STATE-SPACE MODELS AS VISION ENCODERS FOR VLMS: A STUDY ACROSS TRAINING SETTINGS AND INTERFACES

Anonymous authors
Paper under double-blind review

ABSTRACT

Vision-Language Models (VLMs) typically leverage a vision encoder whose image features are mapped into a Large Language Model through a connector. While transformer-based vision encoders remain the standard choice, we ask whether state-space models (SSMs) can provide a viable alternative, given their competitive performance on vision tasks. To answer this question, we evaluate VMamba across four pre-trained setups (i.e., contrastive, classification, detection adaptation, and segmentation adaptation) and find that, under a shared downstream VLM recipe, the resulting models are generally competitive across VQA and referring-expression benchmarks, with the most consistent gains on grounding tasks. Moreover, we investigate the vision-language interface and find that *(i)* adding a normalization layer before the connector can substantially improve the VLM performance, and *(ii)* increasing the depth of an MLP connector produces inconsistent gains, while representation probes show that different connector depths alter the balance of local and global information in the visual tokens. Together, these results position VMamba as a competitive vision encoder alternative for VLMs, and highlight the importance of controlling the connector interface when comparing frozen vision encoders.

1 INTRODUCTION

Modern vision-language models (VLMs) pair a pretrained vision encoder with an autoregressive language model through a connector (Bai et al., 2025a; Wang et al., 2025a; Hong et al., 2025). While recent works have improved VLMs through stronger language backbones, larger training sets, and more instruction tuning (Wang et al., 2025a; Bai et al., 2025b), many VLMs still use Vision Transformer (ViT) (Dosovitskiy et al., 2021) encoders pretrained on image-text pairs (Tong et al., 2024b;a; Radford et al., 2021; Zhai et al., 2023). While ViTs remain the default backbone thanks to their robust performance, State Space Model (SSM)-based vision encoders, such as Vim and VMamba, have shown competitive results in image classification, object detection, and semantic segmentation (Zhu et al., 2024; Liu et al., 2024b). These results raise a natural question: *can SSM-based vision encoders provide similarly effective visual representations when used as frozen backbones in autoregressive VLMs?*

Prior works have shown that the vision encoder has a substantial effect on VLM performance (McKinzie et al., 2024; Tong et al., 2024a), making the visual backbone an important component of the overall architecture. However, existing VLM studies often compare transformer-based encoders while varying their pretraining objectives. As a result, we know relatively little about how alternative encoder architectures behave under a common downstream VLM setup, or whether their relative performance persists across different encoder-training settings and vision-language interfaces. To explore these questions, we compare ViT and VMamba under a fixed VLM recipe, with separate comparisons for each encoder-training dataset and objective, and repeat the comparisons to test whether the performance differences recur across settings.

However, comparing vision encoders is challenging due to the VLM's vast design space (Karamcheti et al., 2024; Laurençon et al., 2024; Tong et al., 2024a). Released encoder checkpoints differ in architecture, pretraining data, and objective (Liu et al., 2025), and connectors, language back-

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.

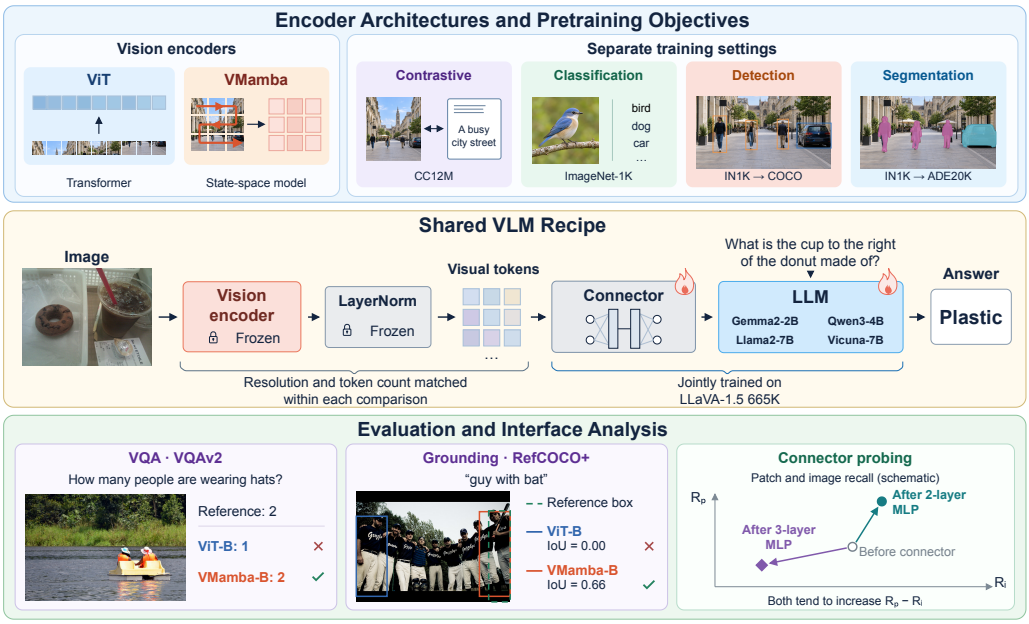


Figure 1: **Overview of our study.** We compare ViT and VMamba within four encoder-training settings using a shared downstream VLM recipe. The vision encoder and LayerNorm remain frozen, while the connector and language model are jointly trained. Selected CC12M examples illustrate VQA and referring-expression grounding evaluation, showing cases where ViT-B is incorrect and VMamba-B is correct. The connector schematic illustrates typical changes in patch-level recall R_p and image-level recall R_i from the connector input to the outputs of two- and three-layer MLPs. Point positions are illustrative, not measured values.

bones, and instruction-tuning stages are added to the VLM design space. These extensive choices make it difficult to determine which changes account for their performance. While large-scale VLM studies have examined released vision encoders, connectors, language backbone, and training data choices (Laurençon et al., 2024; Karamcheti et al., 2024; McKinzie et al., 2024; Tong et al., 2024a), and even isolated the encoder pretraining objectives to assess their effects on VLM performance (Liu et al., 2025; Fan et al., 2025), these studies mostly focus on ViTs. We know less about SSM-based vision encoders in autoregressive VLMs.

In this paper, we study this question using VMamba, with standard ViT as the main comparison and Swin Transformers (Liu et al., 2021) as an additional Transformer baseline. We first examine language-backbone and connector choices on CC12M-pretrained ViT and VMamba encoders, selecting Vicuna-7B v1.5 (Chiang et al., 2023) and a two-layer MLP as a strong common VLM recipe. Holding the VLM recipe fixed, we compare ViT and VMamba separately in four encoder-training settings: CC12M contrastive pretraining, ImageNet-1K classification pretraining, COCO detection adaptation, and ADE20K segmentation adaptation. Across our 13 VQA-oriented and referring-expression grounding scores, VMamba generally achieves a higher overall average than ViT. The largest gains are in grounding, while VQA differences are smaller. Finally, we use local and global probes adapted from Covert et al. (2025) to compare visual tokens before and after the connector.

Our contributions are summarized as follows: 1. We compare VMamba with ViT-based encoders separately within four encoder-training settings using a shared downstream VLM recipe, with ViT as the main baseline, and Swin Transformer as an additional hierarchical Transformer comparison. 2. Across four encoder-training settings under a shared VLM recipe, VMamba generally achieves a higher overall average than standard ViT across our evaluation suite. The largest gains are in grounding. 3. We show that LayerNorm can substantially improve VLM performance with frozen encoders and examine how connector choice affects the architecture comparison. Further probing reveals distinct changes in patch- and image-level recall between the two connector configurations. 4. We will release our code, trained VLM checkpoints, adapted vision-encoder checkpoints, and evaluation configurations to facilitate reproducibility and future work on vision encoders for VLMs.

