# Supplementary material for: Scaling Laws for Native Multimodal Models

Mustafa Shukor[2] Enrico Fini[1] Victor Guilherme Turrisi da Costa[1] Matthieu Cord[2]
Joshua Susskind[1] Alaaeldin El-Nouby[1]
[1]Apple [2]Sorbonne University

This supplementary material is organized as follows:

- Appendix A: contains the implementation details and the hyperparameters used to train our models.
- Appendix B: contains detailed comparison between early and late fusion models.
- Appendix C: contains more details about scaling laws derivation, evaluation and additional results.
- Appendix D: contains discussion about the paper limitations.
- Appendix E: contains more results about MoEs and modality specialization.

## A. Experimental setup

In Table 2, we show the pre-training hyperparameters for different model configurations used to derive the scaling laws. The number of parameters ranges from 275M to 3.7B, with model width increasing accordingly, while the depth remains fixed at 24 layers. Learning rates vary by model size, decreasing as the model scales up. Based on empirical experiments and estimates similar to [10], we found these values to be effective in our setup. Training is optimized using a fully decoupled AdamW optimizer with momentum values $\beta_1 = 0.9$, $\beta_2 = 0.95$, and a weight decay of $1e{-4}$. The batch size is set to 2k samples, which account for 2M tokens, given 1k context length. Gradient clipping is set to 1.0, with a maximum warmup duration of 5k iterations, adjusted for shorter training runs: 1k and 2.5k warmup steps for models trained between 1k–4k and 5k–15k steps, respectively. For MoEs, we found that longer warmup is significantly better, so we adopt a 2.5k warmup for all runs under 20k steps. We use a constant learning rate schedule with cooldown during the final 20% of training, gradually reducing to zero following an inverse square root schedule. For vision processing, image inputs are divided into $(14, 14)$ patches, with augmentations including Random Resized Crop (resizing images to 224px with a scale range of [0.4, 1.0]) and Random Horizontal Flip with a probability of 0.5. We train our models on mixture of interleaved, image captions and text only data Table 1. For late fusion models, we found that using smaller learning rate for the vision encoder significantly boost the performance Table 4, and when both the encoder and decoder are initialized (Appendix B.7) we found that freezing the vision encoder works best Table 3.

| Data type | dataset | #samples | sampling prob. |
|---|---|---|---|
| Image-Caption | DFN [3] | 2B | 27% |
| | COYO [2] | 600M | 11.25% |
| | HQITP[13] | 400M | 6.75% |
| Interleaved | Obelics [7] | 141M Docs | 45% |
| Text | DCLM [8] | 6.6T Toks | 10% |

**Table 1. Pre-training data mixture.** Unless otherwise specified, the training mixture contains 45%, 45% and 10% of image captions, interleaved documents and text-only data.

## B. Late vs early fusion

This section provides additional comparison between early and late fusion models.

### B.1. Scaling FLOPs

Figure 1 compares early-fusion and late-fusion models when scaling FLOPs. Specifically, for each model size, we train multiple models using different amounts of training tokens. The performance gap between the two approaches mainly decreases due to increasing model sizes rather than increasing the number of training tokens. Despite the decreasing gap, across all the models that we train, early-fusion consistently outperform late-fusion.

### B.2. Changing the training data mixture

We analyze how the performance gap between early and late fusion models changes with variations in the training data mixture. As shown in Figure 3 and Figure 2, when fixing the model size, increasing the ratio of text and interleaved data favors early fusion. Interestingly, the gap remains largely unchanged for other data types. We also observe interference effects between different data types. Specifically, increasing the amount of interleaved data negatively impacts performance on image captions and vice versa. Additionally, increasing the proportion of text-only data slightly im-

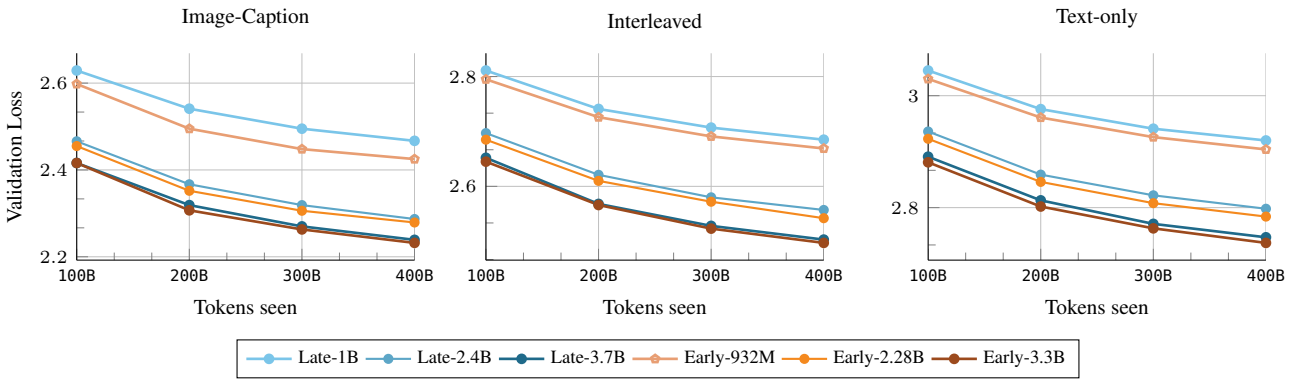


**Figure 1. Early vs late fusion: scaling training FLOPs.** We compare early and late fusion models when scaling both the model size and the number of training tokens. The gap decreases mainly due to scaling models size.

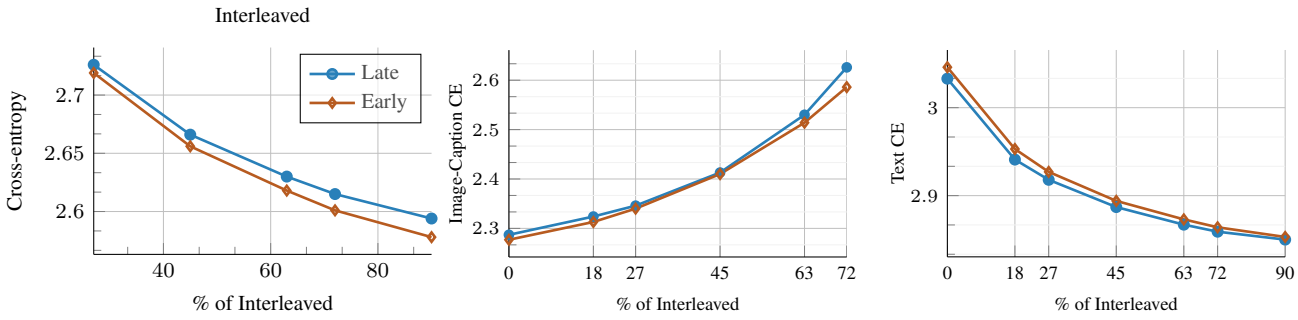


**Figure 2. Early vs late fusion: changing the training mixture.** We vary the training mixtures and plot the final training loss. Early fusion models become better when increasing the proportion of interleaved documents. Early and late fusion has 1.63B and 1.75B parameters respectively.

proves interleaved performance but increases loss on image captions. Overall, we find that text-only and interleaved data are correlated across different setups.

