RESOLUTION WITHOUT RESOLUTION: VISUAL TOKENS AND SCALING IN VISION-LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Vision-language models (VLMs) face a fixed-budget trade-off between processing more visual information for fine-grained perception and using a larger language backbone for complex reasoning. Existing studies do not tell us which combination of backbone size and input resolution to deploy, especially in high-resolution deployments. To address this gap, we propose the Separable Law that describes how VLM performance changes with language backbone size and visual token count. Specifically, we fit the law to measurements from 26 InternVL and QwenVL models, with language backbone sizes from 1B to 72B, on four high-resolution benchmarks with image sizes from 224 pixels to 8K. We find that the questions responding to scaling can be predicted from the skill they require, while a substantial fraction never responds at all. We also find that the two model families gain similarly from a larger backbone, while their gains from more visual tokens differ sharply. Combined with a cost law, the Separable Law gives a closed-form rule for allocating compute between backbone size and visual tokens. When deployment is limited to available configurations, the law identifies model and image sizes that perform close to the best feasible choice under the same budget. We hope our work offers a principled way to decide how much a model should be allowed to see at high resolution, given what it must reason about.

1 INTRODUCTION

Vision-language models (VLMs) have made substantial progress on tasks that require visual understanding and reasoning (Yin et al., 2024; Jin et al., 2025; Deng et al., 2026). Many real-world applications require these models to identify fine-grained details in high-resolution images containing large amounts of visual information (Wu & Xie, 2024; Fu et al., 2024; Zhang et al., 2025b). To preserve these details, VLMs commonly process high-resolution inputs through dynamic resizing (Wang et al., 2024; Kimhi et al., 2026; Liu et al., 2026b) or image tiling (Chen et al., 2024c; Liu et al., 2025; Luo et al., 2025a), and the two strategies produce different visual-token counts as image resolution increases. At the same time, complex reasoning tasks such as cross-region comparison (Wang et al., 2025d; Li & Peng, 2026) and multi-step deduction (Wang et al., 2026a; He et al., 2026) require greater language backbone capacity. Meeting these perception and reasoning demands makes the VLM inference cost depend on the visual token count R and the backbone size N, so a fixed inference budget requires balancing compute between the two. This raises an important question, illustrated in Figure 1: **Given a fixed inference budget, how should compute be allocated between the language backbone and the visual tokens?**

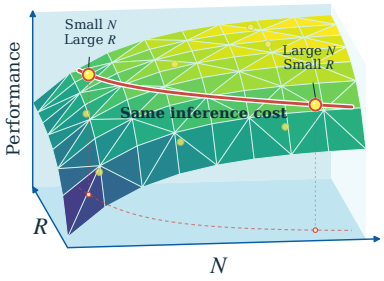


Figure 1: Fixed-budget VLM inference.

Existing studies have improved VLM inference by reducing the cost of individual components. Token pruning and merging reduce the number of visual tokens (Chen et al., 2024a; Shang et al., 2025; Zhang et al., 2024), while efficient vision encoders (Vasu et al., 2025) and lightweight projectors (Cha et al., 2024) reduce visual processing cost. But these efficiency gains do not determine how a fixed inference budget should be divided between a larger backbone and more visual tokens. Recent work models performance jointly over backbone size and visual-token count, but only at a low image resolution of 336 pixels (Li et al., 2025a). Other work jointly adapts visual-token

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

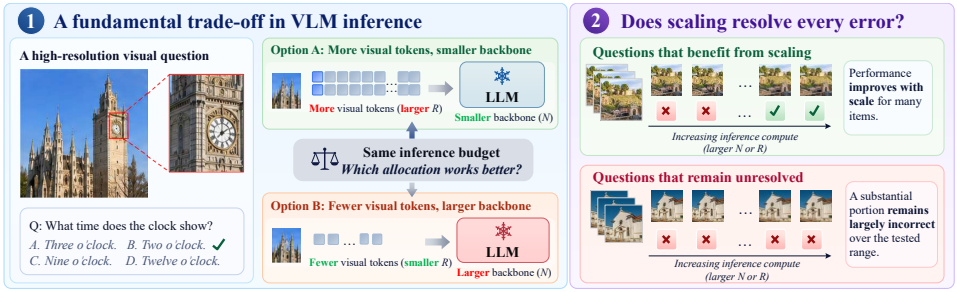


Figure 2: **Motivation. Left:** Language backbone size (N) and visual tokens (R) both consume inference compute, which raises the question of how to split a fixed budget between them. **Right:** Scaling helps many questions but leaves others wrong at every scale we test.

retention and active LLM computation within a given backbone (Wang et al., 2026b). A complementary line of work evaluates test-time scaling strategies, including chain-of-thought prompting and self-consistency, across VLMs and tasks (Sammani et al., 2026). However, these studies remain confined to low-resolution inputs and do not characterize scaling behavior under high-resolution inputs, where how a model turns an image into visual tokens has more room to change the token count. They also leave open how different image-processing strategies affect scaling behavior.

To address this gap, we propose the Separable Law that describes how VLM performance changes with language backbone size and visual token count. Its form is $L(N, R) = e_\infty + A/N^\alpha + B/R^\beta$, where N is the language backbone size, R is the visual token count, and L is the fraction of questions answered incorrectly. The term e_∞ is the error floor that remains even as both the language backbone size and the visual token count grow. Each of the two reducible-error terms depends on only one variable: changing N with R held fixed affects only A/N^α, and changing R with N held fixed affects only B/R^β, so the two components can be estimated separately. To test the law, we evaluate 26 models from the InternVL and QwenVL families, with model sizes from 1B to 72B, on four high-resolution benchmarks covering input resolutions from 224 pixels to 8K. For each combination of model series and benchmark, we fit the law separately and find that it fits the observed errors better than the alternative formulations we evaluate, so the two components can be estimated separately, one for the language backbone and the other for the visual tokens.

With the Separable Law established, we examine its fitted parameters to characterize how VLM performance varies with N and R. **(i) We begin with the fitted error floor** e_∞, and find that some questions remain incorrectly answered across the evaluated range, even as the backbone grows larger and the visual-token count increases (Figure 2, right). Further analysis shows that reasoning tasks have higher fitted error floors than perception tasks, though the gap comes from a minority of task subsets whose error never falls much. Decomposing these floors more finely, we see that the required skill explains more variation than the visual domain does, with the most severe cases at particular pairings of skill and scene. We then turn to **(ii) the two scaling exponents,** α **for how fast error falls with backbone capacity and** β **for how fast it falls with visual tokens**. We find that the two model families agree on capacity (α) but differ sharply on visual tokens (β). This difference arises because, as input image size increases, the visual-token count changes little for InternVL but grows substantially for QwenVL. We also find that the typical ordering of the two exponents reverses between the model families, with $\alpha > \beta$ for QwenVL and $\beta > \alpha$ for InternVL. Thus, a compute-allocation strategy optimized for one architecture may not transfer to another.

To put the Separable Law into practice, we turn it into a framework for allocating compute between the language backbone and the image size under a limited budget. First, we ask how a fixed inference budget should be spent when input resolutions reach 8K. Combining the law with a cost law gives a closed-form answer: larger backbones for InternVL and more visual tokens for QwenVL. Second, we ask which configuration to choose from those available in deployment. Selection by predicted error identifies the best one on average, even when the model and image sizes are held out from the fit. Third, we ask whether a benchmark is difficult in a way that additional scale can resolve. Question-level results show how well a benchmark distinguishes models and whether its questions are more sensitive to backbone size or input image size. We hope this turns visual scaling from something measured case by case into something that can be reasoned about and predicted.

2 METHOD

2.1 PRELIMINARIES

Scaling laws for language models. Scaling laws (Kaplan et al., 2020) describe how model performance changes with controllable resources. For language model pretraining, the Chinchilla formulation (Hoffmann et al., 2022) provides a widely used parametric form. It expresses the loss after training as the sum of an irreducible loss floor and two power-law terms, capturing the effects of model capacity and training data size, respectively:

$$L(N, D) = e_\infty + \frac{A}{N^\alpha} + \frac{B}{D^\beta}. \tag{1}$$

Here, N denotes the number of model parameters and D denotes the number of training tokens. The term e_∞ represents the asymptotic loss floor, while the coefficients A and B, together with the exponents α and β, are fitted constants that characterize the decay rates along the two scaling axes.

Inference-time scaling laws for VLMs. Such scaling laws have recently been extended to VLM inference. One such law models performance as a multiplicative function of language backbone size N and the total number of inference tokens T (Li et al., 2025a),

$$L(N, T) = \frac{A}{N^\alpha} \cdot \frac{B}{T^\beta} + D, \tag{2}$$

where T counts visual tokens retained by a learned compression module at a fixed input resolution. An analogous law has been fit for video VLMs, where the sampled frame count determines the token budget and the model is fine-tuned for each configuration before evaluation (Wang et al., 2025b).

2.2 EXPERIMENTAL SETUP

Models and benchmarks. We evaluate five model series from the QwenVL and InternVL families, with backbone sizes ranging from 1B to 72B. These include Qwen2.5-VL (3B, 7B, 32B, 72B) (Bai et al., 2025b), Qwen3-VL (2B, 4B, 8B, 32B) (Bai et al., 2025a), InternVL2.5 (1B, 2B, 4B, 8B, 26B, 38B) (Chen et al., 2024b), InternVL3 (1B, 2B, 8B, 9B, 14B, 38B) (Zhu et al., 2025), and InternVL3.5 (1B, 2B, 4B, 8B, 14B, 38B) (Wang et al., 2025c). We count only the language backbone in N and leave out the vision encoder and projector, although larger models within a series often differ in those components as well. We test these models on four benchmarks built from high-resolution images, namely HR-Bench (Wang et al., 2025d), MME-RealWorld (Zhang et al., 2025a), TreeBench (Wang et al., 2026a), and V* Bench (Wu & Xie, 2024). To vary the visual input, we resize each image to one of several preset sizes and leave the model's own visual processor unchanged.

Units of analysis. Each evaluation gives a pair (N, R) on one benchmark, which we call a cell. We set the input image size and measure R, the number of visual tokens that enter the language backbone. All evaluations use greedy decoding, and the error L of a cell is one minus its mean multiple-choice accuracy. Each of the 26 models is measured under 25 combinations of benchmark and image size, giving 650 cells and 536,781 valid evaluations. Pairing each model series with each benchmark gives 20 groups, which serve as the units for fitting our Separable Law.

2.3 OUR SEPARABLE INFERENCE SCALING LAW

Existing inference-time laws (Li et al., 2025a; Wang et al., 2025b) establish that downstream error responds to both model scale and the visual token budget. Eq.(2) couples the two axes multiplicatively by construction, while additive alternatives have been evaluated for video VLMs (Wang et al., 2025b). Neither study tests these functional forms on pretrained image VLMs when the input image size itself changes. Moreover, these studies vary the visual token budget through token compression or video input configurations with finetuning the model separately for each configuration, rather than by changing input image size with the model and its visual processor held fixed.

Our formulation instead starts from an accounting observation. For a VLM answering a visual query, the token stream consumed by the language backbone is the inference-time counterpart of the training data D in Eq.(1), and it is overwhelmingly visual. A single high-resolution image expands into thousands of visual tokens and dwarfs the accompanying textual query. We therefore replace

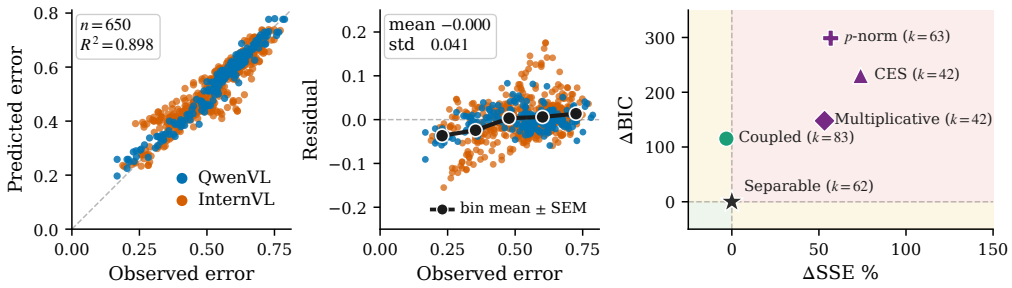


Figure 3: **Fit quality and form comparison for the Separable Law. Left:** predicted vs. observed error, one dot per (N, R) cell, with QwenVL in blue, InternVL in orange, and the line $y{=}x$. **Middle:** residual vs. observed error, with mean $\pm$ standard error of the mean (SEM) in each of five error bins. **Right:** candidate forms in Δ-space relative to the separable form, marked by a charcoal star at the origin. The axes show ΔSSE % and ΔBIC, and k is the nominal number of free parameters.

D directly with the number of visual tokens visible to the backbone and propose a law of the same form, with capacity and visual evidence contributing separately as they did for parameters and data:

$$L(N, R) = e_\infty + \frac{A}{N^\alpha} + \frac{B}{R^\beta}. \tag{3}$$

Here, L denotes downstream error, N is the number of parameters in the language backbone, and R is the number of visual tokens. The parameters α and β are the capacity and visual-token scaling exponents, respectively, governing how the corresponding error terms decay as N and R increase. We call Eq.(3) the Separable Law, since the contributions of N and R enter as two terms that add rather than multiply, and we use this name for it throughout the paper.

2.4 Validating the Separable Form

Before interpreting the fitted components of Eq.(3), we check how well it fits and whether the data favor the separable form over coupled alternatives.

The law fits. We fit the five parameters of Eq.(3) separately for each of the 20 (series, benchmark) groups and pool the predictions across 650 cells. The pooled coefficient of determination (R^2) of 0.898 indicates that the law captures the observed error surface well (Figure 3, left). Residuals are centered at zero with a standard deviation of 0.041, showing little overall bias but some variation across cells. The near-zero mean nevertheless masks a systematic trend, as mean residuals rise from -0.037 in the lowest observed-error bin to $+0.014$ in the highest (Figure 3, middle). Fit quality also varies across groups and is associated with the observed token span within each group. Overall, the fit remains good across the whole grid, with no region where the law clearly breaks down.

The data favor the separable form among the tested candidates. A good fit alone does not establish separability, so we compare Eq.(3) with alternative forms. These include the multiplicative law (Eq.(2)), written for group d as $L_d = D_d + K_d\, N^{-\alpha} R^{-\beta}$, and a coupled form that adds the term $C_d(NR)^{-\gamma}$ to Eq.(3), with group-specific $C_d \geq 0$. We also test two standard ways of letting the two resources interact. A constant-elasticity-of-substitution (CES) form (Arrow et al., 1961) aggregates N and R through a single fitted parameter that controls how far one resource can substitute for the other. A p-norm form aggregates the two error terms and the floor as a generalized mean, so that as p grows error is increasingly set by whichever resource is scarcest. All candidates are fit on the same 650 cells and share their exponents across the 20 groups. We score each candidate by the sum of squared errors (SSE) over those cells, which measures how closely the fitted surface tracks the observed error, and by the Bayesian information criterion (BIC), which adds a penalty for the number of fitted parameters so that a richer form is not favored merely for having more of them. Lower values are better for both scores. Separable and coupled forms use matched unconstrained group intercepts and nonnegative amplitudes. The multiplicative form has 53.2% higher SSE than the separable form and a higher BIC despite fewer parameters (Figure 3, right). Only the coupled form reduces SSE, from 1.221 to 1.182, but raises BIC by 115.2 due to its additional parameters. None of the other candidates earns its extra parameters, so we use the Separable Law throughout.

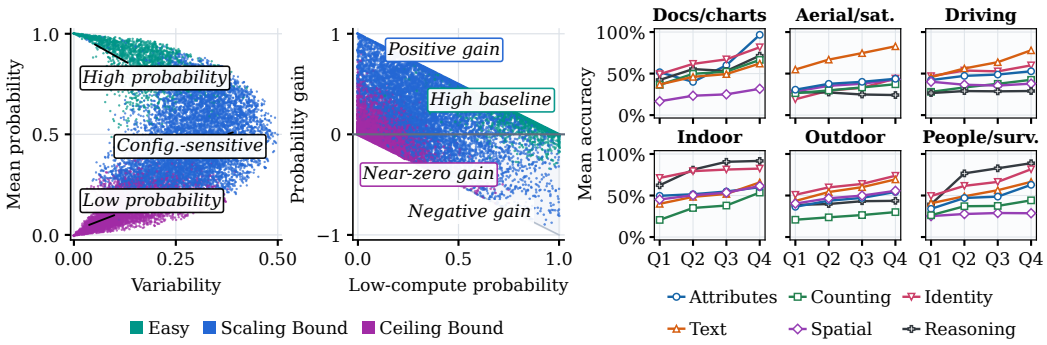


Figure 4: **Question-level scaling heterogeneity and responses by domain and skill. Left:** Mean probability of the correct option versus its variability across (N, R) configurations. **Middle:** Probability gain $\bar{p}_{Q_4} - \bar{p}_{Q_1}$ versus its low-compute value $\bar{p}_{Q_1}$. **Right:** Accuracy across compute quartiles, with one panel per visual domain and one curve per skill.

3 Finding 1: The Cognitive Ceiling

Finding 1: Scaling helps most questions but not all. About a third of them stay wrong at every model size and visual-token count we evaluate, and scaling shifts them in both directions without settling them. Which questions these are depends mainly on the visual skill a question demands, with the worst cases confined to particular pairings of skill and scene.

We study scaling limits through the fitted floor e_∞. Because it is fitted to the average error over many questions, a task can end up with a floor close to zero while a few of its questions stay wrong at the largest model size and highest visual-token count we evaluate. We therefore read this floor at three levels, from the perception and reasoning subsets these benchmarks provide (§3.1) to individual questions (§3.2) and to groups defined by visual scene and required skills (§3.3).

3.1 e_∞ is Higher for Reasoning Tasks Than for Perception Tasks

A minority of task subsets keeps a high floor. We start from the split these benchmarks already provide, labeling each question as perception or reasoning, and ask whether the fitted floor e_∞ differs between the two. We fit Eq.(3) separately for each of the five model series on these subsets, obtaining 40 (series, benchmark, task) fits. Here e_∞ is the error Eq.(3) approaches as N and R both grow without bound, so a high fitted e_∞ means that neither a larger backbone nor more visual tokens brings that task subset near zero error. We call such persistent error a *cognitive ceiling* and flag it with *Saturation*, $e_\infty \geq 0.20$, as opposed to *Floor*, $e_\infty \leq 0.01$, where the fitted error instead decays to essentially zero. In Table 1, 20% of perception fits and 33% of reasoning fits meet Saturation, while 72% and 53% respectively meet Floor. The saturated fits also account for the higher mean e_∞ on reasoning, 0.155 against 0.073, since both medians sit at the lower search bound of 0.001. These fits are aggregates over whole task subsets and hide individual questions that never improve, so we next ask which questions sit behind them.

Table 1: Results of per-task e_∞ across 40 fits.

Task type	Mean e_∞	Median e_∞	Floor	Saturation
Perception	0.073	0.001	**72%**	20%
Reasoning	**0.155**	0.001	53%	**33%**

3.2 Sample-Level Regime Discovery

Scaling response separates questions that the perception and reasoning labels group together. The fits above describe whole task subsets, so we now move to individual questions and ask which constitute the ceiling. We analyze 19,890 trajectories, one per question and model series. Using mean accuracy across configurations, we sort each trajectory into three regimes: 20.0% are Easy, above 0.90 accuracy; 49.4% are Scaling Bound, from 0.20 to 0.90; and 30.7% are Ceiling Bound, below 0.20. For each trajectory we record the probability the model assigns to the correct option, comparing its mean over the lowest and highest compute quartiles. Ceiling Bound tra-

jectories cluster at low probabilities (Figure 4, left) and gain only 0.014 between the two quartiles, against 0.239 for SCALING BOUND trajectories (Figure 4, middle). Gains therefore concentrate in the responsive SCALING BOUND subset, whereas CEILING BOUND trajectories move in both directions, 32.1% gaining more than 0.05 and 27.8% losing more than 0.05. These regimes also cut across the perception and reasoning labels, since reasoning has a higher CEILING BOUND rate, 42.2% against 26.5%, yet most reasoning trajectories still respond to scaling.

3.3 DOMAIN AND SKILL GRANULARITY

Skill captures the broad pattern, with extremes in specific domains. Locating the ceiling requires question labels that are comparable across the four benchmarks, so we re-label each unique question under a unified 6×6 taxonomy of visual domain D and required skill S (Table 2). Domain describes the visual context, while skill specifies the operation needed to answer the question. For each supported (D, S) pairing, we pool observations across the five model series and fit a separate instance of Eq.(3), obtaining 33 fitted floors. The domain panels in Figure 4 (right) show accuracy across four compute quartiles, from the lowest quartile Q_1 to the highest quartile Q_4, and the skill holding the lowest curve changes from one domain to the next, so the same operation is not uniformly difficult. A two-factor decomposition of the 33 floors attributes 36.67% of their variance to skill and 7.84% to domain, placing the broad pattern on the required operation. Marginal means nevertheless miss the most severe pairings, since Aerial/Satellite with Scene Reasoning has a fitted floor of 0.780, against domain and skill marginals of 0.227 and 0.228, and the same pairing also holds the lowest observed accuracy in that domain from the second quartile on. At the other end, most Text Reading and Object Identification pairings reach the lower search bound of 0.001, and the two highest point estimates in this group, 0.080 and 0.180, both stay below the Saturation threshold. Skill therefore gives the broad picture, while the most severe ceilings can be identified only when both the domain and the skill are specified.

Table 2: Unified 6×6 domain and skill taxonomy.

Domain D		Skill S	
D1	Documents, Charts, Slides	S1	Attribute Recognition
D2	Aerial, Satellite	S2	Text Reading
D3	Vehicles, Driving	S3	Counting
D4	Indoor	S4	Spatial Relation
D5	Outdoor	S5	Object Identification
D6	People, Surveillance	S6	Scene Reasoning

4 FINDING 2: ARCHITECTURAL DIVERGENCE

Finding 2: The gains from a larger language backbone and from more visual tokens depend on the architecture. The two families agree on capacity but differ by more than an order of magnitude on visual tokens, because their frontends turn the same pixel budget into very different token counts. Which of the two axes pays off faster reverses between them, so an allocation tuned on one architecture need not carry over to another.

Above the floor studied in Finding 1, error falls at rates set by the two exponents of Eq.(3), α for capacity and β for visual tokens. We compare them across the two families (§4.1), trace their difference to the visual frontends (§4.2), and examine the reversal in their ordering (§4.3).

4.1 CAPACITY EXPONENTS AGREE ACROSS FAMILIES, VISUAL-TOKEN EXPONENTS DO NOT

Capacity exponents differ less than visual-token exponents. For the 20 $(\text{series}, \text{benchmark})$ groups, we search over $\alpha, \beta \in [0.05, 10]$ to reduce fitting instability under limited data. We compare only the fits whose exponents land inside this range, since an exponent stopping at a bound is a lower limit rather than an estimate. For α, twelve InternVL and five of eight QwenVL groups qualify, with medians of 0.19 and 0.59 (Figure 5b). The three QwenVL groups that do not qualify stop at the upper bound, and adding them back raises the QwenVL median to 0.78 (Table 7). The architectural difference is therefore more pronounced in visual-token scaling than in capacity scaling.