2 Related Works

Vision encoders and VLM design. Prior work studies VLM design choices such as preprocessing, encoder pretraining, resolution, token count, connector design, and combining visual encoders (Karamcheti et al., 2024; McKinzie et al., 2024; Tong et al., 2024a), or evaluates encoders trained for contrastive learning, self-supervised learning, detection, segmentation, and OCR, and then studies how to select, align, and combine them (Shi et al., 2025). While these studies provide a basis for fixing the surrounding VLM recipe when comparing encoders, they do not include SSM-based vision encoders. Their encoder comparisons also largely use released checkpoints whose architectures and pretraining recipes vary together; this design examines whether architecture-related performance patterns are shared across training settings. We compare ViT and VMamba across four encoder-training settings, matching the dataset and objective within each setting and keeping the downstream VLM recipe fixed. **Alternative vision architectures.** Several works explore alternatives to plain ViT for autoregressive VLMs. ViTamin compares convolutional, transformer, and hybrid encoders under CLIP training, then evaluates its proposed hybrid encoder in LLaVA-1.5 (Chen et al., 2024). ConvLLaVA uses a hierarchical ConvNeXt encoder and an additional downsampling stage to process high-resolution images with fewer visual tokens (Ge et al., 2024). FastVLM develops a hybrid encoder and studies the trade-off among resolution, visual token count, accuracy, and time to first token (Vasu et al., 2025). CLIMP pairs VMamba with a Mamba text encoder for contrastive image–text learning and evaluates classification and retrieval (Shabtay et al., 2026). MambaXray-VL pretrains a Vim-based encoder on X-ray images and image–report pairs, while R2GenCSR uses VMamba with retrieved context images to guide a language model (Wang et al., 2025b; 2026). While these works establish encoder architecture as an important design choice, we evaluate VMamba as a vision encoder for general-purpose autoregressive VLMs.

Encoder pretraining and VLM transfer. Prior works have also trained CLIP and DINO with the same ViT architecture and image data, then evaluate them in VLMs (Liu et al., 2025). Fan et al. (2025) scale self-supervised and language-supervised ViTs on the same web images, showing that sufficiently scaled visual self-supervision can match the performance of language supervision on VQA. These findings motivate accounting for encoder pretraining. We compare architectures separately within each encoder-training setting and examine whether their performance differences recur across settings. **Connectors and visual representations.** Moreover, other works have introduced convolutional and deformable-attention projectors that preserve local context while controlling the number of visual tokens (Cha et al., 2024), analyze changes across the connector through embedding neighborhoods and feature reconstruction (Li et al., 2025), or fine-tune vision encoders with masked embedding reconstruction and use patch- and image-level probes to evaluate the resulting features to show that the adapter used with these features affects VLM performance (Covert et al., 2025). We examine how normalization and connector design affect the comparison between vision encoders. Using the linear-probe variant of Covert et al. (2025), we further observe distinct changes in patch- and image-level recall between MLP configurations, despite relatively small differences in overall VLM performance.

3 Preliminaries

VLM setup. We use an autoregressive VLM with a vision encoder, a connector, and a decoder-only language model (Karamcheti et al., 2024; Liu et al., 2024a). The encoder produces visual tokens, and the connector maps them to the language model's input space. For VMamba and Swin, we extract features from stage 3 to match ViT's spatial token grid, giving 196 tokens at 224×224 and 1,024 tokens at 512×512. Appendix C compares stage-3 and stage-4 features from the CC12M-pretrained VMamba-B. Unless otherwise stated, we apply LayerNorm (Ba et al., 2016) before the connector, using the pretrained output norm where available and a frozen parameter-free LayerNorm otherwise. The model generates an answer conditioned on the image tokens and text prompt.

Training. The connector is randomly initialized. We jointly train the connector and pretrained language model in one instruction-tuning stage, following Karamcheti et al. (2024). The vision encoder and pre-connector LayerNorm remain frozen throughout VLM training. Within each encoder-training setting, we use the same downstream training recipe and optimize next-token loss on answer tokens. We report one instruction-tuning run per configuration, using a fixed random seed.

Training data and preprocessing. We train on the 665K-example LLaVA-v1.5 instruction-tuning mixture (Liu et al., 2024a). We letterbox each image to the encoder's input size, preserving its aspect ratio, and normalize it using the encoder's preprocessing statistics. Encoder configurations and training details are provided in Appendix A.

Evaluation suite. We report eight VQA-oriented scores: VQA-v2 (Goyal et al., 2017), GQA (Hudson & Manning, 2019), VizWiz (Gurari et al., 2018), TextVQA with and without supplied OCR tokens (Singh et al., 2019), POPE (Li et al., 2023), TallyQA (Acharya et al., 2019), and CV-Bench 2D (Tong et al., 2024a). We report five referring-expression grounding scores: RefCOCO and RefCOCO+ (Kazemzadeh et al., 2014; Yu et al., 2016), RefCOCOg (Mao et al., 2016), OCID-Ref (Wang et al., 2021), and Flickr30k Entities (Plummer et al., 2015). Following Prismatic (Karamcheti et al., 2024), we report grounding accuracy at $\text{IoU} \geq 0.5$ for RefCOCO/+/g and $\text{IoU} \geq 0.25$ for OCID-Ref. For Flickr30k Entities, we use $\text{IoU} \geq 0.5$. Evaluation protocols are detailed in Appendix B. The VQA and grounding averages are the unweighted means of their eight and five scores. The overall average is the unweighted mean of all 13 scores.

4 Contrastive Encoders and a Shared VLM Recipe

In this section, we compare contrastively pretrained ViT and VMamba encoders. We first select the encoder pair in Section 4.1, then use this pair to select a shared VLM recipe in Section 4.2. Finally, we compare their VLM performance under the selected recipe in Section 4.3.

4.1 Selecting the Vision Encoders

We use the CLIP ViT-B/16 checkpoint released by Mu et al. (2022) and the VMamba-B checkpoint from CLIMP (Shabtay et al., 2026). Both are contrastively pretrained on CC12M (Changpinyo et al., 2021) at 224×224, and produce 196 visual tokens. These released checkpoints differ in their pretraining histories; VMamba is initialized from ImageNet-1K, whereas ViT is trained from scratch, and their training schedules and contrastive text encoders also differ. These differences limit attribution of the contrastive result to architecture alone. Nevertheless, the pair provides a useful practical comparison, since both use the same contrastive pretraining dataset and are matched in input resolution and visual-token count.

4.2 Selecting the VLM Recipe

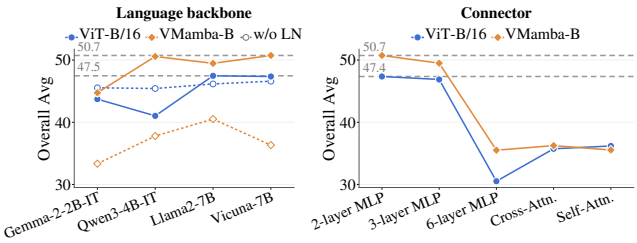


Figure 2: **Selecting a shared VLM recipe with CC12M-pretrained ViT-B/16 and VMamba-B.** Scores are the unweighted mean of the 13 benchmark scores. **Left:** language-backbone comparisons with a two-layer MLP connector. Solid lines use output LayerNorm; dashed lines omit it. **Right:** connector comparisons with Vicuna-7B and output LayerNorm. Per-benchmark results for the left and right panels are reported in Tables 6 and 9, respectively.

With the encoder pair set, we still need to decide the rest of the VLM recipe (i.e., the connector and language backbone to use).

