARS consistently outperforms fixed/adaptive-rank baselines, learning-rate tuning, and equal-budget search methods across diverse model–benchmark settings, while requiring only lightweight calibration and a single fine-tuning run ($11.5\times$ faster than exhaustive search).

2 Related Work and Motivation

Fine-Tuning Strategies for VLMs. State-of-the-art VLMs fine-tune all major components – vision encoder, projector, and LLM backbone – using LoRA [6] to keep adaptation tractable at scale [2, 3, 4, 5, 1]. However, prior work uniformly applies the same LoRA rank across all modules, treating them as homogeneous despite their fundamentally different parameter scales and pre-training origins [14, 8, 15, 9]. To partially compensate, practitioners resort to heuristically tuning module-specific learning rates; however, this process demands substantial manual effort and is acutely sensitive to miscalibration. Figure 1 illustrates the cost of such errors: a much lower VE rate ($lr_{ve} \ll lr_{llm}$) induces a severe performance bottleneck, while a much lower LLM rate ($lr_{llm} \ll lr_{ve}$) destabilizes training entirely. Differential learning rates thus offer only a partial solution; they modulate the speed of gradient updates but leave the fundamental mismatch in adaptation capacity unresolved.

Scaling Laws. Scaling laws have been established across LLM pre-training [11, 16], LLM fine-tuning [12], and multimodal pre-training [13, 17], demonstrating that final task performance scales predictably with compute, data, and model size. However, all of these focus on aggregate, end-to-end performance as the prediction target. No prior work has examined scaling behavior at the level of individual modules, nor studied how convergence dynamics of heterogeneous components, such as the VE and LLM, scale with their LoRA rank during fine-tuning.

Motivation. Adjusting learning rates is limited; hence, adjusting the LoRA rank is a more principled alternative, as the rank directly governs each module's adaptation capacity and implicitly regularizes fine-tuning [10]. However, identifying the appropriate rank pair is far from straightforward due to an inherent *two-fold disparity* between the VE and LLM. First, the two modules operate at vastly different parameter scales, giving each a distinct ceiling on adaptation. Second, each originates from a separate pretrained unimodal model with its own domain gap to the downstream task, making the adaptation effort inherently unequal across modules. These asymmetries call for a principled, data-driven method for rank selection, inspired by the predictive power of scaling laws in characterizing model behavior.

3 Multimodal Adaptive Rank Search (MARS) Methodology

MARS is a pre-fine-tuning procedure that identifies the optimal modality-specific LoRA rank pair $(r^*_{\text{ve}}, r^*_{\text{llm}})$ to align the convergence dynamics of the VE and LLM before task-specific training begins. MARS addresses this by introducing dual scaling laws that characterize how rank governs module behavior: **Scaling Law-C** models the relationship between LoRA rank and each module's convergence speed, while **Scaling Law-P** models the relationship between the rank pair and final task performance (§ 3.1). The guided search algorithm then uses these laws to efficiently find the optimal pair – using Scaling Law-C to prune the space to convergence-aligned candidates and Scaling Law-P to select the best-performing one – without running a single full fine-tuning job, as shown in Algorithm 1 (§ 3.2). The parameters required by the dual scaling laws are empirically estimated through a lightweight calibration phase that precedes the guided search (§ 3.3).

Algorithm 1: Comparison of Naive Search and MARS

Input : Candidate ranks R_{options}, target dataset D_f
Output : Optimal config $(r^*_{\text{ve}}, r^*_{\text{llm}})$
$BestPerformance \leftarrow -\infty$
$CandidatePairs \leftarrow \emptyset$
for $r_{llm} \in R_{options}$ **do**

```
# --- Naive Search: Iterate over all rank pair combinations;
full fine-tune to obtain accuracy. ---
for r_ve ∈ R_options do
    Perf ← FullFineTuning(r_ve, r_llm, D_f)
    if Perf > BestPerformance then
        BestPerformance ← Perf
        (r*_ve, r*_llm) ← (r_ve, r_llm)
# +++ MARS: No inner loop; prune the search space and select
the best VE rank for r_llm. +++
r'_ve ← FindBalancedRank(r_llm)                    // SL-C
Add (r'_ve, r_llm) to CandidatePairs
```

$(r^*_{\text{ve}}, r^*_{\text{llm}}) \leftarrow \text{SelectBest}(CandidatePairs)$ `// SL-P`
return $(r^*_{\text{ve}}, r^*_{\text{llm}})$

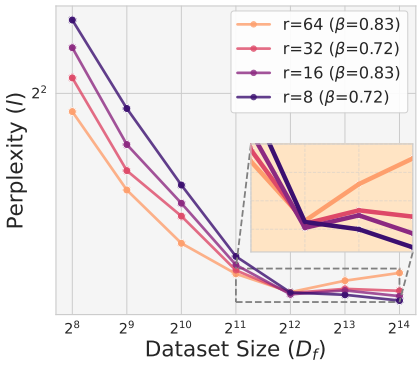
3.1 Dual Scaling Laws

Dual scaling laws are the predictive foundation that makes rank search feasible. In this section, we elaborate on how they are formulated and validated in the VLM fine-tuning setup.

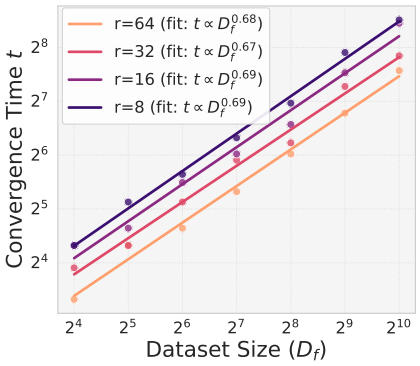
3.1.1 Experimental Setup

To empirically derive our scaling laws, we designed a controlled experimental setup to isolate and measure pure fine-tuning capability. Our experiments are based on the LLaVA-OneVision (OV) architecture [8], but we initialize our models "from scratch." This means we assemble publicly available vision encoder (SigLIP; [18]) and LLM (Qwen2; [15]) checkpoints with a pretrained projector, ensuring the model architecture is consistent while the initial parameters have no prior exposure to the downstream fine-tuning datasets. This controlled initialization allows us to focus solely on the scaling dynamics of the fine-tuning process itself. To study the impact of parameter disparity, we used two variants: LLaVA-OV-0.5B* (minimal VE-LLM parameter size gap) and LLaVA-OV-7B* (significant gap).

Our primary fine-tuning dataset was LLaVA-158K [14, 8], from which we sampled varying subset sizes to study scaling trends. We use validation perplexity as our primary metric to measure fundamental fine-tuning capability (transferability), as it directly aligns with the training objective.



(a) Scaling Law-P: Performance (perplexity) as a function of dataset size for different LLM ranks ($r_{\text{ve}} = 4$). The yellow line shows larger discrepancies between VE and LLM lead to reduced fine-tuning capability.



(b) Scaling Law-C: Convergence time as a function of dataset size for different LLM ranks ($r_{\text{ve}} = 16$). The near-parallelism of the fitted lines supports the separability of the rank and dataset size terms in Equation 2.

Figure 2: The proposed dual scaling laws from LLaVA-OV-0.5B* (VE and LLM have a similar parameter count without task-specific knowledge). This pattern holds consistently across different fixed-rank settings (e.g., $r_{\text{ve}} = 8$ and $r_{\text{ve}} = 32$).

Experimental setup details are in Appendix A. Convergence is defined by a standard early stopping criterion [19, 20]: the model is considered converged if validation performance fails to improve for a set number of steps (five in this paper).

3.1.2 Scaling Law-P for Performance

Formulation. Based on our empirical findings and previous scaling laws on LLM fine-tuning [12], we formulate Scaling Law-P to model the fine-tuning loss (or perplexity, $\hat{L}$) for a VLM. This law extends existing frameworks by incorporating separate LoRA ranks for the VE (r_{ve}) and the LLM (r_{llm}), alongside the dataset size (D_f):

$$\hat{L}(r_{\text{ve}}, r_{\text{llm}}, D_f) = A \cdot \frac{1}{(r_{\text{ve}})^{\alpha_m} \cdot (r_{\text{llm}})^{\alpha_l} \cdot D_f^{\beta}} + E \tag{1}$$

Here, α_m and α_l are scaling exponents for the VE and LLM ranks, reflecting their impact on loss reduction; β is the exponent for dataset size. A and E are fitted constants representing a scaling coefficient and an irreducible error, respectively. This law serves as the objective function in MARS.

Key Observation. Scaling Law-P is grounded in the foundational scaling laws [12], which were originally established in the context of unimodal (LLM) fine-tuning. While this law holds under its original conditions, our work identifies a critical counterexample: ***the law breaks down when there is a non-negligible imbalance in training dynamics across heterogeneous components.***

1) Performance is sensitive to the interplay between VE and LLM ranks. Unlike unimodal studies, where LoRA rank often has a modest effect, VLM fine-tuning performance strongly depends on the combination of ranks chosen for the VE and the LLM. An imbalance between these ranks can undermine training benefits, particularly on large datasets (Figure 2a), highlighting the need to consider both ranks jointly for effective fine-tuning.

