How Far Are We from Removing the Visual Encoder? Scaling Laws for Encoder-Free Multimodal Pretraining

Anonymous authors
Paper under double-blind review

Abstract

Most modern multimodal large language models (MLLMs) build on a pretrained visual encoder that provides a strong visual prior. Encoder-free MLLMs instead learn visual representations directly from raw pixels, offering a simple and unified architecture, but their scaling behavior has not been systematically characterized. To fill this gap, we compare scaling laws for encoder-free and encoder-based MLLMs and report three main findings: **(1)** Removing the visual encoder shifts the compute-optimal allocation for the multimodal objective toward larger models, while leaving that for text nearly unchanged. **(2)** The two architectures exhibit nearly overlapping loss–compute frontiers on the text objective, but diverge on the multimodal objective: encoder-free models underperform at small scales yet are predicted to catch up at around 10^{22} FLOPs, well within practical pretraining budgets. **(3)** Without a visual encoder, the language model learns to take over its role via vision-specific adaptation: bidirectional interactions among visual tokens become increasingly beneficial as training compute grows, visual processing shifts toward earlier layers, and expert routing for visual tokens becomes more concentrated. Overall, our results indicate that the advantage of the visual prior provided by a pretrained encoder diminishes with scale, positioning encoder-free architectures as a promising direction for multimodal pretraining.

1 Introduction

Most modern multimodal large language models (MLLMs) (Liu et al., 2023; Bai et al., 2025; Gemma Team et al., 2025) adopt an encoder-based architecture: a pretrained visual encoder (Radford et al., 2021; Tschannen et al., 2025) supplies the language model with semantically rich visual representations, providing a strong visual prior learned from large-scale image–text data. To achieve a simple and unified architecture, encoder-free MLLMs remove the visual encoder and feed projected image patches directly into the decoder, which must then learn visual representations from raw pixels (Bavishi et al., 2023; Diao et al., 2024; Chen et al., 2024b; Lei et al., 2025; Diao et al., 2026a; Gemma Team et al., 2026; Liu et al., 2026; Thinking Machines Lab, 2026). Although these studies show initial feasibility, the scaling behavior of encoder-free MLLMs has not been systematically characterized.

To this end, we conduct a controlled scaling study of encoder-free and encoder-based MLLMs, in which the two model families share the same sparse decoder ladder, data mixture, optimization setup, and visual-token granularity. We fit scaling laws separately for the text and multimodal objectives to quantify how the efficiency gap between the two architectures evolves with scale. To understand the mechanisms behind these trends, we further probe the decoder's internals and examine how it compensates for the missing visual encoder. Our main findings are as follows.

(1) Compute-optimal encoder-free training favors larger models (§3.1). For the text objective, the two architectures exhibit nearly identical compute-optimal allocation trends. In contrast, the multimodal objective shows a different pattern: removing the visual encoder increases the model allocation exponent from $a = 0.464$ to $a = 0.570$, shifting the optimum toward larger model scale. This shift suggests that encoder-free models require greater decoder capacity to jointly support visual representation learning and language modeling.

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.

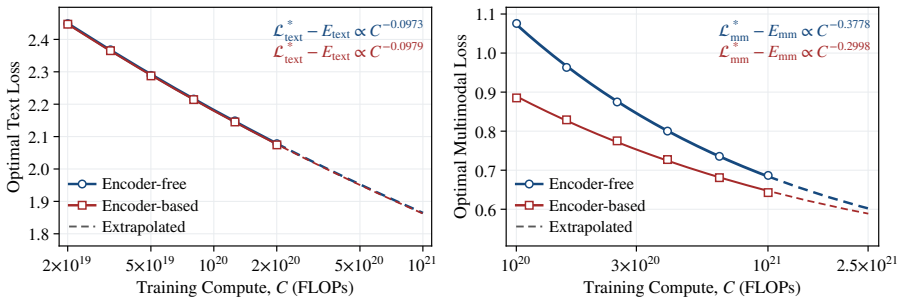


Figure 1: Compute-optimal loss frontiers for encoder-free and encoder-based models. **Left:** The text loss frontiers of the two architectures nearly overlap. **Right:** Encoder-free models have higher multimodal loss over the measured range, but their loss decreases faster with compute.

(2) Encoder-free models are predicted to catch up within practical pretraining budgets (§3.2). As shown in Fig. 1 (Left), the two architectures exhibit nearly overlapping loss–compute frontiers for the text objective. In contrast, Fig. 1 (Right) shows that encoder-free models require more training compute than encoder-based models to reach the same validation loss on the multimodal objective. Nevertheless, their loss decreases more rapidly with compute, narrowing the gap at scale. Extrapolating the fitted scaling laws beyond our measured range predicts that the multimodal crossover occurs on the order of 10^{22} FLOPs under compute-optimal allocation, and at a higher compute budget under $5\times$ overtraining. For reference, the pretraining compute of recent flagship models, such as Kimi K2.5 (Kimi Team et al., 2026), is approximately 10^{25} FLOPs.[1] Furthermore, fits on individual multimodal topics show that the crossover varies by topic, arriving earlier on topics that rely mainly on language and much later on perception-intensive ones.

(3) The decoder takes over visual encoding via vision-specific adaptation (§3.3). The decoder increasingly relies on bidirectional attention among visual tokens, recovering the patch-level contextualization that a visual encoder would otherwise provide. Moreover, visual token representations diverge from their inputs much earlier than in encoder-based models, so the shallow decoder layers effectively serve as an implicit visual encoding stage, whereas text tokens are processed almost identically in both architectures. This vision-specific adaptation further extends to the MoE experts, where the routing of visual tokens becomes more concentrated, consistent with some experts taking over the vision-specific role of the visual encoder. These adaptations suggest that encoder-free models may benefit from decoder architectures designed explicitly for native visual representation learning, rather than directly inheriting designs built for language.

Overall, encoder-free models require more training compute within the fitted range, but their more rapidly improving multimodal frontier predicts an efficiency crossover within practical pretraining budgets. These findings position encoder-free architectures as a promising direction, and we expect this work to encourage broader exploration of encoder-free multimodal pretraining.

2 Preliminaries

2.1 Estimating Scaling Laws

Problem Definition. Let M denote FLOPs per token (Bi et al., 2024) and D the number of objective tokens, so the training budget is $C = MD$. At a target budget C, the compute-optimal allocation (Kaplan et al., 2020; Hoffmann et al., 2022) for objective $\mathcal{L}$ is the feasible point on $C = MD$ that minimizes loss:

$$M_{\text{opt}}(C), D_{\text{opt}}(C) = \arg\min_{M,\,D} \mathcal{L}(M, D) \quad \text{s.t.} \quad MD = C. \tag{1}$$

Across budgets, these optima follow the compute-optimal allocation law:

$$M_{\text{opt}}(C) \propto C^a, \qquad D_{\text{opt}}(C) \propto C^b, \qquad a + b = 1. \tag{2}$$

The corresponding compute-optimal frontiers follow:

$$\mathcal{L}^*(C) = E + K\,C^{-\gamma}, \tag{3}$$

[1] Estimated using $C \approx 6N_{\text{active}}D$.

where $\gamma > 0$ is the loss–compute exponent, $K > 0$ is a fitted prefactor, and E is the entropy floor induced by the data distribution (Hoffmann et al., 2022).

IsoFLOP Profiles. Following Chinchilla (Hoffmann et al., 2022), an IsoFLOP profile at budget C is obtained by varying M, setting $D = C/M$, and fitting validation loss as a quadratic function of $\log M$. The fitted minimum defines $M_{\text{opt}}(C)$. The corresponding $D_{\text{opt}}(C) = C/M_{\text{opt}}(C)$ and $\mathcal{L}^*(C)$ then follow directly. Repeating this procedure across budgets provides the optima used to estimate the allocation law and the compute-optimal frontiers.

2.2 Efficiency Gain

Following MAI-Thinking-1 (Microsoft AI, 2026), let λ be a loss reachable by both systems and let $C_s(\lambda)$ denote the actual training compute required by system $s \in \{\text{tar}, \text{ref}\}$ to reach it under the training regime being compared. The compute efficiency gain is

$$\text{EG}^{C}_{\text{tar}\leftarrow\text{ref}}(\lambda) = C_{\text{ref}}(\lambda)/C_{\text{tar}}(\lambda). \tag{4}$$

Further, let $M_s(\lambda)$ denote the FLOPs per token of the model actually used by system s at this point of equal loss. The model efficiency gain is

$$\text{EG}^{M}_{\text{tar}\leftarrow\text{ref}}(\lambda) = M_{\text{ref}}(\lambda)/M_{\text{tar}}(\lambda). \tag{5}$$

For both metrics, values above 1.0 indicate that the target is more efficient than the reference: the target requires less training compute for EG^C or fewer FLOPs per token for EG^M to reach λ.

2.3 Model Ladder

We compare encoder-free and encoder-based MLLMs on a matched ladder of 11 sparse MoE language models with 1.1B–44B total and 71M–2.4B active non-embedding parameters, sharing the same data mixture, optimization setup, and visual-token granularity. As shown in Fig. 2, the encoder-based model encodes images with a pretrained SigLIP 2 ViT (Tschannen et al., 2025), followed by a ConvPool adapter and a projector, and applies causal attention to all tokens. Following common practice (Bai et al., 2025; Gemma Team et al., 2025), the ViT keeps the same size across decoder scales and is trained jointly with the decoder. The encoder-free model instead maps raw image patches into the decoder through a patch projection (Gemma Team et al., 2026), where visual tokens attend bidirectionally within each image (Gemma Team et al., 2026; Diao et al., 2026a; Liu et al., 2026; Fang et al., 2026) and all other attention remains causal. We also study a fully causal variant. More implementation details are provided in Appendix A.

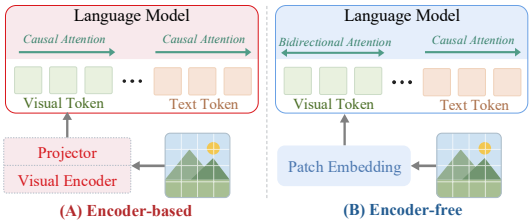


Figure 2: Architectural comparison of encoder-based and encoder-free MLLMs.

2.4 Compute Accounting

The decoder FLOPs per token can be decomposed into a term from matrix multiplications applied to each token, which is independent of sequence length, and a term from self-attention, which grows with sequence length:

$$M_o^{(s)} = M_{\text{base}} + M_{\text{attn}}^{(s)}\,\ell_o, \qquad C_o^{(s)} = M_o^{(s)} D_o, \tag{6}$$

where $s \in \{\text{free}, \text{based}\}$ indexes the model family and $o \in \{\text{text}, \text{mm}\}$ indexes the objective. Here, D_o counts all tokens processed in batches for objective o (D_{mm} includes both visual and text tokens). The mean packed length ℓ_o depends on the objective, while $M_{\text{attn}}^{(s)}$ differs between the two families because they use different attention patterns over visual tokens. The fixed expert activation ratio of $8/256$ keeps M approximately proportional to the number of active parameters.

Since we vary decoder scale with a fixed visual encoder, we use decoder FLOPs as the primary compute measure, focusing the analysis on the tradeoff between decoder capacity and training tokens. Appendix C shows that including visual encoder FLOPs leaves the main conclusions unchanged and further strengthens the relative efficiency of encoder-free models.

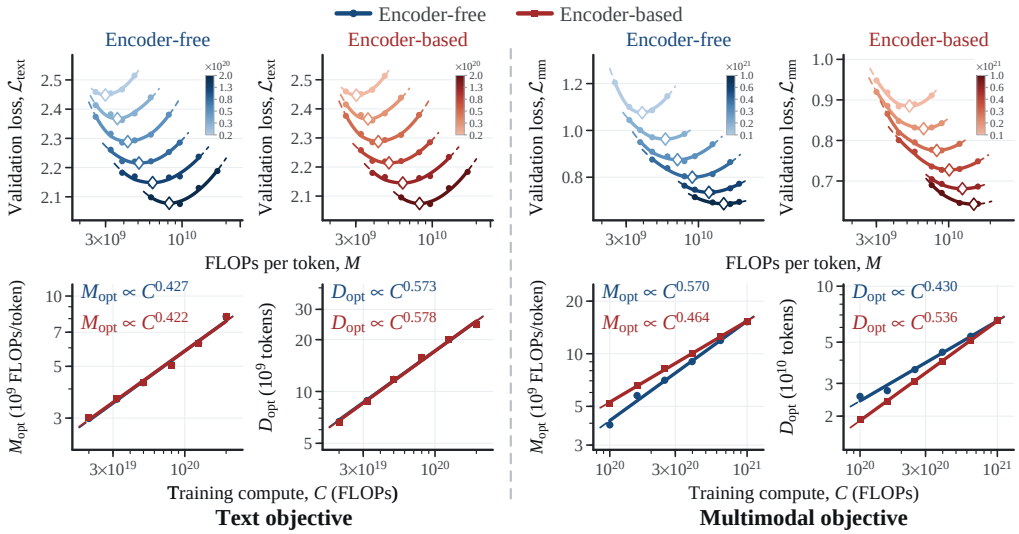


Figure 3: Estimating compute-optimal allocation with IsoFLOP profiles for the text objective (left) and the multimodal objective (right).