### B.3. Scaling image resolution is in favor of early-fusion

We examine how both architectures perform with varying image resolution. We fix the number of model parameters to 1.63B and 1.75B for early and late fusion respecively. All models are trained for 100K steps or 200B tokens. Since the patch size remains constant, increasing the resolution results in a higher number of visual tokens. For all resolutions, we maintain the same number of text tokens. As shown in Figure 4, the early-fusion model consistently outperforms the late-fusion model across resolutions, particularly for multimodal data, with the performance gap widening at higher resolutions. Additionally, we observe that the loss on text and interleaved data increases as resolution increases.

### B.4. Early-fusion is consistently better when matching the late-fusion model size

In this section, we compare the late-fusion model with different configurations of early-fusion one. Specifically, we train early-fusion models that match the late-fusion model in total parameters (Params), text model size (Text), and FLOPs (FLOPs), assuming 45-45-10 training mixture. As shown in Figure 5, early fusion consistently outperforms late fusion when normalized by total parameters, followed by normalization by FLOPs. When matching the text model size, early fusion performs better at higher ratios of interleaved data.

### B.5. Different late-fusion configuration

We examine how this scaling changes with different late-fusion configurations. Instead of scaling both the vision and text models equally, as done in the main paper, we fix the vision encoder size to 300M and scale only the text model. Figure 6 shows that late-fusion models lag behind at smaller model sizes, with the gap closing significantly as the text model scales. This suggests that allocating more parameters

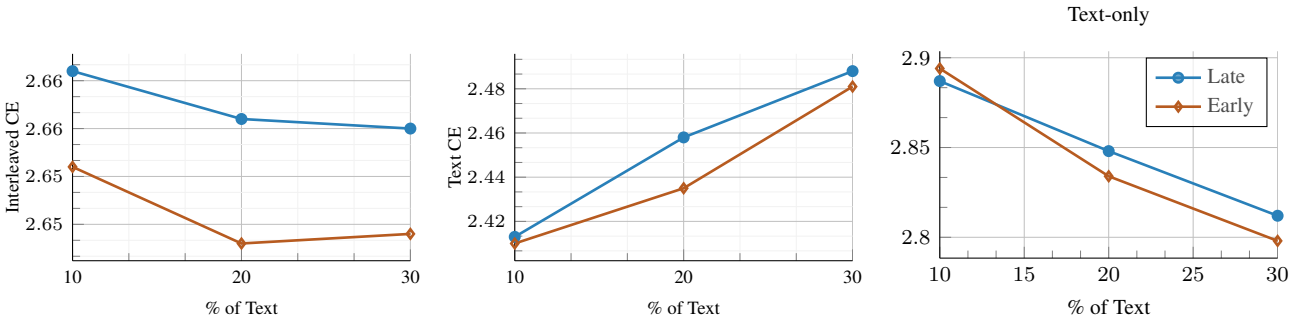


**Figure 3. Early vs late fusion: changing the amount of text-only data in the training mixture (isoFLOPs).** We vary the ratio of text-only data and plot the final training loss. The gap increases with the text data ratio in favor of early fusion model. Early fusion has 1.63B parameters and late fusion 1.75B parameters.

| | | | | | | |
|---|---|---|---|---|---|---|
| **Early-fusion** | | | | | | |
| Params | 275M | 468M | 932M | 1.63B | 2.28B | 3.35B |
| width | 800 | 1088 | 1632 | 2208 | 2624 | 3232 |
| depth | 24 | | | | | |
| Learning rate | 1.5e-3 | 1.5e-3 | 5e-4 | 4.2e-4 | 4e-4 | 3.5e-4 |
| **Late-fusion** | | | | | | |
| Params | 289M | 494M | 1B | 1.75B | 2.43B | 3.7B |
| vision encoder width | 384 | 512 | 768 | 1024 | 1184 | 1536 |
| vision encoder depth | 24 | | | | | |
| width | 768 | 1024 | 1536 | 2048 | 2464 | 3072 |
| depth | 24 | | | | | |
| Learning rate | 1.5e-3 | 1.5e-3 | 5e-4 | 4.2e-4 | 3.8e-4 | 3.3e-4 |
| **Early-fusion MoEs** | | | | | | |
| Active Params | 275M | 468M | 932M | 1.63B | 2.28B | 3.35B |
| width | 800 | 1088 | 1632 | 2208 | 2624 | 3232 |
| depth | 24 | | | | | |
| Learning rate | 1.5e-3 | 1.5e-3 | 5e-4 | 4.2e-4 | 4e-4 | 3.5e-4 |
| Training tokens | 2.5B-600B | | | | | |
| Optimizer | Fully decoupled AdamW [9] | | | | | |
| Optimizer Momentum | $\beta_1 = 0.9, \beta_2 = 0.95$ | | | | | |
| Minimum Learning rate | 0 | | | | | |
| Weight decay | 1e-4 | | | | | |
| Batch size | 2k | | | | | |
| Patch size | (14, 14) | | | | | |
| Gradient clipping | 1.0 | | | | | |
| MAximum Warmup iterations | 5k | | | | | |
| Augmentations: | | | | | | |
| `RandomResizedCrop` | | | | | | |
| `size` | 224px | | | | | |
| `scale` | [0.4, 1.0] | | | | | |
| `RandomHorizontalFlip` | $p = 0.5$ | | | | | |

**Table 2. Pre-training hyperparameters** We detail the hyperaparmeters used for pre-training different model configurations to derive scaling laws.

to shared components is more beneficial, further supporting the choice of early-fusion models.