2) Optimal LoRA rank pair involves a trade-off with dataset size. Performance is non-monotonic: small datasets benefit from higher VE ranks for richer feature extraction (see Appendix D), while large datasets favor moderate ranks to avoid overfitting. Thus, the optimal rank pair balances expressive feature learning with overfitting risk, depending on dataset size.

These findings highlight the need for an adaptive, modality-aware rank search. Additional results, including counterexamples and justifications (e.g., why a multiplicative rather than additive formulation), are provided in Appendix E.

3.1.3 Scaling Law-C for Convergence Time

Formulation. Modeling the relationship between LoRA rank, dataset size, and convergence time (i.e., the number of iterations required for convergence) is a relatively underexplored area. To address this, we propose Scaling Law-C to model the convergence time for each modality-specific module within a VLM. For a given module i, where i represents either the VE or the LLM, we define the convergence time t_i using the following general form:

$$t_i(r_i, D_f) = k_i \cdot (r_i)^{\gamma_i} \cdot D_f^{\delta_i} + E_i \tag{2}$$

Here, t_i is the predicted number of training steps to convergence for module i (e.g., $i \in \{\text{ve}, \text{llm}\}$), with LoRA rank r_i and dataset size D_f. The exponent $\gamma_i < 0$ captures the effect of rank on convergence time, while δ_i reflects the impact of dataset size. Constants k_i and E_i are fitted from data. This model predicts how convergence scales with rank and data, enabling MARS to balance t_{ve} and t_{llm} during pruning.

Key Observation. Our empirical results strongly support Scaling Law-C, showing consistent convergence time trends across varying dataset sizes and LoRA ranks.

1) Increasing dataset size *increases* convergence time. Larger fine-tuning datasets (D_f) require more iterations for the model to converge. This is intuitive, as the model needs additional steps to process and learn from the extra data. The near-linear trend, shown in Figure 2b, aligns with the $D_f^{\delta_i}$ term in Equation 2.

2) Increasing rank size *decreases* convergence time. Modules with higher LoRA ranks (r_i) converge in fewer iterations, as larger ranks provide greater capacity and flexibility for efficient adaptation. Consequently, these modules reach convergence faster than those with lower, constrained ranks. This inverse relationship aligns with the $r_i^{\gamma_i}$ term in Equation 2, with negative γ_i capturing the acceleration.

These two distinct scaling behaviors form the empirical basis for our convergence time predictions, which are integral to MARS's ability to identify convergence-balanced rank pairs. Additional *theoretical* and *empirical* evidence is provided in Appendix F.

3.2 The Guided Search Space Optimization

Once the dual scaling laws are calibrated, MARS performs an automated search for the optimal LoRA rank pair $(r^*_{\text{ve}},\ r^*_{\text{llm}})$ for a target dataset size. The search first prunes the vast space of possible rank combinations using our fitted Scaling Law-C. This strategy is computationally grounded in the ***monotonic smoothness*** and ***local continuity*** of the scaling landscape observed in our evidence (see Figure 2b & Appendix F). Because the training dynamics are smooth and order-preserving without abrupt spikes or discontinuities across the search space, MARS models Scaling Law-C as a continuous power-law function, enabling efficient pruning via interpolation between discrete anchor points.

The guiding principle is to identify rank pairs where the VE and LLM modules achieve aligned convergence times ($t_{\text{ve}} \approx t_{\text{llm}}$). By leveraging the overall smoothness of these scaling curves, we can solve this balance condition to express the ideal VE rank as a direct function of the LLM rank:

$$r_{\text{ve}} \approx \left(\frac{k_l \cdot (r_{\text{llm}})^{\gamma_l} \cdot D_f^{\delta_l} + E_{\text{llm}} - E_{\text{ve}}}{k_v \cdot D_f^{\delta_v}} \right)^{\frac{1}{\gamma_v}} \tag{3}$$

To generate a set of *convergence-aligned* candidates, MARS iterates through a list of possible LLM ranks, uses Equation 3 to calculate the corresponding ideal VE rank for each, and rounds to the closest integer based on the local continuity and smoothness of the scaling landscape. This procedure efficiently creates a pruned set of candidate pairs, $CandidatePairs$, in Algorithm 1. See Appendix G for the full derivation.

The final step is to select the pair with the highest performance potential. We use the fitted Scaling Law-P to predict the fine-tuning perplexity, $\hat{L}$, for each candidate pair, with Θ_{acc} denoting the fitted coefficients. The optimal rank pair is the one that minimizes this predicted perplexity:

$$(r_{\text{ve}}^*, r_{\text{llm}}^*) = \underset{(r_{\text{ve}}, r_{\text{llm}}) \in \texttt{CandidatePairs}}{\text{argmin}} \hat{L}(r_{\text{ve}}, r_{\text{llm}}, D_f; \Theta_{\text{acc}}) \tag{4}$$

The value of our dual scaling law framework lies in its synergistic two-phase approach. A search guided only by a performance predictor can yield *practically unstable* configurations due to the gradient interference caused by imbalanced convergence, while optimizing only for balance does not guarantee the highest accuracy. MARS resolves this trade-off by first using Scaling Law-C to prune the search space to a stable subset of candidates, and then employing Scaling Law-P to select the highest-performing solution from that set.

3.3 Calibration of Dual Scaling Laws

When fine-tuning begins, the MARS procedure starts with a calibration phase that empirically estimates the coefficients of our dual scaling laws. To minimize computational cost, we employ an efficient data collection strategy: for each representative LoRA rank pair (e.g., $r_{\text{ve}}, r_{\text{llm}} \in \{8, 16, 32, 64\}$), we initiate a single fine-tuning run and record performance and convergence metrics at multiple intermediate checkpoints corresponding to smaller effective dataset sizes (e.g., at 2^{10} and 2^{11} fine-tuning steps). This approach yields a rich dataset, $\mathcal{D}$, from a minimal number of training runs. Using this dataset, we determine the coefficients for both scaling laws by framing it as a numerical optimization problem. Following established practices [21, 12], we employ the L-BFGS-B algorithm to find the parameter set that minimizes the Huber loss between our model's predictions and the observed data. Crucially, this calibration stage utilizes the same fine-tuning settings (e.g., optimizer, batch size) as the full fine-tuning, ensuring the identified rank pair remains effective. The result is a reliable predictive model of performance and convergence dynamics, essential for the subsequent guided search.

4 Evaluation

4.1 Evaluation Setup

Evaluation Focus. To ensure a comprehensive assessment, we consider the overall phases of VLM fine-tuning in alignment with current community trends and established practices, which can be categorized into two types:

- **Generalist Fine-tuning.** Fine-tuning the model on a diverse set of general multimodal instructions to build a strong foundational generalist. This stage is often referred to as *pre-training* after the pre-trained VE and LLM are first connected, as these components have not previously been exposed to multimodal tasks.
- **Specialist Fine-tuning.** Fine-tuning a pre-trained generalist model on domain-specific datasets to produce a specialized expert. This stage is generally referred to as *post-training*. Given that the model possesses strong general knowledge, effective fine-tuning must balance acquiring task-specific knowledge while preserving pre-trained representations.

Evaluation Benchmarks and Datasets. To evaluate the effectiveness of MARS, we utilize two distinct benchmarks designed to measure different types of fine-tuning performance. For generalist fine-tuning, we use LLaVA Bench[14], which consists of 15k validation samples from the LLaVA-Instruct dataset and is composed of complex reasoning tasks. For this benchmark, we use validation perplexity as the primary metric. Perplexity, computed directly from cross-entropy loss, serves as a proxy for the model's training objective (minimizing loss) and provides a direct measure of its

Table 1: Comparison with fixed-rank tuning across different learning rates. Best in bold, second-best underlined. OV: LLaVA-OV; Qwen: Qwen2.5-VL.

Model	Bench.	LoRA (lr_{ve}, lr_{llm})			MARS
		(★, 1e-5)	(★, 1e-6)	(★, 1e-7)	
OV-0.5B [8]	LLaVA (↓)	<u>2.7336</u>	2.771	2.8472	**2.7188**
	SciQA (↑)	<u>71.06</u>	61.88	59.28	**72.85**
OV-7B [8]	LLaVA (↓)	2.2317	<u>2.295</u>	2.4346	**2.1875**
	SciQA (↑)	<u>72.26</u>	69.86	67.27	**74.25**
Qwen-3B [9]	LLaVA (↓)	<u>3.6156</u>	3.7415	4.1557	**3.5925**
	SciQA (↑)	<u>78.04</u>	76.45	76.25	**79.24**
Qwen-7B [9]	LLaVA (↓)	<u>3.5032</u>	3.5908	3.8716	**3.3879**
	SciQA (↑)	**79.84**	76.25	74.25	<u>79.64</u>

Table 2: Comparison with adaptive rank search baselines. Variance is low across three seeds (std < 0.3%); See Appendix K for stability details; AdaL.: AdaLoRA; GeoL: GeoLoRA.