3 Scaling Laws for Encoder-Free Multimodal Pretraining

In this section, we explore how removing the pretrained visual encoder changes the scaling behavior of the language model. §3.1 first examines how encoder removal changes compute-optimal allocation. §3.2 then uses the resulting allocation laws to test whether encoder-free models can catch up at scale, under both compute-optimal allocation and overtraining. Finally, §3.3 examines how the decoder takes over visual encoding through vision-specific adaptation.

3.1 How Does Encoder Removal Change Compute-Optimal Allocation?

We estimate compute-optimal allocation for both encoder-free and encoder-based models using the IsoFLOP profiles described in §2.1, with the results shown in Fig. 3. At each budget for each model, the best point in the profile gives M_{opt} and D_{opt}, whose scaling with C yields the allocation exponents a and b.

On the text objective, the two architectures have nearly identical model allocation exponents ($a = 0.427$ for encoder-free and $a = 0.422$ for encoder-based models), which is expected. **On the multimodal objective**, however, removing the encoder increases a from 0.464 to 0.570, indicating that compute-optimal training allocates more compute to model scale. For these multimodal estimates, a bootstrap gives central 80% intervals of $[0.546, 0.595]$ and $[0.458, 0.472]$ for the encoder-free and encoder-based models, respectively (Appendix B.4). The shift persists under causal attention over visual tokens, which gives $a = 0.557$ (Tab. 1; Appendix E.2), suggesting that it is not an artifact of the attention mask. Together, these results suggest that removing the visual encoder increases the decoder's representational burden and favors larger models.

Table 1: Model allocation exponent a of encoder-free models under bidirectional and causal attention over visual tokens.

Attention	Text	Multimodal
Bidirectional	0.427	0.570
Causal	0.436	0.557

Takeaway 1. Removing the visual encoder leaves the compute-optimal allocation on the text objective unchanged, but shifts multimodal training toward larger models (Fig. 3).

3.2 Can Encoder-Free Models Catch Up at Scale?

After estimating compute-optimal allocation, we explore how much compute and decoder scale encoder-free models need to match encoder-based models at equal validation loss. For all metrics in this subsection, encoder-free is the target and encoder-based is the reference. We report two

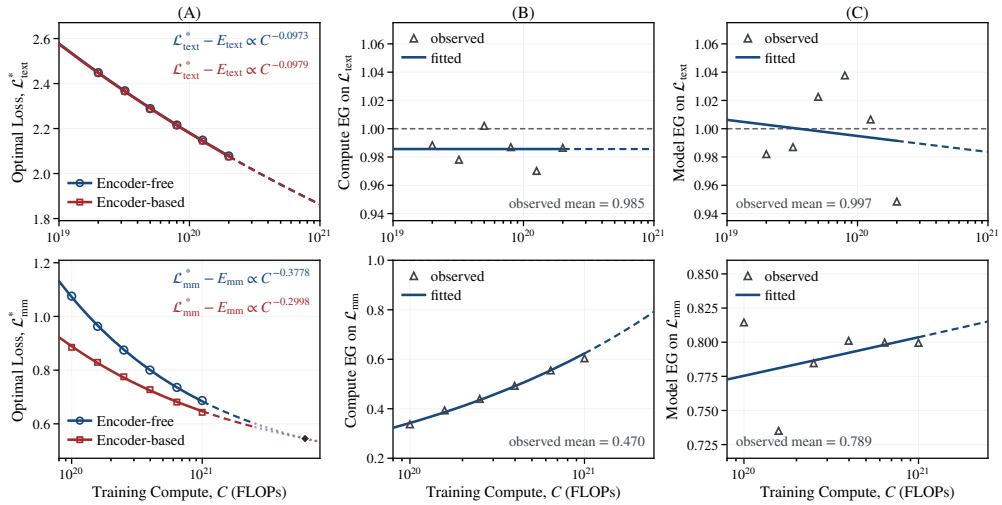


Figure 4: Efficiency gain analysis on the compute-optimal frontier for the text (top) and multimodal (bottom) objectives. **(A)** fitted loss–compute scaling laws, **(B)** compute efficiency gain (EG^C), **(C)** model efficiency gain (EG^M).

ratios. EG^C compares the training compute required at equal loss. Because the ratio is reference over target, values below 1.0 mean encoder-free models need more compute. EG^M compares the FLOPs per token selected at the point of equal loss. Values below 1.0 mean the encoder-free model at equal loss uses a larger decoder. We first evaluate both ratios on the compute-optimal frontier, then examine how they shift under overtraining.

3.2.1 Efficiency Gain Under Compute-Optimal Allocation

We first compare the two systems on their respective compute-optimal frontiers, where each training budget is allocated between model scale and tokens to minimize loss. Fixing a target loss then determines both the required compute and the corresponding optimal FLOPs per token. Fig. 4 summarizes the fitted loss–compute scaling laws and the resulting efficiency gains.

Text Objective. On $\mathcal{L}_{\text{text}}$, the two fitted loss curves are almost indistinguishable. Their loss–compute exponents are also nearly identical (0.0973 versus 0.0979). The fitted and measured EG^C values stay around 0.98, and the fitted EG^C curve is nearly flat. EG^M at equal loss also stays close to 1.0, around 0.99, with only a slight downward drift across the fitted range. Both deviations remain small and nearly constant across the fitted range, so text acts as a nearly matched control rather than a regime with a meaningful encoder-free penalty.

Multimodal Objective. On $\mathcal{L}_{\text{mm}}$, encoder-free models require more training compute to attain the same loss throughout the measured range. However, this gap narrows with scale: the fitted EG^C increases because the encoder-free loss decreases more rapidly with training compute. At equal loss, encoder-free models also favor a larger decoder, with $\text{EG}^M \approx 0.80$, corresponding to approximately $1.25\times$ the FLOPs per token. Extrapolating the fitted scaling laws places the efficiency crossover on the order of 10^{22} FLOPs (Fig. 4A, bottom). The point estimate is 6.1×10^{21} FLOPs, with a conditional bootstrap 80% interval of $[4.2 \times 10^{21}, 1.0 \times 10^{22}]$ (Appendix B.4). This projection assumes that the fitted laws persist beyond the measured range, with the visual encoder held at a fixed size and the irreducible loss determined only by the data distribution. Even accounting for this uncertainty, the crossover remains roughly three orders of magnitude below the pretraining compute of recent flagship models, e.g., approximately 10^{25} FLOPs for Kimi K2.5 (Kimi Team et al., 2026).

3.2.2 Efficiency Gain Under Overtraining

The compute-optimal frontier is not the only practical regime, since deployment models are often overtrained by spending extra tokens at a fixed model scale to reduce inference cost at a target quality (Sardana et al., 2023). We therefore test whether overtraining changes the efficiency gain.

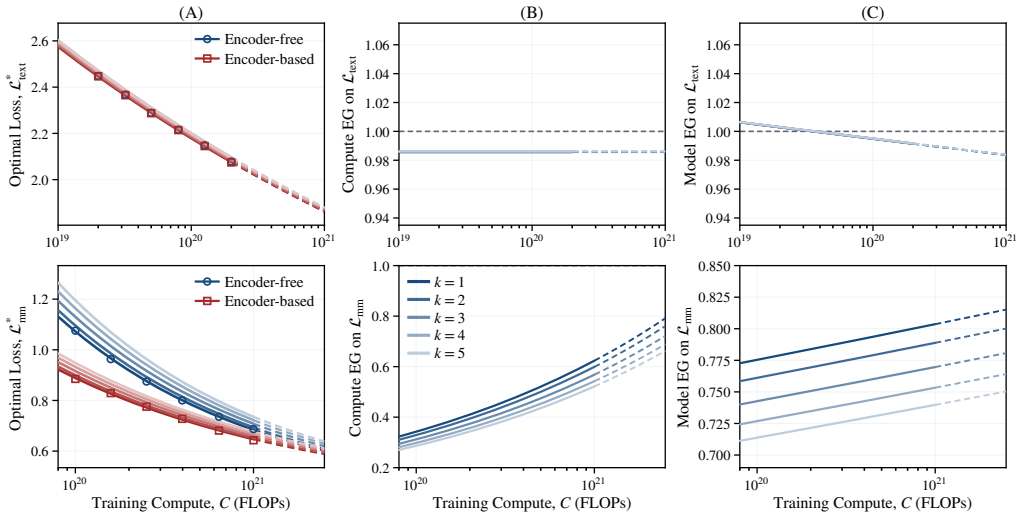


Figure 5: Efficiency gain analysis under overtraining for the text (top) and multimodal (bottom) objectives, with shades denoting overtraining factors $k = 1$–5 from darkest to lightest. **(A)** loss–compute scaling laws, **(B)** compute efficiency gain (EG^C), **(C)** model efficiency gain (EG^M).

Starting from the compute-optimal point at base budget C_{base}, overtraining keeps the model scale fixed and trains on k times as many tokens, where k is the overtraining factor, so the actual training compute is $C_{\text{actual}} = kC_{\text{base}}$.

We model overtraining as a shift in the prefactor of the loss–compute law, following prior scaling analyses (Gadre et al., 2025). Under the separable form $\mathcal{L}(M, D) = E + AM^{-\alpha} + BD^{-\beta}$ with $C_{\text{base}} = MD$, the compute-optimal frontier is $\mathcal{L}^*(C_{\text{base}}) = E + KC_{\text{base}}^{-\gamma}$ with $\gamma = \alpha\beta/(\alpha + \beta)$. Fixing $M = M_{\text{opt}}(C_{\text{base}})$ and setting $D = kD_{\text{opt}}(C_{\text{base}})$ gives

$$\mathcal{L}(C_{\text{base}}, k) = E + g(k)\, K\, C_{\text{base}}^{-\gamma}, \tag{7}$$

where the multiplier $g(k)$ depends on k but not on C_{base}. Overtraining therefore leaves E and γ unchanged and rescales only the reducible term (derivation in Appendix D). In practice, we reuse E, K, and γ from the compute-optimal fit and estimate only an empirical $g(k)$ for each k (Appendix A.3), which requires far fewer runs and lets us run overtraining experiments at smaller model scales.

Fig. 5 shows how overtraining changes the comparison at equal loss. On the multimodal objective, the shift is asymmetric: the same overtraining factor k changes the two systems' prefactors by different amounts because their data exponents and the optimal split between loss terms differ. At $k = 5$, the extrapolated crossover remains on the order of 10^{22} FLOPs but arrives later than under compute-optimal allocation. The point estimate is 1.2×10^{22} FLOPs, with a conditional bootstrap 80% interval of $[8.4 \times 10^{21}, 2.0 \times 10^{22}]$ (Appendix B.4). At the largest fitted budget, overtraining also lowers EG^C from 0.62 to 0.52 and EG^M from 0.80 to 0.74. This is consistent with the allocation shift in §3.1: since encoder-free models favor larger decoders on multimodal data, spending extra compute on tokens at a fixed model scale benefits them less. By contrast, encoder-free and encoder-based models remain nearly matched on the text objective under overtraining, with both ratios changing by less than 1% from their $k = 1$ values.

Takeaway 2. Both architectures nearly overlap on text loss frontiers, while encoder-free models initially lag on multimodal loss but are predicted to catch up within practical budgets (Figs. 4–5).

3.2.3 Analysis by Topic

The aggregate multimodal loss summarizes the overall trend but obscures variation across topics. We therefore repeat the analysis at equal loss on each major multimodal topic (Fig. 6). Appendix F reports the underlying IsoFLOP profiles.

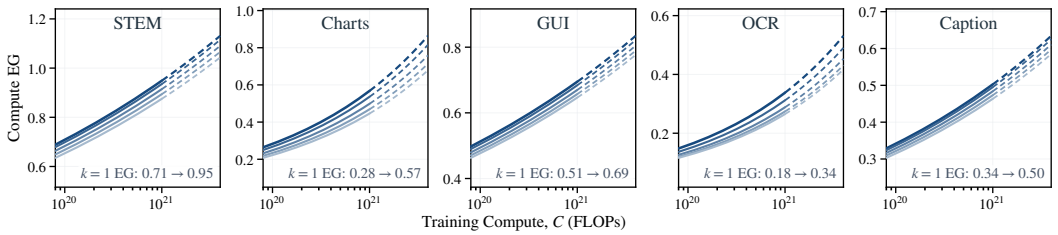


Figure 6: Compute efficiency gain on each multimodal topic. Curves from dark to light correspond to overtraining factors $k = 1$–5. Solid segments span the fitted budgets, and dashed segments extrapolate the fitted laws.