## B.6. Different context lengths

In the paper, we use a 1k context length following [5]. Also following, this paper, we ignore the context length effect, as the model dimension dominates the training compute estimate. Moreover, [11] empirically found that scaling coefficients are robust to context length. Nevertheless, Our initial experiments (Figure 7) indicate that scaling the context length did not significantly affect the comparison between late and early fusion.

| Vision encoder lr scaler | Interleaved (CE) | Image-Caption (CE) | Text (CE) | AVG (CE) | AVG (SFT) (Acc) |
|---|---|---|---|---|---|
| 1 | 2.521 | 2.15 | 2.867 | 2.513 | 43.49 |
| 0.1 | 2.502 | 2.066 | 2.862 | 2.477 | 52.27 |
| 0.01 | 2.502 | 2.066 | 2.859 | 2.476 | 53.76 |
| 0.001 | 2.513 | 2.066 | 2.857 | 2.479 | – |
| 0 (frozen) | 2.504 | 2.061 | 2.856 | 2.474 | 54.14 |

**Table 3. Vision encoder scaler.** Freezing the vision encoder works best when initializing late-fusion models with pre-trained models.

| Vision encoder lr scaler | Interleaved (CE) | Image-Caption (CE) | Text (CE) | AVG (CE) | AVG (SFT) (Acc) |
|---|---|---|---|---|---|
| 0.1 | 2.674 | 2.219 | 3.072 | 2.655 | 34.84 |
| 0.01 | 2.672 | 2.197 | 3.071 | 2.647 | 38.77 |
| 0.001 | 2.674 | 2.218 | 3.073 | 2.655 | 38.46 |

**Table 4. Vision encoder scaler.** Reducing the learning rate for the vision encoder is better when training late-fusion models from scratch.

## B.7. Initializing from LLM and CLIP

We study the case where both late and early fusion models are initialized from pre-trained models, specifically DCLM-1B [8] and CLIP-ViT-L [12] for late fusion. Interestingly, Figure 8 shows that for text and interleaved multimodal documents, early fusion can match the performance of late fusion when trained for longer. However, closing the gap on image caption data remains more challenging. Notably, when considering the overall training cost, including that of pre-trained models, early fusion requires significantly longer training to compensate for the vision encoder's pre-training cost.

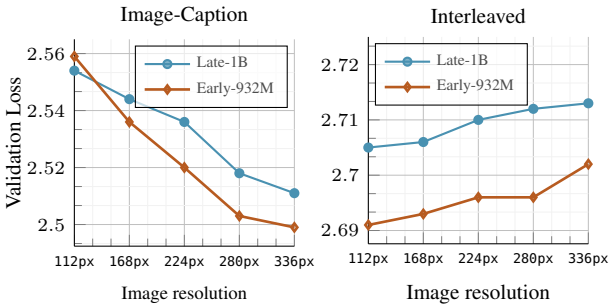


**Figure 4. Early vs late fusion: training with different image resolutions (isoFLOPs).** For the same training FLOPs we vary the image resolution (and thus the number of image tokens) during training and report the final training loss. Increasing resolution, hurts the performance on text and interleaved documents, while helping image captioning. The gap stays almost the same on text and interleaved data while slightly increase on image captioning in favor of early fusion.

# C. Scaling laws

## C.1. Fitting $L = F(N, D)$

Following [4], we determine the parameters that minimize the following objective across all our runs $i$:

$$\min_{a,b,e,\alpha,\beta} \sum_i \text{Huber}_\delta \left( \text{LSE}\left(a - \alpha \log N_i, b - \beta \log D_i, e\right) - \log L_i \right), \tag{1}$$

We perform this optimization across various initialization ranges and select the parameters that achieve the lowest loss across all initializations. Specifically, our grid search spans $\{0, 0.5, 2.5\}$ for $\alpha$ and $\beta$, $\{0, 5, 10, ..., 30\}$ for $a$ and $b$, and $\{-1, -0.5, 1, 0.5\}$ for $e$. We use the L-BFGS algorithm with $\delta = 1e-3$.

## C.2. Fitting $N \propto C^a$, $D \propto C^b$, $D \propto N^d$

While these equations have a closed-form solution [4] for early-fusion models that can be derived from **??**, this is not the case for late-fusion models without specifying either the vision encoder or text model size. To ensure a fair comparison, we derive these equations for both models, by performing linear regression in log space. We found that the regression is very close to the coefficient found with closed-form derivation Table 5. For instance, to derive $N = K_a C^a$, given a FLOP budget $C$ and a set of linearly spaced tokens $D_i$ ranging from 10B to 600B, we compute the model size for each $D_i$ as $N_i = \frac{C}{6D}$ for early fusion and $N_i = \frac{C}{6D} + 0.483 * N_v$ for late fusion (for the 45-45-10 mixture, $D_v = 0.544D$, thus $C = 6D(0.544N_v + N_t)$). We then apply **??** to obtain the loss for each model size and select $N$ that has the minimum loss. We repeat this for all FLOP values corresponding to our runs, resulting in a set of points $(C, N_{opt})$ that we use to regress $a$ and $K_a$. We follow a similar procedure to find $b$ and $d$. For late-fusion models, we regress a linear model to determine $N_v$ given $N$. Notably, even though we maintain a fixed width ratio for late-fusion models, this approach is more accurate, as embedding layers prevent a strictly fixed ratio between text and vision model sizes. We present the regression results in Figure 9.

| Model | $a$ | $b$ | $d$ | $n$ | $dn$ |
|---|---|---|---|---|---|
| Closed form | 0.52649 | 0.47351 | 0.89938 | 1.11188 | -0.05298 |
| Regression | 0.52391 | 0.47534 | 0.90052 | 1.10224 | -0.04933 |

**Table 5. Scaling laws parameters for early-fusion.** Doing regression to derive the scaling laws coefficients leads to very close results to using the closed-form solution.

## C.3. Fitting $L \propto C^c$

To determine the relationship between the final model loss and the compute budget $C$, we begin by interpolating the points corresponding to the same model size and compute the convex hull that covers the minimum loss achieved by all runs for each FLOP. This results in a continuous mapping from the FLOPs to the lowest loss. We consider a range of FLOPs, excluding very small values ($\leq 3e^{19}$), and construct a dataset of $(C, L)$ for linearly spaced compute $C$. Using this data, we find the linear relationship between $L$ and $C$ in the log space and deduce the exponent $c$. We visualize the results in Figure 13.

## C.4. Scaling laws for different target data type

In Figure 14, we derive the scaling laws for different target data types. In general, we observe that the model learns image captioning faster than interleaved data, as indicated by the higher absolute value of the scaling exponent (e.g., 0.062 vs 0.046), despite using the same data ratio for captioning and interleaved data (45% each). Additionally, we find that the model learns more slowly on text-only data, likely due to the smaller amount of text-only data (10%). Across model configurations, we find that early fusion scales similarly to late fusion on image captioning but has a lower multiplicative constant (49.99 vs 47.97). For MoEs, the model learns faster but exhibits a higher multiplicative constant. On text and interleaved data, early and late fusion models scale similarly and achieve comparable performance. However, MoEs demonstrate better overall performance while learning slightly more slowly.