Model	AdaL.	GeoL.	Full-rank Tuning	LoRA (r_{ve}, r_{llm})			MARS
				(★, 8)	(★, 16)	(★, 32)	
LLaVA Bench (perplexity ↓)							
OV-0.5B	2.8973	2.8801	<u>2.7209</u>	2.7582	2.7336	2.7331	**2.7188**
OV-7B	2.5189	2.4888	2.2693	2.3181	<u>2.2317</u>	2.4420	**2.1875**
Qwen-3B	3.6679	3.6676	3.6528	<u>3.5912</u>	3.6156	3.7811	**3.5825**
Qwen-7B	3.6394	3.6378	3.5917	<u>3.5011</u>	3.5032	3.6098	**3.3616**
ScienceQA bench (accuracy (%) ↑)							
OV-0.5B	62.28	63.52	69.66	70.46	<u>71.06</u>	69.86	**72.85**
OV-7B	66.27	67.81	70.46	70.26	72.26	<u>73.65</u>	**74.25**
Qwen-3B	70.06	70.46	75.25	<u>78.24</u>	78.04	78.04	**79.24**
Qwen-7B	73.85	72.85	77.45	**79.84**	79.84	78.24	<u>79.64</u>

fine-tuning capability by indicating how well it can specialize on a downstream dataset. For specialist fine-tuning, we use the ScienceQA benchmark [22]. This benchmark contains multimodal multiple-choice questions across diverse scientific subjects. For our experiments, we use a 16k training split and evaluate on 500 test samples. Following standard convention, we report accuracy. We further extend MARS to a broader set of benchmarks to evaluate its generality: MME [23], MMStar [24], POPE [25], TextCaps [26], AI2D [27], GQA [28]. Benchmark details are provided in Appendix I.

Baselines. To demonstrate the effectiveness of MARS as an adaptive rank search algorithm for convergence coordination, we compare it against several key baselines. The first is a differential learning rate approach, representing the standard heuristic of manually tuning separate learning rates for the VE and LLM. Second, since we argue that using differential ranks between modalities is important, we compare against a set of fixed differential rank pairs (i.e., uniform but different across modalities). Third, we benchmark against AdaLoRA [29] and GeoLoRA [7], two representative adaptive rank allocation methods designed for unimodal models (see Appendix B for details). Finally, we compare against two representative search methods: Random Search and Bayesian Optimization (TPE [30]), where a probabilistic model guides the search. By comparing against these methods, we demonstrate that MARS effectively identifies the optimal rank combination.

Implementation Details. Our experiments are implemented using the Cornstarch MLLM training framework [31]. All models were fine-tuned using the Adam optimizer with a cosine learning rate scheduler, and all experiments were conducted on a single NVIDIA GH200 GPU. Further details on hyperparameters, LoRA settings, and the MARS calibration setup are provided in the Appendix C.

4.2 Main Results

We evaluate MARS across multiple VLM architectures and model scales, comparing it against several strong baselines across various benchmarks. As shown in Tables 1 and 2 and Figure 3, the fine-tuned models with MARS show effectiveness and generality compared to our baselines.

Comparison with Differential Learning Rates. Table 1 compares MARS against the common heuristic of using fixed LoRA ranks while tuning differential learning rates. On the LLaVA Bench, MARS consistently achieves a lower (better) perplexity across all tested models. For instance, on the LLaVA-OV-7B model, MARS achieves a perplexity of 2.1875, outperforming the best-performing differential LR setting (2.295). This trend holds on the more specialized ScienceQA benchmark, where MARS improves the accuracy of the LLaVA-OV-7B model from 72.26% to 74.25%. These results demonstrate that our rank-centric search is a more effective strategy for harmonizing module dynamics than just tuning learning rates. See Appendix H for results on the from-scratch models.

Comparison with Other Baselines. In Table 2, we benchmark MARS against a wider range of fine-tuning strategies. MARS consistently surpasses Full-rank Tuning, highlighting the regularization benefits of our adaptive PEFT approach. More importantly, it outperforms various fixed differential LoRA rank pairs, proving the value of our guided search over simply selecting arbitrary rank

Table 3: Equal-budget comparison on Qwen2.5-VL-3B. AdaLoRA (separate): separate per module; AdaLoRA (unified): treats VLM as unimodal.

		ScienceQA (Acc ↑)		LLaVABench (PPL ↓)	
Method	**Total Cost**	**Params (M)**	**Result**	**Params (M)**	**Result**
Random Search	5 full runs	26–27	77.64%	22–24	3.5928
Bayesian (TPE)	6 full runs	≤27.19	78.64%	≤23.62	3.6882
AdaLoRA (separate)	1 full run	29.97	71.26%	22.79	3.6493
AdaLoRA (unified)	1 full run	25.86	70.06%	25.86	3.6679
MARS	Calib+1 full run	27.19	79.24%	23.62	3.5825

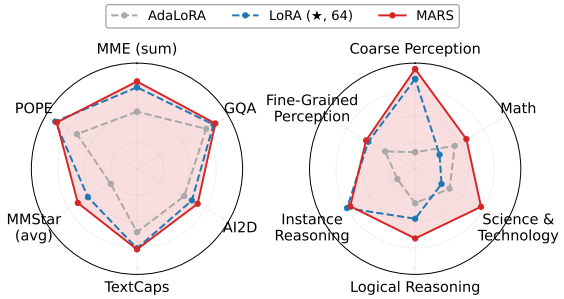


Figure 3: Generalist Capabilities: **(L)** Comparison across diverse multimodal benchmarks. **(R)** Fine-grained breakdown on MMStar. Details in Appendix I.2.

Table 4: Comparison against the exhaustive grid search optimum. ($\mathbf{r_{ve}}$, $\mathbf{r_{llm}}$).

Benchmark	**Model**	**Exhaustive Top-1**	**MARS**
LLaVAB.	LLaVA-OV-7B	(10, 16)	(10, 16)
LLaVAB.	Qwen2.5VL-3B	(10, 8)	(10, 8)
ScienceQA	LLaVA-OV-7B	(24, 32)	(22, 32)
ScienceQA	Qwen2.5VL-3B	(15, 8)	(15, 8)

Table 5: Correlation between Convergence Gap ($|T_{ve} - T_{llm}|$) and Perplexity.

Data Size	**Pearson's r**	**P-value**
0.5K	0.90	0.0020
1K	0.95	0.0003
2K	0.87	0.0052
4K	0.98	0.0000

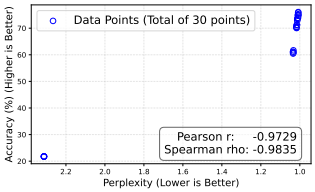


Figure 4: Correlation between Perplexity and Task Accuracy.

combinations. The most critical comparison is with AdaLoRA and GeoLoRA, two representative adaptive-rank methods designed for unimodal models. We evaluate them under the equal-budget setting as MARS-selected ranks. Due to their algorithmic characteristics (e.g., AdaLoRA starts with larger ranks and progressively prunes, while GeoLoRA dynamically expands and shrinks ranks), smaller effective ranks are required to match the same budget. Moreover, their layer-wise importance estimation is not optimal in VLM settings, leading both methods to significantly underperform.

Comparison with Equal-Budget Search Baselines. We compare MARS against equal-budget representative search methods under the equal trainable-parameter budget. As shown in Table 3, MARS outperforms both Random Search (best of 5 trials) and Bayesian Optimization (TPE with 6 trials) with lower search overhead. For AdaLoRA, we find that the per-module variant outperforms unified rank assignment.

Comparison with Exhaustive Grid Search. To verify that the convergence-alignment pruning does not exclude high-performing candidates, we compare MARS against a 64-point exhaustive grid search ($r_{\text{llm}} \in \{4, 8, 16, 32\} \times r_{\text{ve}} \in \{2, 4, \ldots, 32\}$). Table 4 shows that MARS's scaling-law-based alignment effectively identifies or closely approximates optimal rank pairs.

Generality with Broader Benchmark Coverage. To demonstrate the generality of MARS, we evaluate the model on a diverse set of benchmarks, including both generic and domain-specific tasks. Given the limited data diversity of the initial LLaVA dataset [14], we use the updated version, LLaVA-1.5-665K [32], to train LLaVA-OV-7B*. This experiment is generalist fine-tuning with a from-scratch model, isolating pretrained knowledge to assess MARS's generality. As shown in Figure 3, MARS mostly outperforms the strongest LoRA baselines (rank combinations $r_{\text{ve}} \in \{32, 48, 64\}$, $r_{\text{llm}} = 64$) and adaptive methods such as AdaLoRA (target rank= 64) across the benchmarks. For this experiment, we set R_{options} (i.e., candidate ranks for the LLM) to $\{32, 48, 64\}$ to better absorb the knowledge from the large, diverse dataset.

4.3 Computational Efficiency

Beyond its advantages in fine-tuning effectiveness, MARS also achieves a substantial improvement in computational efficiency. As shown in Figure 5, a naive exhaustive search for the optimal rank pair requires extensive compute, often exceeding 100 GPU hours. In contrast, MARS consists of a

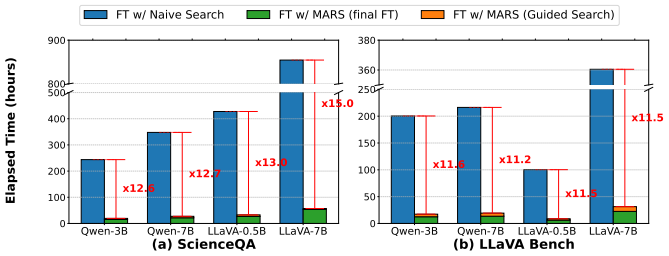


Figure 5: Time comparison with Naive Search. Naive Search explores the grid $\{4, 8, 16, 32\} \times \{4, 8, 16, 32\}$ to find the best rank pair.