Language backbone and LayerNorm: In Figure 2 (left), we first fix the connector to a 2-layer MLP, a default choice in Prismatic Karamcheti et al. (2024), and compare four language backbones (Chiang et al., 2023; Touvron et al., 2023; Yang et al., 2025; Team et al., 2024) with and without LayerNorm. Vicuna-7B with LayerNorm provides a strong common baseline for both encoders. Recent encoder studies also use Vicuna-7B for VLM evaluation (Shi et al., 2025; Yang et al., 2026). ***Connector:*** In Figure 2 (right), we fix Vicuna-7B and LayerNorm and compare different connectors. The MLPs differ in both depth and hidden width; their dimensions are given in Appendix A. These are the two strongest connector configurations under our recipe, with the two-layer MLP slightly better. We therefore use Vicuna-7B, a 2-layer MLP, and LayerNorm before the connector for the remaining encoder comparisons. We further examine LayerNorm and connector choices across all four encoder-training settings in Section 6.1.

Table 1: **Results on contrastive vision encoders.** We evaluate VLMs under matched downstream training (i.e., 224×224 input, 196 visual tokens, 2-layer MLP, Vicuna-7B). OpenAI CLIP ViT-B/16 pretrained on WIT-400M serves as a reference baseline. We show CLIP ViT-B pretrained on CC3M and CC12M, and compare against CLIMP VMamba-B pretrained on CC12M.

				VQA										Grounding					
Visual Encoder	Dataset	Dataset Size	Overall Avg	VQA-v2	GQA	VizWiz	TextVQA	TextVQA +OCR	POPE	TallyQA	CV-Bench^{2D}	Avg	RefCOCO	RefCOCO+	RefCOCOg	OCID-Ref	Flickr30k	Avg
ViT-B	WIT	400M	55.7	72.2	58.4	52.9	32.0	51.0	85.0	60.6	55.7	58.5	64.8	55.6	60.0	28.4	47.2	51.2
ViT-B	CC3M	3.1M	38.4	58.6	48.3	44.9	11.0	43.4	72.5	51.4	45.4	46.9	37.1	25.7	28.6	9.7	23.0	24.8
ViT-B	CC12M	11.0M	47.4	64.9	53.2	44.2	12.3	44.8	80.5	**56.2**	53.1	51.1	55.6	45.2	48.2	20.2	37.4	41.3
VMamba-B	CC12M	11.0M	**50.7**	**66.8**	**55.3**	**46.6**	**12.5**	**44.9**	**82.5**	55.6	**53.3**	**52.2**	**62.8**	**51.7**	**57.2**	**26.2**	**44.2**	**48.4**

Table 2: **VLM performance with ImageNet-1K classification-pretrained encoders.** We show results under the shared VLM recipe, using 224×224 inputs and 196 visual tokens. IN1K top-1 reports the accuracy of the original full classifiers.

				VQA										Grounding					
Visual Encoder	Full Params	IN1K top-1	Overall Avg	VQA-v2	GQA	VizWiz	TextVQA	TextVQA +OCR	POPE	TallyQA	CV-Bench^{2D}	Avg	RefCOCO	RefCOCO+	RefCOCOg	OCID-Ref	Flickr30k	Avg
Swin-T	28M	81.2	44.8	63.2	51.9	48.8	11.7	44.0	78.4	54.2	46.5	49.8	49.0	37.8	43.8	20.4	32.0	36.6
Swin-S	49M	83.2	41.5	63.7	52.6	41.9	**12.4**	44.6	79.9	54.0	**52.0**	50.1	38.7	29.2	34.2	11.6	25.4	27.8
Swin-B	87M	83.5	37.6	61.5	50.8	**51.2**	12.3	43.5	79.9	47.1	41.6	48.5	31.0	21.0	24.8	5.6	18.6	20.2
ViT-S/16	22M	78.8	43.0	62.3	52.3	46.8	12.2	44.4	78.2	51.8	44.7	49.1	46.6	35.8	39.8	13.2	30.7	33.2
ViT-B/16	86M	81.1	43.5	62.6	51.9	46.2	12.2	**44.9**	79.5	52.8	48.8	49.8	46.3	34.7	40.9	13.6	30.6	33.2
VMamba-T	29M	82.6	**47.8**	65.0	54.0	45.0	12.2	44.6	81.7	54.6	50.4	50.9	**58.2**	**46.6**	**51.7**	20.2	**37.8**	**42.9**
VMamba-S	49M	83.6	47.6	**65.2**	**54.1**	45.0	12.2	44.1	**82.2**	**55.5**	50.0	**51.0**	56.5	44.3	49.9	**23.1**	36.8	42.1
VMamba-B	88M	83.9	42.6	64.2	53.2	48.0	12.1	43.8	80.8	54.0	47.4	50.4	42.1	31.4	36.2	15.2	25.2	30.0

4.3 CC12M Encoder Comparison

Table 1 reports both encoders under the selected recipe. VMamba-B scores 50.7 overall, compared with 47.4 for ViT-B/16. Its grounding average is 48.4 versus 41.3, while its VQA-oriented average is 52.2 versus 51.1. VMamba-B leads on all five grounding benchmarks and seven of the eight VQA-oriented scores; ViT-B/16 is slightly higher on TallyQA. The overall advantage therefore spans the evaluation suite, with the largest difference in grounding. To test whether this performance pattern recurs beyond the selected contrastive checkpoint pair, we extend the comparison to classification pretraining and detection and segmentation adaptation in Section 5. We keep the downstream VLM recipe fixed and compare encoders separately within each setting.

5 Comparisons under Other Encoder-Training Settings

In this section, we study encoders trained with other settings. In each setting, we use the same Vicuna-7B, output LayerNorm, and two-layer MLP recipe selected in Section 4. We first compare classification-pretrained encoders in Section 5.1. Then, we compare detection- and segmentation-adapted encoders and examine whether the ViT–VMamba performance pattern persists across input resolutions in Section 5.2.

5.1 ImageNet-1K Classification

We evaluate available ImageNet-1K (Deng et al., 2009) classification checkpoints for ViT (including ViT-S/B), and VMamba (including VMamba-T/S/B). We additionally include Swin (Liu et al., 2021) as another transformer encoder. All use 224×224 inputs and 196 visual tokens. ***VLM performance:*** Table 2 shows that VMamba-T and VMamba-S have the two highest VQA, grounding, and overall averages among all evaluated encoders. Their advantage is largest in grounding. Among

Table 3: **VLM performance after COCO detection adaptation using Mask R-CNN.** All encoders use the shared VLM recipe with 512×512 inputs and 1,024 visual tokens.

					VQA									Grounding					
Visual Encoder	Full Params	Pretraining	Input	Overall Avg	VQA-v2	GQA	VizWiz	TextVQA	TextVQA +OCR	POPE	TallyQA	CV-Bench^{2D}	Avg	RefCOCO	RefCOCO+	RefCOCOg	OCID-Ref	Flickr30k	Avg
Swin-T	28M	IN1K → COCO	512x512	52.9	67.0	54.9	48.2	12.0	**44.9**	84.0	57.6	53.6	52.8	70.3	58.4	64.7	28.1	43.8	53.1
Swin-S	49M			54.7	**67.7**	**55.3**	48.1	12.0	44.6	88.0	59.1	**55.6**	**53.8**	75.2	61.5	69.8	29.6	44.6	56.2
Swin-B	87M			51.9	66.4	54.6	43.6	12.1	44.5	**88.7**	58.4	54.0	52.8	68.9	53.6	62.8	27.1	40.1	50.5
ViT-S/16	22M			52.3	64.7	53.3	47.2	11.5	44.2	84.7	57.6	51.3	51.8	71.7	58.2	65.1	26.6	43.9	53.1
ViT-B/16	86M			52.9	65.8	54.2	45.4	11.7	44.4	84.6	56.3	52.2	51.8	73.4	60.5	67.5	27.6	43.6	54.5
VMamba-T	29M			54.7	67.2	55.0	46.9	11.7	44.0	86.8	**59.4**	54.5	53.2	76.2	**62.9**	69.5	31.8	45.9	57.3
VMamba-S	49M			**55.8**	67.0	55.1	48.6	**12.3**	44.7	88.2	58.8	55.5	**53.8**	**78.0**	62.5	**72.2**	**35.1**	**47.0**	**59.0**
VMamba-B	88M			55.3	66.3	55.1	**49.1**	12.1	44.4	88.6	**59.4**	54.9	53.7	76.8	60.9	71.6	34.2	45.4	57.8

Table 4: **VLM performance after ADE20K segmentation adaptation using UPerNet.** All encoders use the shared VLM recipe with 512×512 inputs and 1,024 visual tokens.