## C.5. Scaling laws for different training mixtures

We investigate how the scaling laws change when modifying the training mixtures. Specifically, we vary the ratio of image caption, interleaved, and text-only data and report the results in Figure 15. Overall, we observe similar scaling trends, with only minor changes in the scaling coefficients. Upon closer analysis, we find that increasing the ratio of a particular data type in the training mixture, leads

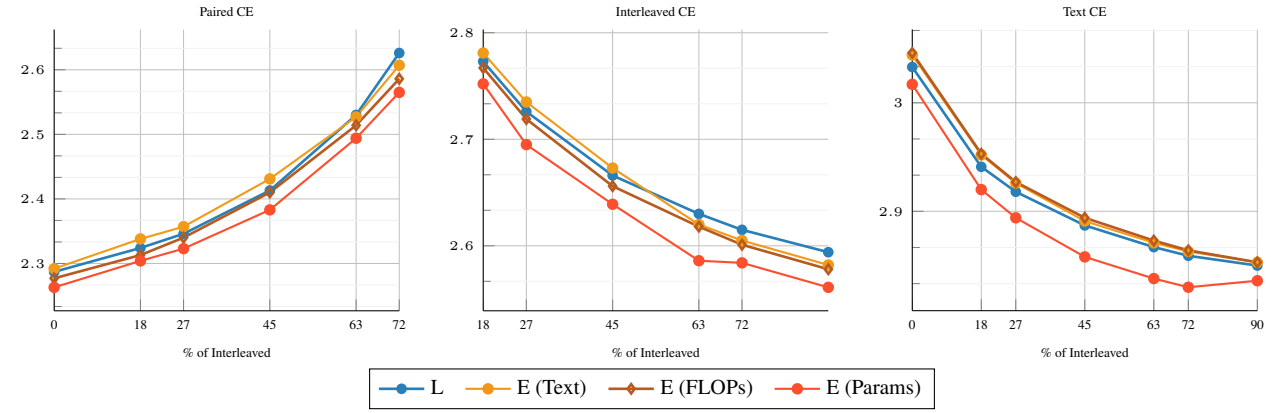


**Figure 5. Early vs late fusion: changing the training mixture and early-fusion configuration.** We vary the training mixtures and plot the final training loss for different configuration of early fusion models. For the same number of total parameters early fusion consistently outperform late fusion.

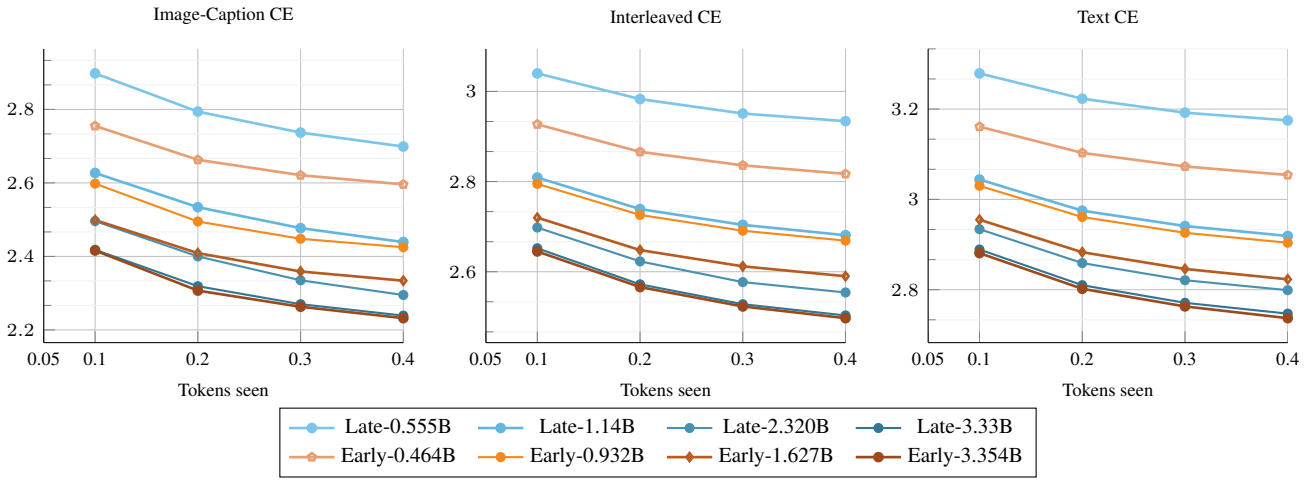


**Figure 6. Early vs late fusion: scaling training FLOPs while fixing the vision encoder size.** We compare early and late fusion models when scaling both the amount of training tokens and model sizes. For late fusion mdoels, we fix the vision encoder size (300M) and scale the text model (250M, 834M, 2B, 3B). The gap between early and late get tighter when scaling the text model.

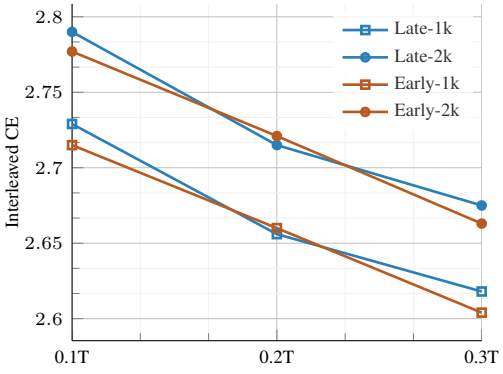


**Figure 7. Early vs late fusion with different context lengths.**

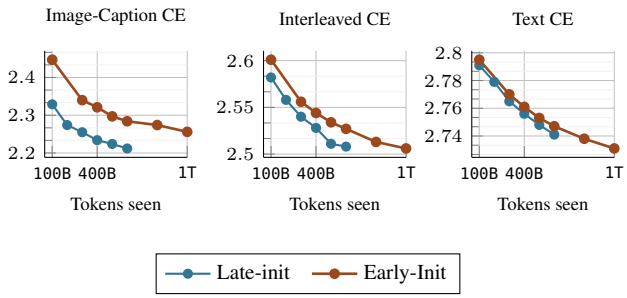


**Figure 8. Early vs late fusion when initializing the encoder and decoder.** Early-fusion can match the performance of late-fusion models when trained for longer. However, the gap is bigger on image-caption data.