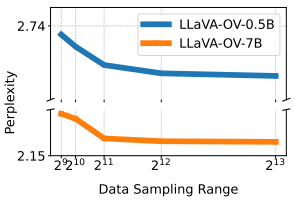


Figure 6: Impact of Data Sampling Range.

lightweight calibration phase followed by a single fine-tuning run. This reduces the total elapsed time by more than 11.5x on average across different models and tasks, with most of the time spent on the actual fine-tuning (which is unavoidable). This substantial reduction in search cost demonstrates that MARS is not only more effective, but also more practical and scalable for VLM fine-tuning.

4.4 Impact of Convergence Alignment on Performance

To validate the hypothesis that aligning the convergence times of the Vision Encoder (VE) and the Large Language Model (LLM) leads to better final performance, we performed a two-part statistical analysis linking convergence synchronization, perplexity, and final accuracy.

Perplexity as a Performance Proxy. First, we establish the reliability of perplexity as a robust, task-agnostic proxy for fine-tuning quality. While prior studies [8, 13] already suggest a link between lower perplexity and improved downstream performance, we empirically validate this relationship in our fine-tuning setup using the ScienceQA dataset. Figure 4 shows a strong negative correlation between perplexity and task accuracy (Pearson's $r = -0.97$, $p < .001$), confirming that optimizing the perplexity proxy effectively maximizes task accuracy. This relationship generalizes across six diverse benchmarks (Pearson r from -0.72 to -0.97; Appendix M).

Linking Convergence Gap to Performance. Building on this proxy, we directly address the relationship between the *Convergence Gap* ($|T_{ve} - T_{llm}|$) and model perplexity. We conducted a statistical correlation analysis across multiple fine-tuning regimes with dataset sizes ranging from $D = 2^9$ to 2^{13}. As in Table 5, we observed statistically significant positive correlations ($r > 0.86$, $p < 0.01$) across all scales. These results confirm that minimizing the convergence gap is a statistically dominant predictor of reduced perplexity and, by extension, superior model performance.

4.5 Functional-Form Transferability.

Table 6 tests whether per-model calibration is necessary and sufficient. Reusing coefficients from a different model leads to a severe mismatch, whereas calibrating on the target model achieves a strong fit. These results suggest that the functional form of the scaling law is transferable across regimes, but the coefficients must be recalibrated for each model, as assumed in MARS (see Appendix L for further analysis).

Table 6: Prediction reliability of Scaling Law-P on pre-trained LLaVA-OV. R^2: fit quality; MAPE: mean abs. percentage error.

Source of the coefficients	R^2	MAPE
From-scratch LLaVA-OV	-5.309	57.23%
Per-model calibration	0.994	0.96%

4.6 Ablation: Impact of Data Sampling Range.

We conducted an ablation study to determine the optimal range of data required for the lightweight calibration phase of MARS. We ran the MARS search procedure multiple times, each time using a different maximum data sampling range for calibration, and then evaluated the final test perplexity achieved by the resulting optimal rank pair. As shown in Figure 6, we observe that increasing the data sampling range up to 2^{11} leads to a clear improvement in the final perplexity, as it allows for a more accurate fitting of the scaling laws. However, performance gains diminish significantly beyond that point, validating our choice of using a data sampling range up to 2^{11} for MARS calibration, as it strikes an effective balance between accuracy and computational efficiency.

5 Discussion

Scalability with Modality Expansion. The current scope of MARS focuses on VLMs, while tri-modal settings (vision–audio–language [33]) are left for future work. However, we believe MARS can help address the combinatorial explosion of multimodal hyperparameter tuning. In contrast to the exponential growth ($O(C^N)$) of naive grid search, MARS transforms the search into $(N-1)$ independent 1D equations by anchoring the LLM rank as a reference. Furthermore, to minimize the overhead of the calibration phase, we employ a Simultaneous Multi-Rank Adapter Tuning strategy (see Appendix J). By leveraging a frozen backbone with multiple parallel adapters, we reduce the calibration cost to near-constant time $O(1)$, ensuring that MARS remains computationally efficient.

Heterogeneous Gains. MARS's gains vary with the degree of convergence imbalance. When the optimal rank pair coincides with a standard configuration (e.g., $r_{\text{ve}} = r_{\text{llm}} = 8$ for Qwen2.5-VL-7B on ScienceQA in Table 2), improvements over exhaustive search are marginal. In such cases, MARS's main benefit is efficiency, identifying optimal configurations without $>11\times$ the cost of grid search.

6 Conclusion

We introduce MARS to address imbalanced convergence in VLM fine-tuning. By leveraging dual scaling laws to predict convergence and prune the LoRA rank search space, MARS identifies optimal rank pairs with balanced dynamics and outperforms common heuristic-based strategies, highlighting the importance of convergence coordination for effective multimodal adaptation. As future work, we aim to systematically characterize fine-grained relationships between domain sensitivities across tasks.

Acknowledgments and Disclosure of Funding

This work was supported by NSF under Grant CNS-2321532. I am deeply grateful to Xueshen Liu, Jiarui Li, Joseph Brewington, Hui Shen, Una Chan, and Boyuan Zheng for their valuable discussions, insightful feedback, and generous support throughout this work.

References

[1] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.

[2] Haotian Zhang, Mingfei Gao, Zhe Gan, Philipp Dufter, Nina Wenzel, Forrest Huang, Dhruti Shah, Xianzhi Du, Bowen Zhang, Yanghao Li, et al. Mm1. 5: Methods, analysis & insights from multimodal llm fine-tuning. *arXiv preprint arXiv:2409.20566*, 2024.

[3] Simon Zhai, Hao Bai, Zipeng Lin, Jiayi Pan, Peter Tong, Yifei Zhou, Alane Suhr, Saining Xie, Yann LeCun, Yi Ma, et al. Fine-tuning large vision-language models as decision-making agents via reinforcement learning. *Advances in neural information processing systems*, 37:110935–110971, 2024.

[4] Maxime Zanella and Ismail Ben Ayed. Low-rank few-shot adaptation of vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 1593–1603, 2024.

[5] Xiaokang Chen, Zhiyu Wu, Xingchao Liu, Zizheng Pan, Wen Liu, Zhenda Xie, Xingkai Yu, and Chong Ruan. Janus-pro: Unified multimodal understanding and generation with data and model scaling. *arXiv preprint arXiv:2501.17811*, 2025.

[6] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022.

[7] Steffen Schotthöfer, Emanuele Zangrando, Gianluca Ceruti, Francesco Tudisco, and Jonas Kusch. Geolora: Geometric integration for parameter efficient fine-tuning. *arXiv preprint arXiv:2410.18720*, 2024.

[8] Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, et al. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*, 2024.

[9] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.

[10] Dan Biderman, Jacob Portes, Jose Javier Gonzalez Ortiz, Mansheej Paul, Philip Greengard, Connor Jennings, Daniel King, Sam Havens, Vitaliy Chiley, Jonathan Frankle, et al. Lora learns less and forgets less. *arXiv preprint arXiv:2405.09673*, 2024.

[11] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

[12] Biao Zhang, Zhongtao Liu, Colin Cherry, and Orhan Firat. When scaling meets llm finetuning: The effect of data, model and finetuning method. *arXiv preprint arXiv:2402.17193*, 2024.

[13] Mustafa Shukor, Enrico Fini, Victor Guilherme Turrisi da Costa, Matthieu Cord, Joshua Susskind, and Alaaeldin El-Nouby. Scaling laws for native multimodal models. *arXiv preprint arXiv:2504.07951*, 2025.

[14] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916, 2023.

[15] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024.

[16] Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 2022.

[17] Shiqi He, Insu Jang, and Mosharaf Chowdhury. Mordal: Automated pretrained model selection for vision language models, 2025.

[18] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 11975–11986, 2023.

[19] Lutz Prechelt. Early stopping-but when? In *Neural Networks: Tricks of the trade*, pages 55–69. Springer, 2002.

[20] PyTorchIgnite. Pytorch ignite: Earlystopping.

[21] Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 2022.

[22] Pan Lu, Swaroop Mishra, Tony Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering. In *The 36th Conference on Neural Information Processing Systems (NeurIPS)*, 2022.

[23] Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, et al. Mme: A comprehensive evaluation benchmark for multimodal large language models. *arXiv preprint arXiv:2306.13394*, 2023.

[24] Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Jiaqi Wang, Yu Qiao, Dahua Lin, et al. Are we on the right way for evaluating large vision-language models? *arXiv preprint arXiv:2403.20330*, 2024.

[25] Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne Xin Zhao, and Ji-Rong Wen. Evaluating object hallucination in large vision-language models. *arXiv preprint arXiv:2305.10355*, 2023.