					VQA									Grounding					
Visual Encoder	Full Params	Pretraining	Input	Overall Avg	VQA-v2	GQA	VizWiz	TextVQA	TextVQA +OCR	POPE	TallyQA	CV-Bench^{2D}	Avg	RefCOCO	RefCOCO+	RefCOCOg	OCID-Ref	Flickr30k	Avg
Swin-T	28M	IN1K → ADE20K	512x512	46.4	64.7	54.3	45.0	11.7	44.0	81.8	54.2	**50.7**	50.8	53.3	41.2	46.6	17.9	37.1	39.2
Swin-S	49M			49.0	65.8	55.6	45.9	12.1	44.2	83.5	52.2	48.8	51.0	59.3	48.3	54.8	24.5	41.3	45.6
Swin-B	87M			47.8	65.6	55.7	44.4	**12.4**	**44.5**	**84.4**	55.6	48.3	**51.4**	54.7	43.6	50.8	21.5	40.0	42.1
ViT-S/16	22M			47.0	62.8	53.3	48.2	11.6	44.0	80.4	53.0	47.4	50.1	57.3	45.6	50.6	18.4	38.9	42.2
ViT-B/16	86M			48.2	64.2	53.7	**48.3**	12.1	43.9	80.8	53.0	46.5	50.3	59.9	48.4	53.8	20.9	41.2	44.8
VMamba-T	29M			49.6	65.4	55.3	40.9	11.8	44.2	83.6	**55.7**	49.4	50.8	62.6	51.1	57.2	24.0	**43.4**	47.7
VMamba-S	49M			**50.4**	**66.4**	55.7	47.4	11.9	44.4	84.0	53.9	46.6	51.3	**64.2**	**54.0**	**59.1**	24.6	43.2	**49.0**
VMamba-B	88M			49.7	66.1	**55.9**	40.2	12.3	**44.5**	**84.4**	53.6	46.5	50.4	64.0	52.5	58.7	**25.9**	41.8	48.6

transformer encoders, Swin-T has the highest grounding and overall averages. ViT-B is also competitive overall and outperforms VMamba-B. ***ImageNet accuracy and VLM performance:*** Within each architecture family, ImageNet-1K top-1 accuracy increases with model size. However, overall VLM performance does not always improve. VMamba-B is the most obvious example: it has the highest top-1 accuracy in its family, but its VLM performance is much lower than that of VMamba-T and VMamba-S. We can observe similar results in Swin-S, and Swin-B. ImageNet top-1 accuracy also does not consistently predict VLM rankings across architecture families. For example, ViT-B has lower ImageNet accuracy than Swin-B but achieves a higher VLM overall average.

5.2 Dense-Prediction Adaptation

We adapt ImageNet-1K-pretrained ViT, VMamba, and Swin encoders to detection and segmentation using the same training code within each task. We include Swin as a hierarchical transformer baseline following Swin-T's competitive VLM performance in Section 5.1. For detection, we use a shared Mask R-CNN $1\times$ recipe (He et al., 2017) at 512×512 with the COCO dataset (Lin et al., 2014). For segmentation, we use UPerNet-adapted encoders (Xiao et al., 2018) at 512×512 on the ADE20K dataset (Zhou et al., 2017). All encoders provide 1,024 visual tokens and use the shared downstream VLM recipe.

COCO detection: Table 3 shows that the VMamba family outperforms the Swin and ViT families in overall and grounding performance. On the VQA suite, VMamba and Swin Transformers achieve comparable performance, and both outperform the ViT family.

ADE20K segmentation: Table 4 shows a trend similar to that of COCO detection-pretrained encoders. The VMamba family consistently outperforms the Swin and ViT families in overall and grounding performance. For the VQA suite, the margins are smaller than in grounding, with VMamba and Swin slightly better than ViT.

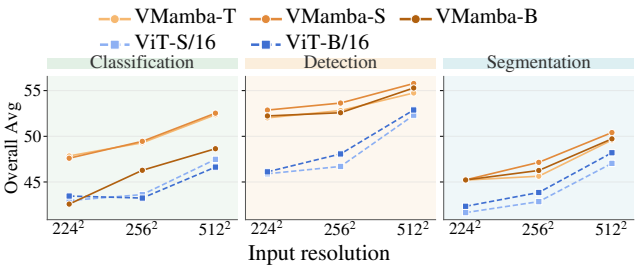


Figure 3: **Overall average across 13 benchmarks for ViT and VMamba.** We show results at 224×224, 256×256, and 512×512, shown separately by encoder-training setting. Detection and segmentation encoders are adapted at each plotted resolution; classification encoders receive additional ImageNet-1K fine-tuning at 256 and 512. Each point uses a separately instruction-tuned VLM. Per-benchmark results are reported in Table 7.

Our detection and segmentation comparisons use 512×512 inputs. To check whether this choice affects the relationship between ViT–VMamba, we repeat the comparisons at 224×224 and 256×256. Detection and segmentation encoders are adapted separately at each resolution. For classification, the 256- and 512-pixel checkpoints receive additional fine-tuning on ImageNet-1 K. We instruction-tune a separate VLM at each resolution using the same recipe. Figure 3 shows that overall performance generally increases across the tested resolution-specific training configurations, with a small decrease for classification-pretrained ViT-B from 224 to 256. VMamba leads ViT throughout the detection and segmentation comparisons. In classification, VMamba-B underperforms ViT-B at 224 but leads at 256 and 512.

6 Vision-Language Interface Analysis

In section 4.2, we show that the choice of LayerNorm and connector can affect VLM performance, even when the encoder checkpoint is fixed. We extend this analysis to all four encoder-training settings to revisit our shared recipe and examine whether the ViT–VMamba comparison persists under different interface choices. We then probe the visual tokens to examine the corresponding changes in their representations.

6.1 Revisiting the Shared Recipe

We compare three configurations for each encoder: a two-layer MLP without LayerNorm, a two-layer MLP with LayerNorm, and a three-layer MLP with LayerNorm. The encoder remains frozen, and each configuration uses the same instruction-tuning recipe.

The effect of LayerNorm and Connector. Figure 4 shows two main patterns. First, LayerNorm improves the overall average in all 17 comparisons, with large gains for several encoders. These gains show that the downstream setup can obtain substantially better results from the same frozen encoder. Normalized features may make the connector and language model easier to optimize during instruction tuning. Normalizing encoder outputs also appears in existing VLMs: Pheye introduces LayerNorm with convergence as one motivation (Carvalho & Martins, 2025), while PaliGemma retains SigLIP's output LayerNorm before its linear connector (Beyer et al., 2024). Second, with LayerNorm retained, changing the connector from a 2-layer MLP to a 3-layer MLP has a smaller effect: the overall average changes by at most 2.32 points and improves 12 of 17 comparisons. The three-layer configuration often scores slightly higher, although it is not uniformly better.