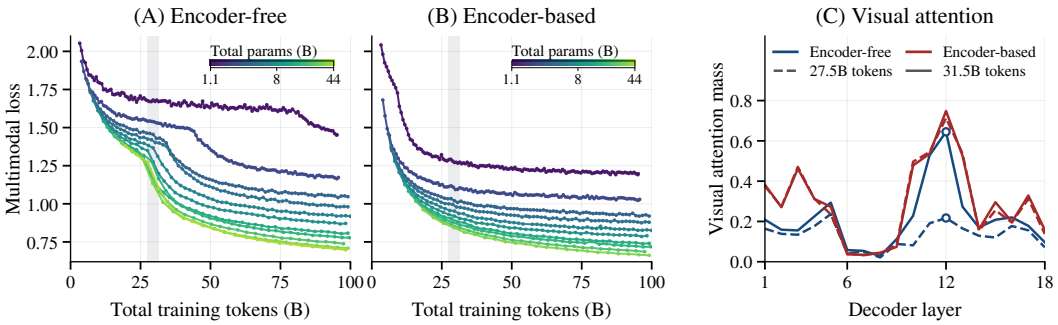


Figure 7: Visual learning across training tokens. **(A, B)** Multimodal validation loss during training. **(C)** Attention mass on visual tokens per decoder layer for the 8B models, before and after the encoder-free loss drop (shaded in A, B).

Within our compute range, encoder-free models remain less compute-efficient than encoder-based models on every topic, yet topics differ markedly in the size of the remaining gap and how quickly it narrows with scale. Extrapolating the fitted laws, encoder-free models catch up first on STEM, which is already close to parity at the largest fitted budget, then on Charts, whose gap narrows quickly, and considerably later on GUI, OCR, and Caption. This ordering is consistent with how strongly each topic relies on pretrained visual representations. The STEM subset consists mainly of text, symbols, and simple diagrams, and chart inputs can often be reduced to symbolic content. Once this content is extracted, prediction depends mainly on language and reasoning. By contrast, captioning requires rich representations of natural images, while GUI and OCR demand detailed spatial and textual perception, for which a pretrained encoder provides a strong prior that encoder-free models must learn from scratch.

Takeaway 3. The crossover varies by topic, arriving earlier on topics that rely mainly on language (e.g., STEM) and much later on perception-intensive ones (e.g., Caption) (Fig. 6).

3.3 How Does the Decoder Take Over Visual Encoding?

Removing the visual encoder shifts visual representation learning into the decoder. To trace this shift, we first compare how the multimodal loss of encoder-free and encoder-based models evolves over training tokens, and then examine three probes inside the decoder: attention over visual tokens, layerwise evolution of visual representations, and expert routing.

Emergence of Visual Encoding During Training. The multimodal loss of encoder-based models decreases smoothly, whereas that of encoder-free models decreases slowly at first, then drops sharply within a short span (Fig. 7A). To understand this drop, we compare attention to visual tokens in the 8B models before and after it: at layer 12, the encoder-free model's attention to visual tokens rises from 0.217 to 0.645, approaching that of the encoder-based model (Fig. 7C). We hypothesize that the slow early phase reflects the decoder bootstrapping its own visual representations. Since the loss is applied only to text tokens, visual tokens receive learning signal only when text attends to them. Initially uninformative and thus ignored, they learn slowly until they become useful enough

to attract attention, after which learning accelerates. A pretrained encoder supplies useful visual representations from the start and thus avoids this stage, consistent with the high attention to visual tokens in the encoder-based model at both checkpoints. At every multimodal IsoFLOP budget, the compute-optimal models have already passed this drop, so the fitted frontier reflects the smooth regime after it.

Encoder-Like Contextualization: Bidirectional Attention. We compare bidirectional and causal attention over visual tokens by rerunning the encoder-free ladder with the causal variant (Fig. 8). Causal attention is slightly better on the text objective at the measured budgets, reaching the same loss for about 1% less compute ($\mathrm{EG}^{C} = 1.011$), but this gain shrinks with compute. On the multimodal objective, causal attention is mildly worse on average ($\mathrm{EG}^{C} = 0.990$), and the multimodal gain of bidirectional attention becomes larger with compute. This pattern is consistent with a transfer of function from encoder to decoder. In encoder-based models, the ViT bidirectionally contextualizes image patches before they reach the decoder. Once the ViT is removed, bidirectional attention among visual tokens allows the decoder to assume part of this role. The increasing multimodal benefit of bidirectional attention with scale suggests that larger decoders exploit these interactions better, while its diminishing cost on text indicates limited interference with language modeling.

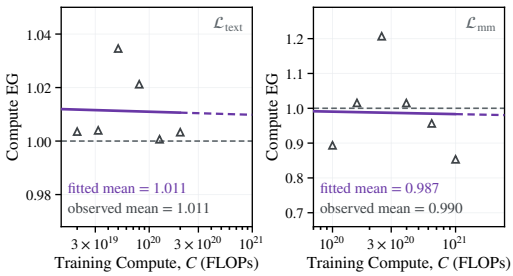


Figure 8: Comparison between causal and bidirectional attention on visual tokens. Values above 1.0 favor causal.

Encoder-Like Early Processing: Layerwise Representation Evolution. In encoder-based MLLMs, the ViT has already transformed visual tokens into semantic representations, so they undergo little additional processing in shallow decoder layers (Fan et al., 2026). If the decoder takes over this transformation, its shallow layers should instead rewrite visual tokens substantially. Fig. 9 tests this using cosine similarity between each layer's token representations and their layer-0 inputs. Without the visual encoder, visual tokens move away from their inputs much earlier (left), while text token trajectories remain close across the two systems (right). The same pattern holds across model scales (Appendix E.1). The shallow decoder layers thus act as an implicit visual encoding stage, performing the transformation that the ViT performs in encoder-based models, and this change is specific to visual tokens.

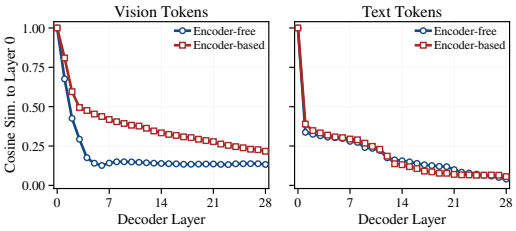


Figure 9: Layerwise representation evolution. Cosine similarity to the input representation at layer 0 across decoder layers for visual tokens (left) and text tokens (right).

Encoder-Like Dedicated Capacity: Expert Routing. Both systems use the same sparse decoder, so routing differences show how the decoder absorbs the changed visual representations. We quantify expert load imbalance using MaxVio (Wang et al., 2024), the relative excess of the most-loaded expert's load over the perfectly balanced load. As shown in Fig. 10, both systems have similarly low overall MaxVio when visual and text tokens are aggregated, although the encoder-free values are slightly higher. Separating tokens by modality reveals substantially greater expert load imbalance for both visual and text tokens than the aggregate suggests. For text tokens, the two architectures remain closely matched. In contrast, across the four largest model sizes, encoder-free models exhibit consistently higher average MaxVio and a wider band for visual tokens throughout training. The close match on text argues against a routing shift across the whole model and localizes the effect to visual processing. This concentration is consistent with the decoder allocating a subset of its experts to play the role of the vision-specific parameters that the ViT previously provided.

Takeaway 4. The decoder adapts to take over visual encoding: bidirectional attention among visual tokens, shallow-layer visual processing, and concentrated expert routing (Figs. 8–10).

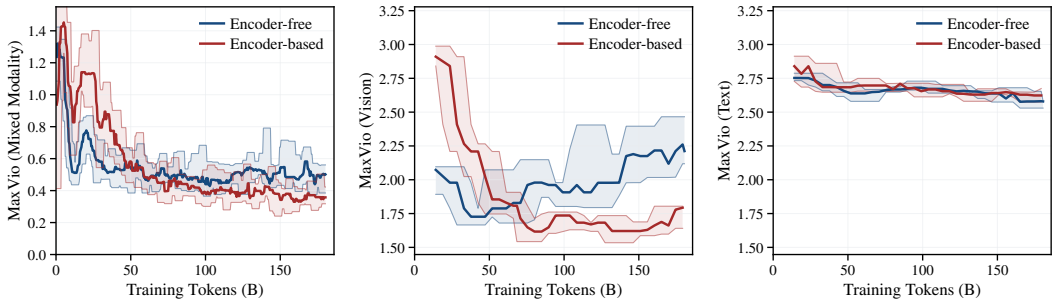


Figure 10: Expert load imbalance over training. MaxVio is computed jointly over visual and text tokens (left), over visual tokens only (middle), and over text tokens only (right). Higher values indicate greater imbalance.

4 Related Work

Encoder-Free MLLMs. Encoder-free MLLMs remove the visual encoder and pass projected image patches directly into the decoder. Early systems such as Fuyu, EVE, and SOLO established the feasibility of this design (Bavishi et al., 2023; Diao et al., 2024; Chen et al., 2024b). Moving beyond feasibility, SAIL systematically studies model and data scalability, cross-modal information flow, and visual representation learning within a single Transformer (Lei et al., 2025). Later work improves visual competence through alignment or distillation (Tao et al., 2025; Wang et al., 2025; Li et al., 2026b; Yang et al., 2025), additional capacity dedicated to each modality (Luo et al., 2025b;a; Diao et al., 2025), native vision-language primitives (Diao et al., 2026a), and extensions to video (Yi et al., 2025; Li et al., 2025a; Diao et al., 2026b), 3D (Tang et al., 2026), segmentation (Zhang et al., 2025), and unified understanding and generation (Chameleon Team, 2024; Li et al., 2025b; 2026a; Diao et al., 2026c; Liu et al., 2026; Wang et al., 2026b). Recent large systems such as the Gemma 4 12B Unified model (Gemma Team et al., 2026) and Inkling (Thinking Machines Lab, 2026) also explore native multimodal input without a visual encoder. Our work complements these efforts by comparing behavior between encoder-free and encoder-based MLLMs.

Scaling Laws. Predicting training behavior at large scale from small runs is now standard practice, spanning loss trends, data scaling, compute-optimal allocation, and training hyperparameters (Kaplan et al., 2020; Hoffmann et al., 2022; Bi et al., 2024; Bjorck et al., 2025; Li et al., 2025c; Clark et al., 2022; Alabdulmohsin et al., 2022; Wang et al., 2026a). Recent work extends this methodology to native multimodal pretraining. In particular, the compute-optimal data requirement grows faster for vision than for language in unified multimodal pretraining (Tong et al., 2026), and native MLLMs trained under data constraints exhibit coupled scaling between the visual encoder and the language model (Tian et al., 2026). Others compare early and late fusion trained from scratch (Shukor et al., 2025) or study how the data mixture affects the scaling of encoder-free MoE models (Wu et al., 2026). In contrast, we compare encoder-free models against an encoder-based baseline on a shared MoE decoder ladder, predict the compute budget at which encoder-free models catch up under both compute-optimal allocation and overtraining, and analyze how the decoder takes over visual encoding via vision-specific adaptation.

5 Conclusion

Encoder-free MLLMs are less compute-efficient on the multimodal objective at the scales we evaluate. However, their compute-optimal loss decreases faster as compute increases. Our fitted scaling laws predict a crossover on the order of 10^{22} FLOPs under compute-optimal allocation and at a higher budget under $5\times$ overtraining. Encoder-free scaling also favors larger decoders, delays the decoder's reliance on visual input, strengthens interactions among visual tokens, shifts visual representation transformation earlier, and concentrates expert routing. In contrast, text scaling remains largely unchanged. These findings point to future work on decoder architectures and training strategies that improve the compute efficiency of native visual input.

AI USE STATEMENT

Generative AI tools were used to improve the grammar, clarity, and organization of the manuscript and to assist with writing, debugging, and refactoring research code, including experimental, analysis, and visualization scripts. The research question, methodology, experimental design, interpretation of results, and scientific conclusions were developed by the authors. All text written with AI assistance was reviewed and revised by the authors. The authors take full responsibility for the final content of this work, including all text, claims, code, and other artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work investigates architectural scaling and does not involve experiments with human subjects or decisions about individuals. Like other research on multimodal pretraining at scale, however, it entails nontrivial computational and environmental costs and may inherit biases, harmful content, privacy concerns, and licensing constraints from its training data. The resulting models could reproduce such content or be used in unintended ways. Our experiments are intended for scientific comparison. Scaling trends based on loss and probes inside the decoder should not be interpreted as evidence that either architecture is safe or suitable for deployment, particularly in high-stakes settings. Any downstream use should include appropriate data governance, privacy, bias, and safety evaluations and comply with applicable licenses and regulations.

REPRODUCIBILITY STATEMENT

To support reproducibility, the main text defines the training objectives, IsoFLOP estimator, the comparison at equal loss, and the compute accounting for each objective. The appendix provides additional implementation details and robustness analyses supporting the scaling law estimates. The encoder-free and encoder-based comparisons use matched model ladders, data, and optimization settings unless explicitly stated otherwise. Exact replication of the full pipeline additionally requires access to the same pretraining corpus and substantial computational resources.

REFERENCES

Ibrahim M Alabdulmohsin, Behnam Neyshabur, and Xiaohua Zhai. Revisiting neural scaling laws in language and vision. *Advances in Neural Information Processing Systems*, 35:22300–22312, 2022.

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*, 2025.

Rohan Bavishi, Erich Elsen, Curtis Hawthorne, Maxwell Nye, Augustus Odena, Arushi Somani, and Sağnak Taşırlar. Introducing our multimodal models, 2023. URL https://www.adept.ai/blog/fuyu-8b.