[26] Oleksii Sidorov, Ronghang Hu, Marcus Rohrbach, and Amanpreet Singh. Textcaps: a dataset for image captioningwith reading comprehension. 2020.

[27] Aniruddha Kembhavi, Mike Salvato, Eric Kolve, Minjoon Seo, Hannaneh Hajishirzi, and Ali Farhadi. A diagram is worth a dozen images. In *European conference on computer vision*, pages 235–251. Springer, 2016.

[28] Drew A Hudson and Christopher D Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering. *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[29] 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. *arXiv preprint arXiv:2303.10512*, 2023.

[30] Shuhei Watanabe. Tree-structured parzen estimator: Understanding its algorithm components and their roles for better empirical performance. *arXiv preprint arXiv:2304.11127*, 2023.

[31] Insu Jang, Runyu Lu, Nikhil Bansal, Ang Chen, and Mosharaf Chowdhury. Cornstarch: Distributed multimodal training must be multimodality-aware. *arXiv preprint arXiv:2503.11367*, 2025.

[32] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 26296–26306, 2024.

[33] Qwen Team. Qwen3. 5-omni technical report. *arXiv preprint arXiv:2604.15804*, 2026.

[34] Qwen Team. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*, 2, 2024.

[35] Wenjun Ke, Jiahao Wang, Peng Wang, Jiajun Liu, Dong Nie, Guozheng Li, and Yining Li. Unveiling lora intrinsic ranks via salience analysis. *Advances in Neural Information Processing Systems*, 37:131575–131595, 2024.

[36] Zhen Dong, Zhewei Yao, Amir Gholami, Michael W Mahoney, and Kurt Keutzer. Hawq: Hessian aware quantization of neural networks with mixed-precision. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 293–302, 2019.

[37] Donghyun Lee, Minkyoung Cho, Seungwon Lee, Joonho Song, and Changkyu Choi. A novel sensitivity metric for mixed-precision quantization with synthetic data generation. In *2021 IEEE International Conference on Image Processing (ICIP)*, pages 1294–1298. IEEE, 2021.

[38] 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.

[39] Abdessalam Ed-dib, Zhanibek Datbayev, and Amine Mohamed Aboussalah. Gelora: Geometric adaptive ranks for efficient lora fine-tuning. *arXiv preprint arXiv:2412.09250*, 2024.

[40] Zeyuan Allen-Zhu and Yuanzhi Li. Physics of language models: Part 3.3, knowledge capacity scaling laws, 2024. *URL https://arxiv. org/abs/2404.05405*, page 2, 2023.

[41] Siqiao Mu and Diego Klabjan. On the convergence rate of lora gradient descent. *arXiv preprint arXiv:2512.18248*, 2025.

[42] Reece Shuttleworth, Jacob Andreas, Antonio Torralba, and Pratyusha Sharma. Lora vs full fine-tuning: An illusion of equivalence. *arXiv preprint arXiv:2410.21228*, 2024.

[43] Kaichen Zhang, Bo Li, Peiyuan Zhang, Fanyi Pu, Joshua Adrian Cahyono, Kairui Hu, Shuai Liu, Yuanhan Zhang, Jingkang Yang, Chunyuan Li, and Ziwei Liu. Lmms-eval: Reality check on the evaluation of large multimodal models, 2024.

Appendix

A Additional Experimental Details for Section 3.1

A.1 Model Setup

Our target model architecture is LLaVA-OneVision (LLaVA-OV; [8]). For these scaling law experiments, all models were initialized "from scratch," a methodology we denote with an asterisk (*). This process involves downloading publicly available, pre-trained vision encoder (VE) and Large Language Model (LLM) checkpoints and connecting them with an MLP projector, without any subsequent instruction tuning. Specifically, we used SigLIP [18] for the VE and Qwen2 [34] for the LLM and attached a 2-layer MLP projector to replicate the original structure.

This approach is intentionally designed to ensure that the model's initial state is *domain-knowledge-free* with respect to multimodal instruction data. By doing so, we can isolate and measure the effects of pure fine-tuning capability as a function of data and model scale. We investigated two architectural variants to analyze the impact of the parameter gap between the VE and LLM components.

- **LLaVA-OV-0.5B*:** Features a *minimal* parameter gap (VE $\sim$400M, LLM $\sim$500M).
- **LLaVA-OV-7B*:** Features a *significant* parameter gap (VE $\sim$400M, LLM $\sim$7B).

A.2 Dataset Setup

Our primary dataset is the LLaVA dataset [14, 8], which, to our knowledge, is one of the largest of its kind available for research. Each sample is centered on images of nature or everyday life. The dataset comprises a mixture of open-ended task types, including basic image captioning, detailed image captioning, and multi-turn conversational question answering.

To study scaling trends, we constructed training subsets from this dataset by varying their size according to a power-of-two progression (i.e., 2^n samples, for n from 3 to 15), while maintaining a consistent distribution of the above task types across all subsets. For validation, we used a fixed subset of 500 samples (non-overlapping with training data) to enable frequent, low-cost perplexity checks during intermediate training cycles.

B Additional Related Work

Adaptive LoRA for Unimodal Models. To move beyond static configurations, adaptive LoRA methods like AdaLoRA [29], SalientLoRA [35], and GeoLoRA [7] were developed. However, these methods are designed for unimodal models and face fundamental limitations in multimodal setups. In the unimodal context, it is often observed that allocating a higher rank to more salient layers correlates with improved transferability and performance [10], paralleling model compression techniques that allocate more resources to higher-importance components [36, 37]. In contrast, we found that for VLMs, simply assigning higher ranks does not invariably lead to better performance. This is due to the complex and crucial inter-modality dependencies, where the optimal configuration for one modality is influenced by the state of the other.

C MARS Implementation Details

C.1 General Training Hyperparameters

Unless otherwise specified, all fine-tuning experiments were conducted with the following hyperparameters. We used the Adam optimizer with its default parameter settings and a batch size of 8. The learning rate was initialized to 1×10^{-5} and followed a cosine decay schedule with a warmup phase corresponding to 10% of the total training steps. These settings were applied consistently across all experiments, with the exception of the peak learning rate in the baseline tests for differential learning rates. Following a prior paper [14], we trained for 3 epochs on the LLaVA task and 12 epochs on the ScienceQA task, reporting the best performance.

C.2 LoRA Configuration

For all experiments utilizing LoRA, we applied LoRA to all linear layers across the entire model, excluding the output heads. The LoRA scaling hyperparameter, α, was consistently set to twice the rank size (i.e., $\alpha = 2 \times r$) [38]. The initial calibration phase for MARS was performed using a representative set of LoRA ranks, $r \in \{8, 16, 32, 64\}$, for both the VE and LLM modules. We treat the projector as part of the vision encoder because it processes the same input stream for the same semantic purpose, so its adapter has the rank size of r_{ve}.

C.3 MARS Edge Case Handling

We consider rare failure cases of the MARS rank search mechanism. Specifically, in very rare cases (0.72% of all runs), the computed rank becomes complex-valued. This occurs when E_{ve} is excessively large due to the inherent entropy of the data distribution. In such cases, we fall back to a default LoRA configuration by setting $r_{\text{ve}} = r_{\text{llm}}$.

D Optimal VE Rank Size in the Small-Dataset Regime

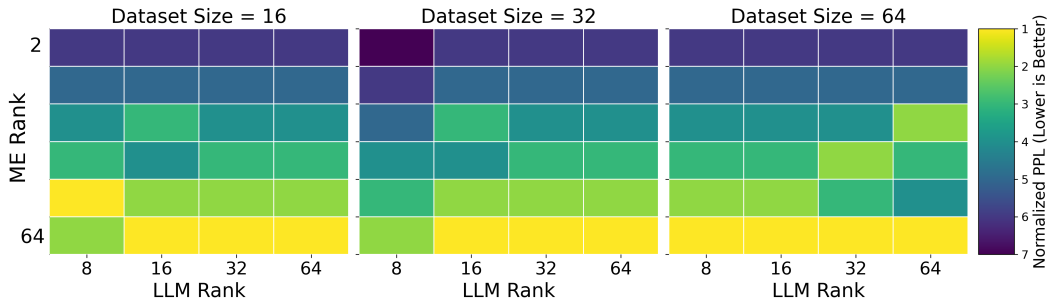


Figure 7: Optimal VE rank size in the small-dataset regime. Lower values (lighter colors) indicate better performance (i.e., lower perplexity).

The optimal VE rank is strongly dependent on the dataset size, revealing a critical trade-off between rapid adaptation and overfitting. As illustrated in the heatmaps (Figure 7), small-dataset regimes clearly prefer higher VE ranks. This is because the VE initially faces a large domain gap, and a higher rank provides the necessary adaptation capacity to quickly align with the LLM. From a capacity perspective, this aggressive approach allows the VE to absorb all available knowledge from limited samples, preventing it from becoming an early learning bottleneck.

Conversely, in mid-to-large dataset regimes, a different pattern emerges: the preference shifts to smaller or more moderate VE ranks. The primary goal becomes reducing overfitting risk and ensuring the efficient utilization of model capacity. With a large and diverse set of examples, extreme adaptation capacity is no longer required. Instead of attempting to absorb all available knowledge, a moderate rank is sufficient to extract essential visual information without over-specializing on training examples. This provides the LLM with stable, moderately adapted inputs, which promotes balanced convergence, prevents overfitting, and allows the model to fully leverage the LLM's capacity for optimal modality harmonization.