Visual-token exponents differ in both magnitude and boundary behavior. For β, the cross-family ordering reverses: the three interior InternVL estimates, all from HR-Bench, have a median of 4.66, versus 0.17 for QwenVL (Figure 5b). These medians cover all eight QwenVL groups but only three of twelve InternVL groups, as the remaining nine reach the upper bound of 10 and carry

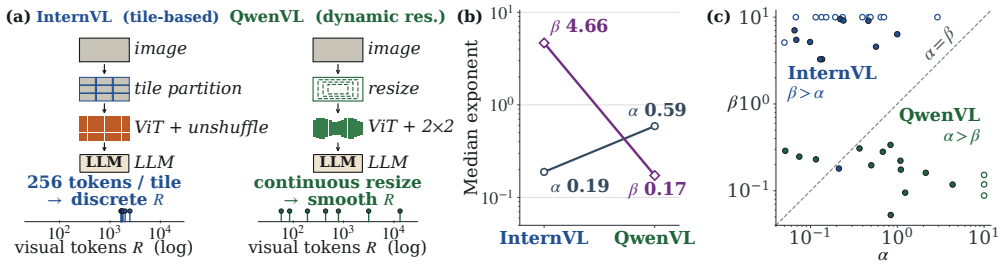


Figure 5: **Visual frontends expose different token ranges and scaling exponents.** **(a)** InternVL uses discrete tiling, whereas QwenVL uses continuous resizing. **(b)** Median exponents reverse, with $\beta > \alpha$ for InternVL and $\alpha > \beta$ for QwenVL. **(c)** Task-level fits in the (α, β) plane, where filled markers denote interior estimates and hollow markers denote boundary-clipped ones.

the family's largest β values, so the gap quoted here is conservative. To understand this contrast, we next examine how far the token ranges of the two visual frontends actually extend.

4.2 Discrete Tiling and Continuous Resizing Expose Different Token Ranges

Discrete tiling exposes a short token range. InternVL picks a tile grid of at most twelve cells by matching the aspect ratio of the input, and uses area only to break ties (Chen et al., 2024c). It then resizes the image to that grid and cuts it into 448×448 tiles, each yielding 256 visual tokens that enter the LLM after pixel unshuffle. Pixel count therefore barely enters the choice of grid, and raising the pixel budget from 224 to 2048 px leaves the tile count unchanged for 70.2% of questions, so the within-group token span has a median of only $1.22\times$ (Figure 5a). Such a narrow range cannot pin down how fast the error decays, which is why nine InternVL β estimates stop at the upper bound.

Continuous resizing exposes a much longer token range. QwenVL resizes the whole image to the target pixel count and reads it with 2D RoPE (Wang et al., 2024). A 2×2 MLP merger compresses every four patches into one visual token, so the token count follows the pixel count continuously rather than in blocks. Raising the pixel budget over the same 224 to 2048 px range therefore moves the within-group token count by $34.09\times$ to $87.79\times$, and the full grid covers roughly 37 to 15,000 visual tokens (Figure 5a). Over such a wide range, the decay is visible, and all eight QwenVL groups return interior β estimates with a median of 0.17, far below the steep values InternVL reaches.

4.3 The α/β Ordering Reverses Across Architectures

On the evaluated grids, the exponent balance reverses across architectures. The ratio α/β compares the two axes within one architecture, with a value above 1 meaning capacity decays faster and below 1 meaning visual tokens do. We compute it on the 40 task-level fits and keep those with both exponents in the search interior, which retains most QwenVL fits but fewer than half of the InternVL ones. The median ratio is 2.55 for QwenVL and 0.0395 for InternVL, and the two families separate clearly in Figure 5c. Almost all of the dropped InternVL fits stop at the upper bound on β, so 0.0395 is an upper bound on the true median. The larger exponent therefore corresponds to capacity in QwenVL and to visual tokens in InternVL, which we next turn into an allocation rule.

5 Finding 3: From Laws to Practice

> **Finding 3:** The best way to spend a fixed budget depends on how much error each axis removes per unit of compute. Combined with a cost law, our law directs extra compute toward a larger backbone for InternVL and toward more visual tokens for QwenVL. The option our law ranks first performs nearly as well as the best available option, even when it was held out from the fit.

We first fit a cost law alongside the Separable Law and solve the two together for a continuous allocation (§5.1), then ask how to choose among the configurations a deployment actually offers (§5.2), and finally turn the same quantities around to describe the benchmarks themselves (§5.3).

5.1 From Laws to a Deployment Decision

A fitted cost law for the two axes. Since N and R are both defined where the visual tokens enter the language backbone, we follow Li et al. (2025a) in measuring allocation cost by the FLOPs the language backbone spends on the prompt, the visual tokens plus the question text. This gives a common basis across the two frontends, and for one cell of the grid the cost is

$$C_{\text{cell}} = 2P_{\text{LLM}}T_{\text{prompt}} = 2P_{\text{LLM}}(R + T_0), \tag{4}$$

where P_{LLM} is the parameter count of the language backbone, T_{prompt} is the length of the prompt, and T_0 comprises the question, template, and special tokens. The vision encoder is left out of this count, so the allocations are optimal under a language-side budget. Cost is linear in N at fixed R, but doubling R does not double the cost, since T_0 does not change with R. Eq.(4) is therefore not an exact power law in (N, R), but a power law approximates it over the observed grid, so we take

$$C(N, R) = \kappa N^p R^q, \tag{5}$$

where κ sets the scale and p and q are the fitted cost elasticities for capacity and visual tokens. We use it as the budget constraint below. Fitting it over the same 650 cells gives $p = 1.00$, matching the linearity above, together with $q = 0.77$ and log-space $R^2 = 0.995$.

Visual–Parameter Exchange balances return against cost. Given the error and cost laws, we now ask how a fixed budget should be split between the two axes. This gives the optimization problem

$$(N^*, R^*) = \arg\min_{N,R>0} L(N, R) \quad \text{subject to} \quad \kappa N^p R^q \le C_0. \tag{6}$$

Its solution follows from two conditions. The first places it on the budget boundary $\kappa N^p R^q = C_0$, since all fitted coefficients and exponents are positive, so error falls along both axes and an optimum never leaves budget unspent. The second fixes where on that boundary it sits, because if one axis returned more error reduction per unit of spending than the other, shifting spending toward it would lower error further, so at the optimum the two returns match,

$$\frac{\alpha A}{p}(N^*)^{-\alpha} = \frac{\beta B}{q}(R^*)^{-\beta}. \tag{7}$$

Both conditions are linear in $\log N$ and $\log R$, so the pair is a 2×2 system with determinant $p\beta + q\alpha$ and right-hand sides $\log \bar{C}$ and $\log K$, where $\bar{C} = C_0/\kappa$ and $K = \alpha A q/(\beta B p)$, and solving gives

$$N^* = \bar{C}^{\frac{\beta}{p\beta+q\alpha}} K^{\frac{q}{p\beta+q\alpha}}, \qquad R^* = \bar{C}^{\frac{\alpha}{p\beta+q\alpha}} K^{-\frac{p}{p\beta+q\alpha}}. \tag{8}$$

We call this the Visual–Parameter Exchange (VPE) allocation, named for the rate at which the two axes trade against each other at the optimum, and it is the unique optimum under the fitted cost law. Since a deployment moves between budgets rather than sitting at one, we differentiate Eq.(8),

$$\frac{\partial \log N^*}{\partial \log C_0} = \frac{\beta}{p\beta + q\alpha}, \qquad \frac{\partial \log R^*}{\partial \log C_0} = \frac{\alpha}{p\beta + q\alpha}. \tag{9}$$

Which axis grows faster is decided by whether β or α is larger, so the reversal of Finding 2 makes InternVL add capacity first and QwenVL add visual tokens first (Figure 6a). Only the exponents enter these two rates, so the direction of expansion is more robust than the point it starts from.

5.2 Choosing Among Available Configurations

The fitted law transfers to configurations it never saw. A deployment picks from a finite catalogue, so the continuous optimum must become a choice among available configurations. Projection takes the continuous optimum, pulls N^* and R^* back into the observed range, and selects the nearest feasible configuration in log coordinates. Direct selection skips the continuous optimum and instead evaluates Eq.(3) at every feasible configuration, taking the lowest. To keep training and evaluation apart, we drop one backbone size and one pixel budget from the performance fit and score allocation only on the configurations left out. Across 2,379 such decisions, projection carries a mean regret of 2.33 pp and a 90th percentile of 7.71 pp. Three static rules that make no use of Eq.(3)—taking the largest backbone, the largest token count, or the log-centre of the range allowed by the budget—have mean regrets from 4.09 to 7.56 pp and 90th-percentile regrets from 12.23 to 20.14 pp (Figure 6b). Direct selection is better still, at 0.76 pp and 2.74 pp, and the regret CDF shows the same ordering across thresholds (Figure 6c). Both read the same law over the same candidates, so this gap comes from the projection step. VPE still earns its place by describing how the balance shifts as the budget grows, whereas direct selection is what we would apply to a fixed catalogue.

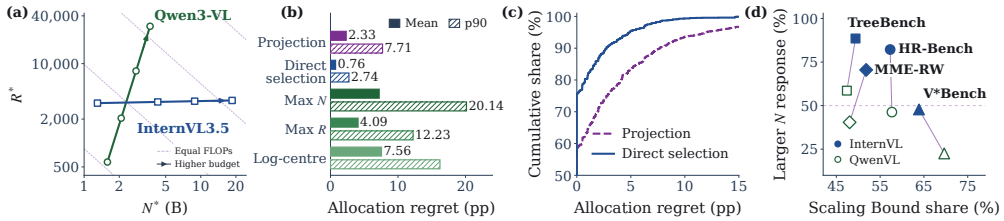


Figure 6: **Allocation and benchmark diagnostics.** **(a)** VPE trajectories (N^*, R^*) for one series per family, arrows marking increasing budget. **(b)** Mean (solid) and p90 (hatched) regret under joint holdout. **(c)** Regret CDFs for projection and direct selection. **(d)** Per benchmark and family, the SCALING BOUND share against the share of those questions responding more to N than to R.

5.3 WHAT A BENCHMARK SEPARATES AND WHICH AXIS IT REWARDS

The same analysis can also be used to examine a benchmark. Only the SCALING BOUND questions of a benchmark separate models, since the other two regimes give the same answer for every model. Among these questions, we call one N-dominated when its accuracy moves further along the backbone axis than along the visual token axis. The dominant axis, however, depends on the frontend, because InternVL's token count barely moves with image size, so InternVL shows a larger N-dominated share than QwenVL on every benchmark (Figure 6d). Even so, both frontends rank the four benchmarks the same way, so a benchmark can still be assessed by how many of its questions separate models and whether they call for a larger backbone or more visual tokens.

6 RELATED WORK

Inference scaling and compute allocation in VLMs. How VLMs should spend inference compute has drawn growing attention as their inputs grow larger. Approaches range from making visual tokens cheaper through nested token representations, adaptive patching, and efficient encoders (Cai et al., 2025; Liu et al., 2026b; Vasu et al., 2025) to choosing the input resolution per task (Luo et al., 2025b; Kimhi et al., 2026) and requesting a sharper view only when needed (Yang et al., 2026; Lee et al., 2026). Test-time methods add a further axis, spending compute on search, verification, or renewed looking (Snell et al., 2024; Wang et al., 2025a; Bai et al., 2026; Avogaro et al., 2026). Closest to our goal, scaling analyses relate accuracy to model size and visual-token count (Li et al., 2025a; Du et al., 2025; Wang et al., 2025b; Li et al., 2025b), with the token count largely set by learned compression, frame sampling, or retraining. Our work fits a law over backbone size and the visual-token count produced at each input size, across frontends that process images differently.

Perceptual bottlenecks and uneven scaling across questions. Understanding where VLMs fail has moved from aggregate scores toward finer diagnosis. Diagnostic benchmarks locate failures in fine-grained perception (Tong et al., 2024; Fu et al., 2024; Wu & Xie, 2024; Zhang et al., 2025a; Wang et al., 2025d), and adaptive perception methods address them by searching or zooming into the region that holds the answer (Wu & Xie, 2024; Wang et al., 2025e; Shen et al., 2025; Liu et al., 2026a). Scaling gains also differ across capabilities in contrastive VLMs (Al-Tahan et al., 2024). Furthermore, question-level analyses track confidence over training (Swayamdipta et al., 2020), separate model ability from question difficulty (Truong et al., 2026), or show that per-question gains and losses offset as video compute grows (Sun et al., 2026). Our work follows each question as the backbone and visual tokens grow together, and relates those that stay wrong to the skill they require.

7 CONCLUSION

In this work, we propose the Separable Law, which describes how VLM performance changes with language backbone size and visual token count. We fit it on four high-resolution benchmarks with input sizes from 224 pixels to 8K, where balancing the two matters most. We find that about a third of questions remain out of reach at every scale we test, and that more visual tokens pay off far more for QwenVL than for InternVL. We hope our work offers a principle for allocating limited inference compute in high-resolution settings, applicable beyond the specific models and benchmarks we test.

AI USE STATEMENT

In this work, we use generative AI tools for two purposes: to support qualitative data analysis and to edit this paper for grammar, wording, and readability. For the first purpose, we use GPT-4o-mini to assign domain and required-skill labels to benchmark questions, and we then inspect these assignments against the taxonomy and decision rules of Appendix C.4. For the second purpose, we check every AI-assisted edit against the underlying results. In both cases, we review all AI-assisted work, and we take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We evaluate publicly released models without modifying them in any way. For the evaluation itself, Section 2.2 and Appendix A describe the models, benchmarks, input sizes, and the evaluation protocol, while Appendix B.1 specifies how the law is fitted. For the analyses built on it, Appendix C.4 documents the domain and skill annotation, while Appendix E documents the allocation and regret evaluation. Since all models and benchmarks are publicly available, the evaluation grid can be reconstructed from these descriptions.

REFERENCES

Haider Al-Tahan, Quentin Garrido, Randall Balestriero, Diane Bouchacourt, Caner Hazirbas, and Mark Ibrahim. Unibench: Visual reasoning requires rethinking vision-language beyond scaling. *Advances in Neural Information Processing Systems*, 37:82411–82437, 2024.

Kenneth J Arrow, Hollis B Chenery, Bagicha S Minhas, and Robert M Solow. Capital-labor substitution and economic efficiency. *The review of Economics and Statistics*, 43(3):225–250, 1961.

Niccolo Avogaro, Nayanika Debnath, Li Mi, Thomas Frick, Junling Wang, Zexue He, Hang Hua, Konrad Schindler, and Mattia Rigotti. Sparc: Separating perception and reasoning circuits for test-time scaling of vlms. *arXiv preprint arXiv:2602.06566*, 2026.

Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-vl: A frontier large vision-language model with versatile abilities. *arXiv preprint arXiv:2308.12966*, 1(2):3, 2023.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025a.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report, 2025b. URL https://arxiv.org/abs/2502.13923.

Tianyi Bai, Zengjie Hu, Fupeng Sun, Qiu Jiantao, Yizhen Jiang, Guangxin He, Bohan Zeng, Conghui He, Binhang Yuan, and Wentao Zhang. Multi-step visual reasoning with visual tokens scaling and verification. *Advances in Neural Information Processing Systems*, 38:74554–74592, 2026.

Mu Cai, Jianwei Yang, Jianfeng Gao, and Yong Jae Lee. Matryoshka multimodal models. In *International Conference on Learning Representations*, volume 2025, pp. 46254–46272, 2025.

Junbum Cha, Wooyoung Kang, Jonghwan Mun, and Byungseok Roh. Honeybee: Locality-enhanced projector for multimodal llm. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13817–13827. IEEE, 2024.

Liang Chen, Haozhe Zhao, Tianyu Liu, Shuai Bai, Junyang Lin, Chang Zhou, and Baobao Chang. An image is worth 1/2 tokens after layer 2: Plug-and-play inference acceleration for large vision-language models. In *European Conference on Computer Vision*, pp. 19–35. Springer, 2024a.

Zhe Chen, Weiyun Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Erfei Cui, Jinguo Zhu, Shenglong Ye, Hao Tian, Zhaoyang Liu, et al. Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling. *arXiv preprint arXiv:2412.05271*, 2024b.

Zhe Chen, Weiyun Wang, Hao Tian, Shenglong Ye, Zhangwei Gao, Erfei Cui, Wenwen Tong, Kongzhi Hu, Jiapeng Luo, Zheng Ma, et al. How far are we to gpt-4v? closing the gap to commercial multimodal models with open-source suites. *Science China Information Sciences*, 67 (12):220101, 2024c.

Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, et al. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 24185–24198, 2024d.

Yihe Deng, Hritik Bansal, Fan Yin, Nanyun Peng, Wei Wang, and Kai-Wei Chang. Openvlthinker: Complex vision-language reasoning via iterative sft-rl cycles. *Advances in Neural Information Processing Systems*, 38:123817–123846, 2026.

Yifan Du, Yuqi Huo, Kun Zhou, Zijia Zhao, Haoyu Lu, Han Huang, Xin Zhao, Bingning Wang, Weipeng Chen, and Ji-Rong Wen. Exploring the design space of visual context representation in video mllms. In *International Conference on Learning Representations*, volume 2025, pp. 14061–14079, 2025.

Xingyu Fu, Yushi Hu, Bangzheng Li, Yu Feng, Haoyu Wang, Xudong Lin, Dan Roth, Noah A Smith, Wei-Chiu Ma, and Ranjay Krishna. Blink: Multimodal large language models can see but not perceive. In *European Conference on Computer Vision*, pp. 148–166. Springer, 2024.

Hang He, Chuhuai Yue, Chengqi Dong, Chengcheng Wan, Ting Su, Haiying Sun, Jiajun Chai, Xiaohan Wang, and Guojun Yin. Vistahop: Benchmarking multi-hop visual reasoning for visual deepsearch. *arXiv preprint arXiv:2606.03273*, 2026.

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.

Yizhang Jin, Jian Li, Tianjun Gu, Yexin Liu, Bo Zhao, Jinxiang Lai, Zhenye Gan, Yabiao Wang, Chengjie Wang, Xin Tan, et al. Efficient multimodal large language models: A survey. *Visual Intelligence*, 3(1):27, 2025.

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.

Moshe Kimhi, Nimrod Shabtay, Raja Giryes, Chaim Baskin, and Eli Schwartz. Cares: Context-aware resolution selector for vlms. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2243–2256, 2026.

Jewon Lee, Wooksu Shin, Seungmin Yang, Ki-Ung Song, DongUk Lim, Jaeyeon Kim, Tae-Ho Kim, and Bo-Kyeong Kim. Ergo: efficient high-resolution visual understanding for vision-language models. In *International Conference on Learning Representations*, volume 2026, pp. 8750–8769, 2026.

Geng Li and Yuxin Peng. Dicobench: Benchmarking multi-image fine-grained perception via differential and commonality visual cues. In *European Conference on Computer Vision*, pp. 169–184. Springer, 2026.

Kevin Li, Sachin Goyal, Joao D Semedo, and Zico Kolter. Inference optimal vlms need fewer visual tokens and more parameters. In *International Conference on Learning Representations*, volume 2025, pp. 96066–96083, 2025a.

Tenghui Li, Guoxu Zhou, Xuyang Zhao, and Qibin Zhao. Scaling capability in token space: An analysis of large vision language model. *Journal of Machine Learning Research*, 26(253):1–61, 2025b.

Anjie Liu, Ziqin Gong, Yan Song, Yuxiang Chen, Xiaolong Liu, Hengtong Lu, Kaike Zhang, Chen Wei, and Jun Wang. The perceptual bandwidth bottleneck in vision-language models: Active visual reasoning via sequential experimental design. *arXiv preprint arXiv:2605.01345*, 2026a.

Wenzhuo Liu, Weijie Yin, Fei Zhu, Shijie Ma, Haiyang Guo, Yi Chen, Xiao-Hui Li, Xiao Liang, Chao Feng, and Cheng-Lin Liu. One patch doesn't fit all: Adaptive patching for native-resolution multimodal large language models. In *International Conference on Learning Representations*, volume 2026, pp. 27594–27608, 2026b.

Zhijian Liu, Ligeng Zhu, Baifeng Shi, Zhuoyang Zhang, Yuming Lou, Shang Yang, Haocheng Xi, Shiyi Cao, Yuxian Gu, Dacheng Li, et al. Nvila: Efficient frontier visual language models. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4122–4134. IEEE, 2025.

Junwei Luo, Yingying Zhang, Xue Yang, Kang Wu, Qi Zhu, Lei Liang, Jingdong Chen, and Yansheng Li. When large vision-language model meets large remote sensing imagery: Coarse-to-fine text-guided token pruning. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9206–9217. IEEE, 2025a.

Weiqing Luo, Zhen Tan, Yifan Li, Xinyu Zhao, Kwonjoon Lee, Behzad Dariush, and Tianlong Chen. Task-aware resolution optimization for visual large language models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 15767–15781, 2025b.

Fawaz Sammani, Tzoulio Chamiti, and Nikos Deligiannis. On test-time scaling for vision-language models. In *European Conference on Computer Vision*, pp. 168–185. Springer, 2026.

Yuzhang Shang, Mu Cai, Bingxin Xu, Yong Jae Lee, and Yan Yan. Llava-prumerge: Adaptive token reduction for efficient large multimodal models. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 22857–22867. IEEE, 2025.

Haozhan Shen, Kangjia Zhao, Tiancheng Zhao, Ruochen Xu, Zilun Zhang, Mingwei Zhu, and Jianwei Yin. Zoomeye: Enhancing multimodal llms with human-like zooming capabilities through tree-based image exploration. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 6613–6629, 2025.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling llm test-time compute optimally can be more effective than scaling model parameters. *arXiv preprint arXiv:2408.03314*, 2024.

Wenzhang Sun, Chunfeng Wang, Xiangchen Yin, Yujia Chen, Hao Li, and Kun Zhan. Stable curves, unstable items: Item-level scaling heterogeneity in video llms. *arXiv preprint arXiv:2608.07014*, 2026.

Swabha Swayamdipta, Roy Schwartz, Nicholas Lourie, Yizhong Wang, Hannaneh Hajishirzi, Noah A Smith, and Yejin Choi. Dataset cartography: Mapping and diagnosing datasets with training dynamics. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 9275–9293, 2020.

Shengbang Tong, Zhuang Liu, Yuexiang Zhai, Yi Ma, Yann LeCun, and Saining Xie. Eyes wide shut? exploring the visual shortcomings of multimodal llms. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9568–9578. IEEE, 2024.

Sang Truong, Yuheng Tu, Rylan Schaeffer, and Sanmi Koyejo. Item response scaling laws: A measurement theory approach for efficient and generalizable neural scaling estimation. *arXiv preprint arXiv:2606.07616*, 2026.

Pavan Kumar Anasosalu Vasu, Fartash Faghri, Chun-Liang Li, Cem Koc, Nate True, Albert Antony, Gokula Santhanam, James Gabriel, Peter Grasch, Oncel Tuzel, et al. Fastvlm: Efficient vision encoding for vision language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19769–19780, 2025.

Feng Wang, Yaodong Yu, Wei Shao, Yuyin Zhou, Alan Yuille, and Cihang Xie. Scaling laws in patchification: An image is worth 50,176 tokens and more. *Proceedings of machine learning research*, 267:65278, 2025a.