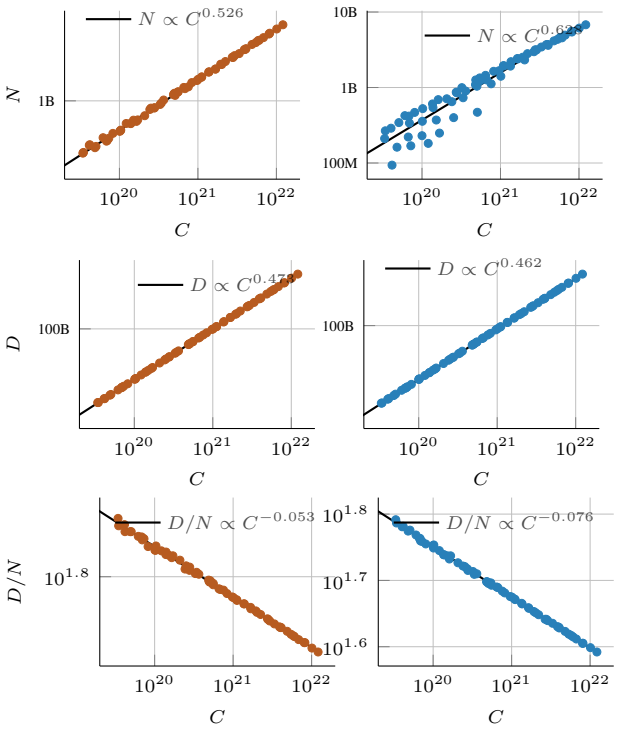


**Figure 9. Regression results of the scaling laws coefficients.** our estimation of the scaling coefficients is close to the closed form solution.

to a corresponding increase in its scaling exponent. For instance, increasing the ratio of image captions from 30% to 40% raises the absolute value of the exponent from 0.056 to 0.061. However, for text-only data, we do not observe significant changes in the scaling coefficients when varying its proportion in the training mixture.

| Parameter | MSE | R2 | MAE (%) |
|---|---|---|---|
| Held-in | 0.0029 | 0.9807 | 0.8608 |
| Held-out | 0.0004 | 0.9682 | 0.5530 |

**Table 6. Scaling laws prediction errors.** We report the mean square error, R2 and mean absolute error for the loss prediction for held-in and held-out (8B model) data.

| Model | E | $\alpha$ | $\beta$ | a | b | d |
|---|---|---|---|---|---|---|
| Avg | 1.80922 | 0.29842 | 0.33209 | 0.54302 | 0.48301 | 0.92375 |
| Std | 0.33811 | 0.10101 | 0.02892 | 0.08813 | 0.05787 | 0.23296 |

**Table 7. Scaling laws sensitivity.** We report the mean and standard deviation after bootstrapping with 100 iterations.

### C.6. Scaling laws evaluation

For each model size and number of training tokens, we compute the loss using the estimated functional form in **??** and compare it to the actual loss observed in our runs. Figure 10, Figure 11, and Table 6 visualizes these comparisons, showing that our estimation is highly accurate, particularly for lower loss values and larger FLOPs. We also assess our scaling laws in an extrapolation setting, predicting performance beyond the model sizes used for fitting. Notably, our approach estimates the performance of an 8B model with reasonable accuracy.

Additionally, we conduct a sensitivity analysis using bootstrapping. Specifically, we sample $P$ points with replacement ($P$ being the total number of trained models) and re-estimate the scaling law coefficients. This process is repeated 100 times, and we report the mean and standard deviation of each coefficient. Table 7 shows that our estimation is more precise for $\beta$ than for $\alpha$, primarily due to the smaller number of model sizes relative to the number of different token counts used to derive the scaling laws.

### C.7. Scaling laws for sparse NMMs.

Similar to dense models, we fit a parametric loss function (**??**) to predict the loss of sparse NMMs based on the number of parameters and training tokens, replacing the total parameter count with the number of active parameters. While incorporating sparsity is standard when deriving scaling laws for MoEs [1, 6, 14], we focus on deriving scaling laws specific to the sparsity level used in our MoE setup. This yields coefficients that are implicitly conditioned on the sparsity configuration.

We also experiment with a sparsity-aware formulation of the scaling law as proposed in [1], and observe consistent trends (Table 8). In particular, the exponents associated with model size ($N$) are substantially larger than those for training tokens ($\beta$), reinforcing the importance of scaling model size in sparse architectures. Additionally, we observe that the terms governing the scaling of active parameters decompose into two components.

## D. Discussion and Limitations

**Scaling laws for multimodal data mixtures.** Our scaling laws study spans different model configurations and training mixtures. While results suggest that the scaling law coefficients remain largely consistent across mixtures, a broader exploration of mixture variations is needed to validate this observation and establish a unified scaling law that accounts for this factor.

**Scaling laws and performance on downstream tasks.** Similar to previous scaling law studies, our analysis focuses on pretraining performance as measured by the validation loss. However, the extent to which these findings translate to downstream performance remains an open question and requires further investigation.

**Extrapolation to larger scales.** The accuracy of scaling law predictions improves with increasing FLOPs Ap-

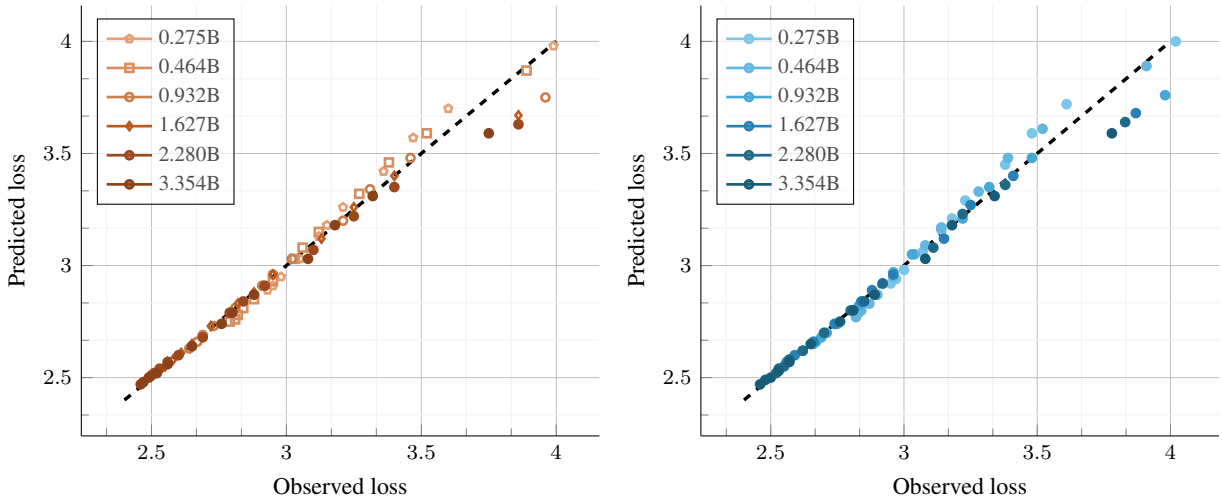


**Figure 10. Observed vs predicted loss.** We visualize the loss predicted by our scaling laws (**??**) and the actual loss achived by each run.