E Further Validation on Scaling Law-P

Formulation. We formulate Scaling Law-P to model the fine-tuning loss (or perplexity, $\hat{L}$) for a VLM. This law extends existing frameworks by incorporating separate LoRA ranks for the vision encoder (r_{ve}) and the LLM (r_{llm}), alongside the dataset size (D_f):

$$\hat{L}(r_{\text{ve}}, r_{\text{llm}}, D_f) = A \cdot \frac{1}{(r_{\text{ve}})^{\alpha_m} \cdot (r_{\text{llm}})^{\alpha_l} \cdot D_f^{\beta}} + E \tag{5}$$

Here, α_m and α_l are scaling exponents for the VE and LLM ranks, reflecting their impact on loss reduction, while β is the exponent for dataset size. A and E are fitted constants representing a scaling coefficient and an irreducible error, respectively.

Scaling Law-P is grounded in the foundational scaling laws proposed by prior work [12], which were originally established in the context of unimodal LLM fine-tuning. Consequently, their scaling formulation resembles Equation 5, but treats r_{ve} and r_{llm} collectively as a single rank parameter r. **While this law holds under its original conditions, our work identifies a critical counterexample**: the law breaks down when there is a non-negligible imbalance in training dynamics across heterogeneous components (e.g., Vision Encoder *vs.* LLM).

Why Multiplicative (i.e., $r_{\text{ve}} \cdot r_{\text{llm}}$)?. We model the interaction as multiplicative because the VE and the LLM form a serial (not parallel) information-processing chain. A capacity bottleneck in the upstream encoder (r_{ve}) imposes an irreversible loss of information that strictly bounds the maximum performance achievable by the downstream LLM, regardless of the LLM's capacity [39]. Also, following [12], we compare the fitting error and find that a multiplicative formulation provides a better fit than an additive one.

E.1 Empirical Evidence as a Counterexample to the Original Scaling Laws

Figure 8 provides further empirical evidence of Scaling Law-P in pre-trained models.

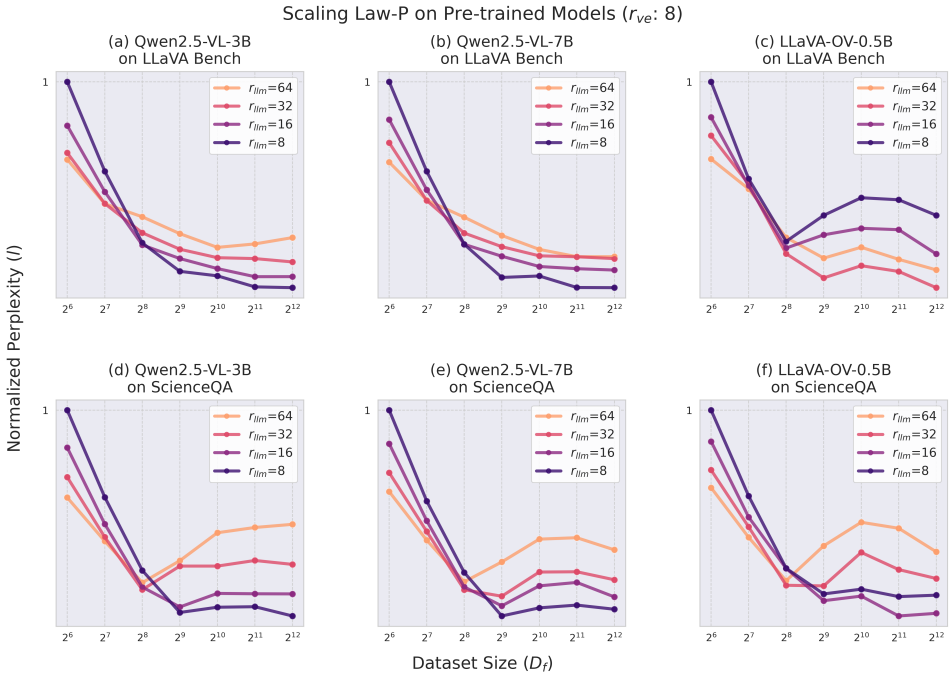


Figure 8: Validating Scaling Law-P on Pre-trained Models

F Further Validation on Scaling Law-C: Theoretical and Empirical Generality

We propose Scaling Law-C to estimate the computational cost of convergence, t. We model t as a tension between the **volume of information** to be absorbed (D_f) and the **condition number** of the optimization landscape (r).

Formulation. Modeling the relationship between LoRA rank, dataset size, and the number of iterations required for convergence is a relatively underexplored area. To address this, we propose Scaling Law-C to model the convergence time for each modality-specific module within a VLM. For

a given module i, where i can represent a vision encoder (VE) or the LLM, we define the number of training steps to convergence, t_i, with the following general form:

$$t_i(r_i, D_f) = k_i \cdot (r_i)^{\gamma_i} \cdot D_f^{\delta_i} + E_i \tag{6}$$

In this formulation, t_i is the predicted training steps to convergence for module i (e.g., $i \in \{\text{ve}, \text{llm}\}$). The LoRA rank for this module is r_i, and D_f is the fine-tuning dataset size. The exponent γ_i links the LoRA rank to convergence time; based on our observations that higher ranks reduce the number of steps, we fit this as a negative value. The exponent δ_i quantifies the impact of dataset size on convergence steps. Finally, k_i and E_i are constants fitted from experimental data for each module. This general formulation allows us to predict how the convergence time of any modality scales with its allocated LoRA rank and the amount of training data.

F.1 Theoretical Justification

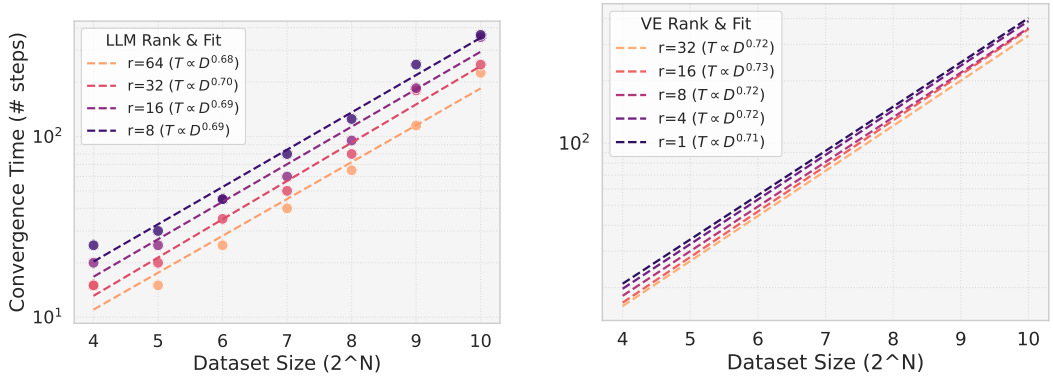
1. **Information Absorption ($t \propto D_f^{\delta}$):** Convergence time scales positively with dataset size ($\delta > 0$). We ground this in *Knowledge Capacity Scaling Laws* [40], which establish that neural networks are bounded by a storage limit of approximately 2 bits of knowledge per parameter. As the dataset size D_f increases, the total information content grows, pushing the fixed-rank adapter toward capacity saturation. To compress this increased entropy into the limited parameter budget, the optimizer requires a longer trajectory to resolve conflicting gradients and reach a generalization minimum, leading to the observed power-law increase in steps.
2. **Geometric Conditioning ($t \propto r^{-\gamma}$):** Convergence time scales inversely with rank ($\gamma > 0$). While [41] suggests that the theoretical worst-case convergence rate is rank-independent under ideal assumptions, our focus is more on the premise that rank fundamentally alters the *geometry* of the loss landscape in practice.
    - *Stationary Point vs. Performance Threshold:* Previous analysis defines convergence as reaching *any* stationary point; under this definition, the convergence rate is rank-independent [41]. However, Scaling Law-C models the time to reach a specific *performance threshold*. Higher ranks are necessary to bridge the *intrinsic dimension* gap, allowing the model to traverse toward the global optimum rather than stalling in rank-deficient local minima [39].
    - *Spectral Stability:* [42] provides spectral evidence that low-rank constraints fundamentally alter the stability of weight updates. They find that low-rank adapters exhibit lower *effective rank* (information density) compared to full fine-tuning. Increasing r increases the effective rank of the update matrix, improving *spectral stability* and ensuring smoother optimization dynamics that approximate full-rank behavior.
    - *Geometric Conditioning:* Furthermore, higher ranks mitigate the formation of *Intruder Dimensions*, which are high-magnitude singular vectors orthogonal to the pre-trained weights that appear in constrained low-rank regimes [42]. Low-rank models rely on these spectral anomalies, resulting in optimization instability and forgetting. Increasing the rank expands the optimization subspace, eliminating these artifacts and improving the condition number of the landscape. This accelerates the effective velocity of convergence ($t \downarrow$) by allowing the update trajectory to align with the smoother, pre-trained spectral manifold.

