POST-CONNECTOR DISTILLATION: AN EFFICIENT RECIPE FOR TRAINING VISION ENCODERS

Anonymous authors
Paper under double-blind review

ABSTRACT

Different efficient vision encoder architectures are needed for different hardware, but developing a new vision encoder requires a large amount of compute, elaborate training pipelines, and repeated downstream fine-tuning to measure its quality on VLM tasks. For fast iteration, we introduce *post-connector distillation*, a simple two-stage recipe for learning from an existing VLM. The first stage is a simple student-teacher reconstruction loss over the final multimodal connector output embeddings, which requires only unlabeled images or videos. This stage enables direct evaluation on downstream VQA benchmarks without additional instruction tuning, significantly accelerating model development. The second stage jointly refines the encoder, connector, and language backbone through supervised fine-tuning (SFT). It closes the remaining capability gaps between the student vision encoder and teacher vision encoder. In our ablations, training loss is a poor indicator of encoder quality, while the early benchmark scores predict which data mixture and architecture perform better after fine-tuning. Paired with a different language backbone, it matches Qwen3.5-2B's ViT across 13 VLM benchmarks while requiring 2.2$\times$ fewer encoder-plus-connector FLOPs, and outperforms all other evaluated encoders.

1 INTRODUCTION

Vision-language models (VLMs) pair a vision encoder with a language backbone through a multimodal connector (Liu et al., 2023; 2024a). The vision encoder alone determines what the VLM can perceive *and* how much computation each image requires. Recent VLMs merge patches in the connector, after the encoder, to reduce the visual tokens processed by the language backbone (Qwen Team, 2026; Kimi Team et al., 2026). While patch merging reduces the language backbone's input, it does not reduce the encoder's cost. Encoder efficiency therefore needs direct improvement, independently of language backbone improvements. Widely used pretrained vision encoders, like CLIP, SigLIP 2, DINOv2, and InternViT, are Vision Transformers (ViTs) (Dosovitskiy et al., 2021; Radford et al., 2021; Tschannen et al., 2025; Oquab et al., 2024; Chen et al., 2024c) whose self-attention costs grow quadratically with the input resolution and quickly become impractical for high-resolution inputs (Vasu et al., 2025; 2024). Convolutional and hybrid architectures substantially reduce the cost (Chen et al., 2024a), but new architectures cannot reuse existing ViTs and must be pretrained from scratch. Yet training a new encoder remains a substantial undertaking.

Contrastive pretraining methods such as CLIP and SigLIP learn strong visual representations from large collections of image–text pairs (Radford et al., 2021; Zhai et al., 2023; Tschannen et al., 2025), but at a high cost; SigLIP 2, for example, is trained on 40B image–text examples. To reduce computation and data requirements, distillation methods like RADIO transfer knowledge from pretrained teachers to a student (Ranzinger et al., 2024; Heinrich et al., 2025; Ranzinger et al., 2026). However, distillation recipes still require substantial data and compute, often alongside multiple teachers, auxiliary objectives, and data augmentation. Such cost limits novel architecture research. Different hardware and deployment settings require entirely different encoder architectures; finding strong architecture means comparing many candidate designs, each of which needs its own pretraining run. Existing recipes make exploring new architectures prohibitively expensive.

Beyond the cost of encoder training itself, evaluating candidate encoders is also expensive. Vision encoders are typically monitored during training using standalone vision benchmarks such as ImageNet

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.

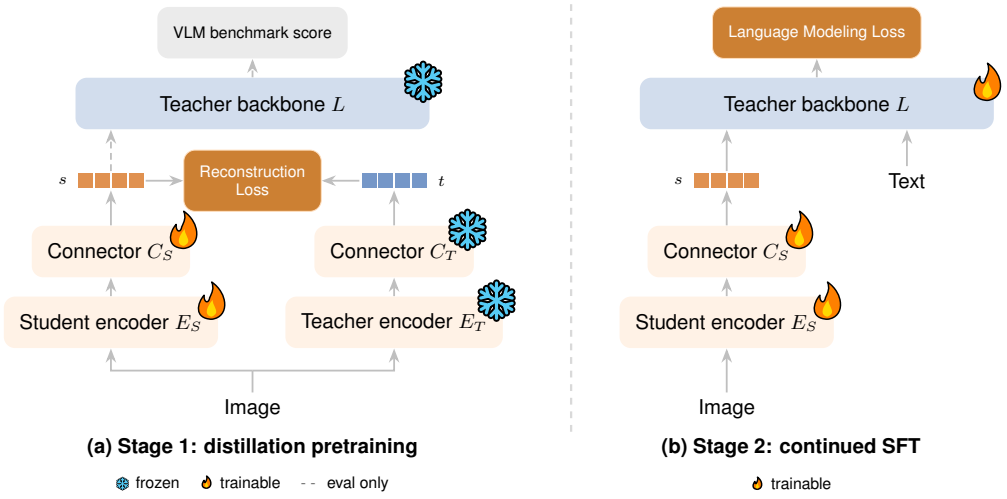

Figure 1: **Two-stage post-connector distillation.** Stage 1 reconstructs a frozen teacher's post-connector tokens without executing the language backbone; that backbone directly evaluates any student checkpoint. Stage 2 removes the teacher branch and jointly fine-tunes the student encoder, connector, and language backbone.

classification and ADE20K semantic segmentation (Deng et al., 2009; Zhou et al., 2017; Tschannen et al., 2025; Ranzinger et al., 2026). These benchmarks are convenient measures of general visual representation quality, but they are becoming saturated and do not directly reflect how well an encoder performs inside a VLM (Tong et al., 2024). Measuring downstream VLM quality therefore requires integrating the encoder into a VLM and performing additional visual instruction tuning (Tong et al., 2024). This makes repeated evaluation across checkpoints and architectures costly, and feedback on a candidate architecture typically arrives only after its pretraining has finished. We therefore ask: *can downstream VLM quality be exposed directly during encoder training?*

In this work, we propose *post-connector distillation*, a two-stage encoder training recipe that learns visual representations directly from a pretrained VLM. The teacher's connector merges and projects encoder features into the tokens consumed by its language backbone; we call these *post-connector visual tokens*. The first stage trains a student encoder and connector to reconstruct these tokens using a simple reconstruction loss, without supervision or data augmentation. Because the targets come entirely from the teacher encoder and connector, training does not run the language backbone, avoiding its computation. At evaluation time, the student encoder and connector simply replace their teacher counterparts while the teacher's language backbone remains unchanged, enabling downstream VLM evaluation at every checkpoint without additional visual instruction tuning.

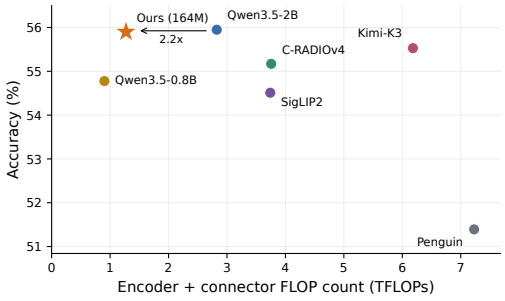

Figure 2: **Performance-efficiency comparison of vision encoders** over 13 VLM benchmarks. We measure encoder and connector FLOPs for a 1024×768 image. Our encoder (star) is the most competitive while being 2.2x fewer FLOPs than Qwen3.5-2B's vision encoder.