Haochen Wang, Xiangtai Li, Zilong Huang, Anran Wang, Jiacong Wang, Tao Zhang, Sule Bai, Zijian Kang, Jiashi Feng, Zhuochen Wang, et al. Traceable evidence enhanced visual grounded reasoning: Evaluation and method. In *International Conference on Learning Representations*, volume 2026, pp. 148769–148794, 2026a.

Peiqi Wang, ShengYun Peng, Xuewen Zhang, Hanchao Yu, Yibo Yang, Lifu Huang, Fujun Liu, and Qifan Wang. Inference compute-optimal video vision language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2345–2374, 2025b.

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.

Pengcheng Wang, Zhiquan Wang, Jayoung Lee, Zhuoyan Xu, Ran Xu, Saurabh Bagchi, Yin Li, and Somali Chaterji. Look less, think faster: Joint token-compute adaptation for multimodal llms. *arXiv preprint arXiv:2607.20357*, 2026b.

Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, et al. Internvl3. 5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *arXiv preprint arXiv:2508.18265*, 2025c.

Wenbin Wang, Liang Ding, Minyan Zeng, Xiabin Zhou, Li Shen, Yong Luo, Wei Yu, and Dacheng Tao. Divide, conquer and combine: A training-free framework for high-resolution image perception in multimodal large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 7907–7915, 2025d.

Wenbin Wang, Yongcheng Jing, Liang Ding, Yingjie Wang, Li Shen, Yong Luo, Bo Du, and Dacheng Tao. Retrieval-augmented perception: High-resolution image perception meets visual rag. *arXiv preprint arXiv:2503.01222*, 2025e.

Penghao Wu and Saining Xie. V*: Guided visual search as a core mechanism in multimodal llms. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13084–13094. IEEE, 2024.

Senqiao Yang, Junyi Li, Xin Lai, Jinming Wu, Wei Li, Zejun MA, Bei Yu, Hengshuang Zhao, and Jiaya Jia. Visionthink: Smart and efficient vision language model via reinforcement learning. *Advances in Neural Information Processing Systems*, 38:95187–95227, 2026.

Shukang Yin, Chaoyou Fu, Sirui Zhao, Ke Li, Xing Sun, Tong Xu, and Enhong Chen. A survey on multimodal large language models. *National Science Review*, 11(12), November 2024. ISSN 2053-714X. doi: 10.1093/nsr/nwae403. URL http://dx.doi.org/10.1093/nsr/nwae403.

YiFan Zhang, Huanyu Zhang, Haochen Tian, Chaoyou Fu, Shuangqing Zhang, Junfei Wu, Feng Li, Kun Wang, Qingsong Wen, Zhang Zhang, et al. Mme-realworld: Could your multimodal llm challenge high-resolution real-world scenarios that are difficult for humans? In *International Conference on Learning Representations*, volume 2025, pp. 89655–89701, 2025a.

Yuan Zhang, Chun-Kai Fan, Junpeng Ma, Wenzhao Zheng, Tao Huang, Kuan Cheng, Denis Gudovskiy, Tomoyuki Okuno, Yohei Nakata, Kurt Keutzer, et al. Sparsevlm: Visual token sparsification for efficient vision-language model inference. *arXiv preprint arXiv:2410.04417*, 2024.

Yusen Zhang, Wenliang Zheng, Aashrith Madasu, Peng Shi, Ryo Kamoi, Hao Zhou, Zhuoyang Zou, Shu Zhao, Sarkar Snigdha Sarathi Das, Vipul Gupta, et al. Hrscene: How far are vlms from effective high-resolution image understanding? In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 22922–22933. IEEE, 2025b.

Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Hao Tian, Yuchen Duan, Weijie Su, Jie Shao, et al. Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models. *arXiv preprint arXiv:2504.10479*, 2025.

APPENDIX CONTENTS

A Extended Experimental Setup **15**

A.1 Models and Visual Frontends . . . 15

A.1.1 InternVL Family . . . 15

A.1.2 QwenVL Family . . . 16

A.1.3 Evaluated Models . . . 16

A.2 Benchmarks . . . 16

A.3 Input Sizes and Visual Token Counting . . . 17

A.4 Units of Analysis and Scale of the Study . . . 18

A.5 Decoding and Scoring . . . 18

B Fitting and Validating the Separable Law **18**

B.1 Fitting Procedure . . . 19

B.2 Per-Group Fits and Diagnostics . . . 19

B.3 Comparison of Functional Forms . . . 20

B.4 Sampling Uncertainty and Precision Weighting . . . 23

B.5 Held-Out Prediction . . . 24

B.6 A Bounded Version of the Law . . . 25

B.7 Parameter Uncertainty and Sensitivity . . . 27

B.8 Input Coverage and Token Aggregation . . . 27

C Finding 1: The Cognitive Ceiling **28**

C.1 Task-Level Floors . . . 28

C.2 Robustness of Regime Assignments . . . 29

C.3 Persistent Errors at High Compute . . . 29

C.4 Domain and Skill Taxonomy . . . 32

C.5 Fits for Each Domain–Skill Pairing . . . 33

D Finding 2: Architectural Divergence **35**

D.1 Exponents by Family, Series, and Task . . . 35

D.2 Token Ranges of the Two Frontends . . . 36

E Finding 3: From Laws to Practice **39**

E.1 Cost Measures and Cost-Law Fits . . . 39

E.2 Derivation of the VPE Allocation . . . 40

E.3 Held-Out Allocation . . . 44

E.4 In-Sample Allocation Analysis . . . 44

E.5 Benchmark Profiles by Family . . . 47

F Case Studies **47**

A Extended Experimental Setup

This section traces each measurement from a model and image to a cell-level error rate. We describe the visual frontends and evaluated models (Appendix A.1), the benchmarks (Appendix A.2), and how input size determines the visual token count (Appendix A.3). We then define each experimental cell, corresponding to one model, input size, and benchmark, and report the scale of the study (Appendix A.4), before detailing decoding and scoring (Appendix A.5).

A.1 Models and Visual Frontends

Our experiments cover 26 models in five InternVL and QwenVL series. InternVL produces a fixed number of tokens per tile, whereas QwenVL varies the token count with input size. This distinction motivates the comparison of β in §4.1. Appendices A.1.1 and A.1.2 describe the frontends, and Appendix A.1.3 lists the models and defines N as the parameter count of the language backbone.

A.1.1 InternVL Family

Since InternVL 1.5, the InternVL series has turned an image into visual tokens in the same way, by cutting it into tiles of a fixed size. Later versions mainly changed the training recipe and the efficiency of inference. Table 3 summarizes these changes from InternVL 1.0 to InternVL3.5.

Table 3: Evolution of the InternVL series, focusing on how each version turns an image into tokens.

Version	Input strategy	Visual token mechanism	Effect on visual token count
InternVL 1.0	Fixed input size	ViT features projected to the language backbone	Fixed count
InternVL 1.5	Dynamic tiling	Up to 40 tiles of 448×448	Changes in whole tiles
InternVL2.5	Dynamic tiling	Pixel unshuffle to 256 tokens per tile	Changes in whole tiles
InternVL3	Dynamic tiling	256 tokens per tile, V2PE positions	Changes in whole tiles
InternVL3.5	Dynamic tiling	256 tokens per tile, ViR only in Flash variants	Changes in whole tiles

Establishing the tiling mechanism (InternVL 1.0→InternVL 1.5). InternVL 1.0 used inputs of a fixed size and encoded them with a vision transformer (ViT) (Chen et al., 2024d). InternVL 1.5 introduced dynamic tiling, which splits an image into as many as 40 tiles of 448×448 pixels (Chen et al., 2024c). The ViT encodes each tile separately, and the resulting features are concatenated before they enter the language backbone. InternVL 1.5 also introduced pixel unshuffle, which compresses the 1,024 ViT output tokens of each tile into 256 visual tokens. All later versions keep this design.

Scaling up and changing the training recipe (InternVL2.5→InternVL3). InternVL2.5 offered a systematic range of model sizes and improved performance through stronger data filtering and more training data (Chen et al., 2024b). InternVL3 replaced the usual two-stage recipe, in which a ViT is attached to a pretrained language model, with native multimodal pre-training (Zhu et al., 2025). The language backbone and InternViT start from pretrained weights, and native multimodal pre-training then trains them jointly on text-only and multimodal data. InternVL3 also introduced Variable Visual Position Encoding (V2PE) to handle long multimodal contexts.

Efficiency features (InternVL3.5). InternVL3.5 adds two features for efficient inference (Wang et al., 2025c). The Visual Resolution Router (ViR), used in the InternVL3.5-Flash variants, chooses a compression rate for the visual tokens of each image. Decoupled Vision-Language Deployment (DvD) places the ViT and the language backbone on separate GPUs. We evaluate the standard InternVL3.5 models without ViR, so every tile yields 256 visual tokens, as in InternVL2.5 and InternVL3.

Tiling setting in our evaluation. We allow at most 12 tiles per image in all three InternVL series. When an image is split into more than one tile, the model also receives a smaller copy of the whole image as one extra tile. An image therefore uses at most 13 tiles, or 3,328 visual tokens. The tile grid is chosen to match the aspect ratio of the input, so for most questions the number of tiles stays the same across our input sizes (§4.2).

A.1.2 QWENVL FAMILY

The QwenVL series moved from a fixed input size to dynamic resolution in Qwen2-VL, which resizes the whole image and produces a visual token count that changes with the image size. Later versions kept this frontend and changed the vision encoder and the training. Table 4 summarizes these changes from Qwen-VL to Qwen3-VL.

Table 4: Evolution of the QwenVL series, focusing on how each version turns an image into tokens.

Version	Input strategy	Visual token mechanism	Effect on visual token count
Qwen-VL	Fixed 448×448 input	Cross-attention adapter to a fixed number of tokens	Fixed count
Qwen2-VL	Dynamic resolution	2D-RoPE and 2×2 MLP patch merger	Changes with input size in small steps
Qwen2.5-VL	Dynamic resolution	Window attention, same patch merger	Changes with input size in small steps
Qwen3-VL	Dynamic resolution	SigLIP-2, DeepStack, same patch merger	Changes with input size in small steps

From fixed to dynamic resolution (Qwen-VL→Qwen2-VL). The original Qwen-VL resized every image to 448×448 and compressed the ViT output into a fixed number of visual tokens through a cross-attention adapter (Bai et al., 2023). Input size therefore had little effect on what reached the language backbone. Qwen2-VL redesigned this frontend in three ways (Wang et al., 2024). It replaced absolute position embeddings with two-dimensional rotary position embedding (2D-RoPE), which lets the ViT accept images of any size. It compressed tokens with a 2×2 MLP patch merger instead of cross-attention. It introduced Multimodal Rotary Position Embedding (M-RoPE), which splits position information into time, height, and width so that one model handles both images and video. As a result, the visual token count changes with the input size in small steps set by the patch grid, much finer than the whole-tile steps of InternVL.

Balancing efficiency and scale (Qwen2-VL→Qwen2.5-VL). Dynamic resolution makes the ViT expensive on large inputs. Qwen2.5-VL reduced this cost with window attention and kept the dynamic resolution frontend and the patch merger (Bai et al., 2025b). It also trained the ViT from scratch and expanded the pre-training data. The way input size maps to visual token count stays the same as in Qwen2-VL.

Deeper fusion of vision and language (Qwen2.5-VL→Qwen3-VL). Qwen3-VL replaced the custom ViT with SigLIP-2 and added three architectural changes (Bai et al., 2025a). DeepStack feeds ViT features from several depths into the matching layers of the language backbone through light residual connections. An interleaved version of M-RoPE spreads the time, horizontal, and vertical position components across frequency bands. Explicit text timestamp tokens mark time in video. The dynamic resolution frontend and the 2×2 MLP merger stay in place, so the visual token count still changes with the input size in small steps.

A.1.3 EVALUATED MODELS

Table 5 lists the 26 models we evaluate, and the same definition of N applies to both families.

Definition of N. N is the exact parameter count of the language backbone of each model. We compute it from the architecture specification, exclude the vision encoder and projector, and do not rely on the size in the model name. Within a series, larger models may also change the vision encoder, the training data, and the training recipe, so differences in N carry these changes as well.

A.2 BENCHMARKS

We evaluate on four benchmarks built from high-resolution images.

HR-Bench. HR-Bench tests fine-grained perception on images at 4K and 8K resolution (Wang et al., 2025d). It targets details that are often lost when an image is downsampled, and it has two subtasks, perception of a single instance and perception across instances. The 4K and 8K releases contain the same 800 questions.

Table 5: Evaluated model series and their visual frontends, 26 models in total. Sizes follow the model names, and N is computed as described above.

Family	Series	Evaluated sizes	Visual frontend
InternVL	InternVL2.5	1B, 2B, 4B, 8B, 26B, 38B	Dynamic tiling
InternVL	InternVL3	1B, 2B, 8B, 9B, 14B, 38B	Dynamic tiling
InternVL	InternVL3.5	1B, 2B, 4B, 8B, 14B, 38B	Dynamic tiling
QwenVL	Qwen2.5-VL	3B, 7B, 32B, 72B	Dynamic resolution
QwenVL	Qwen3-VL	2B, 4B, 8B, 32B	Dynamic resolution

MME-RealWorld. MME-RealWorld is a large benchmark, annotated by humans, that tests multimodal models on high-resolution images of real-world scenes (Zhang et al., 2025a). It contains 13,366 images and more than 29,000 multiple-choice questions. The full benchmark has 43 subtasks in five fields, namely OCR, remote sensing, charts and tables, monitoring, and autonomous driving. We use the MME-RealWorld Lite split, which contains 1,782 unique question IDs in our evaluation pipeline.

TreeBench. TreeBench is a diagnostic benchmark for visual reasoning grounded in the image (Wang et al., 2026a). Its questions focus on small targets in cluttered scenes and ask about spatial hierarchy, interactions between objects, and visual evidence that can be traced in the image. It contains 405 questions on high-resolution images from the SA-1B collection.

V* Bench. V* Bench tests whether a model can locate and reason about tiny details in crowded scenes (Wu & Xie, 2024). It has two question types, recognizing object attributes and judging spatial relations. It contributes 191 questions in our evaluation pipeline.

Table 6: Native image sizes and evaluated input sizes.

Benchmark	Native size ($W \times H$)	Input sizes (long edge, px)
HR-Bench-4K	$\approx 4023 \times 3503$	$224, 336, 512, 768, 1024, 2048$
HR-Bench-8K	$\approx 5727 \times 4430$	4096
MME-RealWorld	$\approx 2000 \times 1500$	$224, 336, 512, 768, 1024, 2048$
TreeBench	$\approx 2152 \times 1615$	$224, 336, 512, 768, 1024, 2048$
V* Bench	$\approx 2246 \times 1582$	$224, 336, 512, 768, 1024, 2048$

A.3 Input Sizes and Visual Token Counting

Input sizes. We resize every image so that its longer edge equals one of a fixed set of input sizes, keeping the aspect ratio. For an input size b and a native size (H, W), the scale factor is $s = b/\max(H, W)$ and the resized image has size $(H', W') = \mathrm{round}(sH, sW)$. All four benchmarks share six input sizes, namely 224, 336, 512, 768, 1024, and 2048 px. For HR-Bench, the six shared sizes use the 4K release, and an additional 4096 px setting uses the separately released 8K split. The two releases contain the same 800 questions but assign them different question IDs. We keep their observations separate for scoring and for building trajectories, and pair matching questions across the two releases when resampling images. The input sizes cover three ranges. The sizes 224 and 336 px match standard ViT inputs, 512 to 1024 px are common in deployment, and 2048 and 4096 px test very high resolution. At 1024 px and below almost every image is shrunk, while at 2048 px many MME-RealWorld images are smaller than the target and are enlarged. We also ran each benchmark at native size but leave this setting out of all fits. At native size the longer edge differs from image to image, so the input size is not fixed and R cannot be compared across questions. Without it, the grid has 650 cells across the five model series.

Measured visual token counts. The same input size produces different numbers of visual tokens in different models, so we measure R instead of computing it from the input size. Every record logs the number of visual tokens that enter the language backbone, and the R of a cell is the mean over

its records. For QwenVL we read this count from the image tokens in the processed input, after the patch merger. For InternVL3 and InternVL3.5 we count the image placeholder tokens that the model inserts into the prompt, one for each visual token. For InternVL2.5 we multiply the logged tile count, including the extra tile for the whole image, by 256 tokens per tile. These measurements also reveal how strongly each frontend changes its visual token count with input size.

A.4 Units of Analysis and Scale of the Study

Cells and groups. A cell is one model evaluated at one input size on one benchmark, and it is the unit of fitting. For bootstrap resampling, the unit is an image together with all its questions across every evaluated configuration, since different cells share the same images. Each benchmark is run at six input sizes, and HR-Bench adds the 4096 px setting, which gives $4 \times 6 + 1 = 25$ combinations of benchmark and input size. The 26 models therefore give $26 \times 25 = 650$ cells. The cells form 20 series–benchmark groups. A QwenVL group contains 24 cells, or 28 on HR-Bench, and an InternVL group contains 36, or 42 on HR-Bench. Each group varies along two axes. Along N it covers the backbone sizes of the series, six for each InternVL series and four for each QwenVL series, and along R it covers six or seven input sizes.

Scale of the study. We recorded $537{,}940$ generations, of which 691 failed during inference. Of the remaining $537{,}249$ scored records, 468 lack usable probabilities for the correct option, which leaves $536{,}781$ records with both signals across the 650 cells. The question-level analysis in §3.2 uses all scored records and keeps the last record when a question ID repeats within a configuration. This gives $515{,}938$ observations that form $19{,}890$ trajectories, where a trajectory is one question evaluated by one model series across its configurations. Each series has $3{,}978$ trajectories, from $1{,}600$ HR-Bench question IDs, $1{,}782$ MME-RealWorld questions, 405 TreeBench questions, and 191 V* Bench questions. The $1{,}600$ HR-Bench IDs consist of 800 from the 4K release, evaluated at the six shared sizes, and 800 from the 8K release, evaluated only at 4096 px.

A.5 Decoding and Scoring

Decoding. We run the public instruction-tuned models in BF16 without modifying them. Prompting is zero-shot, and the system and user prompts ask the model to answer with a single capital option letter. Decoding is greedy, meaning the model always takes the most probable next token without sampling or beam search, so the temperature setting has no effect. Of the 104 runs, one for each model and benchmark, 96 allow at most 48 new tokens with a repetition penalty of 1.05. The other 8 runs, Qwen2.5-VL 3B, 7B, and 32B and InternVL2.5-38B on HR-Bench and MME-RealWorld, allow at most 32 new tokens with no repetition penalty. In total, 429 generations reach the token limit, which is about 0.08% of all generations. Of these, 304 reach 48 tokens and 125 reach 32 tokens, and 51 contain no option letter that the parser can extract.

Scoring. Each scored record carries two predictions. The text prediction is parsed from the generated answer, and the logit prediction is the option with the highest logged confidence. The parser first looks for an answer in parentheses or after an explicit answer phrase, and otherwise takes the last valid option letter that stands alone. Option confidences are normalized over the scored answer labels, and the decoding step at which they are read follows each model's implementation. The two predictions agree on $518{,}644$ of the $536{,}781$ records with both signals, or 96.6%, so we use text accuracy as the main signal. The error in Eq.(3) is $L = 1 -$ text accuracy, computed for each cell over its scored records. The question-level robustness analysis also uses the logged probability of the correct option, p_{gold}. We use this normalized score as a relative measure and do not assume that it is calibrated. For chance normalization we use the actual number of options of each question, after checking that the prompt and the log agree. This normalization leaves the error used in the Separable fits unchanged.

B Fitting and Validating the Separable Law

We describe the fitting procedure and group-level diagnostics (Appendices B.1 and B.2), then compare the Separable law with coupled alternatives (Appendix B.3). We test robustness to image resampling, precision weighting, and held-out configurations (Appendices B.4 and B.5). Because

some edge predictions fall outside $[0, 1]$, we also fit a bounded law and verify that the decisions and exponent ordering are unchanged (Appendix B.6). Finally, we quantify parameter uncertainty (Appendix B.7) and test sensitivity to the HR-Bench 4096 px setting and cell-level token aggregation (Appendix B.8).

B.1 Fitting Procedure

Normalization. We divide both axes by a common reference point, $n = N/\widetilde{N}$ and $r = R/\widetilde{R}$, where $\widetilde{N}$ and $\widetilde{R}$ are the geometric means of N and R over all 650 cells. We write the fitted amplitudes at this reference point as A_n and B_r, so they give the error contributed by each axis at the same point and can be compared across groups. The coefficients A and B of Eq.(3), which the VPE allocation in §5.1 uses, follow as $A = A_n \widetilde{N}^{\alpha}$ and $B = B_r \widetilde{R}^{\beta}$, with N in billions of parameters as in the cost law. The choice of reference point changes only A_n and B_r, and it leaves the exponents, the floor, every predicted error, and the VPE allocation unchanged.

Objective. We fit each group separately by minimizing the unweighted squared error over its cells, so that each configuration counts once,

$$\min_{\alpha,\beta,e_\infty,A_n,B_r} \sum_{i \in g} \Big[L_i - e_\infty - A_n\, n_i^{-\alpha} - B_r\, r_i^{-\beta} \Big]^2, \qquad A_n, B_r \geq 0. \tag{10}$$

Optimization. We optimize the exponents over $(\log \alpha, \log \beta)$ with the Nelder–Mead method, starting from 16 points given by $\alpha_0 \in \{0.1, 0.3, 0.6, 1.0\}$ and $\beta_0 \in \{0.05, 0.15, 0.3, 0.6\}$. At each step, we search e_∞ over a grid of 55 values in $[0.001, 0.999]$. For each candidate e_∞, the nonnegative amplitudes A_n and B_r have a closed-form solution, which we find by checking the four combinations of amplitudes held at zero or left free. A soft quadratic penalty keeps both exponents within $[0.05, 10]$. Nelder–Mead uses a step of 0.4, at most 300 iterations, and tolerances $\mathrm{ftol} = 10^{-8}$ and $\mathrm{xtol} = 10^{-7}$. This objective weights all configurations equally, and its point estimates serve as the baseline for the uncertainty and sensitivity analyses.

B.2 Per-Group Fits and Diagnostics

Per-group parameters. Table 7 reports the fitted parameters, SSE, and R^2 for all 20 groups. Pooled over the 650 cells, R^2 is 0.898 and SSE is 1.088, and the median group has an SSE of 0.024 and an R^2 of 0.892. These numbers describe the point fits, and Table 15 gives bootstrap intervals for α, β, and e_∞. A dagger marks the 9 groups whose β reaches the upper bound of the search range, all from InternVL, and a double dagger marks the 3 QwenVL groups whose α reaches that bound. Overall, the law provides a strong fit across the 20 groups.

Predicted versus observed. Figure 7 plots predicted against observed error for each group, with the same $[0, 0.82]^2$ axes in every panel so that the spread can be compared directly. Seventeen of the 20 groups reach $R^2 \geq 0.65$. The other three are the V* Bench groups of the three InternVL series, with R^2 of 0.594, 0.548, and 0.515. In these groups the measured visual token count spans the narrowest range, with the largest count only about 1.067 times the smallest. Thus, fit quality is high whenever the visual-token axis spans a substantive range.