Tamay Besiroglu, Ege Erdil, Matthew Barnett, and Josh You. Chinchilla scaling: A replication attempt. *arXiv preprint arXiv:2404.10102*, 2024.

Xiao Bi, Deli Chen, Guanting Chen, Shanhuang Chen, Damai Dai, Chengqi Deng, Honghui Ding, Kai Dong, Qiushi Du, Zhe Fu, et al. DeepSeek LLM: Scaling open-source language models with longtermism. *arXiv preprint arXiv:2401.02954*, 2024.

Johan Bjorck, Alon Benhaim, Vishrav Chaudhary, Furu Wei, and Xia Song. Scaling optimal LR across token horizons. In *International Conference on Learning Representations*, volume 2025, pp. 83640–83657, 2025.

Chameleon Team. Chameleon: Mixed-modal early-fusion foundation models. *arXiv preprint arXiv:2405.09818*, 2024.

Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Jiaqi Wang, Yu Qiao, Dahua Lin, et al. Are we on the right way for evaluating large vision-language models? *Advances in Neural Information Processing Systems*, 37:27056–27087, 2024a.

Yangyi Chen, Xingyao Wang, Hao Peng, and Heng Ji. Solo: A single transformer for scalable vision-language modeling. *arXiv preprint arXiv:2407.06438*, 2024b.

Leshem Choshen, Yang Zhang, and Jacob Andreas. A hitchhiker's guide to scaling law estimation. *arXiv preprint arXiv:2410.11840*, 2024.

Aidan Clark, Diego de Las Casas, Aurelia Guy, Arthur Mensch, Michela Paganini, Jordan Hoffmann, Bogdan Damoc, Blake Hechtman, Trevor Cai, Sebastian Borgeaud, et al. Unified scaling laws for routed language models. In *International Conference on Machine Learning*, pp. 4057–4086. PMLR, 2022.

Damai Dai, Chengqi Deng, Chenggang Zhao, RX Xu, Huazuo Gao, Deli Chen, Jiashi Li, Wangding Zeng, Xingkai Yu, Yu Wu, et al. DeepSeekMoE: Towards ultimate expert specialization in mixture-of-experts language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1280–1297, 2024.

Haiwen Diao, Yufeng Cui, Xiaotong Li, Yueze Wang, Huchuan Lu, and Xinlong Wang. Unveiling encoder-free vision-language models. *Advances in Neural Information Processing Systems*, 37: 52545–52567, 2024.

Haiwen Diao, Xiaotong Li, Yufeng Cui, Yueze Wang, Haoge Deng, Ting Pan, Wenxuan Wang, Huchuan Lu, and Xinlong Wang. EVEv2: Improved baselines for encoder-free vision-language models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 21014–21025. IEEE, 2025.

Haiwen Diao, Mingxuan Li, Silei Wu, Linjun Dai, Xiaohua Wang, Hanming Deng, Lewei Lu, Dahua Lin, and Ziwei Liu. From pixels to words–towards native vision-language primitives at scale. In *International Conference on Learning Representations*, volume 2026, pp. 109909–109929, 2026a.

Haiwen Diao, Jiahao Wang, Penghao Wu, Yuhao Dong, Yuwei Niu, Yue Zhu, Zhongang Cai, Weichen Fan, Linjun Dai, Silei Wu, et al. From pixels to words–towards native one-vision models at scale. *arXiv preprint arXiv:2605.28820*, 2026b.

Haiwen Diao, Penghao Wu, Hanming Deng, Jiahao Wang, Shihao Bai, Silei Wu, Weichen Fan, Wenjie Ye, Wenwen Tong, Xiangyu Fan, et al. SenseNova-U1: Unifying multimodal understanding and generation with NEO-unify architecture. *arXiv preprint arXiv:2605.12500*, 2026c.

Yingqi Fan, Junlong Tong, Anhao Zhao, and Xiaoyu Shen. What do visual tokens really encode? Uncovering sparsity and redundancy in multimodal large language models. *arXiv preprint arXiv:2603.00510*, 2026.

Yan Fang, Mengcheng Lan, Zilong Huang, Weixian Lei, Yunqing Zhao, Yujie Zhong, Yingchen Yu, Qi She, Yao Zhao, and Yunchao Wei. Let ViT speak: Generative language-image pre-training. *arXiv preprint arXiv:2605.00809*, 2026.

Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, et al. MME: A comprehensive evaluation benchmark for multimodal large language models. *Advances in Neural Information Processing Systems*, 38, 2026.

Samir Yitzhak Gadre, Georgios Smyrnis, Vaishaal Shankar, Suchin Gururangan, Mitchell Wortsman, Rulin Shao, Jean Mercat, Alex Fang, Jeffrey Li, Sedrick Keh, et al. Language models scale reliably with over-training and on downstream tasks. In *International Conference on Learning Representations*, volume 2025, pp. 67661–67682, 2025.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, et al. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*, 2025.

Gemma Team, Sherif El Abd, Vaibhav Aggarwal, Robin Algayres, Alek Andreev, Olivier Bachem, Ian Ballantyne, Cormac Brick, Victor Cărbune, Michelle Casbon, et al. Gemma 4 technical report. *arXiv preprint arXiv:2607.02770*, 2026.

Tom Henighan, Jared Kaplan, Mor Katz, Mark Chen, Christopher Hesse, Jacob Jackson, Heewoo Jun, Tom B Brown, Prafulla Dhariwal, Scott Gray, et al. Scaling laws for autoregressive generative modeling. *arXiv preprint arXiv:2010.14701*, 2020.

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.

Keller Jordan, Yuchen Jin, Vlado Boza, Jiacheng You, Franz Cesista, Laker Newhouse, and Jeremy Bernstein. Muon: An optimizer for hidden layers in neural networks, 2024. URL https://kellerjordan.github.io/posts/muon/.

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.

Aniruddha Kembhavi, Mike Salvato, Eric Kolve, Minjoon Seo, Hannaneh Hajishirzi, and Ali Farhadi. A diagram is worth a dozen images. In *European Conference on Computer Vision*, pp. 235–251. Springer, 2016.

Kimi Team, Tongtong Bai, Yifan Bai, Yiping Bao, SH Cai, Yuan Cao, Ziwei Chai, Y Charles, HS Che, Cheng Chen, et al. Kimi K2.5: Visual agentic intelligence. *arXiv preprint arXiv:2602.02276*, 2026.

Weixian Lei, Jiacong Wang, Haochen Wang, Xiangtai Li, Jun Hao Liew, Jiashi Feng, and Zilong Huang. The scalability of simplicity: Empirical analysis of vision-language learning with a single transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 20758–20769. IEEE, 2025.

Han Li, Xinyu Peng, Yaoming Wang, Zelin Peng, Xin Chen, Rongxiang Weng, Jingang Wang, Xunliang Cai, Wenrui Dai, and Hongkai Xiong. OneCAT: Decoder-only auto-regressive model for unified understanding and generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 30235–30245, 2026a.

Handong Li, Yiyuan Zhang, Longteng Guo, Xiangyu Yue, and Jing Liu. Breaking the encoder barrier for seamless video-language understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 23167–23176. IEEE, 2025a.

Hao Li, Changyao Tian, Jie Shao, Xizhou Zhu, Zhaokai Wang, Jinguo Zhu, Wenhan Dou, Xiaogang Wang, Hongsheng Li, Lewei Lu, et al. SynerGen-VL: Towards synergistic image understanding and generation with vision experts and token folding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 29767–29779. IEEE, 2025b.

Houyi Li, Wenzhen Zheng, Qiufeng Wang, Hanshan Zhang, Zili Wang, Shijie Xuyang, Yuantao Fan, Zhenyu Ding, Haoying Wang, Ning Ding, et al. Predictable scale: Part I, Step Law–optimal hyperparameter scaling law in large language model pretraining. *arXiv preprint arXiv:2503.04715*, 2025c.

Tianle Li, Yongming Rao, Winston Hu, and Yu Cheng. BREEN: Bridge data-efficient encoder-free multimodal learning with learnable queries. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 5384–5395. IEEE, 2026b.

Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Xin Zhao, and Ji-Rong Wen. Evaluating object hallucination in large vision-language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 292–305, 2023.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in Neural Information Processing Systems*, 36:34892–34916, 2023.

Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, et al. MMBench: Is your multi-modal model an all-around player? In *European Conference on Computer Vision*, pp. 216–233. Springer, 2024.

Zhiheng Liu, Weiming Ren, Xiaoke Huang, Shoufa Chen, Tianhong Li, Mengzhao Chen, Yatai Ji, Sen He, Jonas Schult, Tao Xiang, Wenhu Chen, Ping Luo, Luke Zettlemoyer, and Yuren Cong. TUNA-2: Pixel embeddings beat vision encoders for unified understanding and generation. *arXiv preprint arXiv:2604.24763*, 2026.

Pan Lu, Swaroop Mishra, Tanglin Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering. *Advances in Neural Information Processing Systems*, 35:2507–2521, 2022.

Gen Luo, Wenhan Dou, Wenhao Li, Zhaokai Wang, Xue Yang, Changyao Tian, Hao Li, Weiyun Wang, Wenhai Wang, Xizhou Zhu, et al. Mono-InternVL-1.5: Towards cheaper and faster monolithic multimodal large language models. *arXiv preprint arXiv:2507.12566*, 2025a.

Gen Luo, Xue Yang, Wenhan Dou, Zhaokai Wang, Jiawen Liu, Jifeng Dai, Yu Qiao, and Xizhou Zhu. Mono-InternVL: Pushing the boundaries of monolithic multimodal large language models with endogenous visual pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 24960–24971. IEEE, 2025b.

Ahmed Masry, Jia Qing Tan, Shafiq Joty, Enamul Hoque, et al. ChartQA: A benchmark for question answering about charts with visual and logical reasoning. In *Findings of the Association for Computational Linguistics: ACL 2022*, pp. 2263–2279, 2022.

Minesh Mathew, Dimosthenis Karatzas, and CV Jawahar. DocVQA: A dataset for VQA on document images. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 2199–2208. IEEE, 2021.

Microsoft AI. MAI-Thinking-1: Building a hill-climbing machine. https://microsoft.ai/pdf/mai-thinking-1.pdf, 2026.

Tomer Porian, Mitchell Wortsman, Jenia Jitsev, Ludwig Schmidt, and Yair Carmon. Resolving discrepancies in compute-optimal scaling of language models. *Advances in Neural Information Processing Systems*, 37:100535–100570, 2024.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pp. 8748–8763. PMLR, 2021.

Nikhil Sardana, Jacob Portes, Sasha Doubov, and Jonathan Frankle. Beyond Chinchilla-optimal: Accounting for inference in language model scaling laws. *arXiv preprint arXiv:2401.00448*, 2023.

Mustafa Shukor, Enrico Fini, Victor Guilherme Turrisi da Costa, Matthieu Cord, Joshua Susskind, and Alaaeldin El-Nouby. Scaling laws for native multimodal models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 12–23. IEEE, 2025.

Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. Towards VQA models that can read. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8309–8318. IEEE, 2019.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. RoFormer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

Yiwen Tang, Ziyu Guo, Zhuhao Wang, Renrui Zhang, Qizhi Chen, Junli Liu, Delin Qu, Dong Wang, Bin Zhao, and Xuelong Li. Exploring the potential of encoder-free architectures in 3D LMMs. In *International Conference on Learning Representations*, volume 2026, pp. 150063–150084, 2026.

Chenxin Tao, Shiqian Su, Xizhou Zhu, Chenyu Zhang, Zhe Chen, Jiawen Liu, Wenhai Wang, Lewei Lu, Gao Huang, Yu Qiao, et al. HoVLE: Unleashing the power of monolithic vision-language models with holistic vision-language embedding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14559–14569. IEEE, 2025.

Thinking Machines Lab. Inkling: Our open-weights model. https://thinkingmachines.ai/news/introducing-inkling/, 2026.

Changyao Tian, Hao Li, Gen Luo, Xizhou Zhu, Weijie Su, Hanming Deng, Jinguo Zhu, Jie Shao, Ziran Zhu, Yunpeng Liu, et al. NaViL: Rethinking scaling properties of native multimodal large language models under data constraints. *Advances in Neural Information Processing Systems*, 38: 85618–85646, 2026.

Shengbang Tong, Ellis Brown, Penghao Wu, Sanghyun Woo, Manoj Middepogu, Sai C Akula, Jihan Yang, Shusheng Yang, Adithya Iyer, Xichen Pan, et al. Cambrian-1: A fully open, vision-centric exploration of multimodal LLMs. *Advances in Neural Information Processing Systems*, 37:87310–87356, 2024.

Shengbang Tong, David Fan, John Nguyen, Ellis Brown, Gaoyue Zhou, Shengyi Qian, Boyang Zheng, Théophane Vallaeys, Junlin Han, Rob Fergus, et al. Beyond language modeling: An exploration of multimodal pretraining. *arXiv preprint arXiv:2603.03276*, 2026.

Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, et al. SigLIP 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. *arXiv preprint arXiv:2502.14786*, 2025.