Our choice also simplifies the architecture: distilling *pre*-connector visual tokens requires another adapter to map the student's features to the teacher's feature space. EUPE and AM-RADIO use MLP adapters (Zhu et al., 2026; Ranzinger et al., 2024). The multimodal connector already provides a learned mapping from the encoder to the language backbone; post-connector distillation uses that mapping directly. Evaluating pre-connector-distilled encoders requires the student encoder, the additional adapter, the teacher connector, and the teacher language backbone. This adds parameters and architectural choices to the evaluated system, even though the distillation adapter is later discarded

when integrating the encoder with a new connector. Our recipe instead trains and evaluates the student encoder and its own connector throughout. The second stage then jointly fine-tunes the student encoder, connector, and language backbone on curated instruction data. Our ablations show that this image-language supervision improves VLM quality and adapts the encoder to important downstream task details (Table 3). Figure 1 illustrates the complete two-stage recipe.

Because our recipe provides VLM benchmark scores throughout training, VLM performance is an early diagnostic signal for refining architecture and data. This contrasts with prior vision encoder training recipes for VLMs, where downstream VLM quality is typically assessed only after encoder training and subsequent visual instruction tuning, delaying useful feedback until billions of tokens have been consumed. With our method, every ablation can be judged directly on VLM benchmarks rather than proxy metrics. While proxies show relative improvements over baselines, they are poor absolute metrics and say little about VLM quality. Our contributions:

1. **A simple and efficient two-stage vision encoder training recipe.** Stage 1 uses unlabeled visual data, one teacher, one reconstruction objective, and no language backbone; Stage 2 supplies task-aware refinement through supervised fine-tuning (Section 3).

2. **Direct downstream signal during encoder development.** Stage 1 models show meaningful VLM signal without visual instruction tuning, enabling fast encoder development (Section 4.4).

3. **A strong, efficient vision encoder.** Under a controlled visual instruction tuning recipe with a different language backbone , our hybrid vision encoder is competitive with the strongest public vision encoder in our comparison while requiring substantially fewer FLOPs (Figure 2).

2 Related Work

Efficient VLM encoders. Efficient vision backbones reduce cost through convolutions, linear attention, or state-space token mixing (Vasu et al., 2023; Cai et al., 2023; Hatamizadeh & Kautz, 2025), and FastVLM, MobileCLIP, and ViTamin pretrain efficient encoders with CLIP-style objectives for VLMs (Vasu et al., 2025; 2024; Faghri et al., 2025; Chen et al., 2024a). Other work reduces high-resolution cost with encoders that emit fewer post-connector visual tokens (Ge et al., 2024; Luo et al., 2025; Wei et al., 2025). These encoders still require visual instruction tuning before downstream quality can be measured, so new architectures requires both encoder training and visual instruction tuning. Our work evaluates checkpoints directly through the frozen teacher language backbone, without visual instruction tuning.

Contrastive, self-supervised, and distilled encoders. CLIP and SigLIP learn image–text alignment; DINOv2 learns self-supervised features (Radford et al., 2021; Zhai et al., 2023; Tschannen et al., 2025; Oquab et al., 2024). RADIO combines CLIP, DINO, and SAM (Kirillov et al., 2023) teachers, while ScaleKD and InternVL align teacher representations (Ranzinger et al., 2024; Heinrich et al., 2025; Ranzinger et al., 2026; Fan et al., 2024; Chen et al., 2024c). These methods also hide downstream VLM quality and require visual instruction tuning to evaluate the combined VLM quality.

Concurrent VLM recipes. Penguin-VL and UltraViT are closest to our approach (Zhang et al., 2026; Metaxas et al., 2026). Penguin-VL initializes its vision encoder from a pretrained language model and trains the encoder and connector with a frozen language backbone, first using image captioning and feature distillation and then captioning alone. Subsequent VLM pretraining and supervised fine-tuning jointly train the encoder, connector, and language backbone. UltraViT uses local and global feature distillation, followed by image-captioning with a frozen language backbone. In contrast, we use a randomly initialized encoder and distill post-connector features, without a language backbone during distillation. We avoid the language backbone computation compared to Penguin-VL and enable downstream VLM evaluation of our encoder, without visual instruction tuning, which is deferred to our second stage. Table 1 summarizes related work.

Table 1: **Encoder training recipes.** Comparison of the dominant general-knowledge acquisition stage. Our method has a tight feedback loop via SFT-free VLM evaluation at every checkpoint.

Method	Loss	Teacher	Data (# images)	LLM?	Labels?	Visual Inst. Tuning?
SigLIP2	Contrastive, CE, self-distill.	N/A	40B	No	Yes	Yes
C-RADIOv4	Patch + summary distill.	3 pretrained encoders	700M	No	No	Yes
Penguin-VL	CE, distill.	Pretrained encoder	270M	Yes	Yes	Yes
UltraViT	Patch + summary distill.	Pretrained encoder	150M	No	No	Yes
Ours	Post-connector distillation	Encoder in VLM	100M	No	No	No

3 METHOD

Assume a VLM teacher with a vision encoder E_T, a connector C_T, and a language backbone L_T. For an image x, the teacher exposes

$$t(x) = C_T(E_T(x)) \in \mathbb{R}^{n(x)\times d_L}, \tag{1}$$

where $n(x)$ is the number of visual tokens and d_L is the language hidden dimension. The backbone L_T processes $t(x)$. Our student is an encoder E_S and a connector C_S that maps an image to the same spatial arrangement:

$$s(x) = C_S(E_S(x)) \in \mathbb{R}^{n(x)\times d_L}. \tag{2}$$

3.1 STAGE 1: RECONSTRUCTION PRETRAINING

During Stage 1, we calculate both the amplitude and direction error between the teacher visual tokens $t(x)$ and student visual tokens $s(x)$ for each token i:

$$\ell_{\text{amp},i} = \frac{1}{d_L}\|s_i - t_i\|_1, \qquad \ell_{\text{dir},i} = 1 - \frac{s_i^\top t_i}{\|s_i\|_2\|t_i\|_2 + \epsilon}. \tag{3}$$

We average token losses $\ell_{\text{amp},i}$ and $\ell_{\text{dir},i}$ into sample losses $\mathcal{L}_{\text{amp}}$ and $\mathcal{L}_{\text{dir}}$. The final Stage 1 loss:

$$\mathcal{L}_{\text{S1}} = \mathcal{L}_{\text{amp}} + \mathcal{L}_{\text{dir}}. \tag{4}$$

No caption, questions or answer are required in Equation (4). The teacher language backbone is ignored. Stage 1 computes one frozen teacher-vision forward and one student forward/backward, using visual data without text supervision.

VLM evaluation without SFT training. At checkpoint k, we form $E_S^{(k)} + C_S^{(k)} + L_T$: student post-connector visual tokens replace the teacher's post-connector visual tokens and the frozen language backbone L_T generates answers on VLM benchmarks. No visual instruction tuning is needed.