Fitted curves. Figure 8 overlays the fitted Separable law on the observed errors of each group. β reaches its upper bound mainly in groups whose measured visual token count spans a narrow range. The curves therefore capture the observed scaling trends across language backbone sizes and visual-token quartiles.

Residual diagnostics. Figure 9 pools the residuals of all 650 cells. They center near zero, with a mean of -0.000 and a standard deviation of 0.041, but their tails are heavier than those of a Gaussian with the same mean and variance, with a sample excess kurtosis of 2.74. We show the Gaussian curve only as a visual reference, since the residuals of different cells need not be independent or identically distributed. The residual standard deviation is 0.046 for InternVL and 0.026 for QwenVL, ranging from 0.045 to 0.047 across the three InternVL series and from 0.023 to 0.028

Table 7: Per-group Separable-law point estimates and fit diagnostics. Amplitudes A_n and B_r are evaluated at the common reference point $\widetilde{N} = 6.60$B and $\widetilde{R} = 1{,}135$. † and ‡ mark β and α, respectively, at the upper bound of 10; n is the cell count.

Group		Separable-law parameters					Fit diagnostics		
Series	**Benchmark**	e_∞	A_n	α	B_r	β	**SSE**	n	R^2
InternVL2.5	HR-Bench	0.001	0.308	0.149	1.24	4.66	0.031	42	0.92
InternVL2.5	MME-RealWorld	0.001	0.527	0.058	11.41	$10.0^{\dagger}$	0.043	36	0.77
InternVL2.5	TreeBench	0.001	0.582	0.057	9.65	$10.0^{\dagger}$	0.030	36	0.73
InternVL2.5	V* Bench	0.001	0.089	0.381	56.31	$10.0^{\dagger}$	0.203	36	0.59
InternVL3	HR-Bench	0.001	0.262	0.183	1.05	3.87	0.025	42	0.94
InternVL3	MME-RealWorld	0.36	0.120	0.194	12.48	$10.0^{\dagger}$	0.043	36	0.76
InternVL3	TreeBench	0.06	0.506	0.071	8.57	$10.0^{\dagger}$	0.013	36	0.87
InternVL3	V* Bench	0.01	0.008	1.262	65.96	$10.0^{\dagger}$	0.231	36	0.55
InternVL3.5	HR-Bench	0.001	0.312	0.145	1.42	4.93	0.023	42	0.93
InternVL3.5	MME-RealWorld	0.46	0.008	1.203	13.78	$10.0^{\dagger}$	0.044	36	0.80
InternVL3.5	TreeBench	0.50	0.054	0.418	11.96	$10.0^{\dagger}$	0.010	36	0.83
InternVL3.5	V* Bench	0.001	0.057	0.465	62.21	$10.0^{\dagger}$	0.258	36	0.51
Qwen2.5-VL	HR-Bench	0.001	0.043	0.462	0.372	0.123	0.021	28	0.93
Qwen2.5-VL	MME-RealWorld	0.001	1.3×10^{-6}	$10.0^{\ddagger}$	0.599	0.075	0.010	24	0.93
Qwen2.5-VL	TreeBench	0.24	0.261	0.053	0.092	0.187	0.008	24	0.81
Qwen2.5-VL	V* Bench	0.001	3.8×10^{-6}	$10.0^{\ddagger}$	0.324	0.205	0.041	24	0.91
Qwen3-VL	HR-Bench	0.001	0.036	0.756	0.343	0.162	0.023	28	0.94
Qwen3-VL	MME-RealWorld	0.08	1.5×10^{-8}	$10.0^{\ddagger}$	0.468	0.110	0.005	24	0.96
Qwen3-VL	TreeBench	0.40	0.033	0.800	0.121	0.184	0.003	24	0.95
Qwen3-VL	V* Bench	0.001	0.057	0.589	0.230	0.323	0.023	24	0.96

across the two QwenVL series. Across five bins of observed error with equal width, the mean residuals are -0.037, -0.024, $+0.003$, $+0.006$, and $+0.014$. The law therefore slightly overpredicts error in the lowest bin and underpredicts it in the highest, which is the trend noted in §2.4. Overall, the residuals remain centered near zero across both families.

B.3 Comparison of Functional Forms

Protocol. To compare functional forms fairly, we fit every candidate to the same 650 cells, share its shape parameters across the 20 groups, and let the group coefficients vary. This gives Separable 62 parameters and an SSE of 1.221; fitting separate shape parameters to every group uses 100 parameters and reaches an SSE of 1.088 and $R^2 = 0.898$. Separable and Coupled use the same unconstrained group intercepts, nonnegative amplitudes, and exponents in $[0.05, 10]$, with Coupled reducing to Separable when $C_g = 0$. We treat BIC as a descriptive comparison rather than a formal model-selection test. This protocol isolates the effect of coupling under matched data and constraints.

Candidate forms. The alternatives below differ in how they combine backbone capacity and visual tokens.

$$\text{Multiplicative (Eq.(2)):} \quad e = e_{\infty,g} + K_g\, N^{-\alpha} R^{-\beta} \tag{11}$$

$$\text{Coupled:} \quad e = e_{\infty,g} + A_g\, N^{-\alpha} + B_g\, R^{-\beta} + C_g\, (NR)^{-\gamma} \tag{12}$$

$$\text{CES:} \quad e = e_{\infty,g} + K_g \big[w\, N^{\delta} + (1-w)\, R^{\delta} \big]^{-1/\delta} \tag{13}$$

$$p\text{-norm:} \quad e = \big[(A_g\, N^{-\alpha})^p + (B_g\, R^{-\beta})^p + C_g^p \big]^{1/p}. \tag{14}$$

The multiplicative form has the single product term of Eq.(2) (Li et al., 2025a), with nonnegative K_g. Coupled adds to Separable a nonnegative interaction amplitude C_g for each group and a shared exponent γ, which raises the parameter count from 62 to 83. The CES form (Arrow et al., 1961)

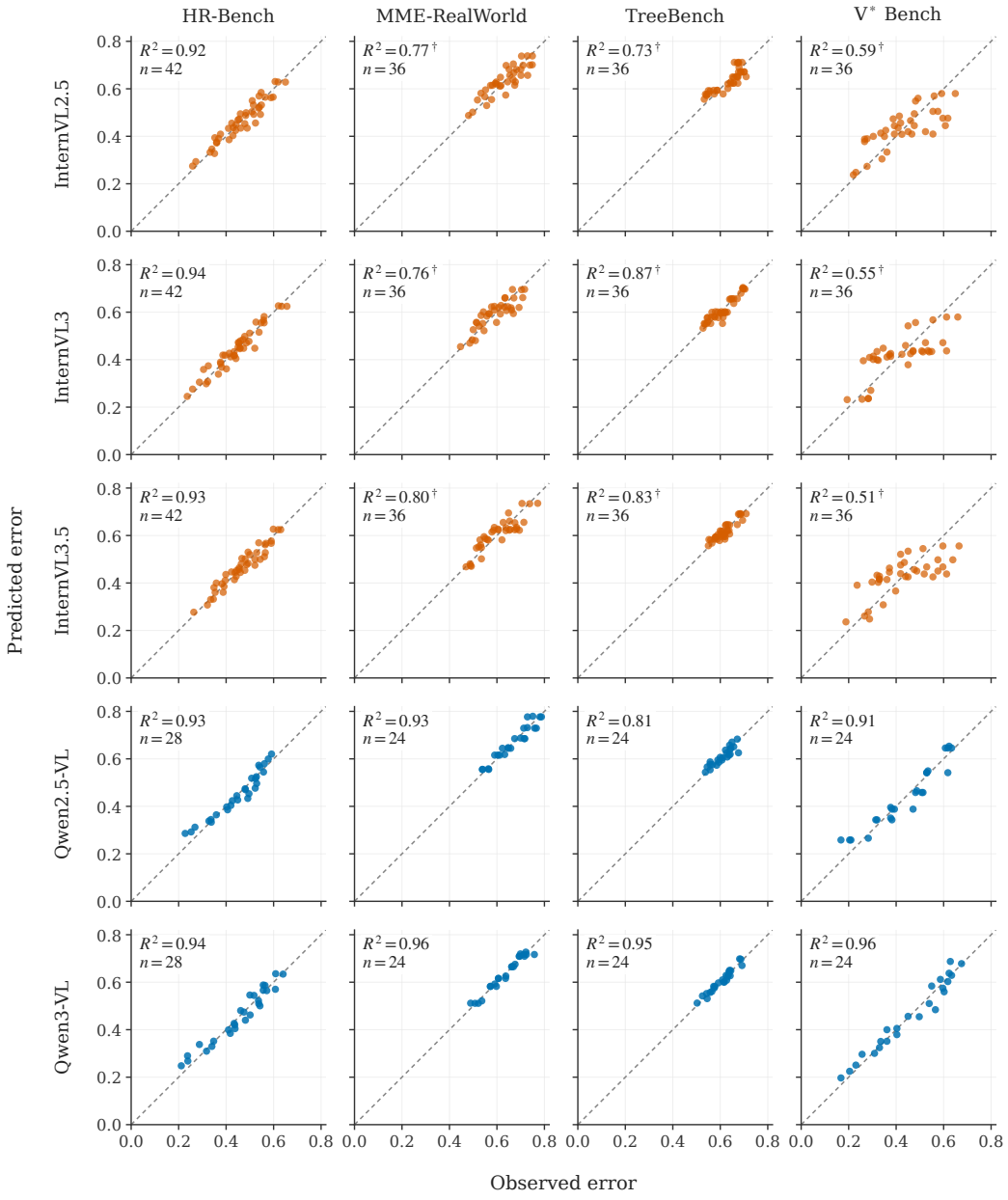


Figure 7: **Predicted versus observed error for each group.** Rows correspond to the five model series and columns to the four benchmarks. Each point is one (N, R) cell, all panels share $[0, 0.82]^2$ axes, and the dashed line marks perfect agreement, $y = x$. Orange denotes InternVL and blue denotes QwenVL; each panel reports its R^2 and cell count n. † marks groups whose fitted β reaches the search upper bound of 10. The three InternVL groups on V* Bench have the lowest R^2 values, consistent with their narrow visual-token ranges.

combines the two axes with nonnegative K_g and shared w and δ, searched over $0 < w < 1$ and $\delta \in [-8, 4]$. Multiplicative and CES use unconstrained group intercepts. The p-norm form has nonnegative A_g, B_g, and C_g, with $\alpha, \beta \in [0.05, 3]$ and $p \in [0.5, 15]$. Its C_g is a nonnegative floor inside the norm and takes the place of the intercept. At $p = 1$ it reduces to a Separable law with a nonnegative intercept, and as $p \to \infty$ it approaches $\max\{A_g N^{-\alpha}, B_g R^{-\beta}, C_g\}$, where error is set by the scarcest resource. Our fitter finds the group amplitudes by least squares on e^p and scores the

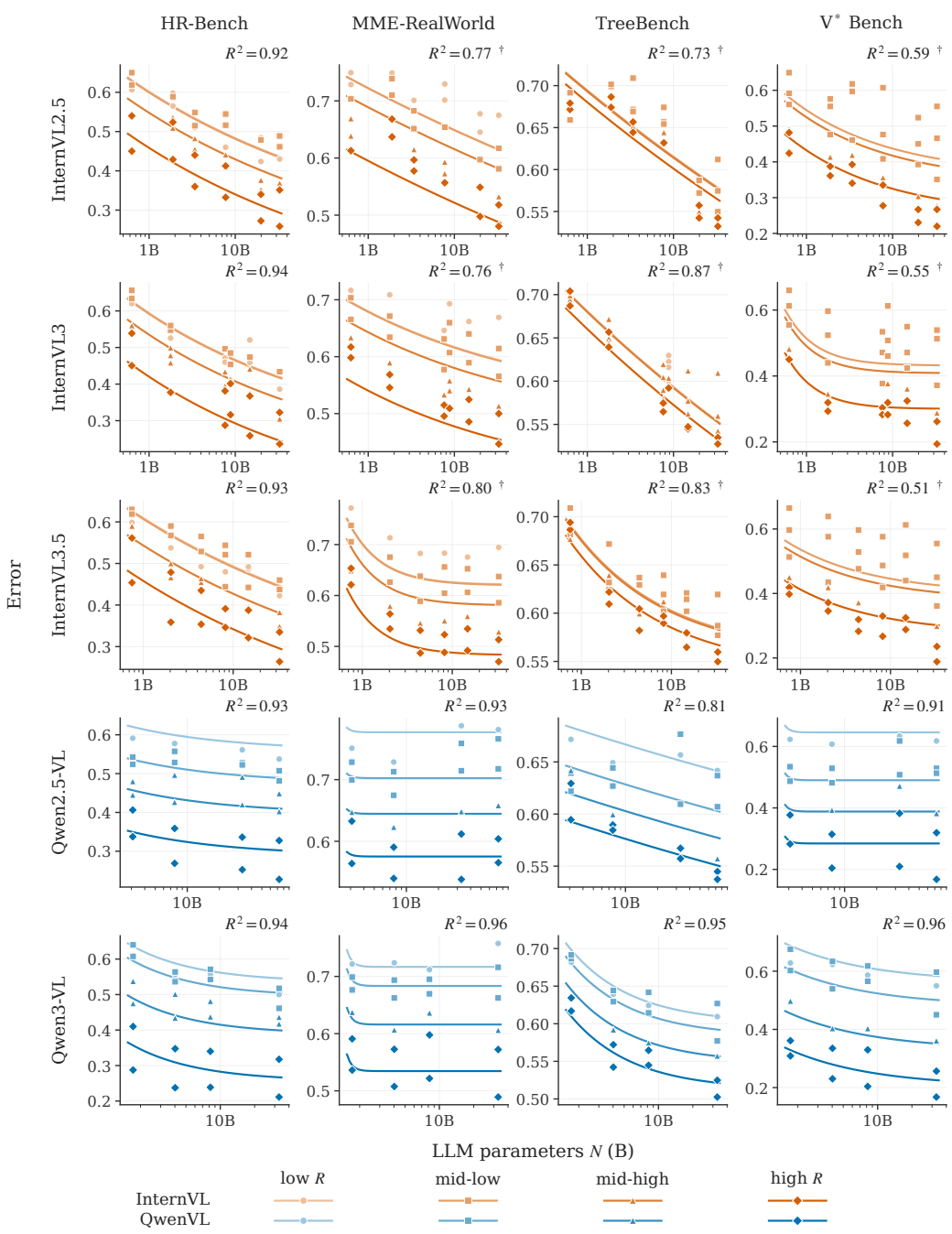


Figure 8: **Separable-law fits across model series and benchmarks.** Rows show model series and columns show benchmarks; each panel plots observed error against N on a logarithmic scale. Points are (N, R) cells, and curves are predictions at the median R of each visual-token quartile; color denotes family and shade or marker denotes quartile. † marks $\beta = 10$; Table 7 reports the parameter estimates.

result on the original error scale. The reported p-norm score is therefore the best fit this procedure finds, which may lie above the lowest SSE the p-norm form can reach.

Results. Table 8 reports the parameter count, pooled SSE, and BIC of each form, and we compute every percentage difference in SSE from the unrounded values. The multiplicative form has 53.2%

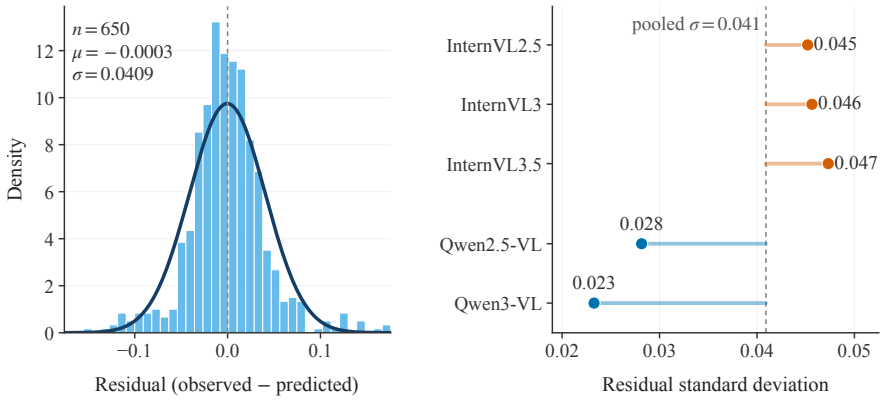


Figure 9: **Residual diagnostics for the per-group Separable-law fits. Left:** Distribution of all 650 residuals (observed minus predicted), with zero marked by the dashed line and a matched Gaussian shown for reference. **Right:** Residual standard deviation by model series; the dashed line gives the pooled value. Residuals are centered near zero, but InternVL shows greater dispersion than QwenVL.

higher SSE and, although it uses 20 fewer parameters, a BIC higher by 147.8. Its fitted exponents are $\alpha = 0.05$, at the lower bound, and $\beta = 0.40$. Coupled is the only alternative with lower training SSE, reducing it from 1.221 to 1.182, or 3.1%, but its 21 extra parameters raise BIC by 115.2. CES and the p-norm form have 73.9% and 56.8% higher SSE. The fitted p sits at its lower bound of 0.5, far from the large values at which error is set by the scarcest resource. Among the tested forms and under these constraints, the Separable law therefore gives the best balance between fit and parameter count.

Table 8: Comparison of functional forms on the same 650 cells, with exponents and shape parameters shared across groups. k counts the parameters of each fitted law.

Form	k	SSE	BIC	ΔSSE	ΔBIC
Separable	62	1.221	**−1834.3**	—	—
Multiplicative	42	1.870	−1686.4	+53.2%	+147.8
Coupled	83	**1.182**	−1719.0	−3.1%	+115.2
CES	42	2.123	−1604.0	+73.9%	+230.3
p-norm	63	1.913	−1535.6	+56.8%	+298.7

Per-group multiplicative comparison. Table 9 refits the multiplicative form with exponents specific to each group, using the same exponent range, nonnegative amplitudes, and floor grid as the Separable fits. It still fits worse than the Separable law in 19 of the 20 groups, with 15.2% higher pooled SSE, 1.254 against 1.088, and a median group R^2 of 0.840 against 0.892. The multiplicative form thus stays behind even when each group has its own exponents.

Table 9: Separable and multiplicative laws with exponents specific to each group.

Form	Pooled SSE	Median group R^2	Groups with lower SSE
Separable	**1.088**	**0.892**	**19/20**
Multiplicative	1.254	0.840	1/20

B.4 Sampling Uncertainty and Precision Weighting

Paired image bootstrap. To test whether correlated questions change the comparison between Separable and Coupled, we resample images within each benchmark while preserving all associ-

ated questions, repeated records, configurations, and model series. The 650 cells are rebuilt from 536,781 records covering 3,975 question IDs, 3,175 distinct questions, and 2,253 images; matched HR-Bench questions receive the same weight in the 4K and 8K releases. Every replicate recomputes the cell error and mean visual token count, and we run 500 in-sample replicates and 200 full held-out replicates with seed 20260909. Separable and Coupled use shared exponents, matched constraints, and verified optimization, with Coupled containing Separable at $C_g = 0$. Coupled lowers unweighted SSE by 3.15%, but the BIC difference is 115.23, with a 95% bootstrap interval of $[93.95, 123.10]$; keeping only the last valid record per question and configuration gives the same preference. The paired bootstrap therefore confirms that the preference for Separable is stable under correlated image sampling.

Precision-weighted comparison. To test whether unequal benchmark sizes affect the comparison, we give more precise cell estimates greater weight. For cell j, let a_{ij} be the number of valid records and z_{ij} the number of errors contributed by image i, with $n_j = \sum_i a_{ij}$, $\widehat{e}_j = \sum_i z_{ij}/n_j$, and m_s images in benchmark s. We estimate the cell variance by

$$\widehat{v}_j = \frac{1}{n_j^2} \sum_s \frac{m_s}{m_s - 1} \left\{ \sum_{i \in s} (z_{ij} - \widehat{e}_j a_{ij})^2 - \frac{\left[\sum_{i \in s} (z_{ij} - \widehat{e}_j a_{ij})\right]^2}{m_s} \right\}. \tag{15}$$

We fit both forms by minimizing $Q = \sum_{j=1}^{650} w_j (\widehat{e}_j - \widehat{L}_j)^2$, where $w_j \propto \widehat{v}_j^{-1}$ is scaled to a mean of one and kept at its original value in every bootstrap refit. The paired bootstrap retains correlation across configurations, and Table 10 summarizes both experiments. Its BIC score combines the logarithm of the Coupled-to-Separable objective ratio over the 650 cells with the penalty for Coupled's 21 additional parameters. Coupled lowers the weighted objective by 4.38%, while the BIC difference is 106.91, with a 95% bootstrap interval of $[89.65, 116.90]$. The precision-weighted experiment therefore confirms the preference for Separable when more reliable cells receive greater influence.

Table 10: Paired image-bootstrap comparison with shared exponents (500 replicates). Cells give point estimates and 95% percentile intervals; differences are Coupled minus Separable. Q denotes unweighted or precision-weighted SSE, and positive ΔBIC favors Separable.

Objective	ΔQ	Relative ΔQ (%)	ΔBIC
Unweighted	-0.0384	-3.15	**115.23**
	$(-0.0823, -0.0277)$	$(-6.27, -1.97)$	$(93.95, 123.10)$
Weighted by precision	-0.0435	-4.38	**106.91**
	$(-0.0758, -0.0314)$	$(-6.88, -2.90)$	$(89.65, 116.90)$

B.5 Held-Out Prediction

Held-out configurations. To test whether the laws predict configurations not used for fitting, we leave out one backbone size (22 folds) or one input size (7 folds), refit Separable and Coupled under matched constraints, and predict the omitted cells using their measured visual-token counts. We evaluate shared and group-specific exponents with nonnegative amplitudes, exponents in $[0.05, 10]$, and continuous floors, and each of 200 paired image bootstraps rebuilds the cells and repeats every fold without initialization from a full-data fit. Table 11 reports pooled RMSE over all 650 cells with predictions clipped to $[0, 1]$ and over the interior levels without clipping, comprising 400 cells from 15 backbone folds and 442 cells from 5 input-size folds. With group-specific exponents, Separable has lower RMSE on the all-cell backbone holdouts and matches Coupled on the input-size holdouts; on interior configurations the Coupled-minus-Separable differences are only -0.012 and -0.011 percentage points. Thus, at the group level used in the main analysis, the simpler Separable law predicts unseen configurations as well as or better than the Coupled law.

Endpoint extrapolation. To distinguish interpolation from extrapolation beyond the observed grid, we inspect the unclipped predictions from folds that omit the smallest or largest level of an axis. Predictions outside $[0, 1]$ occur only in these endpoint folds, whereas every prediction from the interior folds remains within $[0, 1]$. The held-out analysis therefore gives a clean comparison on interior configurations.

Table 11: Held-out RMSE over 200 paired image bootstraps. Differences are Coupled minus Separable, with 95% percentile intervals below; positive values favor Separable. All-cell predictions are clipped to $[0, 1]$, whereas interior predictions are not.