$L(N, D) = E + \frac{A}{N^\alpha} + \frac{B}{D^\beta}$ vs $L(N, D, S) = \frac{A}{N^\alpha} + \frac{B}{D^\beta} + \frac{C}{(1-S)^\lambda} + \frac{d}{(1-S)^\delta N^\gamma} + E$

| Model | E | A | B | $\alpha$ | $\beta$ | $\lambda$ | $\delta$ | $\gamma$ | C | d |
|---|---|---|---|---|---|---|---|---|---|---|
| $L(N, D)$ (**??**) | 2.158 | 381773 | 4659 | 0.710 | 0.372 | – | – | – | – | – |
| $L(N, D, S)$ [1] | 1.0788 | 1 | 4660 | 0.5890 | 0.3720 | 0.2 | 0.2 | 0.70956 | 1.0788 | 381475 |

**Table 8. Scaling laws for sparse native multimodal models**.

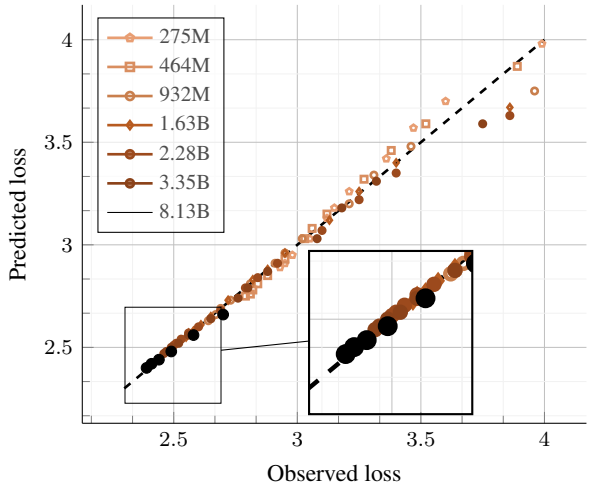


**Figure 11. Observed vs predicted loss.** We visualize the loss predicted by our scaling laws **??** and the actual loss achieved by each run. We can reliably predict the performance of models larger (8B params) than those used to fit the scaling laws.

pendix C. Furthermore, we validate our laws when extrapolating to larger model sizes (Appendix C.6). However, whether these laws can be reliably extrapolated to extremely large model sizes remains an open question.

**High resolution and early-fusion models.** Training early-fusion models with high-resolution inputs leads to a significant increase in vision tokens. While pooling techniques have been widely adopted for late-fusion models, alternative approaches may be necessary for early fusion. Given the similarity of early-fusion models to LLMs, it appears that techniques for extending context length could be beneficial.

**Scaling laws for multimodal MoEs models.** For MoEs, we consider only a single configuration (top-1 routing with 8 experts). We found this configuration to work reasonably well in our setup, and follow a standard MoEs implementation. However, the findings may vary when optimizing more the MoE architecture or exploring different load-balancing, routing strategies or different experts implementations.

## E. Mixture of experts and modality-specific specialization

### E.1. MoEs configuration

We experiment with different MoEs configuration by changing the number of experts and the top-k. We report a sample of these experiments in Table 9.

### E.2. MoEs specialization

We investigate multimodal specialization in MoE architectures. We compute a specialization score as the average difference between the number of text/images tokens as-

| | Accuracy | | | | | | CIDEr | |
|---|---|---|---|---|---|---|---|---|
| | AVG | VQAv2 | TextVQA | OKVQA | GQA | VizWiz | COCO | TextCaps |
| 4-E-top-1 | 40.0552 | 64.068 | 14.284 | 41.948 | 61.46 | 18.516 | 62.201 | 34.08 |
| 8-E-top-1 | 41.6934 | 65.684 | 17.55 | 42.908 | 63.26 | 19.065 | 67.877 | 39.63 |
| 8-E-top-2 | 42.8546 | 66.466 | 19.162 | 45.344 | 63.94 | 19.361 | 65.988 | 41.649 |
| 8-E-top-2 finegrained | 39.904 | 62.76 | 15.58 | 41.88 | 61.6 | 17.7 | 57.52 | 35.42 |

**Table 9. SFT results with different MoEs configurations.** .

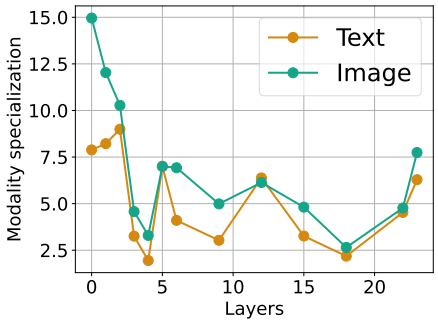


**Figure 12. Modality-specific specialization.** We visualize the experts specialization to text and image modalities. Models are evaluated on Obelics.

signed to each expert and a uniform assignment $(1/E)$. Additionally, we visualize the normalized number of text and image tokens assigned to each expert across layers. Figure 12 shows clear modality-specific experts, particularly in the early layers. Furthermore, the specialization score decreases as the number of layers increases but rises again in the very last layers. This suggests that early and final layers require more modality specialization compared to mid-layers. Additionally, we observe several experts shared between text and image modalities, a phenomenon not present in hard-routed or predefined modality-specific experts.