Theoretically, this parallelism indicates that the geometric conditioning induced by the rank ($r^{-\gamma}$) acts as an independent multiplicative preconditioner to the information complexity of the dataset (D^{δ}). This supports our *Subspace Optimization* hypothesis by decomposing the convergence time into two orthogonal factors.

F.2 Additional Empirical Evidence

To demonstrate that this behavior reflects a fundamental property of low-rank adaptation, we investigate the Scaling Law-C pattern on the LLaVA-OneVision-0.5B and Qwen2.5-VL families (3B and 7B) using the ScienceQA dataset (Figures 9 and 10). Consistent with the results observed in

Figure 2b, the Qwen2.5-VL models exhibit a near-linear relationship on the log–log scale, validating the power-law form $t \propto D^{\delta}$. Moreover, the fitted lines corresponding to different LoRA ranks remain nearly parallel across all architectures, providing strong empirical evidence for the separability of the rank and dataset size terms in the proposed scaling law (Equation 6).



(a) LLM (LLaVA-OV-0.5B) (b) VE (LLaVA-OV-0.5B)

Figure 9: Scaling Law-C Validation: LLaVA-OV-0.5B Models for the ScienceQA Dataset

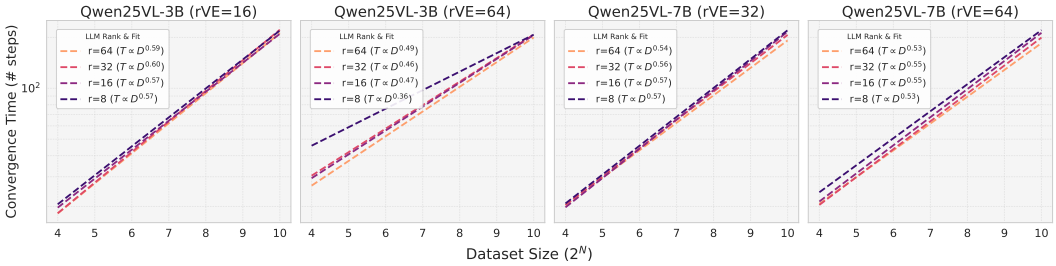


Figure 10: Scaling Law-C Validation: Qwen2.5-VL Models for the ScienceQA Dataset

G Derivation of Equation 3

The objective of MARS is to minimize the gradient conflict caused by asynchronous convergence. By equating the convergence times derived from Scaling Law-C ($t_{ve} \approx t_{llm}$), we analytically solve for the optimal rank ratio. This ensures that both the visual perception and reasoning modules traverse their optimization manifolds at compatible velocities, preventing the fast module from overfitting while the slow module is still acquiring features.

This section provides a step-by-step derivation of Equation 3, which estimates the ideal VE-to-LLM parameter ratio (r_{ve}). The ideal ratio is defined as the value that balances the expected training time of the vision encoder (t_{ve}) and the LLM (t_{llm}) for a given target fine-tuning dataset size.

The derivation starts with our two empirically fitted scaling law models for training time:

$$t_{\text{ve}} = k_v \cdot (r_{\text{ve}})^{\gamma_v} \cdot D_f^{\delta_v} + E_{\text{ve}} \tag{7}$$

$$t_{\text{llm}} = k_l \cdot (r_{\text{llm}})^{\gamma_l} \cdot D_f^{\delta_l} + E_{\text{llm}} \tag{8}$$

where $\Theta_{t_\text{ve}} = \{k_v, \gamma_v, \delta_v, E_{\text{ve}}\}$ and $\Theta_{t_\text{llm}} = \{k_l, \gamma_l, \delta_l, E_{\text{llm}}\}$ are the fitted coefficients.

To find the balance point, we set these two time estimates to be approximately equal ($t_{\text{ve}} \approx t_{\text{llm}}$) for a target dataset size, denoted as D_{target}. This gives the following relationship:

$$k_v \cdot (r_{\text{ve}})^{\gamma_v} \cdot D_{\text{target}}^{\delta_v} + E_{\text{ve}} \approx k_l \cdot (r_{\text{llm}})^{\gamma_l} \cdot D_{\text{target}}^{\delta_l} + E_{\text{llm}} \tag{9}$$

Our goal is to solve for r_{ve}. We begin by isolating the term containing r_{ve} on one side of the equation:

$$k_v \cdot (r_{\text{ve}})^{\gamma_v} \cdot D_{\text{target}}^{\delta_v} \approx k_l \cdot (r_{\text{llm}})^{\gamma_l} \cdot D_{\text{target}}^{\delta_l} + (E_{\text{llm}} - E_{\text{ve}}) \tag{10}$$

Finally, to solve for r_{ve}, we raise both sides to the power of $1/\gamma_v$, which yields the final expression[1]:

$$r_{\text{ve}} \approx \left(\frac{k_l \cdot (r_{\text{llm}})^{\gamma_l} \cdot D_{\text{target}}^{\delta_l} + E_{\text{llm}} - E_{\text{ve}}}{k_v \cdot D_{\text{target}}^{\delta_v}} \right)^{\frac{1}{\gamma_v}} \tag{11}$$

[1] In very rare cases (0.72% of our total runs) where the numerator becomes non-positive, implying that E_{ve} is large due to the inherent entropy of the data distribution, we set $r_{\text{ve}} = r_{\text{llm}}$ to serve as a default LoRA configuration.

H Evaluation on From-Scratch VLMs.

To assess pure fine-tuning capability, we evaluate task performance on from-scratch models whose modules have not been exposed to multimodal training data. Since the original from-scratch models are not publicly available [8, 15], we instead assemble unimodal models and connect them with a projector to replicate the reference architectures (Sec. 3.1.1). As shown in Tables 7 and 8, MARS consistently outperforms the baseline configurations, highlighting its effectiveness in enabling from-scratch models to acquire downstream knowledge.

Table 7: From-scratch: fixed-rank tuning across learning rates. Details as in Table 1.

Model	Bench.	LoRA (lr_{ve}, lr_{llm})			MARS
		(★, 1e-5)	(★, 1e-6)	(★, 1e-7)	
OV-0.5B*	LLaVA (↓)	5.0011	5.0209	5.4407	**4.9547**
	SciQA (↑)	42.49	42.12	39.88	**45.01**
OV-7B*	LLaVA (↓)	3.8701	4.0298	4.3407	**3.5609**
	SciQA (↑)	63.71	64.11	60.24	**65.07**
Q2-VL-7B*	LLaVA (↓)	3.7738	3.7949	4.0315	**3.6214**
	SciQA (↑)	65.47	62.87	59.28	**66.47**

Table 8: From-scratch: adaptive rank search baselines. Details as in Table 2.

Model	AdaLoRA	LoRA (r_{ve}, r_{llm})		MARS
		(★, 8)	(★, 16)	(★, 32)	
LLaVA Bench (perplexity ↓)					
OV-0.5B*	5.0807	5.2819	5.0011	4.9912	**4.9547**
OV-7B*	4.0012	3.6175	3.8701	3.9204	**3.5609**
Q2-VL-7B*	4.1555	3.7480	3.7738	4.1057	**3.6214**
ScienceQA bench (accuracy (%) ↑)					
OV-0.5B*	41.52	38.27	42.49	41.99	**45.01**
OV-7B*	58.28	64.11	64.15	64.07	**65.07**
Q2-VL-7B*	60.08	65.87	65.47	63.67	**66.47**

I Generality of MARS with Broader Benchmark Coverage

In Section 4.2, we evaluate MARS on broader benchmark suites, including generalist suites (MME, POPE, MMStar) and domain-specific tasks (GQA, TextCaps, AI2D), as provided by the *Multimodal Evaluation Toolkit* [43]. Here, we provide a brief description of each benchmark along with the detailed numerical results omitted from Figure 3.

I.1 Benchmark Description

Table 9: Summary of Benchmarks and Evaluation Metrics

Benchmark	Focus	Metric in Figure 3
MME [23]	General perception and cognition	Cognition + Perception
POPE [25]	Object hallucination	F1 Score
MMStar [24]	Visual perception and reasoning	Avg. Accuracy
GQA [28]	Compositional visual reasoning	Accuracy
TextCaps [26]	OCR-based image captioning	ROUGE_L
AI2D [27]	Scientific diagram understanding	Accuracy

I.2 Detailed Numbers for Figure 3

Table 10: Performance comparison on diverse multimodal benchmarks demonstrating the broad generalization of generalist models when MARS is applied to the connected VE and LLM.