Exponents	Held-out axis	Separable	Coupled	ΔRMSE (pp)
All cells				
Shared	Backbone size	0.0655	0.0579	−0.755 (−1.668, 2.940)
Shared	Input size	0.0763	0.0891	1.283 (−0.412, 2.388)
Per-group	Backbone size	0.0577	0.0781	**2.040** (0.488, 5.111)
Per-group	Input size	0.0601	0.0599	**−0.028** (−0.182, 0.044)
Interior levels				
Shared	Backbone size	0.0608	0.0512	−0.962 (−2.004, −0.421)
Shared	Input size	0.0606	0.0544	−0.614 (−1.403, 0.661)
Per-group	Backbone size	0.0447	0.0446	**−0.012** (−0.045, 0.037)
Per-group	Input size	0.0456	0.0455	**−0.011** (−0.037, 0.009)

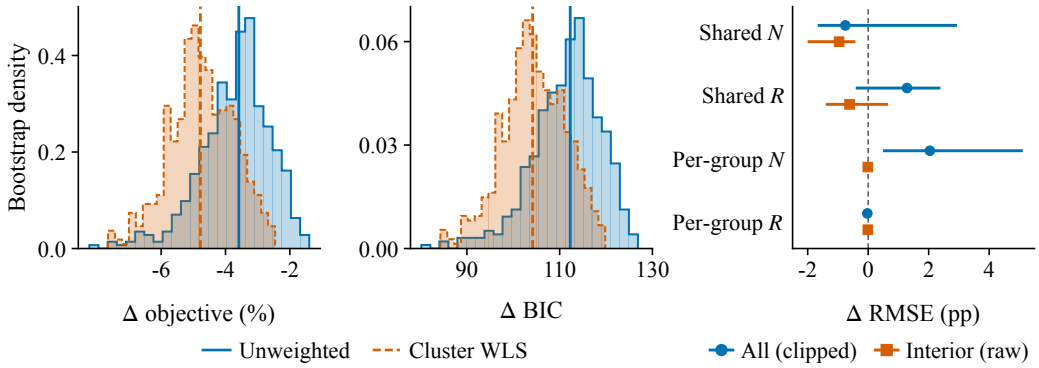


Figure 10: **Bootstrap comparison of Separable and Coupled laws.** Left and middle show bootstrap objective and BIC differences for unweighted (blue) and precision-weighted (orange) fits; right shows held-out RMSE differences with 95% intervals. All differences are Coupled minus Separable: negative objective or RMSE differences favor Coupled, whereas positive BIC differences favor Separable.

B.6 A Bounded Version of the Law

Bounded formulation. To test whether the main conclusions depend on unbounded error predictions, we replace the Separable output with a logistic link,

$$\eta_g(n, r) = c_g + A_g n^{-\alpha_g} + B_g r^{-\beta_g}, \qquad L_g^{\mathrm{bnd}}(n, r) = \sigma(\eta_g(n, r)), \tag{16}$$

where $\sigma(x) = (1 + \exp(-x))^{-1}$ and n and r use the main-fit normalization. We fit the same 650 cells with $A_g, B_g \geq 0$, $\alpha_g, \beta_g \in [0.05, 10]$, and $c_g \geq \mathrm{logit}(0.001)$ by least squares on the logit of observed error. For a matched comparison, we refit the Separable law with a continuous floor, the same five parameters, and the same constraints. The bounded formulation therefore preserves the monotone two-axis structure while guaranteeing predictions in $[0, 1]$.

Configuration selection. To test whether boundedness changes practical allocation, we refit both laws when leaving out one backbone size, one input size, or both together; all 1,798 full-grid and

held-out optimizations converge. For each fold we evaluate direct selection at four FLOP budgets, retain decisions with at least two feasible held-out configurations, and define regret as the measured error above the best feasible configuration. The two laws agree on 100.0% of backbone holdouts, 99.3% of input-size holdouts, and 98.3% of the 2,379 joint-holdout decisions, with nearly identical mean regret. Thus, enforcing bounded predictions leaves the allocation decisions essentially unchanged.

Table 12: Configuration selection is stable under bounded outputs. Mean regret is reported in percentage points (pp), and agreement is the share of decisions for which the two laws select the same configuration. Each decision includes at least two feasible candidates.

Held-out design	Eligible decisions	Mean regret (pp)		Same choice (%)
		Separable	Bounded	
Backbone holdout	297	0.48	0.48	**100.0**
Input-size holdout	411	0.66	0.67	**99.3**
Joint holdout	2,379	0.76	0.75	**98.3**
Joint holdout, two-axis tradeoff	1,284	0.88	0.87	**97.1**

Exponent stability. To test whether boundedness changes the qualitative scaling relation, we fit both laws on the complete grid and compare α and β across all 20 groups. Their ordering is identical in every group: $\beta > \alpha$ in all 12 InternVL groups and in one of the 8 QwenVL groups, while the rank correlations are 0.82 for α and 0.96 for β. Although the numerical values differ across error scales, the family-level ordering of the two resource axes is unchanged.

Table 13: Median exponents and numbers of groups at the search bounds.

Family	Law	Median α	Median β	$\alpha = 0.05$	$\beta = 0.05$	$\beta = 10$
InternVL	Separable	0.196	10.000	0	0	9
InternVL	Bounded	**0.113**	**10.000**	5	0	9
QwenVL	Separable	0.779	0.174	1	0	0
QwenVL	Bounded	**0.664**	**0.074**	2	3	0

Prediction stability. To test whether boundedness changes predictive accuracy, we compare the laws on folds that leave out interior levels and folds that leave out endpoints. Across the three designs, their interior RMSEs differ by at most 0.12 percentage points, while the bounded law keeps every prediction in $[0, 1]$. The bounded formulation therefore preserves interior predictive accuracy while guaranteeing valid predictions at the edges of the grid.

Table 14: **Predictive stability under bounded outputs.** Held-out RMSE is reported in percentage points (pp), together with the number of predictions outside $[0, 1]$. Separable predictions are clipped only when computing RMSE; neither law clips predictions during configuration selection. Highlighted cells mark the lower RMSE or the bounded law's removal of invalid predictions.

Design	Held-out level	Held-out RMSE (pp)		Predictions outside $[0, 1]$	
		Separable	Bounded	Separable	Bounded
Backbone size	Interior	4.47	**4.43**	0	0
Backbone size	Endpoint	7.38	**7.18**	6	**0**
Input size	Interior	4.56	**4.45**	0	0
Input size	Endpoint	**8.29**	11.79	0	0
Joint holdout	Interior	4.59	**4.54**	0	0
Joint holdout	Endpoint	**6.97**	8.26	36	**0**

B.7 Parameter Uncertainty and Sensitivity

Parameter uncertainty. To measure how precisely the scaling parameters are determined, we refit all 20 Separable groups in 500 paired image bootstraps using the original floor grid and exponent range $[0.05, 10]$. The refits reproduce every original-sample SSE to within 1.1×10^{-11}, and Table 15 reports the 95% intervals together with the rate at which β reaches a search bound. QwenVL keeps compact β intervals below 0.51 and rarely reaches a bound, whereas the steep InternVL fits frequently reach the upper limit. The bootstrap therefore makes the frontend contrast explicit. QwenVL has a well-resolved visual-token exponent, while InternVL is consistently much steeper.

Table 15: **Bootstrap uncertainty of the Separable-law parameters.** Entries report 95% percentile intervals from 500 paired image bootstraps and the share of refits in which β reaches either search bound. Intervals are conditional on the original exponent box $[0.05, 10]$; rates of at least 90% are highlighted.

Series	Benchmark	95% bootstrap interval			β at search bound (%)
		α	β	e_∞	
InternVL2.5	HR-Bench	$[0.106, 0.332]$	$[1.654, 10.000]$	$[0.001, 0.140]$	3.2
	MME-RealWorld	$[0.050, 0.068]$	$[10.000, 10.000]$	$[0.001, 0.020]$	**100.0**
	TreeBench	$[0.050, 0.077]$	$[6.695, 10.000]$	$[0.001, 0.140]$	**95.4**
	V* Bench	$[0.201, 0.651]$	$[10.000, 10.000]$	$[0.001, 0.030]$	**99.0**
InternVL3	HR-Bench	$[0.135, 0.601]$	$[1.310, 9.660]$	$[0.001, 0.200]$	1.8
	MME-RealWorld	$[0.050, 0.530]$	$[10.000, 10.000]$	$[0.001, 0.440]$	**100.0**
	TreeBench	$[0.050, 0.588]$	$[1.597, 10.000]$	$[0.001, 0.520]$	88.8
	V* Bench	$[0.379, 10.000]$	$[8.041, 10.000]$	$[0.001, 0.140]$	**90.6**
InternVL3.5	HR-Bench	$[0.112, 0.585]$	$[1.219, 10.000]$	$[0.001, 0.260]$	8.0
	MME-RealWorld	$[0.753, 1.861]$	$[10.000, 10.000]$	$[0.440, 0.480]$	**100.0**
	TreeBench	$[0.050, 2.398]$	$[0.622, 10.000]$	$[0.001, 0.560]$	61.2
	V* Bench	$[0.189, 0.905]$	$[10.000, 10.000]$	$[0.001, 0.040]$	**97.8**
Qwen2.5-VL	HR-Bench	$[0.247, 10.000]$	$[0.099, 0.152]$	$[0.001, 0.001]$	0.0
	MME-RealWorld	$[0.100, 10.000]$	$[0.070, 0.083]$	$[0.001, 0.060]$	0.0
	TreeBench	$[0.050, 10.000]$	$[0.050, 0.454]$	$[0.001, 0.540]$	6.0
	V* Bench	$[0.100, 10.000]$	$[0.169, 0.257]$	$[0.001, 0.001]$	0.0
Qwen3-VL	HR-Bench	$[0.406, 4.459]$	$[0.128, 0.201]$	$[0.001, 0.001]$	0.0
	MME-RealWorld	$[0.051, 10.000]$	$[0.091, 0.228]$	$[0.001, 0.340]$	0.0
	TreeBench	$[0.056, 10.000]$	$[0.050, 0.501]$	$[0.001, 0.520]$	2.6
	V* Bench	$[0.343, 1.317]$	$[0.257, 0.417]$	$[0.001, 0.001]$	0.0

Search-box sensitivity. To test whether the frontend contrast depends on the exponent search range, we refit all 20 groups with $\alpha, \beta \in [0.05, 3]$. All 12 InternVL groups move to the upper limit $\beta = 3$, whereas all 8 QwenVL groups remain below 0.5 under both ranges; the narrower box raises pooled SSE from 1.088 to 1.400. Four InternVL group-level floors move by more than 0.02, indicating some sensitivity to the narrower box. The search-box experiment nevertheless preserves the family-level exponent contrast while showing that the exact large-β and group-level floor estimates depend on the search range.

B.8 Input Coverage and Token Aggregation

Removing the HR-Bench 4096 px level. To test whether the release-specific 4096 px setting drives the conclusions, we remove its 26 cells and refit the remaining 624 with the original normalization and constraints. The α–β ordering and allocation-path direction remain unchanged in all 20 groups, pooled RMSE stays nearly constant at 4.102 percentage points versus 4.092 on the full grid, and the reduced-grid fits reproduce 79 of 80 decisions on the common candidates. The exponent and allocation conclusions therefore do not depend on the additional HR-Bench input size.

Table 16: Fits under the narrower exponent box $[0.05, 3]$. † and ‡ mark upper-bound β and α; pooled SSE is 1.400.

Series	Benchmark	α	β	e_∞	R^2
InternVL2.5	HR-Bench	0.17	**3.00**†	0.001	0.91
InternVL2.5	MME-RealWorld	0.07	**3.00**†	0.001	0.69
InternVL2.5	TreeBench	0.06	**3.00**†	0.001	0.72
InternVL2.5	V* Bench	0.49	**3.00**†	0.001	0.44
InternVL3	HR-Bench	0.21	**3.00**†	0.001	0.94
InternVL3	MME-RealWorld	0.22	**3.00**†	0.300	0.65
InternVL3	TreeBench	0.07	**3.00**†	0.001	0.87
InternVL3	V* Bench	1.58	**3.00**†	0.001	0.39
InternVL3.5	HR-Bench	0.17	**3.00**†	0.001	0.92
InternVL3.5	MME-RealWorld	1.20	**3.00**†	0.380	0.69
InternVL3.5	TreeBench	0.45	**3.00**†	0.420	0.82
InternVL3.5	V* Bench	0.62	**3.00**†	0.001	0.34
Qwen2.5-VL	HR-Bench	0.46	0.12	0.001	0.93
Qwen2.5-VL	MME-RealWorld	3.00‡	0.08	0.001	0.93
Qwen2.5-VL	TreeBench	0.05	0.19	0.240	0.81
Qwen2.5-VL	V* Bench	3.00‡	0.21	0.001	0.91
Qwen3-VL	HR-Bench	0.76	0.16	0.001	0.94
Qwen3-VL	MME-RealWorld	3.00‡	0.11	0.060	0.96
Qwen3-VL	TreeBench	0.80	0.18	0.400	0.95
Qwen3-VL	V* Bench	0.59	0.32	0.001	0.96

Question-level token aggregation. To test whether the cell-mean approximation drives the results, we replace $B(\overline{R})^{-\beta}$ with $B\,\overline{R^{-\beta}}$, computed from the exact visual-token histogram of the scored question records while keeping the same 650 cells, errors, weights, and constraints. The alternative preserves the α–β ordering in all 20 groups and 79 of 80 configuration choices, lowers pooled RMSE from 4.092 to 3.077 percentage points, and removes all upper-bound β solutions. Exact question-level aggregation therefore strengthens the fit while preserving the qualitative scaling and allocation conclusions.

Table 17: Effect of visual-token aggregation on fit quality and exponent ordering.

Parameterization	Pooled RMSE (pp)	β at upper bound	$\alpha > \beta$
Cell mean	4.092	9	7
Question moment	**3.077**	**0**	7

C Finding 1: The Cognitive Ceiling

This section reports the task-level fits by series (Appendix C.1), tests the regime assignments under alternative cutoffs, held-out configurations, and high-compute evaluation (Appendices C.2 and C.3), and examines domain–skill effects and their stability (Appendices C.4 and C.5).

C.1 Task-Level Floors

Reasoning has a higher fitted floor in every series. To test whether the pooled difference of §3.1 holds across architectures, we break the 40 task-level fits down by model series and task type in Table 18. The mean floor is higher for reasoning than for perception in all five series, with pooled means of 0.155 and 0.073. The consistent direction across series shows that the higher reasoning floor is not driven by a single model family.

Table 18: Mean fitted e_∞ by model series and by the perception and reasoning labels, pooling the fits of §3.1. The gap between reasoning and perception is positive in all five series. Gaps are computed from the unrounded means and rounded on their own.

Series	Perception	Reasoning	Gap
InternVL2.5	0.001	0.067	+0.066
InternVL3	0.097	0.187	+0.090
InternVL3.5	0.180	0.194	+0.013
Qwen2.5-VL	0.001	0.187	+0.186
Qwen3-VL	0.085	0.141	+0.056
Pooled	**0.073**	**0.155**	**+0.082**

High floors are concentrated in a few pairings of benchmark and task. To locate the task subsets behind the pooled result, we inspect the fits that meet the Saturation threshold, $e_\infty \geq 0.20$. Ten of the 40 fits saturate, namely five of 25 perception fits and five of 15 reasoning fits, corresponding to 20% and 33% in Table 1. Membership is determined after rounding to two decimals because two estimates lie numerically at the threshold. Six cases cluster in TreeBench reasoning and MME-RealWorld, while the remaining four occur in individual HR-Bench or TreeBench subsets. By contrast, 18 perception fits and 8 reasoning fits reach the Floor criterion, $e_\infty \leq 0.01$. Thus, most task-level fits approach a near-zero floor, while the high floors are localized rather than universal.

The comparison uses matched task-specific fits. To construct the comparison, we fit every combination of model series, benchmark, and task subset, yielding the 40 estimates in Table 19. HR-Bench is divided into Cross and Single, MME-RealWorld and TreeBench into Perception and Reasoning, and V* Bench into Direct Attributes and Relative Position. For MME-RealWorld, we retain the native subtasks and weight every pair of subtask and configuration equally. Each InternVL series therefore contributes 180 points to its perception fit and 144 to its reasoning fit. This construction preserves the benchmark-specific task structure while supporting a consistent perception–reasoning comparison.

C.2 Robustness of Regime Assignments

The low-accuracy stratum persists across cutoff choices. To test whether the three regimes of §3.2 depend on their thresholds, we sweep τ_{easy} over $\{0.85, 0.90, 0.95\}$ and τ_{cog} over $\{0.15, 0.20, 0.25\}$ for all 19,890 trajectories. Table 20 shows that the Ceiling Bound share remains between 25.2% and 33.2%, with the default cutoffs giving 30.7%. Thus, the substantial low-accuracy stratum is not an artifact of one cutoff choice.

The regimes generalize to held-out configurations. To separate regime definition from evaluation, we assign each trajectory using half of its configurations and measure accuracy and scaling response on the other half. Across all five series, held-out accuracy is 7.08%–8.85% for Ceiling Bound and 50.53%–54.79% for Scaling Bound, while their slopes are respectively 0.13–1.24 and 3.13–4.55 percentage points per doubling of NR (Table 21). Among questions shared by all series, 17.95% are Ceiling Bound in every series. The regime labels therefore capture reproducible differences in both accuracy and scaling response.

C.3 Persistent Errors at High Compute

Ceiling Bound errors persist at the highest observed compute. To test whether Ceiling Bound questions remain difficult at high compute, we retain trajectories with sufficient configurations, backbone sizes, valid probabilities, and consistent option sets, yielding 15,830 trajectories from 3,166 questions evaluated by all five series. Of these, 5,392, or 34.1%, belong to $\mathcal{C}_{\text{mean}}$, the Ceiling Bound stratum in this cohort. Within each trajectory, we order configurations by $u_{ij} = N_j R_{ij}^{\text{q}}$ and form tied compute quartiles Q_1 to Q_4. To compare questions with two to eleven

Table 19: **Task-level Separable-law exponents.** The table covers all 40 series–benchmark–task fits; † and ‡ mark the upper and lower exponent bounds, 10 and 0.05. Interior values are point estimates and need not be statistically identified.

Series	Benchmark	Task	Exponent	
			α	β
InternVL2.5	HR-Bench	Cross	0.099	5.18
		Single	0.221	9.50
	MME-RealWorld	Perception	0.068	$\mathbf{10.0}^{\dagger}$
		Reasoning	0.065	7.05
	TreeBench	Perception	0.115	$\mathbf{10.0}^{\dagger}$
		Reasoning	$\mathbf{0.05}^{\ddagger}$	5.11
	V* Bench	Direct Attributes	0.645	$\mathbf{10.0}^{\dagger}$
		Relative Position	0.234	$\mathbf{10.0}^{\dagger}$
InternVL3	HR-Bench	Cross	0.136	3.28
		Single	0.240	9.15
	MME-RealWorld	Perception	0.465	9.08
		Reasoning	0.069	5.48
	TreeBench	Perception	0.134	$\mathbf{10.0}^{\dagger}$
		Reasoning	0.152	$\mathbf{10.0}^{\dagger}$
	V* Bench	Direct Attributes	2.90	$\mathbf{10.0}^{\dagger}$
		Relative Position	0.470	$\mathbf{10.0}^{\dagger}$
InternVL3.5	HR-Bench	Cross	0.129	3.27
		Single	0.193	$\mathbf{10.0}^{\dagger}$
	MME-RealWorld	Perception	0.999	6.36
		Reasoning	0.569	4.57
	TreeBench	Perception	0.701	$\mathbf{10.0}^{\dagger}$
		Reasoning	0.213	0.180
	V* Bench	Direct Attributes	0.491	$\mathbf{10.0}^{\dagger}$
		Relative Position	0.396	$\mathbf{10.0}^{\dagger}$
Qwen2.5-VL	HR-Bench	Cross	0.051	0.289
		Single	1.10	0.175
	MME-RealWorld	Perception	$\mathbf{10.0}^{\dagger}$	0.119
		Reasoning	$\mathbf{10.0}^{\dagger}$	0.088
	TreeBench	Perception	0.074	0.247
		Reasoning	1.23	0.095
	V* Bench	Direct Attributes[a]	0.116	0.230
		Relative Position	$\mathbf{10.0}^{\dagger}$	0.152
Qwen3-VL	HR-Bench	Cross	0.502	0.197
		Single	1.09	0.222
	MME-RealWorld	Perception	2.13	0.161
		Reasoning	4.34	0.118
	TreeBench	Perception	0.681	0.281
		Reasoning	0.840	0.053
	V* Bench	Direct Attributes	0.835	0.338
		Relative Position	0.367	0.307

Table 20: Share of trajectories in each regime (%) under the cutoff sweep, written as EASY / SCALING BOUND / CEILING BOUND. The default cutoffs, $\tau_{\text{easy}} = 0.90$ and $\tau_{\text{cog}} = 0.20$, are in bold.

	$\tau_{\text{easy}} = \mathbf{0.85}$	$\tau_{\text{easy}} = \mathbf{0.90}$	$\tau_{\text{easy}} = \mathbf{0.95}$
$\tau_{\text{cog}} = 0.15$	22.3 / 52.5 / 25.2	20.0 / 54.8 / 25.2	17.4 / 57.4 / 25.2
$\tau_{\text{cog}} = 0.20$	22.3 / 47.0 / 30.7	**20.0 / 49.4 / 30.7**	17.4 / 51.9 / 30.7
$\tau_{\text{cog}} = 0.25$	22.3 / 44.4 / 33.2	20.0 / 46.8 / 33.2	17.4 / 49.3 / 33.2

Table 21: Held-out performance after defining the regimes on the other half of the configurations. Accuracy is in percent; slopes are percentage points per doubling of NR on the held-out half. Light-blue cells mark the held-out CEILING BOUND results.

	CEILING BOUND		SCALING BOUND		EASY	
Series	**Accuracy**	**Slope**	**Accuracy**	**Slope**	**Accuracy**	**Slope**
InternVL2.5	**7.08**	**0.30**	53.40	4.55	96.56	1.01
InternVL3	**7.62**	**0.13**	54.79	4.36	96.48	0.80
InternVL3.5	**8.21**	**0.60**	52.82	3.82	95.35	1.47
Qwen2.5-VL	**7.50**	**0.58**	50.93	3.13	93.28	1.22
Qwen3-VL	**8.85**	**1.24**	50.53	4.49	94.74	0.62

options, we verify the prompts and labels, exclude 12 inconsistent questions, and normalize correctness a_{ij} and correct-option probability p_{ij} relative to the uniform-guessing rate $c_i = 1/K_i$:

$$c_i = \frac{1}{K_i}, \qquad z^*_{ij} = \frac{z_{ij} - c_i}{1 - c_i}, \qquad z \in \{a, p\}, \tag{17}$$

keeping negative values unchanged. In the highest-compute quartile, $4{,}602$ of the $5{,}392$ trajectories, or 85.3%, still have accuracy below 0.20; their mean accuracy is 8.6%, and only 6.6% reach at least 0.50. Meanwhile, 32.1% gain more than 0.05 in correct-option probability from Q_1 to Q_4, while 27.8% lose more than 0.05. The matched within-trajectory experiment therefore confirms that low accuracy persists at the highest observed compute even though individual scaling responses vary.