3.2 STAGE 2: SUPERVISED REFINEMENT

Stage 2 initializes E_S, C_S from Stage 1, attaches the same teacher language backbone L_T, and unfreezes all three. On visual instruction tuning data, it minimizes assistant-token cross-entropy:

$$\mathcal{L}_{\text{S2}} = \frac{1}{|B|}\sum_{b\in B}\frac{1}{|\mathcal{T}_b|}\sum_{i\in\mathcal{T}_b} -\log p_{E_S,C_S,L_T}(y_i \mid y_{<i}, x_b). \tag{5}$$

Stage 2 fine tunes the encoder to attend to details that matter for VLM tasks. In Section 4.1 and Table 3, we show that Stage 2 is necessary to close the gap with the teacher vision encoder.

4 EXPERIMENTS

We claim that cheap access to VLM evaluations during encoder training simplifies novel vision encoder development. To demonstrate, we train a hybrid attention-convolution encoder for a VLM, ablating data, objective, and architecture, using early VLM evaluation signal to inform experiments.

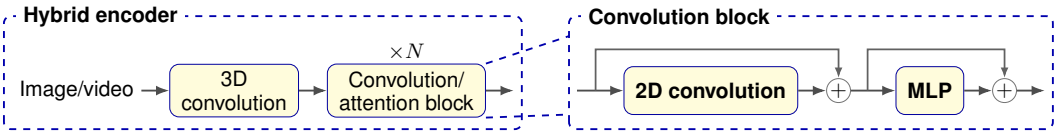

Figure 3: **Hybrid encoder architecture. Left:** Images and video frames are patchified by a 3D convolution (2D spatial, 1D two-frame temporal) and processed by 14 blocks: 8 convolution and 6 grouped-query attention blocks (Ainslie et al., 2023, 16 query heads, 8 key/value heads of dimension 64). **Right:** a convolution block, which mixes tokens with a gated depthwise 3×3 2D convolution followed by an MLP. Our encoder has 164M parameters and is adapted from LFM2.5-230M (Liquid AI, 2026) without its token embeddings.

We use Qwen3.5-2B as our teacher: it has a 306M ViT E_T, a 25M connector C_T and a 2B language backbone L_T (Qwen Team, 2026). We train a 164M hybrid convolution and grouped-query attention vision encoder[1] E_S and a 25M connector C_S with the same output shape as the teacher connector C_T.

In Stage 1, we optimize our newly initialized encoder and connector with per-head Muon (Kimi Team et al., 2026, Sec. 2.5) with a peak learning rate of 10^{-3}, weight decay of 0.1 and a warmup–stable–decay (WSD) learning rate schedule (Hu et al., 2024) for one epoch over 80M images and 20M videos. In Stage 2, we optimize the student encoder, student connector and teacher language backbone with per-head Muon with a peak learning rate of 5×10^{-5} for 17M image and 3M videos, composed of a mix of public and synthetically-generated visual instruction tuning examples.

During training, we evaluate on four VLM benchmarks: InfoVQA (Mathew et al., 2022), RefCOCO-M (Moondream, 2025), OCR-Bench (Liu et al., 2024b) and MMStar (Chen et al., 2024b). Because of our post-connector distillation, we can frequently evaluate our encoder E_S with the connector C_S and the language backbone L_T without any additional training.

Table 2: **Student and teacher VLM comparison.** All benchmark scores are normalized. We evaluate Qwen3.5-2B in comparison to our hybrid convolution-attention encoder E_S and connector C_S connected to a fine-tuned L_T language backbone. Bold indicates 0.1 points within the best.

Benchmark	Teacher	Ours
ChartQA	78.8	**83.2**
DocVQA	93.5	**94.5**
TextVQA	**78.8**	**78.8**
POPE	88.9	**89.3**
InfoVQA	75.2	**76.0**
RealWorldQA	**71.6**	71.5
MME	**75.0**	74.2
BLINK	60.5	**64.7**
MMMU-Pro	**39.7**	27.1
MathVista	**70.0**	65.5
LogicVista	**48.8**	37.6
RefCOCO-M	**91.8**	86.0
MMStar	**66.9**	61.7
Mean	**72.3**	70.0

After Stage 1 and Stage 2, we evaluate on a suite of 13 VLM benchmarks covering chart understanding, document understanding, visual hallucinations, perception, reasoning and grounding.

Training Results. We compare the Stage 2 result (a fine-tuned combination of our new vision encoder and the Qwen3.5-2B language backbone) against the frozen teacher Qwen3.5-2B VLM in Table 2. Our student encoder is stronger on perception benchmarks (ChartQA, BLINK) but weaker on reasoning benchmarks (MMMU-Pro) and grounding benchmarks (RefCOCO-M). Additional experiments with task-intensive data mixes yield improved task-specific scores, but lower scores elsewhere, reflecting a data mix trade-off; our mix favors broad VLM performance.

We note that we intend to combine our trained encoder E_S new, unseen language backbones to enable strong, efficient VLMs. To evaluate our encoder for VLMs, we test encoder quality through integration with an unseen language backbone.

[1]We base our encoder architecture on the Liquid Foundation Model 2 architectures (Amini et al., 2025); specifically the smallest LFM2.5-230M configuration without the 67M token embedding matrix. Figure 3 summarizes our architecture; exact details appear in Section A.

Table 3: **Encoder quality comparison.** Scores for our encoder and other VLM encoders on ChartQA (Masry et al., 2022), DocVQA (Mathew et al., 2021), TextVQA (Singh et al., 2019), POPE (Li et al., 2023), InfoVQA (Mathew et al., 2022), RealWorldQA (xAI, 2024), MME (Fu et al., 2025), BLINK (Fu et al., 2024), MMMU-Pro (Yue et al., 2025), MathVista (Lu et al., 2024), LogicVista (Xiao et al., 2024), RefCOCO-M (Moondream, 2025), and MMStar (Chen et al., 2024b). In addition, we evaluate our encoder after Stage 1 only. Our encoder, after both stages, matches the teacher Qwen3.5 2B's vision encoder and surpasses other encoders.