## References


[1] Samira Abnar, Harshay Shah, Dan Busbridge, Alaaeldin Mohamed Elnouby Ali, Josh Susskind, and Vimal Thilak. Parameters vs flops: Scaling laws for optimal sparsity for mixture-of-experts language models. *arXiv preprint arXiv:2501.12370*, 2025. 6, 7

[2] Minwoo Byeon, Beomhee Park, Haecheon Kim, Sungjun Lee, Woonhyuk Baek, and Saehoon Kim. Coyo-700m: Image-text pair dataset. https://github.com/kakaobrain/coyo-dataset, 2022. 1

[3] Alex Fang, Albin Madappally Jose, Amit Jain, Ludwig Schmidt, Alexander Toshev, and Vaishaal Shankar. Data filtering networks. *arXiv preprint arXiv:2309.17425*, 2023. 1

[4] 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. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, pages 30016–30030, 2022. 4

[5] 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. 3

[6] Jakub Krajewski, Jan Ludziejewski, Kamil Adamczewski, Maciej Pióro, Michał Krutul, Szymon Antoniak, Kamil Ciebiera, Krystian Król, Tomasz Odrzygóźdź, Piotr Sankowski, et al. Scaling laws for fine-grained mixture of experts. *arXiv preprint arXiv:2402.07871*, 2024. 6

[7] Hugo Laurençon, Lucile Saulnier, Léo Tronchon, Stas Bekman, Amanpreet Singh, Anton Lozhkov, Thomas Wang, Siddharth Karamcheti, Alexander Rush, Douwe Kiela, et al. Obelics: An open web-scale filtered dataset of interleaved image-text documents. *Advances in Neural Information Processing Systems*, 36, 2024. 1

[8] Jeffrey Li, Alex Fang, Georgios Smyrnis, Maor Ivgi, Matt Jordan, Samir Gadre, Hritik Bansal, Etash Guha, Sedrick Keh, Kushal Arora, et al. Datacomp-lm: In search of the next generation of training sets for language models. *arXiv preprint arXiv:2406.11794*, 2024. 1, 3

[9] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 3

[10] Brandon McKinzie, Zhe Gan, Jean-Philippe Fauconnier, Sam Dodge, Bowen Zhang, Philipp Dufter, Dhruti Shah, Xianzhi Du, Futang Peng, Anton Belyi, et al. Mm1: methods, analysis and insights from multimodal llm pre-training. In *European Conference on Computer Vision*, pages 304–323. Springer, 2025. 1

[11] Tim Pearce and Jinyeop Song. Reconciling kaplan and chinchilla scaling laws. *arXiv preprint arXiv:2406.12907*, 2024. 3

[12] 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*, pages 8748–8763. PMLR, 2021. 3

[13] Kanchana Ranasinghe, Brandon McKinzie, Sachin Ravi, Yinfei Yang, Alexander Toshev, and Jonathon Shlens. Perceptual grouping in contrastive vision-language models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5571–5584, 2023. 1

[14] Siqi Wang, Zhengyu Chen, Bei Li, Keqing He, Min Zhang, and Jingang Wang. Scaling laws across model architectures: A comparative analysis of dense and MoE models in large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*,

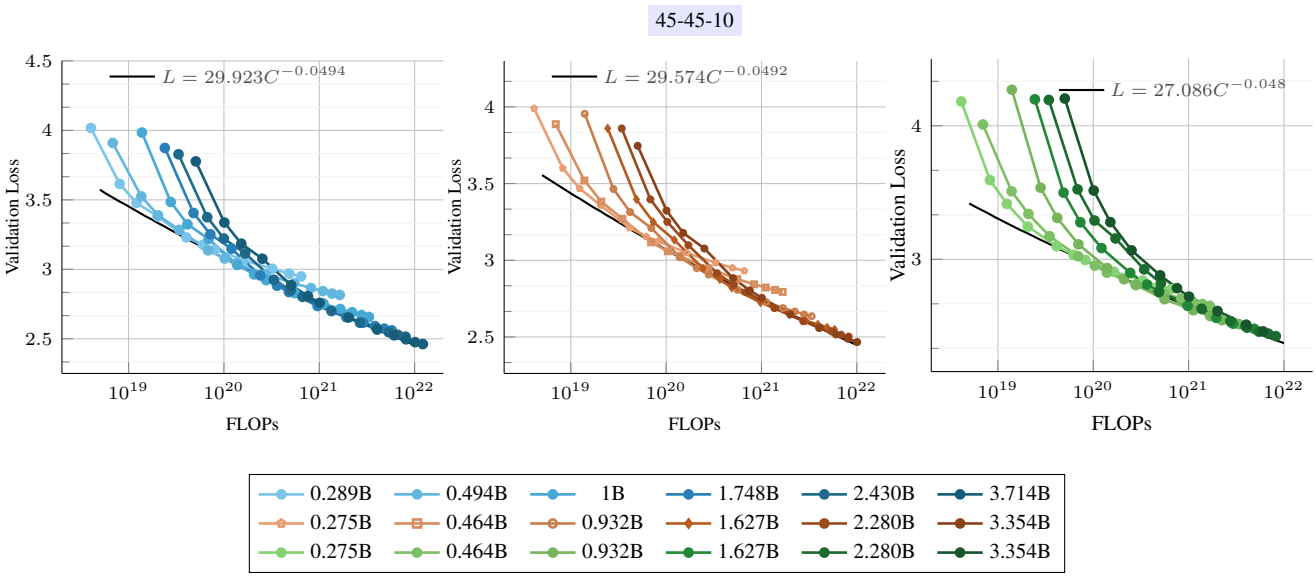


**Figure 13. Scaling laws for native multimodal models.** From left to right: late-fusion (dense), early-fusion (dense) and early-fusion MoEs. The scaling exponents are very close for all models. However, MoEs leads to overall lower loss (smaller multiplicative constant) and takes longer to saturate.

pages 5583–5595, Miami, Florida, USA, 2024. Association for Computational Linguistics. 6