A stricter response-based core lies inside CEILING BOUND. To distinguish low performance at high compute from weak response to scaling, for $z \in \{a, p, a^*, p^*\}$ we write z_q for the mean over quartile q, $\Delta z = z_4 - z_1$, $\Delta_{\text{late}} z = z_4 - z_3$, and b_z for the slope against $\log_2 u$ over all configurations, and compare

$$\begin{aligned} &H_z : z_4 < \tau, \\ &E_z : H_z \wedge |\Delta z| \le \delta, \qquad P_z : H_z \wedge |\Delta_{\text{late}} z| \le \delta, \\ &B_z : H_z \wedge |b_z| \le \eta, \qquad J_z : E_z \cap P_z \cap B_z. \end{aligned} \tag{18}$$

H_z requires low performance at high compute, while J_z additionally requires small endpoint change, small late change, and a small fitted slope. The default margins are $\tau = 0.20$ for the raw measures, $\tau = 0.10$ for the normalized ones, $\delta = 0.05$, and $\eta = 0.01$ per doubling of compute. Applying the joint rule selects $1{,}502$ trajectories, or 9.5%, using raw probability and $1{,}296$, or 8.2%, after chance normalization (Table 22). Of these, 98.9% and 97.1%, respectively, lie inside $\mathcal{C}_{\text{mean}}$. The response-based experiment therefore isolates a smaller persistent-error core while confirming that CEILING BOUND captures nearly all of it.

Table 22: Share of the same $15{,}830$ trajectories selected by each criterion (%). Every row requires low Q_4 performance. The raw columns use $\tau = 0.20$ and the normalized columns $\tau = 0.10$, with $\delta = 0.05$ and $\eta = 0.01$ per doubling. The CEILING BOUND rule selects 34.1% of this cohort, and the 30.7% of §3.2 uses the full cohort instead.

Criterion	a	p	a^*	p^*
Low Q_4 performance H_z	37.5	30.6	39.1	40.0
Small endpoint change E_z	20.8	13.1	20.8	13.0
Small late change P_z	23.5	18.3	23.4	18.1
Small fitted slope B_z	19.0	14.1	18.3	13.9
All three together J_z	**16.4**	**9.5**	**16.3**	**8.2**

Persistent-error rates are highest on MME-RealWorld, TreeBench, and reasoning tasks. To test whether the group-level pattern depends on the definition, Table 23 compares CEILING BOUND with the two joint response criteria on the same observations. Under both criteria, MME-RealWorld

and TreeBench keep higher rates than HR-Bench and V* Bench, and reasoning keeps a higher rate than perception, 11.1% against 6.7% after chance normalization. The broad benchmark and task contrasts therefore persist even when the definition of persistent error becomes stricter.

Table 23: Benchmarks and tasks compared on the same observations (%). The benchmark rows and the task rows are two separate partitions of the same $15{,}830$ trajectories. HR-Bench covers only the six shared input sizes, since the trajectories that exist only at 4096 px are excluded. The CEILING BOUND column uses mean accuracy below 0.20, and J_p and J_{p^*} follow Eq.(18).

Group	Trajectories	CEILING BOUND	J_p	J_{p^*}
HR-Bench	4,000	23.0	4.5	3.9
MME-RealWorld	8,900	39.3	12.4	10.8
TreeBench	1,975	40.8	9.7	8.3
V* Bench	955	17.6	2.4	1.6
Perception	10,525	30.0	7.9	6.7
Reasoning	5,305	42.1	12.7	11.1

A small-response core remains across thresholds and token aggregation. To test sensitivity to the response definition, we sweep the raw and normalized performance cutoffs, the change margin, and the slope margin. The joint rule then selects between 4.2% and 18.4% of trajectories on raw probability and between 3.5% and 15.5% after normalization, so a group with small response remains throughout the sweep while its size depends on the margins. Using the mean visual token count of each configuration instead of the count logged for each question gives 9.7% and 8.4%, close to the defaults of 9.5% and 8.2%. Requiring the four quartile means to span at most 0.05 still leaves 970 candidates, or 6.1%. Thus, a nontrivial small-response core remains across reasonable thresholds and visual-token aggregation choices.

High-compute difficulty replicates across configuration halves. To check internal stability, we split the configurations in every quartile into two outcome-independent halves using a fixed hash ordering with seed 20260909. Selecting J_{p^*} on one half gives $1{,}268$ candidates, of which 824, or 65.0%, also satisfy the rule on the other half, and reversing the halves gives $1{,}180$ candidates with 69.8% replication. Difficulty at high compute itself carries over to the other half for 98.2% and 96.8% of the selected candidates. The main high-compute difficulty signal is therefore highly stable across disjoint configurations, and together the experiments establish persistent error over the evaluated grid without extrapolating to unseen architectures or to configurations beyond our grid.

C.4 DOMAIN AND SKILL TAXONOMY

Why we relabel. The four benchmarks label their questions in ways that do not match. HR-Bench separates perception of a single instance from perception across instances, MME-RealWorld gives subtasks and categories, TreeBench names subtypes of reasoning, and V* Bench separates direct attributes from relative position. These labels work within a benchmark but give no shared vocabulary across benchmarks. To locate the cognitive ceiling, we therefore give every question two labels: the domain D is the visual scene of the image, and the skill S is the operation the question requires. Keeping the two labels separate is what lets us ask whether the floor e_∞ follows the domain, the skill, or a particular domain and skill pairing. Table 2 gives the 6×6 grid, with domains D1 Documents/Charts/Slides, D2 Aerial/Satellite, D3 Vehicles/Driving, D4 Indoor, D5 Outdoor, and D6 People/Surveillance, and skills S1 Attribute Recognition, S2 Text Reading, S3 Counting, S4 Spatial Relation, S5 Object Identification, and S6 Scene Reasoning.

Labeling and coverage. GPT-4o-mini assigns the labels at temperature 0, reading the image at low detail together with the full question text and its answer options, and returning exactly one domain and one skill. We join the labels back to the evaluation records by benchmark and question ID. The labeling run covered the 800 HR-Bench questions evaluated at the six shared input sizes. The records at 4096 px come from the other HR-Bench release and carry no labels, so the fits below use the six shared input sizes. This gives one pair of labels for each of the $3{,}178$ unique questions,

namely 800 from HR-Bench, 1,782 from MME-RealWorld, 405 from TreeBench, and 191 from V* Bench.

Checking the labels. We inspect the assigned labels by hand against the taxonomy and the decision rules. As an automatic check, we compare them with a fixed mapping of the native metadata of MME-RealWorld. The MME-RealWorld metadata does not correspond one to one with our taxonomy, so the agreement rate measures how consistent the two label sets are rather than how accurate ours is.

Labeling prompt. The box below abridges the system prompt. The full prompt also gives ordered rules for choosing a domain, rules for skills that overlap, and instructions on the output format. Afterwards we only parse the returned labels and map them to their canonical names, and we never change an assignment.

> You are labeling an image for a vision-language scaling-law study. Look at the image and read the question text, then output TWO labels.
> DOMAIN – pick exactly one: D1 Documents/Charts/Slides, D2 Aerial/Satellite, D3 Vehicles/Driving, D4 Indoor, D5 Outdoor-General, or D6 People/Surveillance.
> SKILL – pick exactly one: S1 Attribute Recognition, S2 Text Reading, S3 Counting, S4 Spatial Relation, S5 Identification, or S6 Reasoning over Scene.
> Output a single JSON line, no other text: `{"domain": "D?", "skill": "S?"}`.

The prompt writes D5 as Outdoor-General, S5 as Identification, and S6 as Reasoning over Scene, and the paper writes them as Outdoor, Object Identification, and Scene Reasoning. Labels are stored as the codes D1 to D6 and S1 to S6, so the wording changes nothing in the analysis.

C.5 Fits for Each Domain–Skill Pairing

Bootstrap uncertainty in the variance decomposition. To compare persistent error across domains and skills, we fit Eq.(3) separately to every supported domain–skill pairing at the six shared input sizes. Each fit pools the five model series and all contributing benchmarks, uses the same error definition and parameterization as the main analysis, and weights eligible benchmark–configuration–domain–skill subcells equally. Retaining subcells with at least five questions yields 10,135 subcells across 624 configurations and 33 supported pairings. This common protocol makes the fitted floors directly comparable across the domain–skill grid.

High floors are concentrated in specific domain–skill pairings. To locate the strongest ceiling effects, we compare the 33 fitted floors and their domain and skill marginals. Table 25 lists ten pairings with a fitted floor of at least 0.30, spanning several visual domains. The highest is Aerial/Satellite with Scene Reasoning, at $e_\infty = 0.780$. Table 24 averages the floors over the supported domain–skill pairings to give one marginal mean for each skill and one for each domain. The skill means run from 0.001 for Text Reading to 0.432 for Attribute Recognition, and the domain means from 0.041 for Documents/Charts/Slides to 0.272 for People/Surveillance. Text Reading and Object Identification remain near the lower floor across almost every supported domain. The largest ceilings are therefore localized to particular combinations of scene and required skill rather than entire benchmarks or domains.

Skill accounts for more of the spread than domain. To separate the two sources of variation, we fit least-squares models to the 33 floors using domain alone, skill alone, and both factors together. All three weight the 33 domain–skill pairings equally and leave out the three unsupported pairings, rather than filling them in. They explain 9.26%, 38.09%, and 44.51% of the variance of the floors. Either factor can enter the model with both terms first or second, so we average its share over the two orders,

$$V_S = \tfrac{1}{2}\big[R_S^2 + (R_{D+S}^2 - R_D^2)\big], \qquad V_D = \tfrac{1}{2}\big[R_D^2 + (R_{D+S}^2 - R_S^2)\big]. \tag{19}$$

This assigns 36.67% of the variance to skill and 7.84% to domain, a ratio of 4.68. Skill therefore captures substantially more of the broad variation in fitted floors, while specific pairings account for the remaining structure.

Table 24: Marginal fitted floors by skill and domain. Each mean equally weights the supported domain–skill pairings; Pairings gives their number.

Skill marginals			Domain marginals		
Skill	**Mean e_∞**	**Pairings**	**Domain**	**Mean e_∞**	**Pairings**
S1 Attribute Recognition	**0.432**	5	D1 Documents/Charts/Slides	**0.041**	4
S2 Text Reading	**0.001**	6	D2 Aerial/Satellite	0.227	6
S3 Counting	0.250	6	D3 Vehicles/Driving	0.204	6
S4 Spatial Relation	0.213	5	D4 Indoor	0.124	6
S5 Object Identification	0.044	6	D5 Outdoor	0.214	6
S6 Scene Reasoning	0.228	5	D6 People/Surveillance	**0.272**	5

Table 25: The ten domain–skill pairings with a fitted floor of at least 0.30, under the taxonomy of Table 2 and the protocol above.

Pairing	**Domain × Skill**	e_∞
D2×S6	**Aerial/Satellite × Scene Reasoning**	**0.780**
D6×S4	People/Surveillance × Spatial Relation	0.580
D3×S3	Vehicles/Driving × Counting	0.540
D5×S1	Outdoor × Attribute Recognition	0.540
D2×S1	Aerial/Satellite × Attribute Recognition	0.520
D5×S4	Outdoor × Spatial Relation	0.480
D6×S3	People/Surveillance × Counting	0.480
D3×S1	Vehicles/Driving × Attribute Recognition	0.400
D4×S1	Indoor × Attribute Recognition	0.400
D6×S1	People/Surveillance × Attribute Recognition	0.300

Cluster resampling preserves the domain–skill comparison. To assess whether the ordering between V_S and V_D depends on particular images, we repeat the full analysis under two image-cluster resampling schemes. Both schemes retain the original fitting choices, namely the 55-point floor grid, $\alpha, \beta \in [0.05, 10]$, nonnegative amplitudes, equal subcell weights, and the original scoring rule. Within each benchmark, a cluster contains one image together with all of its questions, models, and input sizes. Each replicate reweights or resamples these clusters, recomputes the subcell errors and measured visual token counts, refits every domain–skill floor that remains defined, and repeats the equal-weight two-factor decomposition. The ordinary bootstrap yields 397 complete-support replicates out of 500, while a positive-weight bootstrap preserves all 33 pairings in all 500 replicates. The skill share exceeds the domain share in 82.6% of ordinary-bootstrap replicates and 81.6% of positive-weight replicates, with closely aligned percentile ranges (Table 26). Although the ratio intervals cross one and do not pin down the point-estimate ratio of 4.68, both schemes support the same skill-dominant ordering. Cluster resampling therefore preserves the main domain–skill comparison under complementary checks with and without support loss in rare pairings.

Table 26: Image-cluster bootstrap stability of the domain–skill variance decomposition. Intervals are the 2.5th–97.5th percentiles; ordinary-bootstrap intervals use the 397 replicates retaining all 33 pairings.

Resampling	**Complete**	**95% percentile interval**			$V_S > V_D$ **(% replicates)**
		V_S **(%)**	V_D **(%)**	V_S/V_D	
Ordinary cluster	397/500	10.60–40.89	6.25–29.07	0.60–4.27	**82.6**
Positive-weight cluster	500/500	11.94–39.13	6.28–28.78	0.57–4.65	**81.6**

Most domain–skill fits have interior exponents. To check whether the pairing-level fits are driven by the exponent search limits, we inspect the optimizer solutions across all 33 pairings. β lies away from the bounds in 29 fits, and both exponents are interior in 28. D2×S1 has α at the upper

bound of 10, and D2×S3, D2×S6, D3×S6, and D5×S3 have β at the lower bound of 0.05. For D2×S4 and D2×S6 the fitted amplitude A_n is zero, so the matching α has no effect on the fitted curve at any observed configuration and is not determined, even when the optimizer returns a value away from the bounds. Thus, most pairing-level fits are not pinned to the exponent limits, although interior estimates alone do not establish statistical identification.

D Finding 2: Architectural Divergence

We first report the two fitted exponents by family and series and compare their ratio on the task-level fits (Appendix D.1). Because β reaches the upper bound in most InternVL groups, we then measure how far the visual token count of each frontend actually moves, test effective pixels as an alternative visual measure, and vary source-image quality at fixed token sequences (Appendix D.2).

D.1 Exponents by Family, Series, and Task

The backbone capacity exponent is interior in all InternVL groups and most QwenVL groups. To compare backbone capacity scaling across the two families, we inspect α in all 20 series–benchmark fits and summarize the estimates that lie inside the search bounds. All 12 InternVL fits are interior, with a median of 0.19, while 5 of the 8 QwenVL fits are interior, with a median of 0.59; including the three boundary estimates raises the QwenVL median to 0.78. The boundary cases are Qwen2.5-VL on MME-RealWorld and V* Bench and Qwen3-VL on MME-RealWorld. In all three, the fitted amplitude at the reference point is $A_n \leq 4 \times 10^{-6}$. A small amplitude there does not make the backbone capacity term negligible across the observed range, since that term is $A_n(N/\widetilde{N})^{-\alpha}$ and N falls well below $\widetilde{N}$ for the smaller models. Backbone capacity scaling is therefore represented by an interior exponent in most groups, with greater boundary sensitivity in QwenVL.

The visual-token exponent sharply separates the two families. To compare visual-token scaling, we examine β under the same group fits. Nine of the 12 InternVL fits reach the upper bound of 10, and the three that do not are all on HR-Bench, at 3.87, 4.66, and 4.93. All 8 QwenVL fits stay away from the bounds, with a median of 0.17 (0.173 before rounding). Although individual interior estimates can still be imprecise, the group fits consistently distinguish the steep InternVL response from the gradual QwenVL response.

The β contrast repeats in every model series. To test whether the family difference is driven by one release, Table 27 summarizes the four benchmark fits of each series. Every InternVL series has exactly one estimate away from the bounds, always its HR-Bench fit, and that value lies between 3.87 and 4.93 in all three series. Both QwenVL series have all four estimates away from the bounds, with medians of 0.155 and 0.173. The backbone capacity exponent varies more across releases, but the visual-token exponent preserves the same family ordering in every series. The architectural divergence is therefore repeated across releases rather than produced by a single model generation.

Table 27: Group-level exponents by series, with four benchmark fits each. Interior means strictly inside $[0.05, 10]$. For β, InternVL reports its sole interior HR-Bench estimate and QwenVL reports the median of four interior estimates.

Series	Backbone capacity exponent α			Interior β
	Median	Range	At bound	
InternVL				
InternVL2.5	0.10	0.057–0.381	0	**4.66**
InternVL3	0.19	0.071–1.262	0	**3.87**
InternVL3.5	0.44	0.145–1.203	0	**4.93**
QwenVL				
Qwen2.5-VL	0.26	0.053–0.462	2	**0.155**
Qwen3-VL	0.76	0.589–0.800	1	**0.173**

The exponent ratio reverses between QwenVL and InternVL. To compare the relative response to backbone capacity and visual tokens, we compute α/β on the 40 task-level fits. Both exponents lie away from the bounds in 13 of the 16 QwenVL fits and in 11 of the 24 InternVL fits, and 12 of the 13 InternVL estimates at a bound sit at the upper bound on β. Among the fits with both exponents away from the bounds, the median ratio is 2.55 for QwenVL, with $\alpha > \beta$ in 10 of 13, and 0.0395 for InternVL, with $\beta > \alpha$ in 10 of 11. The one InternVL exception is InternVL3.5 on TreeBench Reasoning. One QwenVL fit has $A_n = 0$, so its α is not determined; dropping it leaves 12 fits with a median ratio of 3.72. On reasoning tasks alone, the QwenVL and InternVL medians are 14.47 and 0.0685. The task-level ratios therefore reinforce the same family-specific balance between the two scaling axes.

D.2 Token Ranges of the Two Frontends

InternVL tiling keeps the visual-token range narrow. To determine how strongly input size changes the resource seen by the language backbone, we compare InternVL3.5-8B and Qwen3-VL-8B on $3{,}166$ equally weighted questions for which both models record valid token counts at all six shared input sizes and receive identical input pixels. InternVL selects at most 12 tiles by matching the input aspect ratio, uses pixel count only to break ties, and converts each 448×448 tile into 256 visual tokens after pixel unshuffle (Chen et al., 2024c). We find that $2{,}223$ questions keep the same tile count at the two endpoints and $2{,}214$, or 69.9%, keep it at all six sizes. Across all input sizes within each group, including 4096 px on HR-Bench, the maximum token count is only 1.07 to 1.55 times the minimum, with a median span of 1.22. Consistent with this limited intervention range, 9 of the 12 InternVL fits place β at its upper bound, while the three HR-Bench estimates are 3.87, 4.66, and 4.93. The tiling frontend therefore exposes a consistently narrow visual-token range over the evaluated input-size grid, as visualized in Figure 5a.

QwenVL dynamic resizing exposes a broad visual-token range. To compare the two frontends on the same resource measure, we measure how QwenVL token counts change across the evaluated input sizes. QwenVL resizes the whole image and compresses each 2×2 block of patches into one visual token, so we compute the mean token count in every cell and its span within each group. The count changes at the much finer granularity of the patch grid, although it need not rise at every step. For Qwen3-VL 2B on HR-Bench, the mean decreases from 74.875 at 224 px to 73.480 at 336 px. Across the full grids, the maximum count is 34 to 360 times the minimum and the cells cover roughly 37 to $15{,}000$ tokens; over the six shared input sizes alone, the spans remain 34.09 to 87.79. All 8 QwenVL fits consequently place β away from the bounds, with a median of 0.17. Continuous resizing therefore supplies broad, fine-grained token variation from which a gradual visual-token response can be estimated, as visualized in Figure 5a.

Effective pixels explain InternVL variation that token count alone misses. To test whether token count alone captures the image detail preserved by each frontend, we refit V* Bench after replacing $L(N, R)$ with $L(N, P_{\text{eff}})$. For every question, we define $P_{\text{eff}} = \min(P_{\text{input}}, P_{\text{grid}})$, where P_{input} is the resized input area and P_{grid} is the spatial area represented by the frontend grid; for InternVL, the repeated thumbnail is excluded. We compute this quantity before cell aggregation, retain the original fitting constraints, and compare in-sample fit, BIC, and whole-level prediction under R, P_{eff}, and both coordinates together. For InternVL2.5, InternVL3, and InternVL3.5, replacing R by P_{eff} raises R^2 from 0.594, 0.548, and 0.515 to 0.943, 0.940, and 0.963, respectively, and none of the effective-pixel exponents reaches a search bound. For InternVL3.5, holding out one backbone size at a time reduces RMSE from 8.95 to 3.32 percentage points, while holding out one input-size level at a time reduces it from 10.50 to 3.16 points. BIC likewise favors P_{eff} for all three InternVL series, whereas Qwen2.5-VL is essentially unchanged and Qwen3-VL decreases from $R^2 = 0.959$ to 0.914. Adding R after P_{eff} produces almost no further increase in R^2 for InternVL and introduces boundary estimates or zero amplitudes, so the current grid cannot separately identify their conditional effects. Figure 11 summarizes these comparisons. The experiment therefore identifies effective pixels as a substantially more informative and complementary visual measure for InternVL, especially when token counts are fixed or change only slightly.

Raising the visual token count on native images improves accuracy on several benchmarks. To test whether visual tokens are a genuine intervention rather than only a fitted quantity, we hold

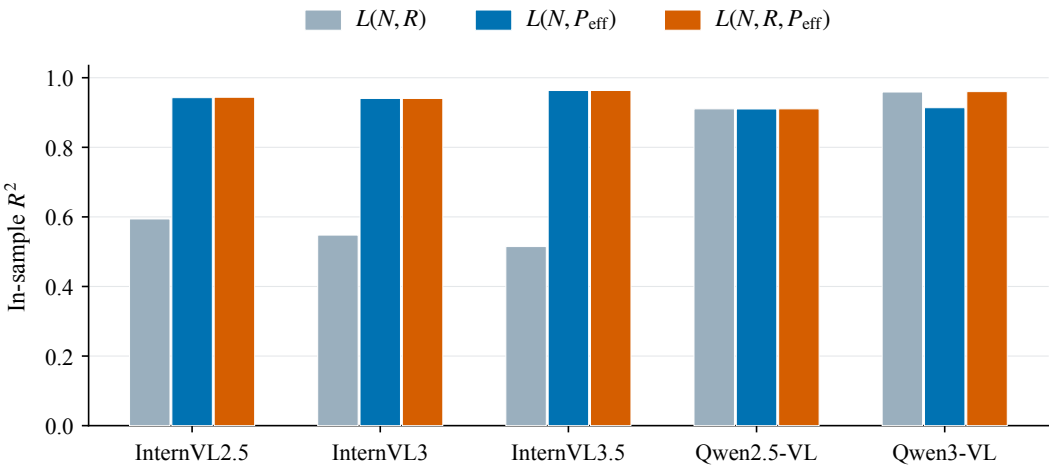


Figure 11: **Effective pixels as an alternative visual scaling measure on V* Bench.** In-sample R^2 for fits using (N, R), (N, P_{eff}), and (N, R, P_{eff}) under the same constraints. Effective pixels substantially improve the InternVL fits but not the Qwen3-VL fit. For InternVL, adding R after P_{eff} yields almost no further increase in R^2; together with boundary estimates or zero amplitudes, this means that the three-variable fits do not statistically separate their contributions.

model size fixed at 8B, retain the native source images, and vary only the number of visual tokens. For InternVL3.5-8B, we sweep `max_num` over $\{1, 2, 4, 6, 12, 24, 36\}$. For Qwen3-VL-8B, pixel caps set token counts from 64 to 16,384. We compare the largest and smallest token counts with paired accuracy differences, pointwise 95% percentile intervals from 20,000 image bootstraps, and Holm correction over the 16 prespecified primary tests. Table 28 reports the endpoint tests, while Figure 12 shows the full accuracy curves against the measured token count. The endpoint gain is significant for InternVL on V* Bench at 30.37 points, and for QwenVL on HR-Bench, MME-RealWorld, and V* Bench at 30.00, 17.00, and 42.41 points. These controlled sweeps confirm that raising the visual token count can produce substantial accuracy gains for both frontends.