Model Parameters	Stage 2 164M	Stage 1 164M	Qwen3.5 2B 306M	Qwen3.5 0.8B 88M	SigLIP2 412M	C-RADIO v4 431M	Kimi-K3 401M	Penguin 441M
ChartQA	70.2	69.8	69.6	70.6	69.8	68.7	**72.3**	65.0
DocVQA	**82.7**	81.2	82.3	81.7	73.5	73.9	81.4	73.4
TextVQA	71.3	71.1	70.7	66.4	**71.6**	70.3	70.0	65.5
POPE	85.6	86.2	**86.9**	84.4	86.4	86.2	86.3	85.9
InfoVQA	47.7	49.8	45.1	**50.6**	39.7	43.4	50.1	40.4
RealWorldQA	58.8	**60.5**	58.8	53.9	55.7	58.0	54.1	54.4
MME	56.8	54.4	56.5	51.6	53.2	**59.4**	52.4	51.4
BLINK	45.0	43.6	45.5	43.5	**47.4**	45.8	42.6	39.1
MMMU-Pro	18.3	16.8	19.0	18.2	18.3	**19.9**	18.4	16.3
MathVista	47.5	45.2	44.0	46.0	**47.7**	45.8	46.0	40.8
LogicVista	26.6	25.5	26.4	**28.2**	26.6	26.6	26.4	23.7
RefCOCO-M	73.1	69.1	**82.6**	76.5	75.4	77.4	80.0	74.3
MMStar	43.1	41.0	40.0	40.7	**43.3**	41.9	41.9	37.9
Mean	**55.9**	54.9	**55.9**	54.8	54.5	55.2	55.5	51.4

4.1 THE RECIPE PRODUCES A STRONG ENCODER

Because our encoder learns to reconstruct the post-connector visual-tokens for Qwen3.5's LLM backbone, we test whether our encoder is useful beyond Qwen3.5 by pairing it with a different LLM backbone; we choose LFM2.5-230M (Liquid AI, 2026). We compare our 164M hybrid encoder against Qwen3.5-2B's 306M encoder, Qwen3.5-0.8B's 88M encoder, a SigLIP2 400M encoder, a C-RADIOv4 400M encoder, Kimi K3's 401M encoder and Penguin-VL's 441M encoder. No encoders were pretrained in conjunction with the LFM2.5-230M LLM. Therefore, comparing their VLM scores with this backbone assesses encoder transfer quality in a common setting, rather than within each encoder's original training setup.

Setup. For each encoder, we initialize a connector and follow 2 stages (Liu et al., 2023; 2024a):

1. Train only the newly initialized multimodal connector parameters for 1M images.
2. Train all parameters from the encoder, connector and LFM2.5-230M backbone for 6M samples.

During both stages, we use per-head Muon and visual instruction tuning data. We then evaluate the VLMs on 13 VLM benchmarks across charts, documents, text recognition, perception, reasoning, and grounding. All VLMs use the same stage visual instruction tuning recipe.

Results. Figure 2 shows a efficiency-quality tradeoff and Table 3 reports individual benchmarks. Our hybrid encoder and Qwen3.5-2B's vision encoder both score 55.9. Despite our simple encoder training recipe, our small, efficient encoder matches or surpasses larger encoders with more complex training recipes. Among these evaluated models, our encoder offers the best quality-efficiency tradeoff. We also measure whether Stage 2 Section 3.2 is necessary by following the same LFM2.5-230M integration recipe with our encoder after Stage 1 only (reconstruction only). We find that Stage 2 consistently improves performance across a diverse set of benchmarks, highlighting the need for vision-language alignment, even during encoder training (Liu et al., 2025).

4.2 WHY RECONSTRUCTION?

Prior work combines next-token cross-entropy with feature reconstruction against a pretrained teacher (Zhang et al., 2026), or uses multi-teacher distillation (NVIDIA, 2026); we instead use a simple reconstruction objective alone (Section 3.1). Here, we empirically justify our choice and compare our proposed reconstruction objective in Stage 1 with a language modeling cross-entropy objective.

Setup. We use a web-scale paired image-caption dataset, with image-caption pairs from DataComp-1B (Gadre et al., 2023) and scanned documents with detailed OCR transcriptions (Kydlíček et al., 2025). We train 3 newly initialized 164M hybrid encoders, as in Section 4, with 3 different objectives, using an internal VLM checkpoint as the teacher:

1. Our proposed reconstruction objective (Section 3.1).
2. Cross-entropy language modeling, where we keep the teacher language backbone L_T frozen and tune the student encoder E_S and connector C_S to minimize cross entropy on the image caption or document transcription, conditioned on the image.
3. The sum of both the above reconstruction and cross-entropy objectives.

Results. We evaluate on InfoVQA, MMStar, OCRBench, and RefCOCO-M and report results in Table 4. Reconstruction leads to stronger VLM benchmark scores than both cross-entropy-based objectives *and* is more compute efficient (because it needs no LLM backbone). We hypothesize that the low-quality text and homogeneous distribution (web quality image captions and noisy OCR transcriptions) lead to poor instruction-following capabilities. During evaluation, we see models respond by captioning the image, instead of answering the question.

Follow-up. To test our hypothesis, we do the same experiment with a high-quality, diverse visual instruction tuning dataset. We repeat the above experiment with this smaller, higher-quality dataset and compare the reconstruction and the combined cross-entropy and reconstruction objectives, reporting results on the same four benchmarks in Table 5. With this high-quality, diverse, visual instruction tuning data, we see that including cross-entropy improves VLM benchmark scores.

Takeaway. Based on these findings, we use pure reconstruction for two reasons. First, the language backbone is expensive. For faster iteration speed, we want to use as little compute as possible. Second, unlabeled image data is much easier to collect at scale than high-quality instruction-following multimodal data. Thus, reconstruction is an ideal Stage 1 objective.

Table 4: **Stage 1 objective ablation on caption/OCR data.** We compare reconstruction, cross-entropy language modeling and a simple sum of both objectives. "LLM?" indicates whether the language backbone is required; "FLOPs" measures total training FLOPs over 100M caption/OCR training samples. Objectives using cross entropy require the language backbone and thus, significantly more training FLOPs. "Mean" is the mean score of the Stage 1 checkpoint over InfoVQA, MMStar, OCRBench, and RefCOCO-M. We hypothesize that the caption/OCR data's low-quality text labels, in combination with a language modeling objective, degrade VLM performance. We explore this hypothesis in Section 4.2 and Table 5.

Objective	LLM?	FLOPs (10^{20})	Mean
Reconstruction	No	7	**59.1**
Cross Entropy	Yes	9	39.1
Both	Yes	12	54.7

Table 5: **Stage 1 objective ablation on visual instruction tuning data.** As in Table 4, we compare Stage 1 objectives: reconstruction and a sum of reconstruction and cross-entropy language modeling. Table 4 explains the columns. This data is a small (17M) high-quality, diverse visual instruction tuning dataset. As a result, training with cross entropy loss improves VLM scores, but requires much more compute.