Comparison between VMamba and ViT. With LayerNorm, VMamba outperforms ViT across contrastive, detection, and segmentation settings for both MLP configurations. In classification, VMamba-T also leads with both configurations, while the ViT-B–VMamba-B ordering changes: ViT-B leads with the two-layer MLP, and VMamba-B leads with the three-layer MLP.

6.2 Probing the Visual Tokens

With the encoder frozen during visual instruction tuning, we examine how the trained connector transforms visual features for the language model. To do so, we adapt the linear-probe variant of Covert et al. (2025), who use probing to evaluate how their locality alignment method changes encoder features. We fit separate linear probes to the tokens before and after the connector. The patch-level probe predicts which COCO object categories occur in each patch. The image-level

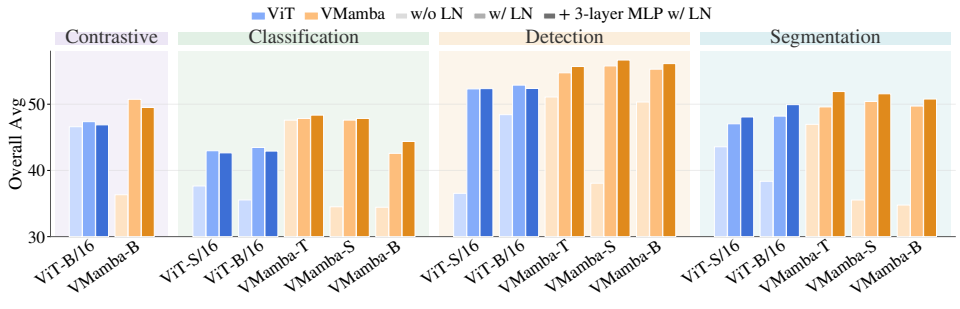


Figure 4: **Overall average for a two-layer MLP without LayerNorm, a two-layer MLP with LayerNorm, and a three-layer MLP with LayerNorm.** Comparisons are made within each frozen encoder and within each encoder-training setting. Per-benchmark results are reported in Table 8.

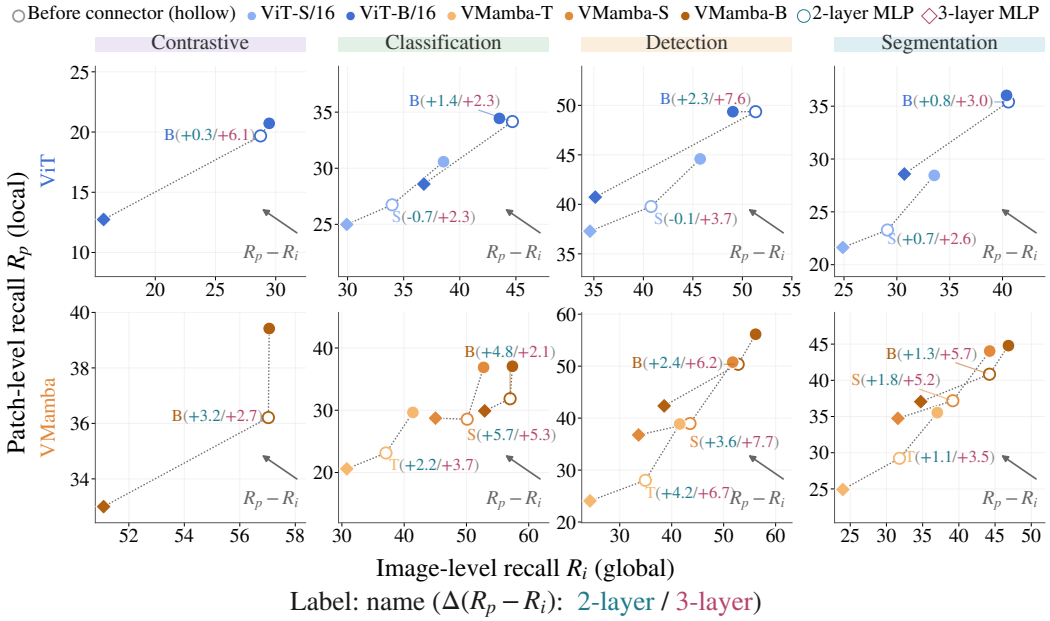


Figure 5: **Image-level recall R_i and patch-level recall R_p before and after the connector, with LayerNorm in place.** Hollow circles denote pre-connector features; filled circles and diamonds denote outputs of the two- and three-layer MLPs, respectively. Colors identify the encoders. Each annotation reports the change in $R_p - R_i$ for the two-layer and three-layer connectors, respectively. Gray diagonal lines mark a constant gap through each pre-connector point.

probe predicts which categories occur anywhere in the image using mean-pooled tokens. We denote their macro recalls by R_p and R_i, respectively. Probing details are provided in Appendix D.

Probing results and observations. Figure 5 shows two distinct patterns: the two-layer MLP usually raises both recalls, with a larger increase in patch-level recall; and the three-layer MLP usually lowers both recalls, with a larger decrease in image-level recall. Both therefore tend to increase $R_p - R_i$, despite their different effects on the absolute scores. In the detection and segmentation settings at 512×512, the three-layer MLP tends to produce a larger $R_p - R_i$ than the two-layer MLP and usually gives a small VLM performance gain (Figure 4). These probe changes could guide encoder pretraining and connector design by revealing how different interfaces reshape local and global visual information.

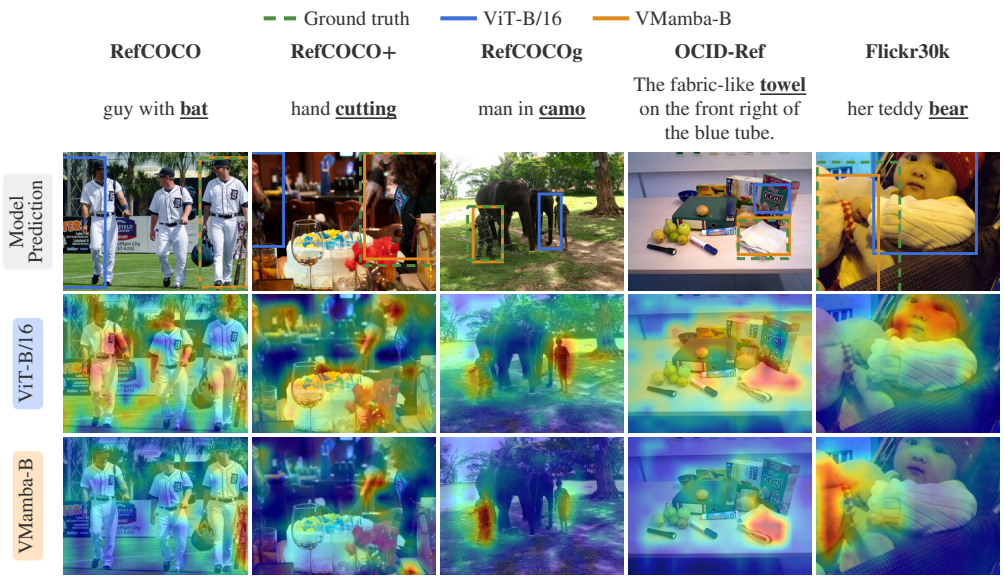


Figure 6: **Grounding examples from the CC12M ViT-B/16 and VMamba-B VLMs.** Top: reference boxes (green dashed) and predictions (ViT: blue; VMamba: orange). Below: token–region similarity maps for the underlined words; warmer colors indicate higher similarity.

7 Qualitative Results and Profiling

Qualitative examples. Figure 6 shows examples from the five grounding benchmarks. In these selected examples, VMamba-B's boxes more closely match the reference regions, and its similarity maps are more concentrated around regions relevant to the referring expression.