Han Wang, Yongjie Ye, Bingru Li, Yuxiang Nie, Jinghui Lu, Jingqun Tang, Yanjie Wang, and Can Huang. Vision as LoRA. *arXiv preprint arXiv:2503.20680*, 2025.

Lean Wang, Huazuo Gao, Chenggang Zhao, Xu Sun, and Damai Dai. Auxiliary-loss-free load balancing strategy for mixture-of-experts. *arXiv preprint arXiv:2408.15664*, 2024.

Pingjie Wang, Zechen Hu, Peiru Yang, Fu Guo, and Debing Zhang. Smooth scaling laws hide stepwise token learning. *arXiv preprint arXiv:2606.29858*, 2026a.

Xinlong Wang, Yufeng Cui, Jinsheng Wang, Fan Zhang, Yueze Wang, Xiaosong Zhang, Zhengxiong Luo, Quan Sun, Zhen Li, Yuqi Wang, et al. Multimodal learning with next-token prediction for large multimodal models. *Nature*, 650(8101):327–333, 2026b.

Haoyuan Wu, Aoqi Wu, Hai Wang, Jiajia Wu, Jinxiang Ou, and Bei Yu. Scaling native multimodal pre-training from scratch. *arXiv preprint arXiv:2607.22043*, 2026.

xAI. RealWorldQA, 2024. URL https://huggingface.co/datasets/xai-org/RealworldQA.

Rui Yang, Lin Song, Yicheng Xiao, Runhui Huang, Yixiao Ge, Ying Shan, and Hengshuang Zhao. HaploVL: A single-transformer baseline for multi-modal understanding. *arXiv preprint arXiv:2503.14694*, 2025.

Jinhui Yi, Syed Talal Wasim, Yanan Luo, Muzammal Naseer, and Juergen Gall. Video-Panda: Parameter-efficient alignment for encoder-free video-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 24119–24128. IEEE, 2025.

Biao Zhang and Rico Sennrich. Root mean square layer normalization. *Advances in Neural Information Processing Systems*, 32, 2019.

Tao Zhang, Xiangtai Li, Zilong Huang, Yanwei Li, Weixian Lei, Xueqing Deng, Shihao Chen, Shunping Ji, and Jiashi Feng. Pixel-SAIL: Single transformer for pixel-grounded understanding. *URL https://arxiv. org/abs/2504.10465*, 2025.

APPENDIX

A Implementation Details **16**

A.1 Visual Front End Architectures 16

A.2 Decoder Architecture and Model Ladder 16

A.3 Training Setup 16

A.4 Training and Validation Data 16

A.5 Scaling Law Fitting Setup 17

B Robustness of the Scaling Law Estimates **17**

B.1 Comparison with the Validation Curve Envelope 17

B.2 Extrapolation Error Analysis 18

B.3 Sensitivity to the Irreducible Loss 18

B.4 Conditional Bootstrap Uncertainty 20

C Compute Accounting for the Visual Front End **21**

C.1 Compute-Optimal Allocation 22

C.2 Loss–Compute Exponent 22

C.3 Summary 22

D Derivation of Eq. 7 **22**

E Probes Inside the Decoder **23**

E.1 Layerwise Visual Representation Evolution Across Scales 23

E.2 Compute-Optimal Allocation Under Causal Attention 23

F IsoFLOP Profiles by Topic **25**

F.1 Pure Text Topics 25

F.2 Multimodal Topics 26

G Downstream Evaluation **26**

G.1 Benchmarks 26

G.2 Evaluation Protocol 26

G.3 Results 26

A Implementation Details

A.1 Visual Front End Architectures

Encoder-Free Front End. Both front ends produce one visual token per 32×32 pixel region. Our encoder-free front end adapts the patch projection pipeline used in the Gemma 4 12B Unified model (Gemma Team et al., 2026). In the original pipeline, 16×16 patches are merged in 3×3 spatial groups, yielding visual tokens that each cover a 48×48 pixel region. To ensure a controlled comparison, we instead use 2×2 merging, matching the encoder-based front end in both visual-token granularity and token count. Each merged patch of raw pixels passes through a LayerNorm, a linear layer, and a second LayerNorm. Learned factorized 2D position embeddings, obtained by summing separate embeddings for the two spatial axes, are then added, followed by a third LayerNorm, an RMSNorm, and a linear projection to the decoder width.

Encoder-Based Front End. The encoder-based front end uses a pretrained SigLIP 2 ViT (Tschannen et al., 2025) with 27 layers, width 1152, patch size 16, and AnyRes processing. A 2×2 ConvPool adapter and a projector match the encoder-free visual-token granularity. We choose this roughly 400M encoder scale as a representative practical setting used by recent advanced MLLMs (Gemma Team et al., 2025; Bai et al., 2025; Kimi Team et al., 2026). The ViT is trained jointly with the decoder in every run, while its architecture and parameter count remain fixed across decoder scales.

Matched Comparison Protocol. Both front ends use the same preprocessing pipeline, consume the same pixels, and pass the same number of visual tokens to the decoder. The main comparison therefore holds visual content and token count fixed while varying the representation attached to each token and the attention pattern over visual tokens described in §3.3. Fixing the ViT across the ladder isolates decoder scaling and avoids introducing joint scaling of the encoder and decoder as an additional variable. Consequently, the ViT accounts for a smaller fraction of total model capacity as the decoder grows, and the reported scaling trends are conditional on this regime with a fixed encoder size. Jointly scaling the visual encoder would define a different allocation problem, requiring a separate sweep that balances potential representation gains against additional compute in the front end.

A.2 Decoder Architecture and Model Ladder

The two systems share a complete ladder of 11 rungs. Model depth and width vary across rungs, while the MoE topology remains fixed. Every rung uses 256 routed experts, activates the top 8 experts for each token, and includes one shared expert (Dai et al., 2024). The activation ratio of routed experts is therefore fixed at $8/256 = 1/32$ across model scales, keeping all rungs within a consistent architectural family. The first layer is dense, followed by MoE layers. All models use RMSNorm (Zhang & Sennrich, 2019) and RoPE (Su et al., 2024).

A.3 Training Setup

Optimization and Hyperparameters. All runs use a sequence length of 4,096, the Muon optimizer (Jordan et al., 2024), and 2,000 warmup steps. Training hyperparameters are selected as a function of model scale using an internal scaling law. In particular, the batch size and learning rate vary across the model ladder. At each rung, identical hyperparameters are used for the encoder-based and encoder-free MLLMs.

Overtraining Runs and Multiplier Fitting. For both model families, we train compute-optimal model sizes at $k \in \{2, 3, 4, 5\}$. To allow for deviations from the idealized separable loss model underlying Eq. 7, we keep E, K_s, and γ_s fixed to their compute-optimal estimates and fit an empirical multiplier $g_s^{\mathrm{emp}}(k)$ using the available overtraining runs. At $k = 5$, the fitted multipliers are 0.645 for encoder-free and 0.692 for encoder-based models. These empirical estimates are used for the overtraining analysis in Fig. 5.

A.4 Training and Validation Data

Training Mixture. The corpus is a 1:1 mixture of text and multimodal data. Since we focus on architectural scaling rather than effects of the data mixture, we keep this ratio fixed rather than

Table 2: Scaling exponents estimated by IsoFLOP and validation curve envelope fitting.

Objective	System	Estimator	Loss γ	Model a	Data b
Text	Encoder-free	IsoFLOP	0.0973	0.427	0.573
Text	Encoder-free	Envelope	0.0905	0.430	0.570
Text	Encoder-based	IsoFLOP	0.0979	0.422	0.578
Text	Encoder-based	Envelope	0.0915	0.435	0.565
Multimodal	Encoder-free	IsoFLOP	0.3778	0.570	0.430
Multimodal	Encoder-free	Envelope	0.3668	0.583	0.417
Multimodal	Encoder-based	IsoFLOP	0.2998	0.464	0.536
Multimodal	Encoder-based	Envelope	0.3050	0.475	0.525

treating it as a study variable. The balanced mixture provides sufficient multimodal training tokens for robust estimation of multimodal scaling. Multimodal sources cover tasks such as captioning, charts, grounding, GUI, OCR, STEM, and knowledge. Text sources cover domains such as STEM, code, books, and wikis.

Validation Losses. Validation data are disjoint from training and identical across systems. We separately track losses on pure text and on multimodal topics: text losses are computed on sequences containing no visual tokens, whereas multimodal losses are computed over text prediction tokens in sequences conditioned on images, with visual tokens masked out from the loss, and are grouped by topic. These masked visual positions remain included in D_{mm} and in decoder FLOP accounting. The primary validation view weights major topics uniformly while preserving the training mixture weights of sources within each topic.

A.5 Scaling Law Fitting Setup

Following prior scaling law work (Kaplan et al., 2020; Hoffmann et al., 2022), we use held-out validation loss as the primary metric and fit separate laws for the text and multimodal objectives, $\mathcal{L}_{\text{text}}$ and $\mathcal{L}_{\text{mm}}$. IsoFLOP fits use six logarithmically spaced budgets for each objective, covering 2×10^{19} to 2×10^{20} FLOPs for $\mathcal{L}_{\text{text}}$ and 1×10^{20} to 1×10^{21} FLOPs for $\mathcal{L}_{\text{mm}}$.

B Robustness of the Scaling Law Estimates

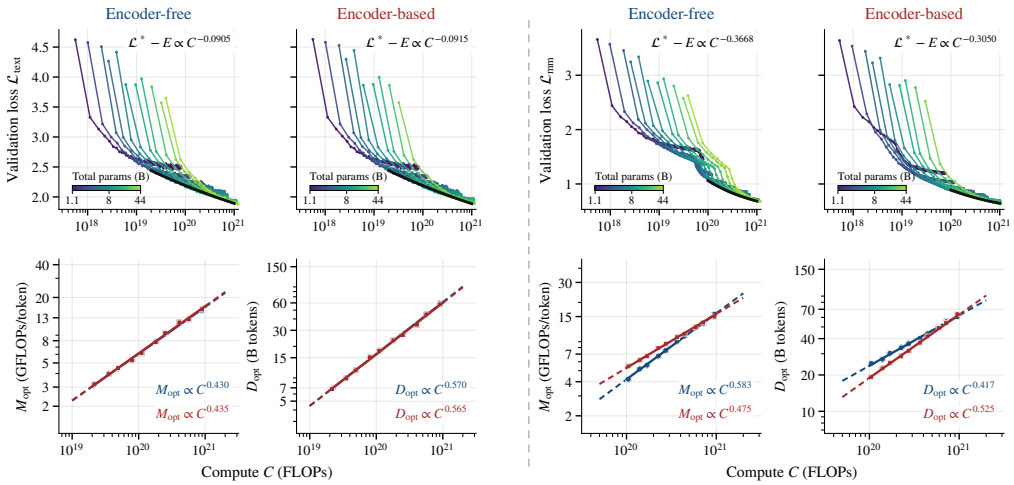
In this section, we assess the robustness of the scaling law estimates. We begin by checking the IsoFLOP estimator against a validation curve envelope estimator in Appendix B.1. Next, we evaluate held-out extrapolation accuracy in Appendix B.2. Appendix B.3 tests the sensitivity of the loss–compute fits to the shared irreducible loss. Appendix B.4 quantifies conditional fitting uncertainty with a residual bootstrap that preserves pairs matched by budget.

B.1 Comparison with the Validation Curve Envelope

This subsection compares the IsoFLOP estimates with a validation curve envelope. We first describe the estimator, then report the text and multimodal fits in Tab. 2.

Validation Curve Envelope. We adapt the envelope approach of Chinchilla (Hoffmann et al., 2022), which uses training loss curves, to validation loss trajectories: the held-out validation loss of each run, measured at its intermediate checkpoints. At each of 200 logarithmically spaced compute budgets, we linearly interpolate each trajectory in log compute and fit a local quadratic in $\log M$ to the five model scales around the lowest observed loss on the ladder. The quadratic vertex gives a continuous estimate of the optimum derived from the trajectories. Because adjacent budgets are highly correlated, we aggregate these estimates by their median within 24 equal bins in log compute before fitting the allocation and loss laws. We use Huber regression in log space for the allocation laws to limit the influence of isolated errors from local interpolation. Unlike the six IsoFLOP profiles, this estimator constructs a dense frontier from points along the validation trajectories.

Text Objective. Fig. 11(a) shows the text fits. Over its broader trajectory range, the envelope estimator gives $a = 0.430$ for encoder-free and $a = 0.435$ for encoder-based models, close to the



(a) Text objective **(b) Multimodal objective**

Figure 11: Validation curve envelopes grouped by objective: (a) text and (b) multimodal. Within each group, the top row shows unsmoothed encoder-free and encoder-based validation trajectories. The bottom row shows representative estimates of compute-optimal M_{opt} and D_{opt} derived from the trajectories and binned in log compute, with fitted allocation laws.

IsoFLOP values of 0.427 and 0.422. Its loss–compute exponents of 0.0905 and 0.0915 are also close to the IsoFLOP estimates.