Method	MME (sum)	POPE	MMStar (avg)	TextCaps (Val)	AI2D	GQA
AdaLoRA [29]	1524.0	79.7	28.6	38.0	45.5	52.7
LoRA (★, 64)	1662.4	86.8	36.1	42.6	48.0	55.1
MARS	1695.6	86.3	39.4	42.9	49.9	55.7

Table 11: Fine-grained capability breakdown on MMStar [24]

Method	Coarse Per.	Fine-Grained Per.	Instance Reason.	Logical Reason.	Sci & Tech	Math
AdaLoRA	34.7	23.2	27.8	29.9	24.4	31.5
LoRA (★, 64)	55.5	27.8	49.8	32.1	22.2	29.0
MARS	58.3	28.4	48.4	34.9	33.0	33.4

J Simultaneous Multi-Rank Tuning for Scalability

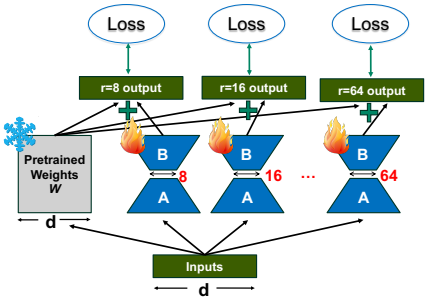


Figure 11: Simultaneous Multi-Rank Tuning Mechanism

Figure 11 illustrates our proposed optimization for the calibration phase, leveraging the frozen nature of the pre-trained backbone weights (W). In standard fine-tuning, the computationally expensive forward pass through the large backbone (W) must be recomputed for every distinct rank configuration tested. To eliminate this redundancy, we construct a computational graph where the backbone features are computed only once per batch. These shared features are then branched into multiple, lightweight parallel LoRA adapter heads (e.g., initialized with ranks $r = \{8, 16, 32, 64\}$). Since the LoRA parameters constitute a negligible fraction of the total model size ($< 1\%$), the computational overhead of running these parallel branches is minimal. This strategy allows us to gather perplexity and convergence statistics for all candidate ranks simultaneously in a single training run, in the best case, effectively reducing the calibration search cost from $\mathcal{O}(K^N)$ to $\mathcal{O}(1)$ (where K is the number of rank candidates and N is the number of modalities). This optimization ensures that MARS remains highly efficient even as the search space expands.

K Stability Across Runs

We report mean±std over three seeds for MARS and the strongest LoRA baseline in Table 12. Variance is low (std $< 0.3\%$) and rankings are stable across all settings.

Table 12: Stability of MARS across three random seeds. Performance is reported as mean±std.

Model	Benchmark	Item	$(r_{\text{ve}}, r_{\text{llm}})$	Performance
LLaVA-OV-7B	LLaVABench	Baseline	(16, 16)	2.2323±0.0038
LLaVA-OV-7B	LLaVABench	MARS	(10, 16)	2.1875±0.0025
Qwen2.5-VL-3B	LLaVABench	Baseline	(16, 8)	3.5914±0.0021
Qwen2.5-VL-3B	LLaVABench	MARS	(10, 8)	3.5825±0.0017
LLaVA-OV-7B	ScienceQA	Baseline	(16, 32)	73.64±0.29%
LLaVA-OV-7B	ScienceQA	MARS	(22, 32)	74.30±0.25%
Qwen2.5-VL-3B	ScienceQA	Baseline	(16, 8)	78.32±0.18%
Qwen2.5-VL-3B	ScienceQA	MARS	(15, 8)	79.28±0.30%

L Transferability Analysis of Scaling Law Coefficients

To clarify the scope of transferability in MARS, we separately analyze the coefficients of Scaling Law-P and Scaling Law-C.

Scaling Law-P Coefficients. Table 13 reports the coefficient of variation (CV) for each Scaling Law-P parameter. The dataset-size exponent β and irreducible error E transfer robustly across model families (CV $< 4\%$), indicating that asymptotic scaling behavior is architecture-agnostic. The rank exponents α_v and α_l are model-specific (cross-family CV $> 30\%$) and require per-model calibration.

Table 13: Coefficient of Variation (CV) analysis for **Scaling Law-P** coefficients.

Coefficient	Within-Family CV	Cross-Family CV	Transfers?
β (dataset-size)	2–3%	3.7%	Yes
E (irreducible loss)	$\sim$1%	1.1%	Yes
α_v (VE rank)	6–7%	31.1%	No (need calibration)
α_l (LLM rank)	4–6%	35.0%	No (need calibration)

Scaling Law-C Coefficients. Table 14 reports the CV for the convergence exponent δ, which appears in Scaling Law-C (Equation 2). The high cross-family CV is precisely the justification for MARS's per-model calibration design.

Table 14: Coefficient of Variation (CV) analysis for **Scaling Law-C** convergence exponent δ.

Grouping	Mean δ	CV
Within LLaVA-OV (pre-trained)	0.697	3.0%
Within Qwen2.5-VL (pre-trained)	0.570	2.5%
Cross-family (all pre-trained)	0.646	11.0%
From-scratch vs. Pre-trained	0.633	14.1%

M Perplexity–Performance Correlation Across Benchmarks

To validate that perplexity is a reliable optimization proxy beyond ScienceQA, we report Pearson correlations between validation perplexity and downstream task performance across six diverse benchmarks. As shown in Table 15, consistent negative correlations (r ranging from -0.72 to -0.97) confirm that lower perplexity reliably predicts better task performance across both specialist and generalist benchmarks.

Table 15: Correlation between validation perplexity and downstream task performance across diverse benchmarks.

Benchmark	Fine-tuning Data	Pearson r	p-value
ScienceQA	ScienceQA	-0.97	< 0.001
MMStar	LLaVA-665K	-0.89	< 0.001
AI2D	LLaVA-665K	-0.79	< 0.01
MME	LLaVA-665K	-0.78	< 0.001
GQA	LLaVA-665K	-0.74	< 0.01
POPE	LLaVA-665K	-0.72	< 0.01

N Limitations

We acknowledge several limitations consistent with the empirical nature of prior scaling law research [11, 12]:

1. **Perplexity proxy scope.** While we demonstrate consistent negative correlations between perplexity and task performance across seven diverse benchmarks (Appendix M), the proxy has been validated primarily within our target evaluation suite. Its reliability on other task types (e.g., embodied AI, video understanding) requires further verification.
2. **Multi-modality scalability (> 2 modalities).** Our tri-module experiment demonstrates $O(N)$ scalability, but exhaustive validation beyond two modalities (e.g., vision + audio + language [33]) remains future work.

O Broader Impacts

As VLMs become more integrated into real-world applications, their adoption through fine-tuning is becoming increasingly popular across a wider range of downstream domains. The optimal rank pair identified by MARS reduces fine-tuning costs by eliminating the repetitive fine-tuning process caused by laborious trial-and-error hyperparameter searches. Ultimately, MARS will accelerate the development cycle and lower the carbon footprint for large-scale VLM fine-tuning.

P License of Assets

Code and Frameworks. Our implementation is built on the Cornstarch framework[2] (Apache-2.0). For evaluation, we use LMMs-Eval[3] (MIT). We compare against AdaLoRA[4] (MIT) and GeoLoRA[5] (MIT).

Models. We use the following pre-trained models: LLaVA-OneVision[6] (Apache-2.0), and Qwen2-VL / Qwen2.5-VL[7] (Apache-2.0).

Datasets and Benchmarks. We utilize the following datasets and benchmarks: LLaVA-Instruct[8] (Apache-2.0), ScienceQA[9] (MIT), MME[10] (academic research use), MMStar[11] (research use), POPE[12] (MIT), TextCaps[13] (CC BY 4.0), AI2D[14] (CC BY-SA), and GQA[15] (CC BY 4.0). All assets are used in accordance with their respective licenses and intended for research purposes.

[2] https://github.com/cornstarch-org/Cornstarch
[3] https://github.com/EvolvingLMMs-Lab/lmms-eval
[4] https://github.com/QingruZhang/AdaLoRA
[5] https://github.com/ScSteffen/Publication-GeoLoRA-Geometric-integration-for-parameter-efficient-fine-tuning
[6] https://github.com/LLaVA-VL/LLaVA-NeXT
[7] https://github.com/QwenLM/Qwen2-VL
[8] https://github.com/haotian-liu/LLaVA
[9] https://github.com/lupantech/ScienceQA
[10] https://github.com/BradyFU/Awesome-Multimodal-Large-Language-Models
[11] https://github.com/MMStar-Benchmark/MMStar
[12] https://github.com/RUCAIBox/POPE
[13] https://textvqa.org/textcaps/
[14] https://prior.allenai.org/projects/diagram-understanding
[15] https://cs.stanford.edu/people/dorarad/gqa/

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Our core claim about imbalanced training dynamics reflects MARS's contributions and scope.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Limitations are discussed in Appendix N.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: See Appendices G for the proofs and F.1 for the theoretical justification of our scaling laws.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: See Sec.4.1 and AppendicesA and C.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We will release the source code and user guide in a personal GitHub repository.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: All training and evaluation details are discussed in Appendices A.2 (in particular, the data splits) and C (in particular, the hyperparameters and training configurations).

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: See Appendix K. Due to the fine-tuning experiments and the large scope of evaluations, we performed statistical significance tests on representative cases only.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provide details on the computational resources in Sec.4.1 (NVIDIA GH200 GPU) and Figure5.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Our research conforms to the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: See Appendix O.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: Our work does not introduce new data or models. Thus, MARS poses no such risks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: See Appendix P.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We will release the source code and user guide in a personal GitHub repository (CC-BY 4.0 or Apache-2.0 license).

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: MARS does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: MARS does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: The LLM is used only for writing, editing, literature review, coding, and formatting. The core methodological development in this research does not involve LLMs.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.