Objective	LLM?	FLOPs (10^{20})	Mean
Reconstruction	No	6	59.9
Both	Yes	11	**61.0**

4.3 DATA-COMPOSITION ABLATIONS

Data curation and mixing is critical for improving model capabilities (Xie et al., 2023; Ye et al., 2025; Diao et al., 2025). However, different training distributions will have different training loss magnitudes, and prior methods for VLM encoder design require visual instruction tuning to measure

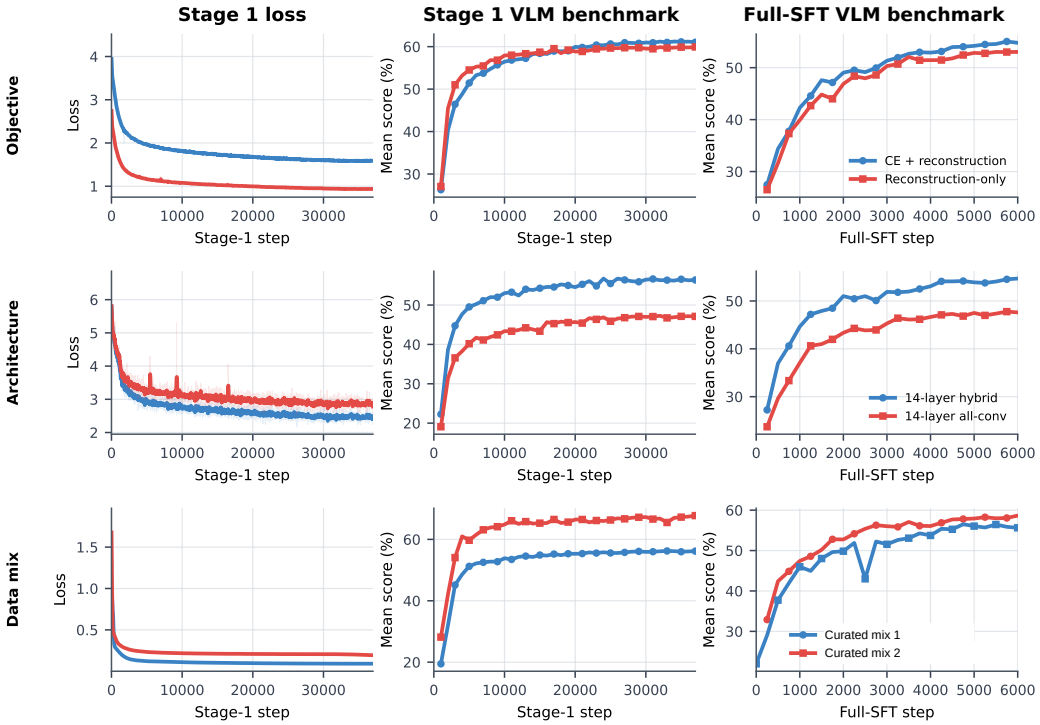

Figure 4: **Objective, Architecture and Data Ablations.** We compare Stage 1 training loss (Section 3.1, left), VLM quality (InfoVQA, MMStar, OCRBench and RefCOCO-M, middle) during Stage 1, and VLM quality during visual instruction tuning (right). We argue that Stage 1 VLM quality is a better signal than training loss with three experiments. **Top:** Objectives have different loss magnitudes (left); loss is not always indicative of encoder quality (right). **Middle:** Architectures can have similar losses (left), but VLM evals (center) highlight gaps in encoder quality (right). **Bottom:** Different data have different loss scales (left); again, Stage 1 VLM evals (middle) better show final encoder quality (right). Our post-connector distillation exposes VLM benchmarks during training, enabling faster encoder iterations.

VLM quality. Our proposed post-connector distillation method enables data ablations for encoder training without expensive visual instruction tuning. To demonstrate, we compare two curated image data mixtures in the bottom row of Figure 4. While Mix 1 has a better training loss (Figure 4, bottom left), Mix 2 has better VLM benchmark scores after the expensive visual instruction tuning stage with LFM2.5-230M (Figure 4, bottom right). Thus, we hypothesize that Mix 1 is a simpler, less diverse dataset, leading to lower training loss. However, since we use our proposed post-connector distillation method, we can evaluate VLM benchmark scores during training before the expensive visual instruction tuning (Figure 4, bottom middle). VLM benchmark scores show that Mix 2 beats Mix 1 early in training. In addition to early signal, post-connector distillation shows individual benchmark scores. Figure 5 shows that InfoVQA and OCRBench scores are weak, prompting us to investigate Mix 1's dataset distribution to inform Mix 2's distribution.

Takeaway. Post-connector distillation enables VLM benchmarks for diagnosing and understanding encoder quality during training, instead of after an expensive visual instruction tuning stage.

4.4 Post-connector distillation enables easy architecture development

In Sections 4.2 and 4.3, we highlight that post-connector distillation enables efficient VLM encoder development. Now we demonstrate that post-connector distillation also enables simple architecture ablations. We compare the 14-layer hybrid vision encoder with its 14-layer all-convolution counterpart in the middle row of Figure 4. Again, we evaluate our encoder checkpoints during training on four VLM benchmarks: InfoVQA, MMStar, OCRBench and RefCOCO-M. The all-convolution encoder

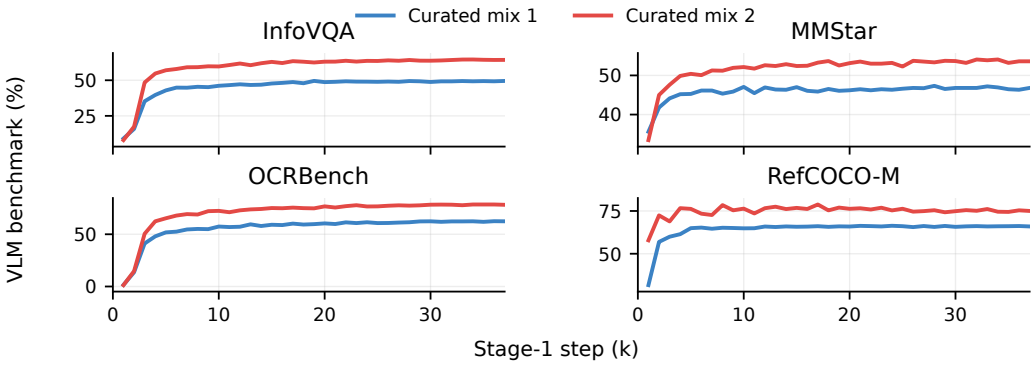

Figure 5: **Data-ablation VLM benchmark breakdown.** Our proposed post-connector distillation method enables cheap encoder evaluation during training; we plot a breakdown of Figure 4's bottom middle plot on InfoVQA, MMStar, OCRBench and RefCOCO-M (during Stage 1 training). This breakdown highlights that Mix 1 lacks chart data, due to the poor InfoVQA and OCRBench scores. *Without* post-connector distillation, an expensive visual instruction tuning stage is necessary before such insights.

has a slightly higher training loss, but is significantly weaker on VLM benchmarks during training. Indeed, this gap in VLM quality carries over after the expensive visual instruction tuning stage.

Takeaway. Post-connector distillation makes architecture ablations, and in fact all of our proposed ablations, very simple. Instead of using proxy metrics like training loss, validation loss, linear probing or k-nearest-neighbors classifiers, we can directly use the VLM benchmarks we care about. **There's no proxy metric better than the benchmark itself.** Concretely, the all-convolution encoder has slightly worse training loss, but the VLM benchmarks degradation is too much to justify the improved architectural efficiency. While training loss and other proxy metrics are useful relative metrics between a strong baseline and a proposed variant, they are poor absolute metrics and are insufficiently informative of final VLM quality.

In summary, post-connector distillation drastically accelerates vision encoder research and development with its simple setup and rich diagnostic signal.

5 Limitations

Like all distillation methods, our post-connector distillation needs a strong teacher, requires a forward pass of the (potentially expensive) teacher on every sample, and inherits the (potentially unknown) teacher weaknesses and biases. Second, we still use the teacher language backbone for VLM benchmarks during training; large teacher VLMs like Kimi K3 (3T parameters) are infeasible for efficient encoder development. Finally, post-connector distillation requires the student to reproduce the teacher's post-connector visual token layout, architectures that emit a different number or different ordering of tokens (Heinrich et al., 2026) need an explicit mapping.