Multimodal Objective. Fig. 11(b) shows the corresponding multimodal fits. The envelope estimator gives $a = 0.583$ for encoder-free and $a = 0.475$ for encoder-based models, close to the IsoFLOP values of 0.570 and 0.464. Its loss–compute exponents of 0.3668 and 0.3050 are also close to the IsoFLOP values of 0.3778 and 0.2998, preserving the same ordering: the encoder-free frontier is steeper than the encoder-based frontier.

B.2 Extrapolation Error Analysis

We test whether the compute-optimal fits forecast held-out IsoFLOP optima when the target budget is a modest multiple of the largest fitted budget. For each objective, we fit separate loss–compute scaling laws for each architecture to the first four IsoFLOP optima, then evaluate the forecast at a held-out budget beyond the fitting window. Held-out optima are estimated independently with the quadratic IsoFLOP procedure from §2.1. All parameters of each architecture are estimated using only the fitting subset at lower compute.

Text Objective. For this check, we additionally run a separate IsoFLOP profile at 4×10^{20} FLOPs, outside the main fitting range of the scaling laws, and use it only as a held-out extrapolation target. The forecasting fit at lower compute spans 2×10^{19} to 8×10^{19} FLOPs, making the held-out target a $5\times$ extrapolation beyond the largest fitted budget. The held-out profiles contain seven encoder-free and six encoder-based model scales, with both estimated minima lying inside the sampled model range. Fig. 12 shows signed relative errors of -1.03% and -0.87%.

Multimodal Objective. The fitting window spans IsoFLOP optima through 4×10^{20} FLOPs. We forecast the optimum at 1×10^{21} FLOPs, a $2.5\times$ extrapolation beyond the largest fitted budget. Fig. 13 shows signed relative errors of -0.01% for encoder-free models and $+2.19\%$ for encoder-based models. Across both objectives, the same fitting procedure forecasts held-out compute-optimal losses over these extrapolation factors with relative errors within about 2%.

B.3 Sensitivity to the Irreducible Loss

This subsection specifies the fit with a common E used for the loss–compute laws, then tests how the estimates move when E is perturbed (Fig. 14 and Tab. 3).

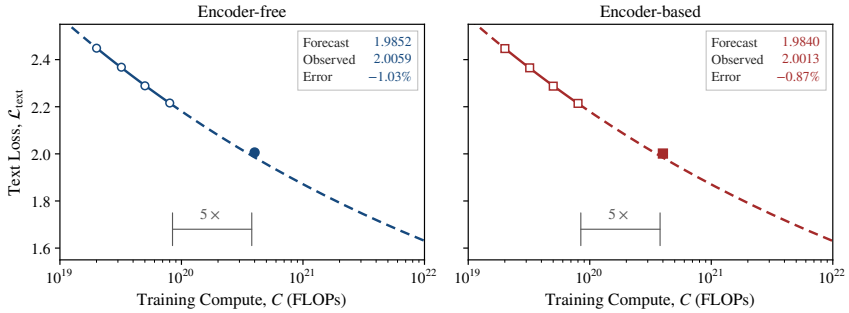


Figure 12: Extrapolation error analysis on text loss. Laws fitted through 8×10^{19} FLOPs are evaluated against held-out IsoFLOP optima at 4×10^{20} FLOPs, a $5\times$ extrapolation. Dashed segments indicate extrapolation.

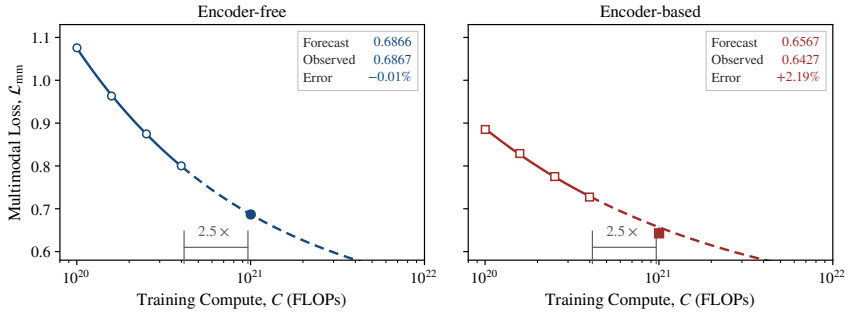


Figure 13: Extrapolation error analysis on multimodal loss. Laws fitted through 4×10^{20} FLOPs are evaluated against held-out IsoFLOP optima at 1×10^{21} FLOPs, a $2.5\times$ extrapolation. Dashed segments indicate extrapolation.

Common Irreducible Loss. Following prior scaling law work (Henighan et al., 2020; Hoffmann et al., 2022), we interpret the irreducible term E as the conditional entropy floor induced by the data distribution and prediction objective. Because the encoder-free and encoder-based systems are trained and evaluated on the same examples with the same next-token objective, their Bayes-optimal loss floor is shared. We therefore impose one E_o per objective across architectures, while allowing the finite compute terms K_s and γ_s to differ.

This constraint of a common E is a structural assumption, not a claim that our observations at finite scale identify the asymptote. Its interpretation further assumes that both model families can eliminate approximation error that depends on architecture as scale grows. In our setting, with six multimodal frontier points per system over approximately one decade of compute, E trades off strongly with the fitted slope. Separate floors for each architecture can therefore absorb differences within the finite range without reliably identifying distinct entropy limits. More broadly, asymptotic parameters of scaling laws are known to be sensitive to the fitting sample and specification (Besiroglu et al., 2024; Choshen et al., 2024; Porian et al., 2024). We consequently treat the constraint as theoretically motivated and test below which conclusions depend on it.

For objective o and system $s \in \{\text{free}, \text{based}\}$, we jointly solve

$$\min_{E_o, \{K_s, \gamma_s\}} \sum_s \sum_i \left[\mathcal{L}_{s,i} - E_o - K_s C_{s,i}^{-\gamma_s}\right]^2, \tag{8}$$

where $K_s > 0$ and $\gamma_s > 0$. The observations $(C_{s,i}, \mathcal{L}_{s,i})$ are the compute-optimal IsoFLOP vertices. The fitted floors are $\hat{E}_{\text{text}} = 0.607$ and $\hat{E}_{\text{mm}} = 0.403$.

Sensitivity. We then fix E_o on a grid from $0.05\hat{E}_o$ to $1.45\hat{E}_o$ and, at every grid point, refit (K_s, γ_s) by constrained least squares. The two objectives are swept independently. Fig. 14 reports the resulting loss–compute exponents, crossover points, and joint fitting error. The exponent ordering does not depend on the assumed floor: on the multimodal objective, γ_{free} exceeds γ_{based} at every grid point, whereas the two text exponents remain nearly identical throughout. Within $\pm 10\%$ of the fitted floor, $E_{\text{mm}}/\hat{E}_{\text{mm}} \in [0.90, 1.10]$, the multimodal exponent gap stays between 0.077 and 0.078 while the

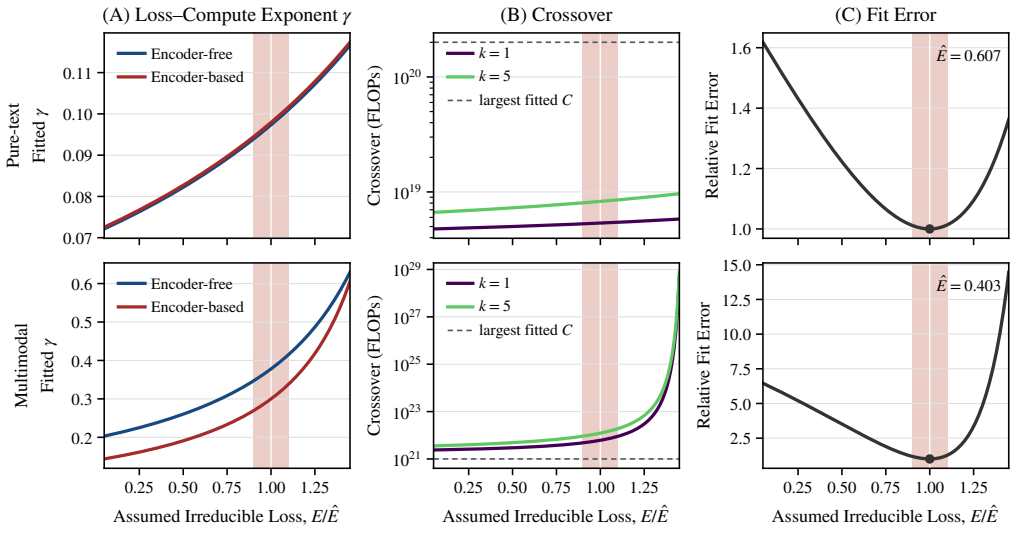


Figure 14: Sensitivity to the irreducible loss of each objective. Each row fixes $E_o/\hat{E}_o$ and jointly refits the encoder-free and encoder-based loss–compute scaling laws with a common E_o. **(A)** Fitted loss–compute exponent γ. **(B)** Implied crossover compute under compute-optimal allocation ($k = 1$) and $5\times$ overtraining ($k = 5$). The dashed line marks the largest fitted budget. **(C)** Joint SSE on raw loss, normalized by its minimum. The red band marks $\pm 10\%$ around the fitted floor.

joint fitting error rises by at most 27%. The gap between the two objectives is likewise not an artifact of an overestimated floor. Lowering E reduces every fitted exponent, yet even at $E_o = 0.05\hat{E}_o$, close to a pure power law, the encoder-based multimodal exponent (0.144) remains twice the text exponent (0.073). The fitted exponent values themselves, and any extrapolated crossover, are more sensitive to E (Tab. 3). These are sensitivity ranges rather than statistical confidence intervals.

Table 3: Multimodal sensitivity to perturbations of the fitted shared irreducible loss.

$E_{\text{mm}}/\hat{E}_{\text{mm}}$	E_{mm}	$\gamma_{\text{free}} - \gamma_{\text{based}}$	Compute-optimal	$5\times$ overtraining
$[0.90, 1.10]$	$[0.363, 0.444]$	$[0.077, 0.078]$	$[4.8, 9.0] \times 10^{21}$	$[0.9, 1.9] \times 10^{22}$

B.4 Conditional Bootstrap Uncertainty

In this section, we use a conditional residual bootstrap with paired resampling across architectures to characterize fitting uncertainty in the multimodal loss–compute and allocation exponents, their differences between architectures, and the extrapolated crossover budgets under compute-optimal allocation and $5\times$ overtraining.

Bootstrap Protocol. For each architecture, we compute the residuals between the six estimated optimal losses from the IsoFLOP profiles and the corresponding values of the fitted compute law, then center them by subtracting the mean for each architecture. At each bootstrap replicate, we sample six matched residual pairs with replacement and add them to the fitted losses at the six fixed compute budgets. We then jointly refit the two loss–compute scaling laws by least squares on the untransformed loss scale, estimating a common E_{mm} anew in each replicate. This paired resampling preserves the association at each budget between the encoder-free and encoder-based residuals.

We apply the same sampled budget indices to the centered residual pairs from the fits of the allocation laws. Each bootstrap replicate then yields estimates of the loss–compute and allocation exponents, their differences between architectures, and the crossover budgets under compute-optimal allocation and $5\times$ overtraining. For the $5\times$ crossover, each replicate reuses the original estimates of the multipliers $g_s(5)$ rather than estimating them again.

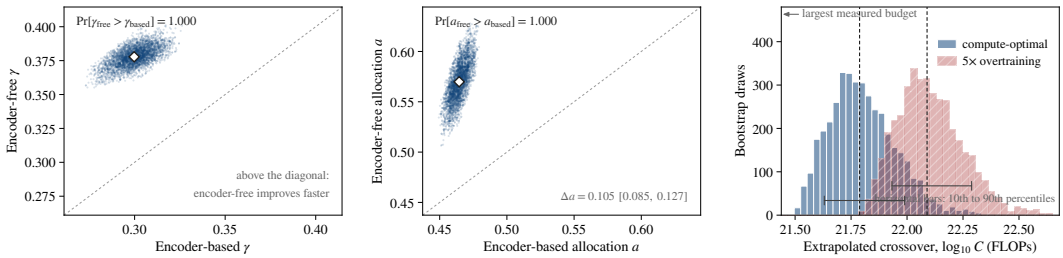


Figure 15: Conditional bootstrap uncertainty of the multimodal scaling fits and extrapolated crossover. **Left:** joint distribution of the encoder-free and encoder-based loss–compute exponents γ. **Middle:** joint distribution of their model allocation exponents a. **Right:** distributions of the extrapolated crossover compute under compute-optimal allocation and $5\times$ overtraining. Horizontal bars span the 10th to 90th percentiles, and dashed lines mark point estimates.

Table 4: Conditional bootstrap uncertainty of the multimodal scaling exponents. Estimates are the fitted IsoFLOP exponents. Intervals are central 80% bootstrap percentile intervals (10th to 90th percentiles).

	Encoder-free		Encoder-based	
Exponent	Estimate	80% interval	Estimate	80% interval
Loss γ	0.3778	[0.3686, 0.3873]	0.2998	[0.2882, 0.3112]
Model a	0.570	[0.546, 0.595]	0.464	[0.458, 0.472]
Data b	0.430	[0.405, 0.454]	0.536	[0.528, 0.543]

Following the reporting convention of Chinchilla (Hoffmann et al., 2022), we report central 80% bootstrap percentile intervals, bounded by the 10th and 90th percentiles of the bootstrap distribution.