High-resolution applications. We further examine the efficiency of VMamba, whose complexity grows linearly with the number of visual tokens (Liu et al., 2024b). At low resolutions, the larger language model can dominate inference cost. However, applications such as whole-slide pathology and satellite image analysis may require much higher resolutions (Ahmed et al., 2024; Soni et al., 2025), making encoder efficiency more important. We measure the median encoder latency after warm-up across input resolutions on a single H200 GPU, using a batch size of 1 and BF16 precision. Figure 7 shows that VMamba has lower latency than ViT at high resolutions, suggesting potential benefits for applications with large visual inputs.

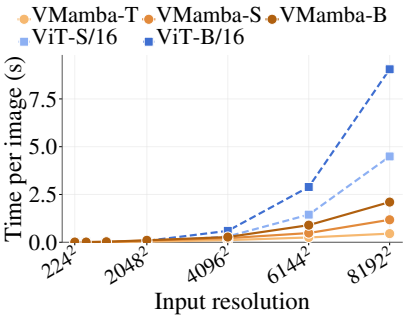


Figure 7: **Encoder latency as input resolution increases.** Lower is better.

8 Conclusion

We explored VMamba as a vision encoder for autoregressive VLMs, comparing it with ViT-based encoders under a shared downstream VLM recipe across four encoder-training settings. VMamba generally achieves higher overall performance, with the largest gains in grounding and smaller gains on VQA benchmarks. This pattern recurs across the tested resolutions and connector configurations. We also show that ImageNet classification accuracy alone does not reliably predict these VLM results. Our interface analysis shows that LayerNorm can substantially improve performance, whereas changes to the connector configurations have a smaller effect. Further probing reveals that two- and three-layer connectors affect patch- and image-level recall in different ways. These findings motivate further study of SSM-based vision encoders and their interfaces with language models. Whether the observed advantages persist at web-scale pretraining remains an open question.

AI USE STATEMENT

In this work, we used generative AI to aid in or polish writing and to check and edit software code. We have not used generative AI tools to propose or refine hypotheses, design or provide feedback on research methodology or experiments, implement methods, support qualitative and thematic data analysis, or interpret results. We have reviewed all AI-assisted work and take responsibility for the final content, including text, claims, and artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We provide the information needed to reproduce our experiments throughout the paper and appendices. The main text specifies the downstream VLM training procedure, preprocessing, datasets, and evaluation suite through Sections 3 to 6. Appendix A reports encoder configurations and training details, Appendix B describes the evaluation protocols, Appendix C provides additional analysis of the extracted encoder features, and Appendix D details the representation-probing setup. We will release our source code, trained VLM checkpoints, adapted vision-encoder checkpoints, and evaluation configurations upon acceptance.

REFERENCES

Manoj Acharya, Kushal Kafle, and Christopher Kanan. Tallyqa: Answering complex counting questions. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pp. 8076–8084, 2019.

Faruk Ahmed, Andrew Sellergren, Lin Yang, Shawn Xu, Boris Babenko, Abbi Ward, Niels Olson, Arash Mohtashamian, Yossi Matias, Greg S Corrado, et al. Pathalign: A vision-language model for whole slide images in histopathology. *arXiv preprint arXiv:2406.19578*, 2024.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, Mei Li, Kaixin Li, Zicheng Lin, Junyang Lin, Xuejing Liu, Jiawei Liu, Chenglong Liu, Yang Liu, Dayiheng Liu, Shixuan Liu, Dunjie Lu, Ruilin Luo, Chenxu Lv, Rui Men, Lingchen Meng, Xuancheng Ren, Xingzhang Ren, Sibo Song, Yuchong Sun, Jun Tang, Jianhong Tu, Jianqiang Wan, Peng Wang, Pengfei Wang, Qiuyue Wang, Yuxuan Wang, Tianbao Xie, Yiheng Xu, Haiyang Xu, Jin Xu, Zhibo Yang, Mingkun Yang, Jianxin Yang, An Yang, Bowen Yu, Fei Zhang, Hang Zhang, Xi Zhang, Bo Zheng, Humen Zhong, Jingren Zhou, Fan Zhou, Jing Zhou, Yuanzhi Zhu, and Ke Zhu. Qwen3-vl technical report, 2025a. URL https://arxiv.org/abs/2511.21631.

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

Lucas Beyer, Andreas Steiner, André Susano Pinto, Alexander Kolesnikov, Xiao Wang, Daniel Salz, Maxim Neumann, Ibrahim Alabdulmohsin, Michael Tschannen, Emanuele Bugliarello, et al. Paligemma: A versatile 3b vlm for transfer. *arXiv preprint arXiv:2407.07726*, 2024.

Miguel Carvalho and Bruno Martins. Efficient architectures for high resolution vision-language models. In *Proceedings of the 31st International Conference on Computational Linguistics*, pp. 10520–10530, 2025.

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

Soravit Changpinyo, Piyush Sharma, Nan Ding, and Radu Soricut. Conceptual 12m: Pushing web-scale image-text pre-training to recognize long-tail visual concepts. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3557–3567. IEEE, 2021.

Jieneng Chen, Qihang Yu, Xiaohui Shen, Alan Yuille, and Liang-Chieh Chen. Vitamin: Designing scalable vision models in the vision-language era. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12954–12966. IEEE, 2024.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality, March 2023. URL https://lmsys.org/blog/2023-03-30-vicuna/.

Ian Covert, Tony Sun, James Y Zou, and Tatsunori Hashimoto. Locality alignment improves vision-language models. In *International Conference on Learning Representations*, volume 2025, pp. 83127–83165, 2025.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009. doi: 10.1109/CVPR.2009.5206848.

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://openreview.net/forum?id=YicbFdNTTy.

David Fan, Shengbang Tong, Jiachen Zhu, Koustuv Sinha, Zhuang Liu, Xinlei Chen, Michael Rabbat, Nicolas Ballas, Yann LeCun, Amir Bar, et al. Scaling language-free visual representation learning. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1–13. IEEE, 2025.

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.

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6904–6913, 2017.

Danna Gurari, Qing Li, Abigale J. Stangl, Anhong Guo, Chi Lin, Kristen Grauman, Jiebo Luo, and Jeffrey P. Bigham. Vizwiz grand challenge: Answering visual questions from blind people. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

Kaiming He, Georgia Gkioxari, Piotr Dollar, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

Wenyi Hong, Wenmeng Yu, Xiaotao Gu, Guo Wang, Guobing Gan, Haomiao Tang, Jiale Cheng, Ji Qi, Junhui Ji, Lihang Pan, et al. Glm-4.5 v and glm-4.1 v-thinking: Towards versatile multimodal reasoning with scalable reinforcement learning. *arXiv preprint arXiv:2507.01006*, 2025.

Drew A. Hudson and Christopher D. Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

Siddharth Karamcheti, Suraj Nair, Ashwin Balakrishna, Percy Liang, Thomas Kollar, and Dorsa Sadigh. Prismatic vlms: Investigating the design space of visually-conditioned language models. In *Forty-first International Conference on Machine Learning*, 2024.

Sahar Kazemzadeh, Vicente Ordonez, Mark Matten, and Tamara Berg. ReferItGame: Referring to objects in photographs of natural scenes. In Alessandro Moschitti, Bo Pang, and Walter Daelemans (eds.), *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 787–798, Doha, Qatar, October 2014. Association for Computational Linguistics. doi: 10.3115/v1/D14-1086. URL https://aclanthology.org/D14-1086/.

Hugo Laurençon, Léo Tronchon, Matthieu Cord, and Victor Sanh. What matters when building vision-language models? *Advances in Neural Information Processing Systems*, 37:87874–87907, 2024.