6 Conclusion

Post-connector distillation trains strong vision encoders with one reconstruction target, unlabeled visual data, no language-backbone compute in Stage 1 and short visual instruction tuning in Stage 2. The resulting 164M hybrid encoder matches the teacher Qwen3.5-2B vision encoder on VLM tasks, while using 2.2$\times$ fewer FLOPs. Our method drastically accelerates the vision encoder research and development process by enabling practitioners to measure VLM performance even in early training stages.

AI USE STATEMENT

In this work, we used generative AI tools to implement methods, create charts, edit code, draft and edit parts of the paper, summarize existing literature and search for information. We have reviewed all AI-assistent work: we verified LLM code and experimental results, reviewed charts and the other contents of the paper, and performed our own literature review. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

Sections 3 and 4 specify the training objective and experimental design. The appendices provide architecture descriptions, hyperparameters, evaluation sample counts, and additional per-task results. The method inherits pretrained teacher weights and internal-only training corpora; While reproducing our work exactly would require our own dataset, open datasets are sufficient to reproduce our conclusions.

REFERENCES

Joshua Ainslie, James Lee-Thorp, Michiel de Jong, Yury Zemlyanskiy, Federico Lebrón, and Sumit Sanghai. GQA: Training generalized multi-query transformer models from multi-head checkpoints. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pp. 4895–4901, 2023. URL https://aclanthology.org/2023.emnlp-main.298/.

Alexander Amini et al. LFM2 technical report. *arXiv preprint arXiv:2511.23404*, 2025. URL https://arxiv.org/abs/2511.23404.

Han Cai, Junyan Li, Muyan Hu, Chuang Gan, and Song Han. EfficientViT: Lightweight multi-scale attention for high-resolution dense prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023. URL https://openaccess.thecvf.com/content/ICCV2023/html/Cai_EfficientViT_Lightweight_Multi-Scale_Attention_for_High-Resolution_Dense_Prediction_ICCV_2023_paper.html.

Jieneng Chen, Qihang Yu, Xiaohui Shen, Alan Yuille, and Liang-Chieh Chen. ViTamin: Designing scalable vision models in the vision-language era. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024a. URL https://openaccess.thecvf.com/content/CVPR2024/html/Chen_ViTamin_Designing_Scalable_Vision_Models_in_the_Vision-Language_Era_CVPR_2024_paper.html.

Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Jiaqi Wang, Yu Qiao, Dahua Lin, and Feng Zhao. Are we on the right way for evaluating large vision-language models? In *Advances in Neural Information Processing Systems*, volume 37, pp. 27056–27087, 2024b. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/2f8ee6a3d766b426d2618e555b5aeb39-Abstract-Conference.html.

Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, Bin Li, Ping Luo, Tong Lu, Yu Qiao, and Jifeng Dai. InternVL: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024c. URL https://openaccess.thecvf.com/content/CVPR2024/html/Chen_InternVL_Scaling_up_Vision_Foundation_Models_and_Aligning_for_Generic_CVPR_2024_paper.html.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *CVPR*, 2009. URL https://ieeexplore.ieee.org/document/5206848.

Shizhe Diao, Yu Yang, Yonggan Fu, Xin Dong, Dan Su, Markus Kliegl, Zijia Chen, Peter Belcak, Yoshi Suhara, Hongxu Yin, Mostofa Patwary, Yingyan Lin, Jan Kautz, and Pavlo Molchanov. Nemotron-CLIMB: CLustering-based iterative data mixture bootstrapping for language model pre-training. In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/24a8968affe71ffe4067d022b9d16566-Abstract-Datasets_and_Benchmarks_Track.html.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. URL https://iclr.cc/virtual/2021/oral/3458.

Fartash Faghri, Pavan Kumar Anasosalu Vasu, Cem Koc, Vaishaal Shankar, Alexander T. Toshev, Oncel Tuzel, and Hadi Pouransari. MobileCLIP2: Improving multi-modal reinforced training. *Transactions on Machine Learning Research*, 2025. URL https://openreview.net/forum?id=WeF9zolng8.

Jiawei Fan, Chao Li, Xiaolong Liu, and Anbang Yao. ScaleKD: Strong vision transformers could be excellent teachers. In *Advances in Neural Information Processing Systems*, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/73bda9a20f6f9f6074ce822e76f126bb-Abstract-Conference.html.

Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, Yunsheng Wu, Rongrong Ji, Caifeng Shan, and Ran He. MME: A comprehensive evaluation benchmark for multimodal large language models. In *Advances in Neural Information Processing Systems (Datasets and Benchmarks Track)*, 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/d79a27cf2772fe00be7f341efc0eb517-Abstract-Datasets_and_Benchmarks_Track.html.

Xingyu Fu, Yushi Hu, Bangzheng Li, Yu Feng, Haoyu Wang, Xudong Lin, Dan Roth, Noah A. Smith, Wei-Chiu Ma, and Ranjay Krishna. BLINK: Multimodal large language models can see but not perceive. In *Proceedings of the European Conference on Computer Vision*, 2024. URL https://eccv.ecva.net/virtual/2024/poster/2230.

Samir Yitzhak Gadre, Gabriel Ilharco, Alex Fang, Jonathan Hayase, Georgios Smyrnis, Thao Nguyen, Ryan Marten, Mitchell Wortsman, Dhruba Ghosh, Jieyu Zhang, Eyal Orgad, Rahim Entezari, Giannis Daras, Sarah Pratt, Vivek Ramanujan, Yonatan Bitton, Kalyani Marathe, Stephen Mussmann, Richard Vencu, Mehdi Cherti, Ranjay Krishna, Pang Wei Koh, Olga Saukh, Alexander Ratner, Shuran Song, Hannaneh Hajishirzi, Ali Farhadi, Romain Beaumont, Sewoong Oh, Alex Dimakis, Jenia Jitsev, Yair Carmon, Vaishaal Shankar, and Ludwig Schmidt. DataComp: In search of the next generation of multimodal datasets. In *Advances in Neural Information Processing Systems*, 2023. URL https://papers.neurips.cc/paper_files/paper/2023/hash/56332d41d55ad7ad8024aac625881be7-Abstract-Datasets_and_Benchmarks.html.

Chunjiang Ge, Sijie Cheng, Ziming Wang, Jiale Yuan, Yuan Gao, Jun Song, Shiji Song, Gao Huang, and Bo Zheng. ConvLLaVA: Hierarchical backbones as visual encoder for large multimodal models. *arXiv preprint arXiv:2405.15738*, 2024.

Ali Hatamizadeh and Jan Kautz. MambaVision: A hybrid Mamba-transformer vision backbone. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Hatamizadeh_MambaVision_A_Hybrid_Mamba-Transformer_Vision_Backbone_CVPR_2025_paper.html.