Loss–Compute Exponents. On the multimodal objective, the encoder-free loss–compute exponent has a bootstrap median of 0.3781, with a central 80% interval of $[0.3686, 0.3873]$, whereas the encoder-based exponent has a median of 0.3002, with an interval of $[0.2882, 0.3112]$. Their difference, $\Delta\gamma = \gamma_{\text{free}} - \gamma_{\text{based}}$, has a median of 0.0781 and an interval of $[0.0687, 0.0876]$. The difference is positive across all 4,000 conditional bootstrap replicates.

Compute-Optimal Allocation Exponents. The model allocation exponent a_{free} has a bootstrap median of 0.570, with a central 80% interval of $[0.546, 0.595]$, whereas a_{based} has a median of 0.465, with an interval of $[0.458, 0.472]$. Their difference, $\Delta a = a_{\text{free}} - a_{\text{based}}$, has a median of 0.105 and an interval of $[0.085, 0.127]$, and is positive across all conditional bootstrap replicates. The corresponding data exponents have medians of 0.430 and 0.536 for the encoder-free and encoder-based systems, with central 80% intervals of $[0.405, 0.454]$ and $[0.528, 0.543]$, respectively. The left and middle panels of Fig. 15 show the joint bootstrap distributions, while Tab. 4 summarizes the marginal estimates and differences between architectures.

Crossover. The compute-optimal crossover has a median of 6.1×10^{21} FLOPs and an interval of $[4.2 \times 10^{21}, 1.0 \times 10^{22}]$, and the crossover under $5\times$ overtraining has a median of 1.2×10^{22} FLOPs and an interval of $[8.4 \times 10^{21}, 2.0 \times 10^{22}]$. All replicates produce finite crossovers within $[10^{21}, 10^{23}]$ FLOPs, and refits that each leave out one budget place the compute-optimal crossover between 4.3×10^{21} and 9.8×10^{21} FLOPs. The right panel of Fig. 15 shows the two distributions.

C Compute Accounting for the Visual Front End

This appendix adds the training cost of each visual front end. Let ϕ denote the resulting ViT share of encoder-based training compute. Because the loss of every trained configuration is unchanged, we estimate the encoder-based frontier under full accounting from a parametric fit $\mathcal{L}(M, D) = E + AM^{-\alpha} + BD^{-\beta}$ to its IsoFLOP points, minimizing it over M at each total budget.

C.1 Compute-Optimal Allocation

Under full accounting, the resulting compute-optimal allocation follows $M_{\text{opt}} \propto C^{0.273}$ and $D_{\text{opt}} \propto C^{0.727}$ for encoder-based models, compared with $a = 0.464$ and $b = 0.536$ when only decoder FLOPs are counted, while the encoder-free exponents remain $a = 0.570$ and $b = 0.430$ (Tab. 5). Because $M_{\text{full}} = M_{\text{dec}} + \text{const}$, we have $d \ln M_{\text{full}} / d \ln M_{\text{dec}} = 1 - \phi$, so the encoder-based exponent measured in total FLOPs per token is compressed while ϕ is large and approaches the decoder-accounting value as $\phi \to 0$. Under both conventions, encoder-free training still favors larger models.

C.2 Loss–Compute Exponent

Refitting $\mathcal{L}^* = E + KC^{-\gamma}$ to the six encoder-based optima under full accounting, with the shared $\hat{E}_{\text{mm}}$ fixed, gives $\gamma_{\text{based}} = 0.362$ with a root mean square error of 0.003, while the encoder-free exponent remains 0.378 (Tab. 5). The exponent gap narrows from 0.078 to 0.016 but keeps its sign. The narrowing is a finite-scale effect: charging the ViT raises the cost of small encoder-based runs relatively more than that of large ones, which steepens the frontier in total compute. As $\phi \to 0$ at larger scales, the encoder-based exponent approaches the decoder-accounting value of 0.300, and the gap approaches 0.078.

C.3 Summary

Neither conclusion of the main text depends on the accounting convention. For allocation, encoder-free training favors larger models than encoder-based training under both conventions, and encoder-free models retain the larger loss–compute exponent. For the crossover, charging the fixed $\sim$400M ViT only adds cost to encoder-based runs and leaves the encoder-free frontier unchanged, so it shifts every comparison at equal loss toward encoder-free models. At the crossover under decoder accounting in §3.2, encoder-free models therefore already require less total compute, and the crossover under full accounting occurs no later than 6.1×10^{21} FLOPs. We do not refit a numerical crossover under this convention.

Table 5: Multimodal scaling exponents under decoder and full compute accounting.

Accounting	System	Model a	Data b	Loss γ
Decoder only	Encoder-free	0.570	0.430	0.378
Decoder only	Encoder-based	0.464	0.536	0.300
Full	Encoder-free	0.570	0.430	0.378
Full	Encoder-based	0.273	0.727	0.362

D Derivation of Eq. 7

We derive Eq. 7 from §3.2.2. Here k multiplies the compute-optimal token count at fixed model scale: $M = M_{\text{opt}}(C_{\text{base}})$, $D = kD_{\text{opt}}(C_{\text{base}})$, and $C_{\text{actual}} = kC_{\text{base}}$.

Eliminating D via $D = C_{\text{base}}/M$, the excess loss at fixed C_{base} becomes

$$\mathcal{L}(M, C_{\text{base}}/M) - E = AM^{-\alpha} + BC_{\text{base}}^{-\beta} M^{\beta}.$$

This expression is strictly convex in $\log M$, so its unique stationary point is the global optimum. Differentiating with respect to M and setting the derivative to zero yields

$$-\alpha A M^{-\alpha-1} + \beta B C_{\text{base}}^{-\beta} M^{\beta-1} = 0,$$

hence

$$M^{\alpha+\beta} = \frac{\alpha A}{\beta B} C_{\text{base}}^{\beta}.$$

Writing $\rho = \alpha + \beta$ and $m = (\alpha A/\beta B)^{1/\rho}$, we obtain

$$M_{\text{opt}}(C_{\text{base}}) = m C_{\text{base}}^{\beta/\rho}.$$

The corresponding token count is $D_{\text{opt}}(C_{\text{base}}) = C_{\text{base}}/M_{\text{opt}}(C_{\text{base}})$, which simplifies to

$$D_{\text{opt}}(C_{\text{base}}) = m^{-1} C_{\text{base}}^{\alpha/\rho}.$$

Keeping the model scale fixed at $M_{\text{opt}}(C_{\text{base}})$ and multiplying the optimal token count by k, direct substitution yields

$$\begin{aligned}\mathcal{L}(C_{\text{base}}, k) - E &= A\left(mC_{\text{base}}^{\beta/\rho}\right)^{-\alpha} + B\left(km^{-1}C_{\text{base}}^{\alpha/\rho}\right)^{-\beta} \\ &= \left(Am^{-\alpha} + Bm^{\beta}k^{-\beta}\right) C_{\text{base}}^{-\gamma},\end{aligned}$$

where $\gamma = \alpha\beta/\rho$. Let

$$H(k) = Am^{-\alpha} + Bm^{\beta}k^{-\beta}.$$

The loss–compute scaling law on the compute-optimal frontier at $k = 1$ implies $H(1) = K$. Thus, defining the dimensionless multiplier $g(k) = H(k)/H(1)$ gives

$$\mathcal{L}(C_{\text{base}}, k) = E + g(k) K C_{\text{base}}^{-\gamma}.$$

Since $H(k)$ depends on k but not on C_{base}, overtraining changes only the prefactor and leaves the compute exponent unchanged.

E Probes Inside the Decoder

E.1 Layerwise Visual Representation Evolution Across Scales

§3.3 shows that, without a visual encoder, visual tokens move away from their layer-0 inputs at shallower decoder layers, while text token trajectories remain close across the two systems. Fig. 16 repeats this comparison at 20, 24, and 28 layers. The same pattern holds at every scale: encoder-free visual states diverge earlier, whereas the text curves stay closely matched. The earlier visual rewriting is therefore not an artifact of a single model size.

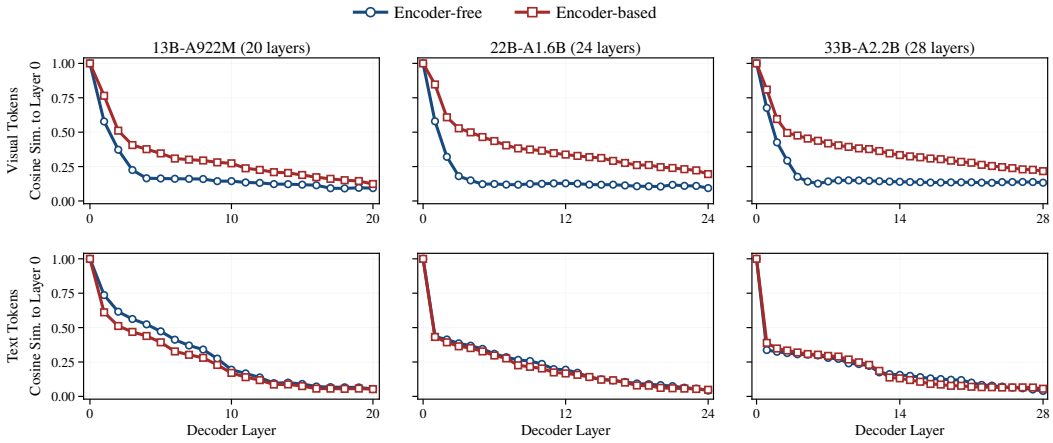


Figure 16: Layerwise similarity to the decoder input across three scales. Encoder-free visual states diverge earlier at every scale, while text trajectories remain closely matched.

E.2 Compute-Optimal Allocation Under Causal Attention

To test whether bidirectional attention among visual tokens causes the encoder-free allocation shift in §3.1, we repeat the IsoFLOP procedure of §2.1 on the causal encoder-free runs from Fig. 8. The two settings differ only in how visual tokens attend within each image. The data mixture, optimizer, and training schedule are otherwise matched, and we do not retune hyperparameters for causal attention.

Fig. 17 compares the two attention settings. Under causal attention, the text objective gives $M_{\text{opt}} \propto C^{0.436}$ and $D_{\text{opt}} \propto C^{0.564}$, while the multimodal objective gives $M_{\text{opt}} \propto C^{0.557}$ and $D_{\text{opt}} \propto$

$C^{0.443}$. These M exponents differ by only 0.009 and 0.013 from the bidirectional results (0.427 for text and 0.570 for multimodal). The largest multimodal vertex is extrapolated and should be read with caution. Even so, the multimodal fit still favors much larger models, so the encoder-free allocation trend is not explained by bidirectional attention among visual tokens.

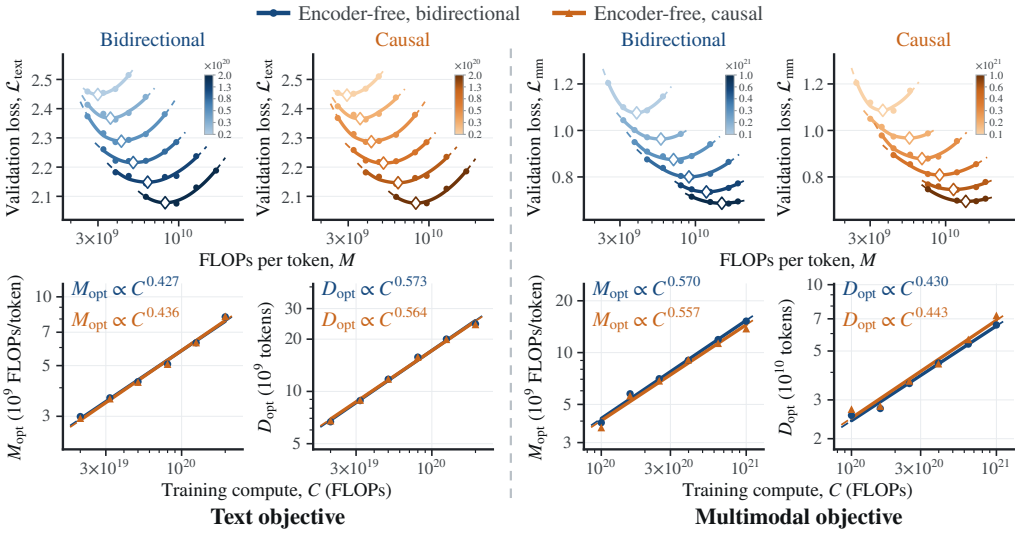


Figure 17: Compute-optimal allocation of encoder-free models under bidirectional and causal attention over visual tokens, for the text objective (left) and the multimodal objective (right). The top row shows IsoFLOP profiles, and the bottom row shows the fitted M_{opt} and D_{opt} laws. Diamonds mark fitted vertices and dashed segments indicate extrapolation.