Wenyan Li, Raphael Tang, Chengzu Li, Caiqi Zhang, Ivan Vulic, and Anders Søgaard. Lost in embeddings: Information loss in vision-language models. In *EMNLP (Findings)*, pp. 22676–22693, 2025.

Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Xin Zhao, and Ji-Rong Wen. Evaluating object hallucination in large vision-language models. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 292–305, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.20. URL https://aclanthology.org/2023.emnlp-main.20/.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pp. 740–755. Springer, 2014.

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, June 2024a.

Yiming Liu, Yuhui Zhang, Dhruba Ghosh, Ludwig Schmidt, and Serena Yeung-Levy. Data or language supervision: What makes clip better than dino? *arXiv preprint arXiv:2510.11835*, 2025.

Yue Liu, Yunjie Tian, Yuzhong Zhao, Hongtian Yu, Lingxi Xie, Yaowei Wang, Qixiang Ye, Jianbin Jiao, and Yunfan Liu. Vmamba: Visual state space model. *Advances in neural information processing systems*, 37:103031–103063, 2024b.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *2021 IEEE/CVF international conference on computer vision (ICCV)*, pp. 9992–10002. Ieee, 2021.

Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, Alan L Yuille, and Kevin Murphy. Generation and comprehension of unambiguous object descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 11–20, 2016.

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*, pp. 304–323. Springer, 2024.

Norman Mu, Alexander Kirillov, David Wagner, and Saining Xie. Slip: Self-supervision meets language-image pre-training. In *European conference on computer vision*, pp. 529–544. Springer, 2022.

Bryan A Plummer, Liwei Wang, Chris M Cervantes, Juan C Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *Proceedings of the IEEE international conference on computer vision*, pp. 2641–2649, 2015.

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.

Nimrod Shabtay, Itamar Zimerman, Eli Schwartz, and Raja Giryes. Climp: Contrastive language-image mamba pretraining. In *European Conference on Computer Vision*, pp. 229–248. Springer, 2026.

Min Shi, Fuxiao Liu, Shihao Wang, Shijia Liao, Subhashree Radhakrishnan, Yilin Zhao, De-An Huang, Hongxu Yin, Karan Sapra, Yaser Yacoob, et al. Eagle: Exploring the design space for multimodal llms with mixture of encoders. In *International Conference on Learning Representations*, volume 2025, pp. 92628–92646, 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 (CVPR)*, June 2019.

Sagar Soni, Akshay Dudhane, Hiyam Debary, Mustansar Fiaz, Muhammad Akhtar Munir, Muhammad Sohail Danish, Paolo Fraccaro, Campbell D Watson, Levente J Klein, Fahad Shahbaz Khan, et al. Earthdial: Turning multi-sensory earth observations to interactive dialogues. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14303–14313. IEEE, 2025.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, et al. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*, 2024.

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, 2024a.

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

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

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

Ke-Jyun Wang, Yun-Hsuan Liu, Hung-Ting Su, Jen-Wei Wang, Yu-Siang Wang, Winston Hsu, and Wen-Chin Chen. OCID-ref: A 3D robotic dataset with embodied language for clutter scene grounding. In Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tur, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou (eds.), *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 5333–5338, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.419. URL https://aclanthology.org/2021.naacl-main.419/.

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

Xiao Wang, Fuling Wang, Yuehang Li, Qingchuan Ma, Shiao Wang, Bo Jiang, and Jin Tang. Cxpmrg-bench: Pre-training and benchmarking for x-ray medical report generation on chexpert plus dataset. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5123–5133. IEEE, 2025b.

Xiao Wang, Yuehang Li, Fuling Wang, Shiao Wang, Chuanfu Li, and Bo Jiang. R2gencsr: Mining contextual and residual information for llms-based radiology report generation, 2026. URL https://arxiv.org/abs/2408.09743.

Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Lihe Yang, Zhen Zhao, and Hengshuang Zhao. SigLIP-HD by fine-to-coarse supervision. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=XeLrfKEOZS.

Licheng Yu, Patrick Poirson, Shan Yang, Alexander C Berg, and Tamara L Berg. Modeling context in referring expressions. In *European conference on computer vision*, pp. 69–85. Springer, 2016.

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

Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

Lianghui Zhu, Bencheng Liao, Qian Zhang, Xinlong Wang, Wenyu Liu, and Xinggang Wang. Vision mamba: Efficient visual representation learning with bidirectional state space model. In *International Conference on Machine Learning*. PMLR, 2024.

A Experimental Details

VLM training. Throughout the paper, Vicuna-7B denotes the `lmsys/vicuna-7b-v1.5` checkpoint. We train for one epoch on the LLaVA-1.5 665K instruction mixture, using AdamW with a learning rate of 2×10^{-5}, weight decay of 0.1, a global batch size of 128, and gradient norm clipping at 1.0. The learning rate uses 3% linear warm-up followed by cosine decay. All runs use seed 7 and four NVIDIA H200 GPUs with FSDP and BF16 mixed precision. The connector and language model are trained jointly; the vision encoder and pre-connector normalization receive no gradient updates.

Feature extraction and connectors. We use the penultimate transformer block for ViT (Karamcheti et al., 2024), excluding the class token, and the stage-3 output for VMamba and Swin. Let D denote the visual feature dimension and L the language-model hidden dimension. The two-layer MLP has dimensions $D \to L \to L$, while the three-layer MLP has dimensions $D \to 4D \to L \to L$. Both apply GELU between linear layers. The six-layer MLP maps D to L and uses five further $L \to L$ layers, with GELU between layers. The attention connectors use either self-attention over visual tokens or cross-attention with 196 learned queries.

Encoder adaptation. We adapt all ViT, VMamba, and Swin encoders from ImageNet-1K checkpoints, using the same training code within each task. Detection adaptation uses Mask R-CNN with a 12-epoch schedule, and segmentation adaptation uses UPerNet. The ViT and VMamba checkpoints in the resolution study are adapted separately at each resolution.

B Evaluation Protocols

We build on the Prismatic evaluation framework (Karamcheti et al., 2024) and use the same evaluation protocol across encoder configurations. Open-ended answers use greedy decoding. For POPE, TallyQA, and CV-Bench 2D, we select the answer option with the highest first-token probability. TallyQA uses its test split with 16 options representing counts from 0 to 15. TextVQA is evaluated separately with and without supplied OCR tokens.

CV-Bench 2D. We evaluate counting and spatial-relation questions. The reported score is the unweighted mean of the accuracies on the COCO and ADE20K image sources.

Flickr30k Entities. We evaluate 11,430 phrase–image pairs from the 1,000-image test split. Each phrase is associated with the union of its annotated boxes, and a prediction is correct at $\mathrm{IoU} \geq 0.5$. Phrases without visual boxes are excluded. The evaluation code specifies the prompts, coordinate conversion, and dataset filtering.

C Feature Extraction Stage

We compare stage-3 and stage-4 features from the same CC12M-pretrained VMamba-B checkpoint, using the same downstream VLM recipe. Stage 3 shows higher overall performance and the largest difference in grounding. This comparison changes both feature depth and spatial token count: stage 3 provides 196 tokens, while stage 4 provides 49.

Table 5: VLM performance using stage-3 or stage-4 features from the same CC12M-pretrained VMamba-B checkpoint. Stage 3 provides 196 visual tokens, and stage 4 provides 49.