Table 28: Endpoint gains in the native-source visual-token sweeps. Gains and intervals are in percentage points. Light-blue rows mark gains with Holm-adjusted $p < 0.05$.

Series	Benchmark	Questions	Gain	95% interval	Holm p
InternVL3.5-8B	HR-Bench	100	14.00	[4.00, 24.00]	0.0692
InternVL3.5-8B	MME-RealWorld	100	7.00	[−1.00, 15.00]	0.225
InternVL3.5-8B	TreeBench	100	8.00	[1.00, 16.00]	0.225
InternVL3.5-8B	V* Bench	191	**30.37**	[23.56, 37.17]	$\mathbf{2.62\times10^{-14}}$
Qwen3-VL-8B	HR-Bench	100	**30.00**	[19.00, 41.00]	$\mathbf{3.21\times10^{-6}}$
Qwen3-VL-8B	MME-RealWorld	100	**17.00**	[8.00, 26.00]	**0.00554**
Qwen3-VL-8B	TreeBench	100	10.00	[0.00, 20.00]	0.225
Qwen3-VL-8B	V* Bench	191	**42.41**	[34.55, 50.26]	$\mathbf{4.15\times10^{-20}}$

Source-image quality amplifies the return to more visual tokens. To separate source information from input-sequence length, we cross the seven InternVL3.5-8B tile limits with native images and versions whose long side is reduced to 1024 or 224 pixels and then restored to the native canvas. Across all 3,437 combinations of question and cap, the three quality conditions have identical actual token counts and identical input-token-sequence hashes, so the factorial changes available pixel information while preserving the model input structure. On V* Bench, the native condition gains 30.37 points between the smallest and largest token counts, from 53.40% to 83.77%, whereas the 224-pixel condition gains only 6.81 points, from 46.07% to 52.88%. Source quality and token count therefore interact, by 23.56 points (95% interval [14.66, 32.46], Holm-adjusted $p = 4.32\times10^{-6}$), and at the largest token count the native condition leads the 224-pixel condition by 30.89 points ([24.08, 38.22], Holm-adjusted $p = 4.01\times10^{-14}$). The corresponding quality effects at the largest

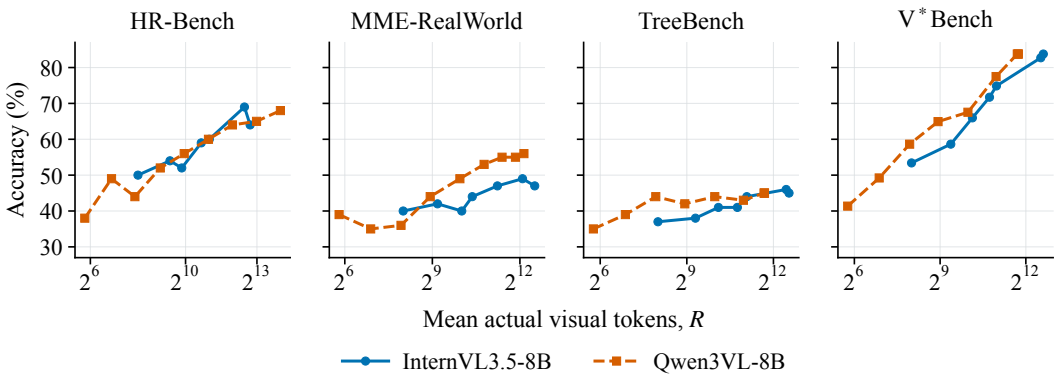


Figure 12: **Visual-token sweeps on native source images.** Accuracy against the mean actual visual-token count for InternVL3.5-8B and Qwen3-VL-8B on the four benchmarks. The two frontends expose different token ranges, so differences between their endpoint gains do not by themselves isolate an architectural effect.

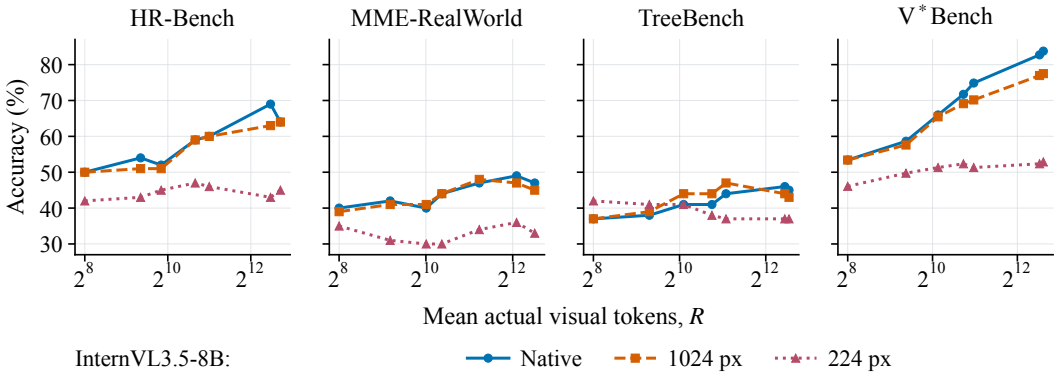


Figure 13: **Source-image information quality modulates the return to visual tokens.** InternVL3.5-8B accuracy under native, 1024-pixel, and 224-pixel source-quality conditions crossed with seven tile limits. Actual visual-token counts and input-sequence hashes match across quality conditions for every combination of question and cap, so the curves vary pixel information while holding the input sequence structure fixed.

token count are also significant at 19 points on HR-Bench and 14 on MME-RealWorld, while the 8-point TreeBench effect is not. Figure 13 shows the complete factorial curves for all four benchmarks. The factorial therefore confirms that high-quality source information enables models to make substantially better use of additional visual tokens.

Controlled sweeps support a frontend-dependent, grid-specific visual response. To identify what the fitted frontend contrast represents, we combine the effective-pixel comparison in Figure 11, the native-source intervention in Table 28 and Figure 12, and the source-quality factorial in Figure 13. The sweeps establish R as a genuine resource axis, while the factorial shows that its return depends on the usable image information summarized by P_{eff}. Together, these results provide stronger evidence for a frontend-dependent visual response than the fitted β values alone. They do not identify an intrinsic architectural exponent reversal, because the two families traverse different token and image-information ranges and the fixed-8B sweeps contain no variation in N from which to recover the ordering of α and β across model sizes. The VPE allocation rates $\beta/(p\beta + q\alpha)$ and $\alpha/(p\beta + q\alpha)$ should therefore be read as conditional on the evaluated grids. The two frontends therefore induce different fitted resource responses on the grids we evaluate, rather than fixed architecture-wide constants.

E FINDING 3: FROM LAWS TO PRACTICE

We first define the cost measure and test sensitivity to the FLOPs of the vision encoder and projector (Appendix E.1), then derive the continuous VPE allocation in a machine-checkable form (Appendix E.2). We next evaluate discrete choices on held-out configurations and in sample by series (Appendices E.3 and E.4), before conditioning the benchmark profiles on model family (Appendix E.5).

E.1 COST MEASURES AND COST-LAW FITS

Cost measure. We measure allocation cost in terms of the FLOPs the language backbone spends on the prompt,

$$C \;=\; 2\,P_{\mathrm{LLM}}\,T_{\mathrm{prompt}}, \tag{20}$$

where P_{LLM} is the exact parameter count of the language backbone and T_{prompt} is the logged prompt length, which includes the question text, the visual tokens, and the chat and special tokens. This is the standard $2NT$ measure, and it gives one accounting basis for both visual frontends. A variant that adds the short generation phase and the attention FLOPs, $C' = 2P_{\mathrm{LLM}}(T_{\mathrm{prompt}} + T_{\mathrm{gen}}) + C_{\mathrm{attn}}$, is available for the same 650 cells and leads to the same conclusions. The original FLOP logs cover the vision encoder and projector for only 257 and 150 of the 650 cells, under conventions that differ across series, so the main budget leaves them out. The visual-side sensitivity analysis below reconstructs these FLOPs for all 650 cells and tests their effect on the fitted allocation paths. The main allocation remains a language-side budget, and the expanded measure is a sensitivity analysis rather than an end-to-end system cost. The question-level robustness checks instead order configurations by $u = NR^{\mathrm{q}}$, which ranks the configurations of one trajectory and leaves out the question and template tokens that enter the budget here.

Fitted cost law. We fit the cost law $C = \kappa N^p R^q$ of §5.1 to all 650 cells pooled together and to each series separately, using the FLOPs above and the measured wall-clock latency as two candidate cost measures. The allocations in this paper use the pooled fit of the FLOPs measure together with the performance fit of each group, and the fits for each series are diagnostic rather than a source of budgets. Table 29 shows that the FLOPs measure follows this law in all five series, with R^2 in log space from 0.986 to 1.000. The fitted p is 1.00 in every series, as expected for the size of a transformer backbone. The fitted q is 0.96 in all three InternVL series and 0.75 to 0.78 in the two QwenVL series. The prompt also carries the question and the template, so cost grows less than proportionally with R over the wide token range of QwenVL, while the narrow range of InternVL keeps q near one. Wall-clock latency is not a stable cost axis across the two families. Its R^2 runs from 0.019 to 0.764 across series, which reflects scheduler noise, transfers between CPU and GPU, differences in batching, and implementation overheads that N and R do not capture. We therefore use the FLOPs measure as the common cost axis and treat latency as a property of the implementation rather than of the architecture.

Table 29: Cost-law fits $C = \kappa N^p R^q$, with N in billions of language backbone parameters and R in visual tokens. The law is fitted to the FLOPs measure and, separately, to wall-clock latency, and R^2 is computed in log space. Only the pooled fit of the FLOPs measure sets budgets, and the fits for each series and for latency are diagnostic.

		FLOPs of the language backbone				**Latency**		
Scope	n	κ	p	q	R^2	p	q	R^2
Pooled	650	1.24×10^{10}	**1.00**	**0.77**	**0.995**	0.07	0.54	0.366
InternVL2.5	150	2.95×10^{9}	1.00	0.96	1.000	-0.04	-0.15	0.019
InternVL3	150	2.96×10^{9}	1.00	0.96	1.000	0.14	-0.53	0.029
InternVL3.5	150	2.93×10^{9}	1.00	0.96	1.000	0.24	1.04	0.150
Qwen2.5-VL	100	1.38×10^{10}	1.00	0.75	0.986	0.03	0.43	0.764
Qwen3-VL	100	1.14×10^{10}	1.00	0.78	0.991	-0.14	0.45	0.543

Adding visual-side cost preserves the family-level allocation direction. To test whether omitting vision computation changes the allocation result, we extend the prompt cost to

$$C_\lambda = C_{\text{LLM,prompt}} + \lambda\big(C_{\text{vision}} + C_{\text{projector}}\big), \qquad \lambda \in \{0, 0.5, 1, 2\}, \tag{21}$$

over all 650 configurations, 26 models, and five series. For Qwen2.5-VL, the vision cost includes both window and global attention, and for Qwen3-VL it includes global attention and the three DeepStack mergers. For each question, we use the actual temporal, height, and width of its visual patch grid to determine the tensor shapes, compute the visual cost, and then average within a cell. For InternVL, we retain the original parameter-times-token estimator, which reproduces the historical component accounting on all 150 InternVL3 cells. We then keep the fitted performance surfaces fixed and recompute their finite-candidate allocation paths under each C_λ. On the same 101 log-spaced budgets supported by all four values of λ, every one of the 12 InternVL groups has a larger growth slope in $\log N$ than in $\log R$, while all eight QwenVL groups show the reverse. An oracle path constructed directly from the observed errors agrees in 19 of the 20 groups, with only Qwen3-VL on TreeBench reversing direction. At the original four absolute budgets and $\lambda = 1$, 10 of the 80 decisions become infeasible and 39 of the 70 commonly feasible decisions select the same configuration as the language-only cost. Thus the family-level allocation direction is stable after adding the visual-side cost, while absolute budgets and individual choices must be recalibrated under the new cost definition. This in-sample check excludes language-model attention and decoding, preprocessing, communication, and measured latency, and therefore does not claim an end-to-end system cost. Because fixed α and β already determine the ordering of the continuous elasticities, the comparison here is made on the finite candidate paths.

E.2 Derivation of the VPE Allocation

What this derivation covers. The fitting procedure estimates the parameters of the Separable Law and of the cost law from data. Given those two laws and positive coefficients and exponents, the derivation below produces the continuous VPE solution and its budget elasticities. It therefore says nothing about how well the two laws fit, or about what happens outside the observed grid.

From the fitted amplitudes to physical coordinates. The fits store the amplitudes at the common reference point $(\widetilde{N}, \widetilde{R})$,

$$L(N, R) = e_\infty + A_n \left(\frac{N}{\widetilde{N}}\right)^{-\alpha} + B_r \left(\frac{R}{\widetilde{R}}\right)^{-\beta}, \tag{22}$$

while the cost law is written in the resource variables themselves. Absorbing the reference scales into the amplitudes, $A = A_n \widetilde{N}^\alpha$ and $B = B_r \widetilde{R}^\beta$, leaves every predicted error unchanged and returns Eq.(3) in physical coordinates. The derivation below uses A and B, and N carries the same unit as in the pooled cost fit, namely billions of parameters. Changing that unit to $N' = sN$ takes $\widetilde{N}$ to $s\widetilde{N}$ and κ to κ/s^p, which leaves the allocation itself unchanged. This conversion leaves the allocation invariant to the unit chosen for backbone size.

The problem. We take positive resources, a positive budget, and strictly positive coefficients and exponents on both the error side and the cost side,

$$A, B, \alpha, \beta, p, q, \kappa, C_0 > 0. \tag{23}$$

The continuous allocation problem then minimizes error over the positive resource pairs that the budget allows,

$$\min_{N,R>0} \left[e_\infty + AN^{-\alpha} + BR^{-\beta}\right] \quad \text{subject to} \quad \kappa N^p R^q \le C_0. \tag{24}$$

Error falls in both N and R while cost rises in both, so any optimum spends the whole budget and the inequality becomes an equality. The search therefore runs along the boundary of fixed cost.

For the closed form, write the rescaled budget and two positive constants,

$$\overline{C} = \frac{C_0}{\kappa}, \qquad D = p\beta + q\alpha, \qquad K = \frac{\alpha A q}{\beta B p}. \tag{25}$$

The assumptions give $\overline{C} > 0$, $D > 0$, and $K > 0$, so the candidate below is well defined and assigns positive resources to both axes,

$$N^* = \overline{C}^{\beta/D} K^{q/D}, \qquad R^* = \overline{C}^{\alpha/D} K^{-p/D}. \tag{26}$$

The closed form and its uniqueness. In log coordinates $x = \log N$ and $y = \log R$, the budget constraint becomes affine, and the positive (N, R) domain maps to all of $\mathbb{R}^2$. On the boundary of fixed cost,

$$px + qy = \log \overline{C}. \tag{27}$$

At a stationary allocation, the error removed per proportional unit of spending is equal on the two axes,

$$\frac{\alpha A}{p} N^{-\alpha} = \frac{\beta B}{q} R^{-\beta}, \tag{28}$$

and taking logarithms turns this balance into a second linear relation,

$$\alpha x - \beta y = \log K. \tag{29}$$

Since $D = p\beta + q\alpha > 0$, the system of Eqs.(27) and (29) has the single solution

$$x^* = \frac{\beta}{D} \log \overline{C} + \frac{q}{D} \log K, \qquad y^* = \frac{\alpha}{D} \log \overline{C} - \frac{p}{D} \log K, \tag{30}$$

and exponentiating it recovers Eq.(26). This gives one stationary candidate on the boundary, and global optimality still needs an argument.

To supply it, eliminate y through the constraint, $y = (\log \overline{C} - px)/q$, and write the objective on the boundary as a function of x alone,

$$G(x) = e_\infty + Ae^{-\alpha x} + B\,\overline{C}^{-\beta/q} e^{(\beta p/q)x}. \tag{31}$$

The floor does not depend on the allocation, and differentiating the other two terms twice gives

$$G''(x) = \alpha^2 Ae^{-\alpha x} + \left(\frac{\beta p}{q}\right)^2 B\,\overline{C}^{-\beta/q} e^{(\beta p/q)x} > 0, \tag{32}$$

so G is strictly convex along the whole boundary. It also diverges at both ends, since $G(x) \to +\infty$ as $x \to -\infty$ and as $x \to +\infty$, so a minimizer exists and strict convexity makes it unique. Its stationarity condition is Eq.(28), so the minimizer is exactly the pair in Eq.(26).

Budget elasticities. Holding the fitted parameters fixed and taking logarithms of Eq.(26) gives

$$\log N^* = \frac{\beta}{D}\big(\log C_0 - \log \kappa\big) + \frac{q}{D} \log K, \tag{33}$$

$$\log R^* = \frac{\alpha}{D}\big(\log C_0 - \log \kappa\big) - \frac{p}{D} \log K. \tag{34}$$

Both are affine in the log budget, and their slopes are the proportional response of each optimal resource to a proportional increase in budget,

$$\frac{\partial \log N^*}{\partial \log C_0} = \frac{\beta}{p\beta + q\alpha}, \qquad \frac{\partial \log R^*}{\partial \log C_0} = \frac{\alpha}{p\beta + q\alpha}. \tag{35}$$

The level of the allocation depends on the coefficients as well, while these two rates depend only on α, β, p, and q. They describe continuous growth rather than a choice among available configurations.

Lean 4 certifies the algebraic core of the allocation. To verify the symbolic derivation independently of the fitting code, we formalize its log-coordinate core as a 2×2 linear system. The Lean 4 source below records the positivity assumptions, checks the budget and balance identities, proves that their intersection is unique, and verifies the finite-difference form of the two budget elasticities. It compiles with Lean 4.34.0 and the matching Mathlib release.

Lean 4 — VPE certificate

```
import Mathlib

noncomputable section
open Real

```

Lean 4 — VPE certificate (continued)

```
namespace VPE

structure Params where
  A      : Real
  B      : Real
  alpha  : Real
  beta   : Real
  p      : Real
  q      : Real
  kappa  : Real
  C0     : Real
  hA     : 0 < A
  hB     : 0 < B
  ha     : 0 < alpha
  hb     : 0 < beta
  hp     : 0 < p
  hq     : 0 < q
  hk     : 0 < kappa
  hC0    : 0 < C0

def D (t : Params) : Real :=
  t.p * t.beta + t.q * t.alpha

def Cbar (t : Params) : Real :=
  t.C0 / t.kappa

def K (t : Params) : Real :=
  t.alpha * t.A * t.q /
    (t.beta * t.B * t.p)

lemma D_pos (t : Params) : 0 < D t := by
  unfold D
  exact add_pos
    (mul_pos t.hp t.hb)
    (mul_pos t.hq t.ha)

lemma Cbar_pos (t : Params) : 0 < Cbar t := by
  unfold Cbar
  exact div_pos t.hC0 t.hk

lemma K_pos (t : Params) : 0 < K t := by
  unfold K
  exact div_pos
    (mul_pos (mul_pos t.ha t.hA) t.hq)
    (mul_pos (mul_pos t.hb t.hB) t.hp)

def xStar (t : Params) : Real :=
  t.beta / D t * Real.log (Cbar t) +
  t.q / D t * Real.log (K t)

def yStar (t : Params) : Real :=
  t.alpha / D t * Real.log (Cbar t) -
  t.p / D t * Real.log (K t)

theorem log_budget (t : Params) :
    t.p * xStar t + t.q * yStar t =
      Real.log (Cbar t) := by
  unfold xStar yStar
  have hD : D t ≠ 0 := ne_of_gt (D_pos t)
  field_simp [hD]
  unfold D
  ring
```

Lean 4 — VPE certificate (continued)

```
68
69  theorem log_balance (t : Params) :
70      t.alpha * xStar t - t.beta * yStar t =
71        Real.log (K t) := by
72    unfold xStar yStar
73    have hD : D t ≠ 0 := ne_of_gt (D_pos t)
74    field_simp [hD]
75    unfold D
76    ring
77
78  theorem unique_log_solution
79      (t : Params) {x y : Real}
80      (hb :
81        t.p * x + t.q * y = Real.log (Cbar t))
82      (he :
83        t.alpha * x - t.beta * y = Real.log (K t)) :
84      x = xStar t /\ y = yStar t := by
85
86    have hD : D t ≠ 0 := ne_of_gt (D_pos t)
87
88    have hx :
89        D t * x =
90          t.beta * Real.log (Cbar t) +
91          t.q * Real.log (K t) := by
92      unfold D
93      linear_combination t.beta * hb + t.q * he
94
95    have hy :
96        D t * y =
97          t.alpha * Real.log (Cbar t) -
98          t.p * Real.log (K t) := by
99      unfold D
100     linear_combination t.alpha * hb - t.p * he
101
102   constructor
103   · unfold xStar
104     field_simp [hD]
105     nlinarith [hx]
106
107   · unfold yStar
108     field_simp [hD]
109     nlinarith [hy]
110
111 def logNAtBudget (t : Params) (s : Real) : Real :=
112   t.beta / D t * s +
113   t.q / D t * Real.log (K t)
114
115 def logRAtBudget (t : Params) (s : Real) : Real :=
116   t.alpha / D t * s -
117   t.p / D t * Real.log (K t)
118
119 theorem logN_budget_increment
120     (t : Params) (s h : Real) :
121     logNAtBudget t (s + h) - logNAtBudget t s =
122       (t.beta / D t) * h := by
123   unfold logNAtBudget
124   ring
125
126 theorem logR_budget_increment
127     (t : Params) (s h : Real) :
128     logRAtBudget t (s + h) - logRAtBudget t s =
129       (t.alpha / D t) * h := by
```

Lean 4 — VPE certificate (continued)

```
  unfold logRAtBudget
  ring

end VPE
```

The listing certifies the algebraic core in log coordinates, namely that the closed form satisfies the budget equation and the balance equation, that their intersection is unique, and that the two log-budget responses have the stated slopes. The convexity and the limits that give global optimality come from the argument above rather than from the listing. The values of A_n, B_r, α, β, p, q, and κ remain outputs of the fitting pipeline, which the listing says nothing about. A continuous optimum also does not name the best configuration in a finite set, so discrete selection must be evaluated separately. The certificate therefore provides an independent, machine-checked validation of the allocation algebra while keeping its empirical and discrete-selection claims clearly separated.