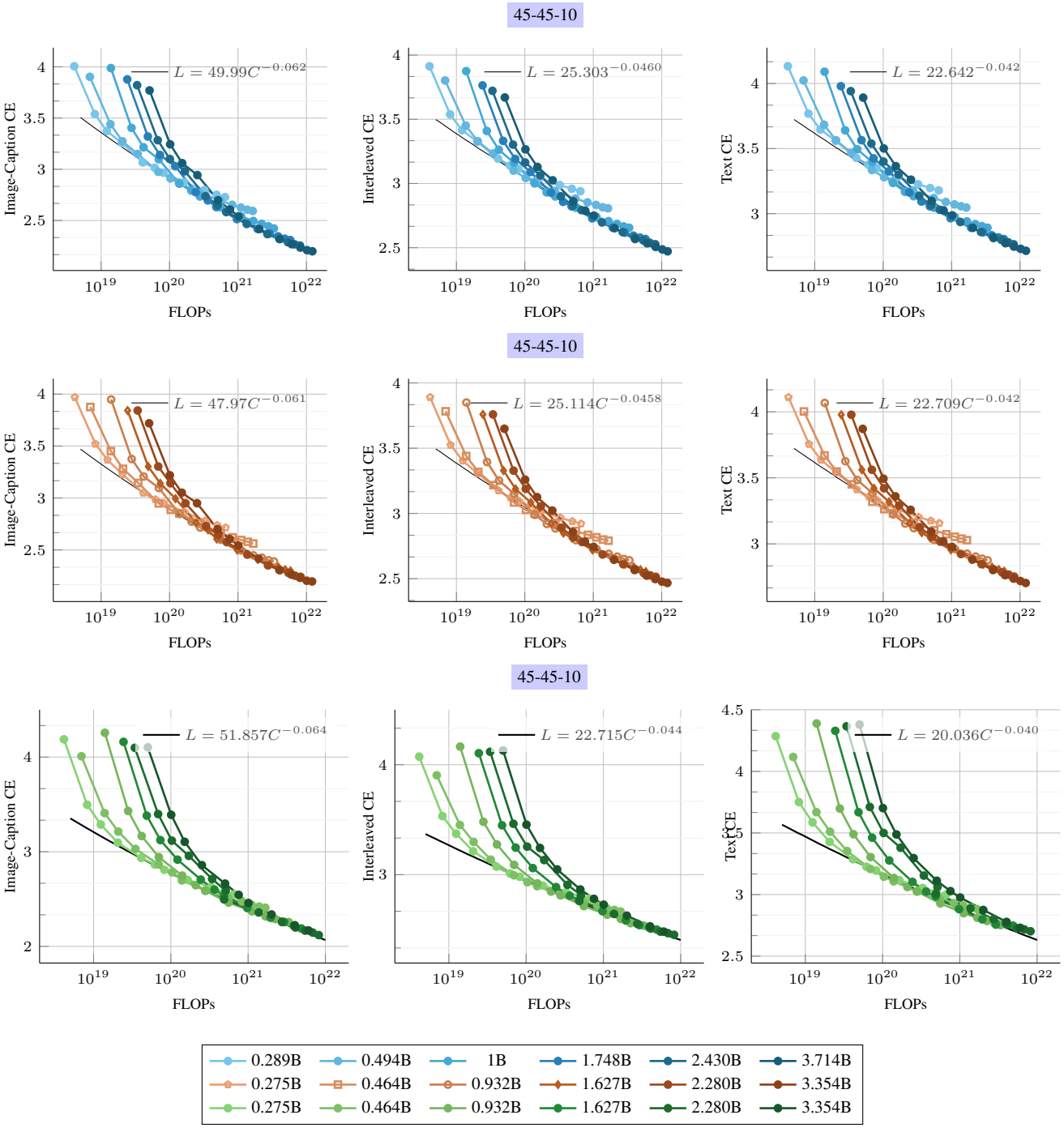


**Figure 14. Scaling laws for native multimodal models.** From top to bottom: late-fusion (dense), early-fusion (dense) and early-fusion MoEs. From left to right: cross-entropy on the validation set of image-caption, interleaved and text-only data.

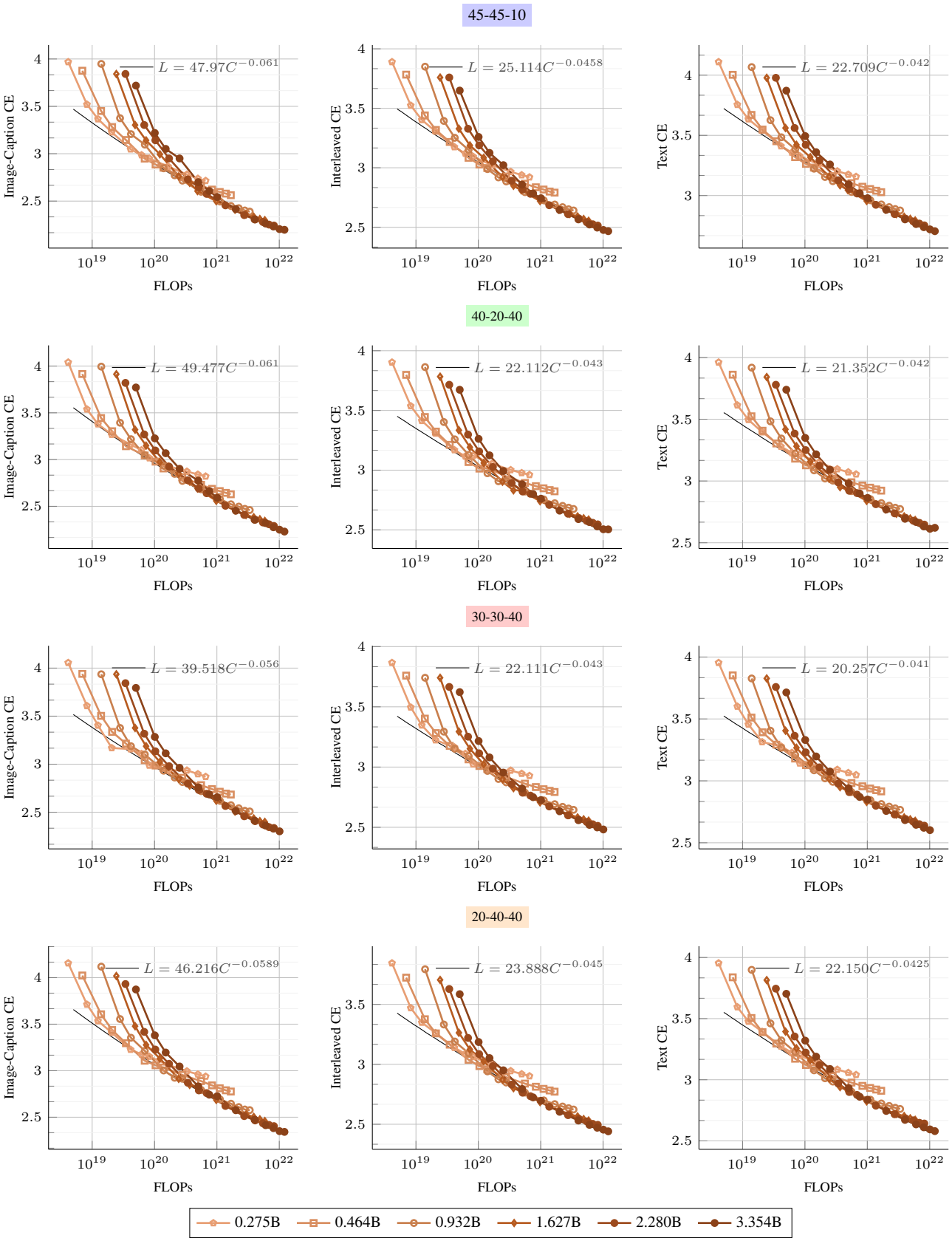


**Figure 15. Scaling laws for early-fusion native multimodal models.** Our runs across different training mixtures (Image-caption-Interleaved-Text) and FLOPs. We visulize the final validation loss on 3 data types: HQITP (left), Obelics (middle) and DCLM (right).