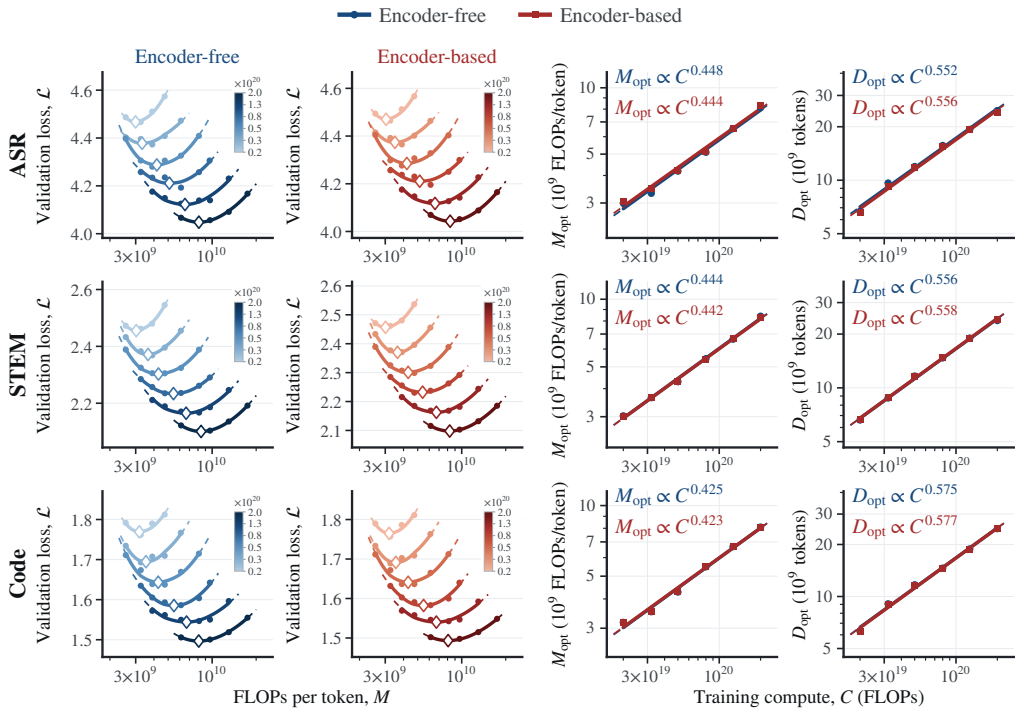


Figure 18: IsoFLOP profiles and fitted compute-optimal allocation laws for three pure text sets. The encoder-free and encoder-based exponents remain close on every topic.

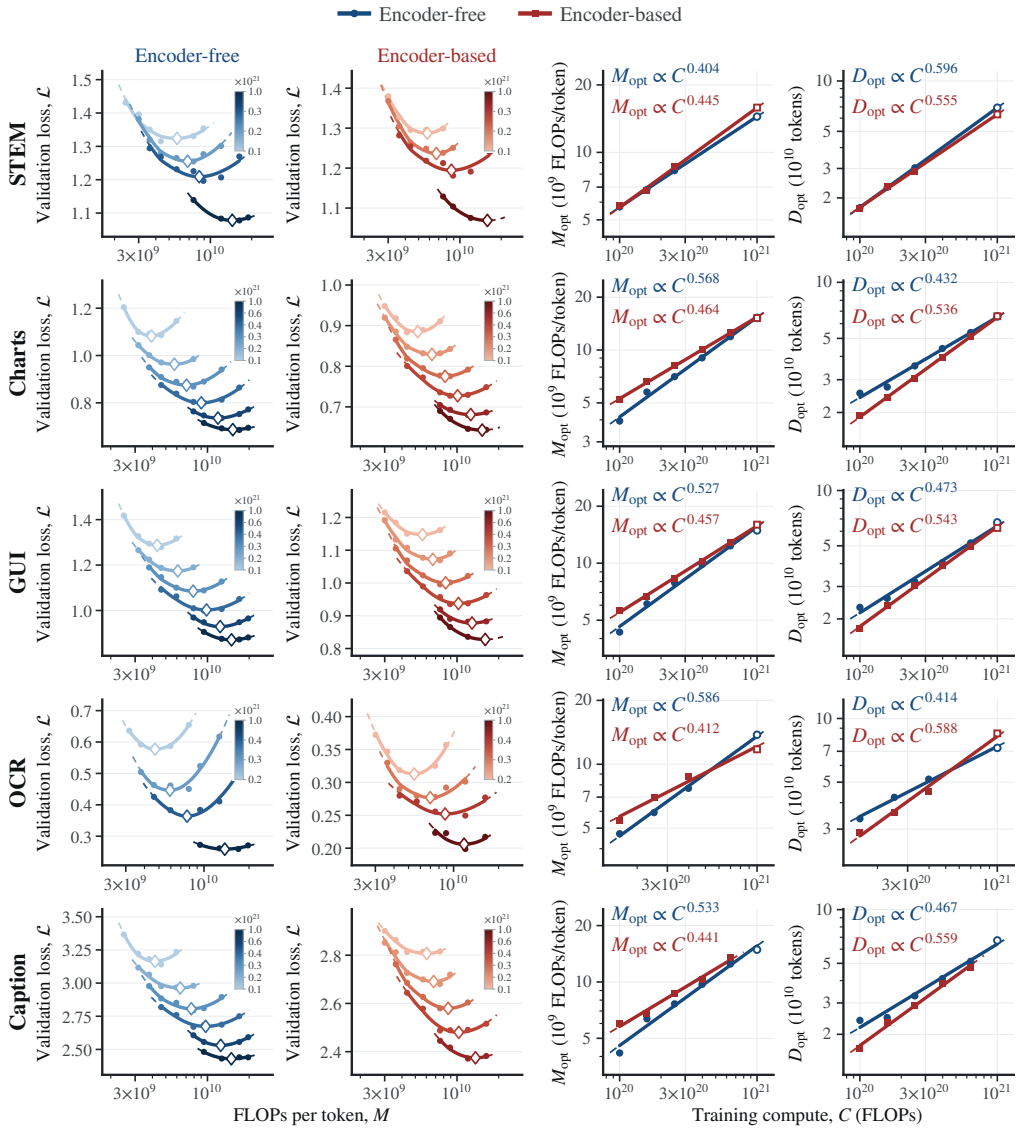


Figure 19: IsoFLOP profiles by topic and fitted compute-optimal allocation laws for five multimodal topics. On most topics, encoder-free models favor larger model scale.

F IsoFLOP Profiles by Topic

The main text fits allocation laws to the aggregate validation loss. Here we repeat the analysis on each validation topic separately. No new models are trained. Instead, we evaluate the existing checkpoints on each topic and fit IsoFLOP profiles as in §2.1. Because the model ladder was chosen for the aggregate loss, the optimum for some topics falls near or beyond its edge. We keep mild extrapolations, shown with open markers, and drop vertices that lie far outside the sampled range.

F.1 Pure Text Topics

We start with pure text sets, where the visual encoder plays no role and the two architectures should behave alike. Fig. 18 confirms this on ASR, STEM, and code. The allocation laws of the two architectures nearly coincide, with M exponents differing by at most 0.004. This control indicates that the differences in the multimodal topics below come from how visual inputs are processed, not from the two training runs themselves.

F.2 Multimodal Topics

We now turn to the five multimodal topics from the main text (Fig. 19). Unlike the text sets, these topics show a clear gap between the two architectures. On most of them, encoder-free models again favor larger M and smaller D, matching the aggregate result, although the size of the shift varies.

G Downstream Evaluation

Our scaling analyses are based on validation loss. To check whether the same trends hold on downstream tasks, we evaluate the pretrained checkpoints on a set of multimodal benchmarks, without any further training.

G.1 Benchmarks

Perception. CV-Bench (Tong et al., 2024) probes basic visual abilities, covering two-dimensional spatial relationships and counting as well as three-dimensional depth and distance. POPE (Li et al., 2023) measures object hallucination by asking whether a queried object appears in the image, and we pool its random, popular, and adversarial subsets. MME (Fu et al., 2026) poses concise yes/no questions spanning perception and cognition. We report accuracy on all three benchmarks.

Document Understanding. ChartQA (Masry et al., 2022) requires extracting values from charts and reasoning over them visually or logically. We report relaxed accuracy, which applies the standard relative tolerance to numerical answers. DocVQA (Mathew et al., 2021) evaluates question answering over document images, where both textual content and layout matter. We report average normalized Levenshtein similarity, which gives partial credit to near-miss answers. AI2D (Kembhavi et al., 2016) tests reasoning about the components and relationships in scientific diagrams, and we report multiple-choice accuracy. TextVQA (Singh et al., 2019) requires reading text in natural images and relating it to the surrounding scene, and we report the VQA consensus score against human answers.

General VQA. RealWorldQA (xAI, 2024) tests understanding of real-world scenes, including images captured from vehicles. MMStar (Chen et al., 2024a) consists of questions curated so that answering them requires visual evidence, spanning a range of perception and reasoning abilities. MMBench (Liu et al., 2024) offers broad coverage of multimodal capabilities, and we use its English development set. ScienceQA-IMG (Lu et al., 2022) is the subset of ScienceQA whose school-level science questions come with images. We evaluate answer selection on it without providing the reference explanations. We report accuracy on all four benchmarks.

G.2 Evaluation Protocol

We evaluate every pretrained checkpoint in the same 3-shot setting: each query is preceded by three solved examples, and all models share the same examples, prompt, and image preprocessing. The examples are drawn from the training split when one exists. Otherwise, we draw them from three images in the evaluation set and exclude all questions about those images from scoring. Images are resized with their aspect ratio preserved, to at most 192 visual tokens per example and 768 for the query, and the full prompt is kept within $4{,}096$ tokens. We use no external OCR, reference explanations, or test-time fine-tuning. For multiple-choice and yes/no questions, the model selects the candidate answer to which it assigns the highest likelihood. For open-ended questions, it decodes greedily for up to 32 tokens.

G.3 Results

Encoder-free models still score below encoder-based models at the scales we evaluate, but the gap narrows as training compute increases (Fig. 20). At the largest token budget, the gap also tends to narrow as the model grows (Tab. 6). Both trends are consistent with the loss-based findings.

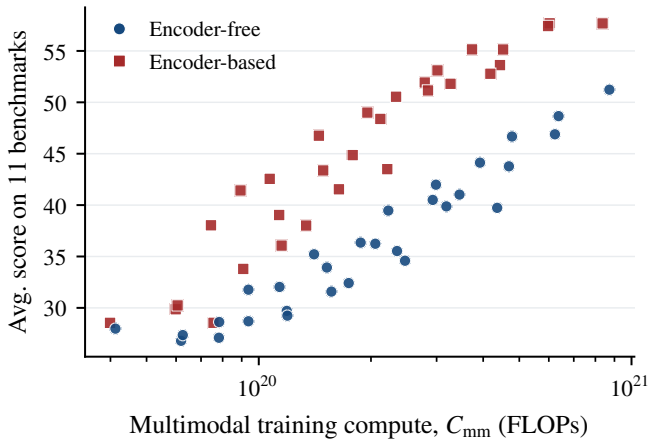


Figure 20: Downstream benchmark performance against multimodal training compute. The y-axis is the unweighted average 3-shot score over the 11 benchmarks.

Table 6: 3-shot benchmark scores at about 100B training tokens. "Based" and "Free" denote encoder-based and encoder-free models, respectively.

	6.1B-A425M		8B-A577M		13B-A922M		16B-A1B		22B-A1.6B		33B-A2.2B	
Benchmark	Based	Free	Based	Free	Based	Free	Based	Free	Based	Free	Based	Free
Perception												
CV-Bench	42.5	46.9	48.2	43.5	46.1	50.4	49.5	44.1	51.5	48.7	56.2	53.8
POPE	66.8	60.3	64.9	61.5	75.2	57.0	69.9	53.0	67.2	66.7	71.6	61.8
MME	62.1	58.0	64.3	53.1	60.9	57.1	67.7	57.9	67.9	58.7	60.7	58.8
Document												
ChartQA	34.8	10.1	48.5	29.7	51.9	29.4	47.4	35.6	53.1	37.8	54.3	42.7
DocVQA	39.6	15.4	59.8	36.4	67.3	46.5	63.9	47.4	70.6	51.0	68.9	57.0
AI2D	44.8	40.8	46.5	41.7	50.2	45.7	50.7	47.2	55.4	49.5	57.3	53.5
TextVQA	21.2	21.5	52.9	22.2	54.7	33.6	44.4	23.7	58.7	39.5	49.9	43.8
General VQA												
RealWorldQA	28.5	23.2	47.1	35.4	39.4	36.1	41.6	43.7	43.4	47.1	42.7	39.9
MMStar	32.2	27.3	34.4	28.3	35.8	28.1	36.9	29.5	35.8	29.2	37.6	31.7
MMBench-EN	55.6	40.1	55.5	41.6	64.7	46.3	60.8	47.2	67.1	48.3	69.1	57.6
ScienceQA-IMG	50.5	47.1	49.1	52.0	60.5	55.3	57.0	52.3	63.9	58.8	66.1	62.9
Average	43.5	35.5	51.9	40.5	55.2	44.1	53.6	43.8	57.7	48.7	57.7	51.2