E.3 Held-Out Allocation

Joint holdouts test whether the fitted law can allocate unseen configurations. This is the evaluation behind §5.2. For each of the 20 groups we jointly hold out one backbone size and one input size, removing the full row and column they span, and refit the performance law using only the remaining cells. The resulting 650 folds provide withheld candidate configurations, which we evaluate at budgets equal to the 25th, 50th, 75th, and 90th percentiles of each group's language-side cell costs. Regret is the measured error of the selected configuration minus the lowest measured error among the feasible candidates, in percentage points. After excluding 44 decisions with no feasible candidate and 177 with only one, the evaluation retains 2,379 of the 2,600 planned decisions. We compare two rules that use the fitted law with three static baselines. Projection clips the continuous optimum of Eq.(26) to the observed coordinate ranges and selects the nearest feasible candidate in log coordinates, whereas direct selection evaluates Eq.(3) over every feasible candidate and takes the lowest predicted error. The static rules select the largest feasible backbone at the smallest visual token count, the largest feasible visual token count at the smallest backbone, or the candidate nearest the log-center of the feasible ranges; ties are broken by cell order without measured errors. Because the closed form degenerates when either fitted amplitude approaches zero, projection falls back to direct selection when $A_n < 10^{-8}$ or $B_r < 10^{-8}$, which occurs in 133 decisions. The same folds and budgets also support the comparison between the Separable and bounded laws.

The fitted law substantially reduces held-out allocation regret. Table 30 gives the regret of the five rules over the 2,379 decisions. Direct selection carries a mean regret of 0.76 pp and a 90th percentile of 2.74 pp, and projection carries 2.33 pp and 7.71 pp. The three static rules run from 4.09 to 7.56 pp in the mean and from 12.23 to 20.14 pp at the 90th percentile, so both rules that read the fitted law stay below all three on those two measures. The worst-case result differs: projection reaches 34.55 pp, compared with 27.00 pp for the rule that takes the largest visual token count, while direct selection stays at 19.37 pp. Both law-based rules read the same fitted law over the same candidates, so the gap between them comes from the projection step rather than from the law. Under the bounded-law robustness check, direct selection has nearly identical mean and 90th-percentile regrets of 0.75 pp and 2.74 pp.

E.4 In-Sample Allocation Analysis

Projection outperforms static allocation rules on the complete grid. To diagnose allocation behavior by series and identify the largest-regret cases, we retain an earlier in-sample evaluation in which the fit and evaluation use the same cells; it is descriptive and does not support the held-out claim in §5.2. For each of the 20 groups, four budgets C_0 sit at the 25th, 50th, 75th, and 90th percentiles of the cell costs of that group, which gives 80 decisions. The cost is the language-side FLOP measure computed from the logged prompt lengths. All decisions use the pooled cost fit together with the performance fit of their group. For every budget we clip the two coordinates of the continuous optimum separately to the ranges observed in the group, and then pick the observed cell nearest in Euclidean distance in $(\log N, \log R)$ among the cells with $C_{\text{cell}} \le C_0$. Clipping need not

Table 30: Allocation regret in percentage points over the 2,379 joint-holdout decisions. All five rules see the same candidates under the same budgets, the 90th percentile uses linear interpolation, and Worst is the largest regret of a single decision. The two rules above the horizontal line read the fitted Separable law, and the three below it do not.

Rule	Mean	90th pct.	Worst
Direct selection	**0.76**	**2.74**	**19.37**
Projection	2.33	7.71	34.55
Largest token count	4.09	12.23	27.00
Largest backbone	7.21	20.14	42.93
Log-center	7.56	16.22	38.74

keep the fitted cost equality, and picking the nearest cell need not minimize the fitted error over the finite grid. The comparator is the cell with the lowest measured error in the same feasible set. Regret is the error of the chosen cell minus the lowest feasible error, multiplied by 100 to give percentage points, and all 80 regrets are nonnegative. The three static rules choose the largest feasible backbone, the largest feasible visual token count, or the log-center of the feasible ranges. Ties are broken by dataset order, quantiles use linear interpolation, and all summaries use unrounded regrets. Projection has lower mean, median, 90th-percentile, and worst-case regret than every static rule and a higher exact-hit rate (Table 31). Its mean regret is 1.98 pp, its median is 0.00 pp, its 90th percentile is 6.34 pp, and its worst case is 11.52 pp, compared with the best static mean of 6.51 pp from the largest-token-count rule. Although the static-rule ordering changes by benchmark, projection has the lowest median regret on all four benchmarks (Table 32). Thus the fitted allocation gives substantially closer choices than fixed resource heuristics on the complete grid.

Table 31: In-sample allocation regret in percentage points over the same 80 decisions, for projection and the three static rules. Exact hit counts the decisions that select the designated best cell, with ties broken by dataset order.

Rule	Mean	Median	90th pct.	Worst	Exact hit
Projection	**1.98**	**0.00**	**6.34**	**11.52**	**51.25%**
Largest token count	6.51	6.10	13.45	19.37	27.50%
Log-center	8.05	7.70	16.24	20.00	1.25%
Largest backbone	15.40	14.23	28.32	41.36	0.00%

Table 32: Median allocation regret in percentage points by benchmark, 20 decisions each, from the same evaluation. Each entry pools four budgets across five model series.

Benchmark	Projection	Largest token count	Log-center	Largest backbone
HR-Bench	**0.00**	5.63	11.06	14.23
MME-RealWorld	**0.07**	3.54	6.52	17.77
TreeBench	**2.49**	6.09	4.10	4.85
V* Bench	**0.00**	7.85	9.16	27.23

Projection remains close to the optimum across series and families. To determine whether the aggregate gain depends on a narrow subset of groups, we measure exact recovery and regret tolerance and break the same 80 decisions down by series, benchmark, and family. Projection selects the designated best cell in 41 decisions, or 51.25%, while 42, or 52.50%, reach the lowest feasible error; the difference is one tied decision for InternVL3 on V* Bench at the 75th-percentile budget, where 9B and 8B at 2048 px have the same error. Model sizes here follow the model names, and the number after the size is the input size. Table 33 shows that 52 decisions, or 65.00%, lie within 2 pp of the optimum and 77, or 96.25%, lie within 8 pp; all 32 QwenVL decisions and 45 of the 48 InternVL decisions meet the 8 pp threshold. Table 34 reports the four decisions in each of the 20 groups, and

Eq.(36) shows how their unrounded group means produce the pooled mean of 1.98 pp in Table 31; the held-out means in §5.2 come from a separate evaluation, and pooled medians and percentiles are computed directly over the 80 regrets. InternVL has a mean regret of 2.16 pp, a median of 0.00 pp, a 90th percentile of 7.06 pp, and a worst case of 11.52 pp over 48 decisions, while QwenVL has 1.71 pp, 0.00 pp, 4.70 pp, and 7.85 pp over 32 decisions. For family f, the empirical distribution function is $\widehat{F}_f(t) = n_f^{-1} \sum_{i \in f} \mathbf{1}\{\text{regret}_i \le t\}$, with $n_{\text{InternVL}} = 48$ and $n_{\text{QwenVL}} = 32$; the two curves cross, so the family ordering depends on the threshold, but projection remains within 8 pp of the optimum for nearly every decision in both families.

$$\overline{\text{regret}}_{\text{pooled}} = \frac{1}{20} \sum_{g=1}^{20} \left(\frac{1}{4} \sum_{b=1}^{4} \text{regret}_{g,b} \right) = 1.979870 \text{ pp} \simeq 1.98 \text{ pp}. \tag{36}$$

Table 33: Share of projection decisions with regret at or below each threshold. The denominators are 80 decisions pooled, 48 for InternVL, and 32 for QwenVL. The zero-regret row counts every tied minimum, whichever cell was chosen.

Threshold	Pooled	InternVL	QwenVL
≤ 0 pp	52.50%	52.08%	53.13%
≤ 1 pp	58.75%	58.33%	59.38%
≤ 2 pp	65.00%	64.58%	65.63%
≤ 5 pp	85.00%	81.25%	90.63%
≤ 8 pp	96.25%	93.75%	100.00%
≤ 10 pp	98.75%	97.92%	100.00%

Table 34: Projection regret in percentage points by series and benchmark, from the same 80 decisions as Table 31. Each row summarizes four decisions, computed from unrounded regrets, and the 90th percentile uses linear interpolation. Hits count the selections of the designated best cell, with ties broken by dataset order. For InternVL3 on V* Bench, all four decisions reach the lowest feasible error and one of them selects a different cell tied with the designated best.

Series	Benchmark	Mean	Median	90th pct.	Worst	Hits
InternVL2.5	HR-Bench	2.16	0.00	6.04	8.63	3/4
InternVL2.5	MME-RealWorld	3.44	2.02	8.03	9.74	2/4
InternVL2.5	TreeBench	3.11	2.74	6.07	6.97	1/4
InternVL2.5	V* Bench	4.19	2.62	9.63	11.52	2/4
InternVL3	HR-Bench	0.44	0.00	1.23	1.76	3/4
InternVL3	MME-RealWorld	2.16	0.68	5.51	7.30	2/4
InternVL3	TreeBench	3.98	3.61	6.89	7.71	0/4
InternVL3	V* Bench	0.00	0.00	0.00	0.00	3/4
InternVL3.5	HR-Bench	0.00	0.00	0.00	0.00	4/4
InternVL3.5	MME-RealWorld	1.51	0.32	3.94	5.42	1/4
InternVL3.5	TreeBench	3.52	4.31	5.25	5.47	1/4
InternVL3.5	V* Bench	1.44	1.05	3.19	3.66	2/4
Qwen2.5-VL	HR-Bench	3.38	3.31	6.24	6.88	1/4
Qwen2.5-VL	MME-RealWorld	1.66	1.20	3.68	4.22	2/4
Qwen2.5-VL	TreeBench	3.11	3.23	3.91	3.98	0/4
Qwen2.5-VL	V* Bench	3.53	3.14	7.38	7.85	2/4
Qwen3-VL	HR-Bench	1.06	0.06	2.93	4.13	2/4
Qwen3-VL	MME-RealWorld	0.72	0.00	2.01	2.87	3/4
Qwen3-VL	TreeBench	0.19	0.00	0.52	0.75	3/4
Qwen3-VL	V* Bench	0.00	0.00	0.00	0.00	4/4

Large regrets are rare and localized to three InternVL2.5 decisions. Exactly three decisions exceed 8 pp, and all three come from InternVL2.5. On V* Bench at the 75th-percentile budget, the rule selects 26B at 512 px against a best feasible 8B at 2048 px, with a regret of 11.52 pp and $\beta = 10.00$. On MME-RealWorld at the 25th-percentile budget, it selects 2B at 512 px against 1B at 2048 px, with 9.74 pp and $\beta = 10.00$. On HR-Bench at the 25th-percentile budget, it selects 2B at 768 px against 1B at 4096 px, with 8.63 pp and $\beta = 4.6623$, under the same budget for both cells. The HR-Bench case has a β away from the bounds, so these tails are not confined to the fits that reach the upper bound. The in-sample gains are therefore broad rather than hiding many large failures, although these three cases do not identify an architectural cause because both fitted-surface error and the clipping and projection step can produce regret on a finite grid.

E.5 Benchmark Profiles by Family

Family-specific profiles reveal different dominant resource axes. To test whether pooled benchmark profiles conceal frontend-specific scaling behavior, we split the 19,890 trajectories by benchmark and model family, report the three regime shares in Table 35, and within the Scaling Bound subset compare the range of mean accuracy along the backbone and visual-token axes in Table 36. We call a trajectory N-dominated when its accuracy range is larger along the backbone axis and R-dominated when it is larger along the visual-token axis. This measures the magnitude of movement rather than whether adding the resource improves accuracy. Of the 193 R-dominated QwenVL trajectories on V* Bench, for instance, 9 are less accurate and 11 are equally accurate at the largest input size as at the smallest. HR-Bench requires an additional check because 1,423 Scaling Bound trajectories exist only at 4096 px, 1,007 from InternVL and 416 from QwenVL, so their visual-token range is zero and they are classified as N-dominated by construction. After retaining only trajectories with at least two input sizes, the N-dominated, R-dominated, and tie shares are 82.2%, 11.6%, and 6.2% for InternVL and 46.3%, 45.8%, and 7.9% for QwenVL, so InternVL is more backbone-dominated under either definition. Across the accuracy cutoffs, MME-RealWorld and TreeBench have the largest Ceiling Bound shares in both families, from 38.1% to 42.0%, and the benchmark contrast also holds under the response-based criteria even though the exact shares depend on the criterion and cohort. The family difference is clearest in the dominant axis. On every benchmark the R-dominated share of QwenVL exceeds that of InternVL, from 31.6% versus 6.7% on HR-Bench to 72.6% versus 38.5% on V* Bench, and on MME-RealWorld QwenVL is majority R-dominated at 53.5% even though the pooled profile is N-dominated. Family-level analysis therefore gives a consistent result. QwenVL trajectories are more sensitive to the visual-token axis, whereas InternVL trajectories are more often separated by backbone size, so benchmark scaling profiles should be interpreted together with the visual frontend.

Table 35: Regime shares for each benchmark over the 19,890 trajectories, under the accuracy cutoffs of §3.2. A separate response-based analysis uses a smaller matched cohort and stricter criteria.

Benchmark	Easy	Scaling Bound	Ceiling Bound
HR-Bench	33.5%	46.5%	**20.0%**
MME-RealWorld	10.3%	50.3%	**39.4%**
TreeBench	10.4%	48.6%	**41.0%**
V* Bench	16.2%	**66.2%**	17.6%

F Case Studies

We close with two sets of examples, one that holds the model fixed and raises the input size, and one that holds the input size fixed and compares model sizes. These are individual questions chosen to show what changes along each axis rather than averages over the full grid. The outputs alone also leave open why an answer changes.

Sensitivity to input size across model sizes. Figure 14 shows selected outputs of the 2B, 4B, 8B, and 32B models of Qwen3-VL at input sizes from 224 to 2048 px. The two 32B examples become correct at 768 and 512 px and stay correct at the larger input sizes. The 8B example asks the

Table 36: Regime shares and dominant axis for each benchmark and family, over the same 19,890 trajectories. Regime shares are fractions of all trajectories of that family and benchmark, and the dominant axis is computed over the SCALING BOUND trajectories only. The HR-Bench rows include the trajectories that exist only at 4096 px, which the text discusses.

		Regime share			Dominant axis within SCALING BOUND		
Family	**Benchmark**	**EASY**	**SCALING**	**CEILING**	***N*-dominated**	***R*-dominated**	**Tie**
InternVL	HR-Bench	29.0%	49.6%	21.4%	89.7%	6.7%	3.6%
InternVL	MME-RealWorld	10.1%	51.8%	38.1%	70.5%	20.7%	8.8%
InternVL	TreeBench	8.6%	49.4%	42.0%	88.5%	7.0%	4.5%
InternVL	V^* Bench	16.8%	63.9%	19.4%	47.8%	38.5%	13.7%
QwenVL	HR-Bench	40.3%	41.8%	17.9%	63.0%	31.6%	5.5%
QwenVL	MME-RealWorld	10.7%	48.0%	41.2%	40.4%	53.5%	6.1%
QwenVL	TreeBench	13.1%	47.4%	39.5%	58.6%	35.9%	5.5%
QwenVL	V^* Bench	15.4%	69.6%	14.9%	22.6%	72.6%	4.9%

model to identify a face, and it is answered incorrectly from 224 to 768 px and correctly at 1024 and 2048 px. Of the two 4B examples, one becomes correct at 512 px and the remote-sensing example at 768 px; both stay correct above. In these five examples a larger input helps on questions that turn on a small detail. The 2B example moves in both directions instead, since it is answered correctly at 224 and 336 px, incorrectly from 512 to 1024 px, and correctly again at 2048 px. Raising the input size therefore need not help every individual question.

Sensitivity to model size across input sizes. Figure 15 compares the four models on selected questions, with the input size held fixed within each example. At 224 px the question asks for the color of a shirt, and the 8B and 32B models answer it correctly while the 2B and 4B models do not. The same pattern holds for the text on a billboard at 512 px, the location on a map at 768 px, the fax number at 1024 px, and the color of a bicycle basket at 2048 px. At 336 px the question asks for the color of a flag, and there the 4B model answers correctly as well. These examples show errors that persist in the smaller models even at a large input size. Each panel fixes the input size and different panels show different questions, so they say nothing about how the gap between model sizes changes with the input size.

Data Source: MME-RealWorld-Lite
Model: Qwen3-VL-2B
Category: Perception
Visual Domains: OCR



Question: What is the discount band for the product "Paseo" in Germany in the bottom right table?

A) Low
B) High
C) None
D) Medium
E) The image does not feature the number.

Ground Truth: C

Answer:

Resolution: 224 C
Resolution: 336 C
Resolution: 512 A
Resolution: 768 A
Resolution: 1024 A
Resolution: 2048 C

Data Source: TreeBench
Model: Qwen3-VL-4B
Category: Reasoning
Visual Domains: Natural Scenes



Question: From the viewpoint of the male individual Gressed in loose-fitting dark attire, who is holding a beverage in his left hand, what is the relative directional position of the novelty trash can shaped like a clown?

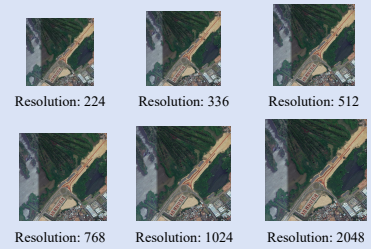
A) Front left
B) Front right
C) Left rear
D) Right rear

Ground Truth: A

Answer:

Resolution: 224 C
Resolution: 336 C
Resolution: 512 A
Resolution: 768 A
Resolution: 1024 A
Resolution: 2048 A

Data Source: MME-RealWorld-Lite
Model: Qwen3-VL-4B
Category: Perception
Visual Domains: Remote Sensing



Question: Where is the green rectangular grassland with a white circular pattern in the picture?

A) In the lower right corner of the picture, near the mound.
B) In the upper left corner of the picture.
C) In the center of the picture.
D) In the lower left corner of the picture.
E) This image doesn't feature the position.

Ground Truth: A

Answer:

Resolution: 224 C
Resolution: 336 C
Resolution: 512 C
Resolution: 768 A
Resolution: 1024 A
Resolution: 2048 A

Figure 14: **Sensitivity to input size across model sizes.** Selected outputs of Qwen3-VL at 2B, 4B, 8B, and 32B and at input sizes from 224 to 2048 px. Green marks the correct option and red marks an incorrect answer. This page shows the 2B and 4B models. The 2B example moves in both directions; one 4B example becomes correct at 512 px and the remote-sensing example at 768 px.

Data Source: HR-Bench
Model: Qwen3-VL-8B
Category: Reasoning
Visual Domains: Natural Scenes



Question: Which president's face is highlighted in the red rectangle on Mount Rushmore?

A) Theodore Roosevelt
B) George Washington
C) Abraham Lincoln
D) Thomas Jefferson

Ground Truth: D

Answer:
Resolution: 224 B
Resolution: 336 B
Resolution: 512 A
Resolution: 768 A
Resolution: 1024 D
Resolution: 2048 D

Data Source: Vstar Bench
Model: Qwen3-VL-32B
Category: Perception
Visual Domains: Natural Scenes



Question: Does the kid have curly hair or straight hair?

A) curly hair
B) straight hair

Answer with the option's letter from the given choices directly.

Ground Truth: A

Answer:
Resolution: 224 B
Resolution: 336 B
Resolution: 512 B
Resolution: 768 A
Resolution: 1024 A
Resolution: 2048 A

Data Source: MME-RealWorld-Lite
Model: Qwen3-VL-32B
Category: Reasoning
Visual Domains: Natural Scenes



Question: This image shows the front view of the ego car. What is the future state of the person wearing a white shirt in the middle?

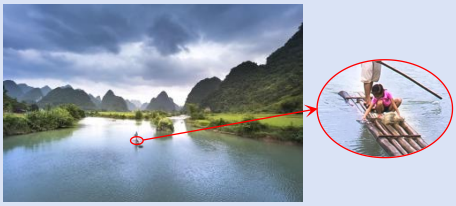
A) Keep going straight
B) Turn left
C) Stationary
D) Turn right
E) The image does not feature the object.

Ground Truth: C

Answer:
Resolution: 224 E
Resolution: 336 E
Resolution: 512 C
Resolution: 768 C
Resolution: 1024 C
Resolution: 2048 C

Figure 14 (continued): Sensitivity to input size for the 8B and 32B models. The 8B answer becomes correct at 1024 px, and the two 32B answers become correct at 768 and 512 px.

Data Source: HR-Bench
Resolution: 224
Category: Perception
Visual Domains: Natural Scenes



Question: What color is the girl's shirt?

A) Pink
B) Blue
C) Yellow
D) White

Ground Truth: A

Answer:
Qwen3-VL-2B: B Qwen3-VL-4B: B
Qwen3-VL-8B: A Qwen3-VL-32B: A

Data Source: VStar Bench
Resolution: 336
Category: Perception
Visual Domains: Natural Scenes



Question: What is the color of the flag?

A) red and yellow
B) blue and yellow
C) blue and red
D) green and white
Answer with the option's letter from the given choices directly.

Ground Truth: B

Answer:
Qwen3-VL-2B: A Qwen3-VL-4B: B
Qwen3-VL-8B: B Qwen3-VL-32B: B

Data Source: MME-RealWorld-Lite
Resolution: 512
Category: Perception
Visual Domains: OCR



Question: What is the largest black text on the billboard behind the golden sculpture in the middle of the picture?

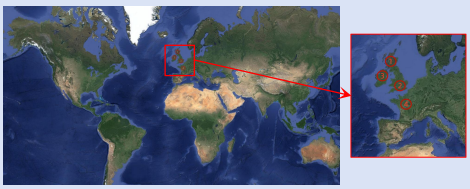
A) RUNDKINO
B) NUR BEI ANS
C) SACHSEN DER FRAUEN
D) NUR BEI UNS
E) The image doesn't feature the content.

Ground Truth: D

Answer:
Qwen3-VL-2B: A Qwen3-VL-4B: E
Qwen3-VL-8B: D Qwen3-VL-32B: D

Figure 15: **Sensitivity to model size across input sizes.** Each panel compares the Qwen3-VL models on one question at a fixed input size. These three examples use 224, 336, and 512 px, and the question differs from panel to panel.

Data Source: HR-Bench
Resolution: 768
Category: Reasoning
Visual Domains: Remote Sensing



Question: On the provided map, which two numbered locations belong to the same country?

A) Locations 1 and 3
B) Locations 1 and 2
C) Locations 2 and 4
D) Locations 2 and 3

Ground Truth: B

Answer:
Qwen3-VL-2B: A Qwen3-VL-4B: A
Qwen3-VL-8B: B Qwen3-VL-32B: B

Data Source: MME-RealWorld-Lite
Resolution: 1024
Category: Perception
Visual Domains: OCR



Question: What is the fax number described at the bottom of the image promotional page?

A) -515-280-7037
B) -551-280-7037
C) -515-280-7073
D) -515-288-7037
E) This image doesn't feature the number.

Ground Truth: A

Answer:
Qwen3-VL-2B: C Qwen3-VL-4B: C
Qwen3-VL-8B: A Qwen3-VL-32B: A

Data Source: TreeBench
Resolution: 2048
Category: Reasoning
Visual Domains: Natural Scenes



Question: What color is the basket of the second bicycle from right to left?

A) White
B) Black
C) Blue
D) Pink

Ground Truth: B

Answer:
Qwen3-VL-2B: A Qwen3-VL-4B: A
Qwen3-VL-8B: B Qwen3-VL-32B: B

Figure 15 (continued): Model comparisons at 768, 1024, and 2048 px. In these three examples the 8B and 32B models answer correctly while the 2B and 4B models do not.