Greg Heinrich, Mike Ranzinger, Hongxu Yin, Yao Lu, Jan Kautz, Andrew Tao, Bryan Catanzaro, and Pavlo Molchanov. RADIOv2.5: Improved baselines for agglomerative vision foundation models. In *CVPR*, 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Heinrich_RADIOv2.5_Improved_Baselines_for_Agglomerative_Vision_Foundation_Models_CVPR_2025_paper.html.

Greg Heinrich, Mike Ranzinger, Collin McCarthy, Natan Bagrov, Eugene Khvedchenya, Bryan Catanzaro, Jan Kautz, Andrew Tao, and Pavlo Molchanov. RADIO1d: Elastic representations for condensed vision modeling. In *Forty-third International Conference on Machine Learning*, 2026. URL https://icml.cc/virtual/2026/poster/65887.

Shengding Hu, Yuge Tu, Xu Han, Ganqu Cui, Chaoqun He, Weilin Zhao, Xiang Long, Zhi Zheng, Yewei Fang, Yuxiang Huang, Xinrong Zhang, Zhen Leng Thai, Chongyi Wang, Yuan Yao, Chenyang Zhao, Jie Zhou, Jie Cai, Zhongwu Zhai, Ning Ding, Chao Jia, Guoyang Zeng, Dahai Li, Zhiyuan Liu, and Maosong Sun. MiniCPM: Unveiling the potential of small language models with scalable training strategies. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=3X2L2TFr0f.

Kimi Team et al. Kimi K3: Open frontier intelligence. arXiv preprint arXiv:2607.24653, 2026. URL https://arxiv.org/abs/2607.24653.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything. In *ICCV*, 2023. URL https://openaccess.thecvf.com/content/ICCV2023/html/Kirillov_Segment_Anything_ICCV_2023_paper.html.

Hynek Kydlíček, Guilherme Penedo, and Leandro von Werra. FinePDFs. https://huggingface.co/datasets/HuggingFaceFW/finepdfs, 2025.

Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne 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*, 2023. URL https://aclanthology.org/2023.emnlp-main.20/.

Liquid AI. LFM2.5-230M. Hugging Face model card, 2026. URL https://huggingface.co/LiquidAI/LFM2.5-230M.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *Advances in Neural Information Processing Systems*, volume 36, pp. 34892–34916, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/6dcf277ea32ce3288914faf369fe6de0-Abstract-Conference.html.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 26296–26306, 2024a. URL https://openaccess.thecvf.com/content/CVPR2024/html/Liu_Improved_Baselines_with_Visual_Instruction_Tuning_CVPR_2024_paper.html.

Yiming Liu, Yuhui Zhang, Dhruba Ghosh, Ludwig Schmidt, and Serena Yeung-Levy. Data or language supervision: What makes CLIP better than DINO? In *Findings of the Association for Computational Linguistics: EMNLP 2025*, 2025. URL https://aclanthology.org/2025.findings-emnlp.98/.

Yuliang Liu, Zhang Li, Mingxin Huang, Biao Yang, Wenwen Yu, Chunyuan Li, Xucheng Yin, Cheng-lin Liu, Lianwen Jin, and Xiang Bai. OCRBench: On the hidden mystery of OCR in large multimodal models. *Science China Information Sciences*, 67(12):220102, 2024b. doi: 10.1007/s11432-024-4235-6. URL https://link.springer.com/article/10.1007/s11432-024-4235-6.

Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. MathVista: Evaluating mathematical reasoning of foundation models in visual contexts. In *International Conference on Learning Representations*, 2024. URL https://iclr.cc/virtual/2024/poster/18900.

Gen Luo, Yiyi Zhou, Yuxin Zhang, Xiawu Zheng, Xiaoshuai Sun, and Rongrong Ji. Feast your eyes: Mixture-of-resolution adaptation for multimodal large language models. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/d2781f7df8825bbde6cef55adfbcd283-Abstract-Conference.html.

Ahmed Masry, Do Xuan Long, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. ChartQA: A benchmark for question answering about charts with visual and logical reasoning. In *Findings of the Association for Computational Linguistics: ACL 2022*, 2022. URL https://aclanthology.org/2022.findings-acl.177/.

Minesh Mathew, Dimosthenis Karatzas, and C. V. Jawahar. DocVQA: A dataset for VQA on document images. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 2200–2209, 2021. URL https://openaccess.thecvf.com/content/WACV2021/html/Mathew_DocVQA_A_Dataset_for_VQA_on_Document_Images_WACV_2021_paper.html.

Minesh Mathew, Viraj Bagal, Rubèn Tito, Dimosthenis Karatzas, Ernest Valveny, and C. V. Jawahar. InfographicVQA. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 1697–1706, 2022. URL https://openaccess.thecvf.com/content/WACV2022/html/Mathew_InfographicVQA_WACV_2022_paper.html.

Ioannis Maniadis Metaxas, Adrian Bulat, Alberto Baldrati, Anestis Zaganidis, Yassine Ouali, Hyeonuk Kim, and Georgios Tzimiropoulos. Ultravit: Latency-optimized on-device vision encoder for large vision-language models. In *ECCV*, 2026. URL https://eccv.ecva.net/virtual/2026/poster/4831.

Moondream. RefCOCO-M. Hugging Face dataset card, 2025. URL https://huggingface.co/datasets/moondream/refcoco-m.

NVIDIA. C-RADIOv4-SO400M. Hugging Face model card, 2026. URL https://huggingface.co/nvidia/C-RADIOv4-SO400M.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jégou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024. URL https://openreview.net/forum?id=a68SUt6zFt.

Qwen Team. Qwen3.5-2B. Hugging Face model card and configuration, 2026. URL https://huggingface.co/Qwen/Qwen3.5-2B.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *ICML*, 2021. URL https://proceedings.mlr.press/v139/radford21a.html.

Mike Ranzinger, Greg Heinrich, Jan Kautz, and Pavlo Molchanov. Am-radio: Agglomerative vision foundation model – reduce all domains into one. In *CVPR*, 2024. URL https://openaccess.thecvf.com/content/CVPR2024/html/Ranzinger_AM-RADIO_Agglomerative_Vision_Foundation_Model_Reduce_All_Domains_Into_One_CVPR_2024_paper.html.

Mike Ranzinger, Greg Heinrich, Collin McCarthy, Jan Kautz, Andrew Tao, Bryan Catanzaro, and Pavlo Molchanov. C-RADIOv4 (tech report). *arXiv preprint arXiv:2601.17237*, 2026. URL https://arxiv.org/abs/2601.17237.

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*, 2019. URL https://openaccess.thecvf.com/content_CVPR_2019/html/Singh_Towards_VQA_Models_That_Can_Read_CVPR_2019_paper.html.

Shengbang Tong, Ellis Brown, Penghao Wu, et al. Cambrian-1: A fully open, vision-centric exploration of multimodal llms. In *NeurIPS*, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/9ee3a664ccfeabc0da16ac6f1f1cfe59-Abstract-Conference.html.

Michael Tschannen, Alexey Gritsenko, Xiao Wang, et al. Siglip 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. *arXiv preprint arXiv:2502.14786*, 2025. URL https://arxiv.org/abs/2502.14786.

Pavan Kumar Anasosalu Vasu, James Gabriel, Jeff Zhu, Oncel Tuzel, and Anurag Ranjan. FastViT: A fast hybrid vision transformer using structural reparameterization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023. URL https://openaccess.thecvf.com/content/ICCV2023/html/Vasu_FastViT_A_Fast_Hybrid_Vision_Transformer_Using_Structural_Reparameterization_ICCV_2023_paper.html.

Pavan Kumar Anasosalu Vasu, Hadi Pouransari, Fartash Faghri, Raviteja Vemulapalli, and Oncel Tuzel. MobileCLIP: Fast image-text models through multi-modal reinforced training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024. URL https://openaccess.thecvf.com/content/CVPR2024/html/Vasu_MobileCLIP_Fast_Image-Text_Models_through_Multi-Modal_Reinforced_Training_CVPR_2024_paper.html.

Pavan Kumar Anasosalu Vasu, Fartash Faghri, Chun-Liang Li, Cem Koc, Nate True, Albert Antony, Gokul Santhanam, James Gabriel, Peter Grasch, Oncel Tuzel, and Hadi Pouransari. FastVLM: Efficient vision encoding for vision language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Vasu_FastVLM_Efficient_Vision_Encoding_for_Vision_Language_Models_CVPR_2025_paper.html.

Haoran Wei, Yaofeng Sun, and Yukun Li. DeepSeek-OCR: Contexts optical compression. *arXiv preprint arXiv:2510.18234*, 2025.

xAI. Grok-1.5 vision preview. RealWorldQA benchmark announcement, 2024. URL https://x.ai/news/grok-1.5v.

Yijia Xiao, Edward Sun, Tianyu Liu, and Wei Wang. LogicVista: Multimodal llm logical reasoning benchmark in visual contexts. *arXiv preprint arXiv:2407.04973*, 2024. URL https://arxiv.org/abs/2407.04973.

Sang Michael Xie, Hieu Pham, Xuanyi Dong, Nan Du, Hanxiao Liu, Yifeng Lu, Percy S Liang, Quoc V Le, Tengyu Ma, and Adams Wei Yu. DoReMi: Optimizing data mixtures speeds up language model pretraining. In *Advances in Neural Information Processing Systems*, volume 36, pp. 69798–69818, 2023. URL https://papers.neurips.cc/paper_files/paper/2023/hash/dcba6be91359358c2355cd920da3fcbd-Abstract-Conference.html.

Jiasheng Ye, Peiju Liu, Tianxiang Sun, Jun Zhan, Yunhua Zhou, and Xipeng Qiu. Data mixing laws: Optimizing data mixtures by predicting language modeling performance. In *International Conference on Learning Representations*, volume 2025, pp. 82263–82287, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/cc84bfabe6389d8883fc2071c848f62a-Abstract-Conference.html.

Xiang Yue, Tianyu Zheng, Yuansheng Ni, Yubo Wang, Kai Zhang, Shengbang Tong, Yuxuan Sun, Botao Yu, Ge Zhang, Huan Sun, Yu Su, Wenhu Chen, and Graham Neubig. MMMU-Pro: A more robust multi-discipline multimodal understanding benchmark. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics*, 2025. URL https://aclanthology.org/2025.acl-long.736/.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *ICCV*, 2023. URL https://openaccess.thecvf.com/content/ICCV2023/html/Zhai_Sigmoid_Loss_for_Language_Image_Pre-Training_ICCV_2023_paper.html.

Boqiang Zhang, Lei Ke, Ruihan Yang, Qi Gao, Tianyuan Qu, Rossell Chen, Dong Yu, and Leoweiliang. Penguin-vl: Exploring the efficiency limits of vlm with llm-based vision encoders. *arXiv preprint arXiv:2603.06569*, 2026. URL https://arxiv.org/abs/2603.06569.

Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ADE20K dataset. In *CVPR*, 2017. URL https://openaccess.thecvf.com/content_cvpr_2017/html/Zhou_Scene_Parsing_Through_CVPR_2017_paper.html.

Chenchen Zhu, Saksham Suri, Cijo Jose, Maxime Oquab, Marc Szafraniec, Wei Wen, Yunyang Xiong, Patrick Labatut, Piotr Bojanowski, Raghuraman Krishnamoorthi, and Vikas Chandra. Efficient universal perception encoder. *arXiv preprint arXiv:2603.22387*, 2026. URL https://arxiv.org/abs/2603.22387.

A HYPERPARAMETERS

This appendix lists the hyperparameters for the experiments in Section 4.

Encoder and connector. Our encoder has 164,206,336 parameters, width 1024, and SwiGLU intermediate width 2560. It uses the block sequence C,C,A,C,A,C,A,C,A,C,A,C,A,C, where C is a convolution block and A is a bidirectional full-attention block with 2D RoPE. The six attention blocks use 16 query heads and eight key/value heads of dimension 64; the eight convolution blocks use gated input/output projections and a depthwise 3×3 convolution. The connector has hidden width 4096; it has 25,174,016 parameters when mapping to the Qwen3.5-2B backbone and 20,978,688 parameters when mapping to LFM2.5-230M.

Stage 1 and Stage 2. Table 6 lists the main training hyperparameters. The Stage-1 schedule decays over its final 3700 updates to 0.1 times the peak learning rate, with a global budget of 12.288M unmerged patches per update. The image processor uses patch size 16, temporal patch size 2, spatial merge size 2, minimum image area 65,536 pixels, and maximum image area 16,777,216 pixels.

Table 6: Stage-1 and Stage-2 hyperparameters.

Setting	Stage 1	Stage 2
Updates	37,000	6,000
GPUs	32	32
Optimizer	per-head Muon	per-head Muon
Compute dtype	BF16	BF16
Optimizer masters	FP32	FP32
Peak learning rate	10^{-3}	5×10^{-5}
Warmup	370	63
Schedule	warmup–stable–decay	cosine to 0.08×
Weight decay	0.1	0.1
Checkpoint/eval interval	1,000	250
Objective	amplitude + direction	assistant-token CE
Language backbone in training	no	yes, trainable

LFM2.5-230M integration (Section 4.1). Each encoder receives a freshly initialized connector and is trained with per-head Muon on 16 GPUs, with 40,960 tokens per rank and two accumulation steps, for a global budget of 1,310,720 tokens per update. The connector-only phase uses 1000 updates, learning rate 10^{-4}, weight decay 10^{-6}, and 250 warmup updates. Full SFT uses 6000 updates, peak learning rate 5×10^{-5}, 63 warmup updates, and cosine decay to 0.08 times the peak.

Ablations (Sections 4.2 to 4.4). Each Stage-1 ablation run trains for 37k updates. Within an ablation, all runs share the initialization seed, data order, teacher, connector, and optimizer schedule. The architecture ablation uses the reconstruction-only objective. The objective ablation uses an internal VLM checkpoint as the teacher. The downstream curves in Figure 4 use the LFM2.5-230M integration schedule above.

Evaluation. All VLM benchmark evaluations use greedy decoding (temperature zero) with an 8192-token generation ceiling.