				VQA										Grounding					
Visual Encoder	Feature Stage	Tokens	Overall Avg	VQA-v2	GQA	VizWiz	TextVQA	TextVQA +OCR	POPE	TallyQA	CV-Bench^{2D}	Avg	RefCOCO	RefCOCO+	RefCOCOg	OCID-Ref	Flickr30k	Avg
VMamba-B	stage 3	196	**50.7**	**66.8**	**55.3**	**46.6**	**12.5**	**44.9**	**82.5**	**55.6**	**53.3**	**52.2**	**62.8**	**51.7**	**57.2**	**26.2**	**44.2**	**48.4**
VMamba-B	stage 4	49	41.6	63.5	52.7	45.8	12.1	44.7	80.3	54.8	52.2	50.8	38.5	28.6	33.6	9.3	24.6	26.9

D PROBING DETAILS

Data and labels. We sample 8,000 images from COCO train2017, using 7,200 for probe training and 800 for model selection. Evaluation uses the 5,000 val2017 images. The probes predict the 80 COCO object categories. A patch is positive for a category if its cell intersects that category's segmentation mask after image preprocessing. Padding cells have no positive labels. Image-level labels indicate which categories occur in the processed image.

Features and fitting. We fit separate linear probes before and after the trained connector. Image-level features are obtained by averaging the tokens at the corresponding location; after the connector, projection therefore precedes pooling. Each probe uses independent sigmoid outputs and binary cross-entropy loss. We train for 20 epochs with AdamW, weight decay 0.01, and cosine learning-rate decay. The learning rate is selected from $\{10^{-2}, 3 \times 10^{-3}, 10^{-3}, 3 \times 10^{-4}\}$ using macro recall on the held-out training split.

Scoring. We predict a category when its probability exceeds 0.5. Recall is computed separately for each category and then averaged across categories. Patch-level recall pools predictions across evaluation patches; image-level recall pools predictions across images.

E DETAILED BENCHMARK RESULTS

Table 6: Per-benchmark results for the language-backbone comparison in Figure 2 (left). We evaluate CC12M-pretrained ViT-B/16 and VMamba-B with four language backbones, with and without LayerNorm before the connector. All configurations use a two-layer MLP connector.

				VQA										Grounding					
LLM	Visual Encoder	LN	Overall Avg	VQA-v2	GQA	VizWiz	TextVQA	TextVQA +OCR	POPE	TallyQA	CV-Bench^{2D}	Avg	RefCOCO	RefCOCO+	RefCOCOg	OCID-Ref	Flickr30k	Avg
Gemma-2-2B-IT	ViT-B/16	w/o	45.5	63.5	52.3	39.8	11.6	39.8	77.9	52.8	51.1	48.6	56.1	46.3	48.2	19.0	33.7	40.6
		w/	43.7	62.7	51.8	36.8	11.2	40.4	76.0	52.2	48.6	47.5	52.4	41.6	44.3	17.6	32.7	37.7
	VMamba-B	w/o	33.3	59.1	48.1	34.2	11.8	39.0	73.2	47.3	45.8	44.8	24.9	14.5	17.4	3.8	14.3	15.0
		w/	44.8	64.3	52.2	34.4	12.1	40.4	80.3	55.0	50.8	48.7	53.4	42.3	45.7	17.0	33.9	38.5
Qwen3-4B-IT	ViT-B/16	w/o	45.4	63.7	52.6	32.2	11.7	45.6	80.4	56.5	52.1	49.3	53.4	43.0	44.8	18.8	35.8	39.2
		w/	41.0	63.4	52.3	30.0	11.7	45.0	80.9	54.8	48.3	48.3	44.5	32.4	34.0	9.9	26.2	29.4
	VMamba-B	w/o	37.8	61.7	51.2	28.2	11.6	44.5	82.8	52.5	50.2	47.8	33.4	22.6	26.0	5.7	20.9	21.7
		w/	50.6	66.4	54.6	28.8	12.3	**46.0**	**83.7**	**59.5**	51.8	50.4	**66.9**	**55.5**	**60.8**	**26.4**	**44.8**	**50.9**
Llama2-7B	ViT-B/16	w/o	46.2	65.0	53.9	48.7	12.5	43.7	80.8	53.0	51.2	51.1	51.7	42.2	44.2	17.2	36.0	38.3
		w/	47.5	64.9	53.9	47.0	12.3	44.6	80.3	52.7	**53.7**	51.2	55.9	45.0	48.1	20.8	38.0	41.6
	VMamba-B	w/o	40.5	64.1	52.8	48.0	12.6	43.0	79.9	51.6	49.5	50.2	37.8	26.8	29.6	7.8	23.3	25.1
		w/	49.5	**66.8**	55.2	39.2	**12.8**	43.8	82.2	56.1	52.5	51.1	61.5	51.0	56.6	21.6	43.6	46.9
Vicuna-7B	ViT-B/16	w/o	46.6	65.0	53.4	49.0	12.4	44.4	80.0	56.2	52.9	51.6	52.4	42.4	45.0	17.4	35.4	38.5
		w/	47.4	64.9	53.2	44.2	12.3	44.8	80.5	56.2	53.1	51.1	55.6	45.2	48.2	20.2	37.4	41.3
	VMamba-B	w/o	36.3	60.9	50.3	**49.3**	12.4	44.2	76.3	52.7	50.4	49.6	25.3	14.3	17.1	3.1	15.8	15.2
		w/	**50.7**	**66.8**	**55.3**	46.6	12.5	44.9	82.5	55.6	53.3	**52.2**	62.8	51.7	57.2	26.2	44.2	48.4

Table 7: Per-benchmark results for the resolution study in Figure 3. ViT and VMamba are evaluated at three input resolutions within the classification, detection, and segmentation settings. Each resolution uses a separately trained VLM.

				VQA										Grounding					
Pretraining	Visual Encoder	Input	Overall Avg	VQA-v2	GQA	VizWiz	TextVQA	TextVQA +OCR	POPE	TallyQA	CV-Bench^{2D}	Avg	RefCOCO	RefCOCO+	RefCOCOg	OCID-Ref	Flickr30k	Avg
Classification	ViT-S/16	224x224	43.0	62.3	52.3	46.8	12.2	44.4	78.2	51.8	44.7	49.1	46.6	35.8	39.8	13.2	30.7	33.2
		256x256	43.6	62.5	52.4	44.9	12.0	43.7	78.8	53.8	50.5	49.8	46.6	35.7	41.1	14.3	30.9	33.7
		512x512	47.5	64.6	54.3	50.5	12.2	44.8	81.0	53.4	47.5	51.0	55.5	44.1	48.6	21.4	39.2	41.8
	ViT-B/16	224x224	43.5	62.6	51.9	46.2	12.2	**44.9**	79.5	52.8	48.8	49.8	46.3	34.7	40.9	13.6	30.6	33.2
		256x256	43.2	62.6	51.9	44.2	12.3	44.5	79.0	50.1	44.6	48.6	48.1	37.0	41.4	14.2	32.3	34.6
		512x512	46.6	64.6	54.0	47.0	12.4	44.6	81.2	53.1	46.4	50.4	53.4	42.4	47.8	20.8	38.6	40.6
	VMamba-T	224x224	47.8	65.0	54.0	45.0	12.2	44.6	81.7	54.6	50.4	50.9	58.2	46.6	51.7	20.2	37.8	42.9
		256x256	49.3	65.7	54.5	45.6	12.4	44.8	81.8	56.1	**51.3**	51.5	60.8	49.5	54.7	22.6	40.8	45.7
		512x512	52.4	**68.1**	**56.3**	46.6	12.2	44.6	84.1	**58.2**	48.8	52.4	**67.7**	**57.1**	**62.1**	29.8	45.1	**52.4**
	VMamba-S	224x224	47.6	65.2	54.1	45.0	12.2	44.1	82.2	55.5	50.0	51.0	56.5	44.3	49.9	23.1	36.8	42.1
		256x256	49.4	65.9	54.9	51.1	12.6	44.7	82.9	54.3	49.2	52.0	59.6	47.6	53.1	26.5	40.4	45.4
		512x512	**52.5**	67.9	**56.3**	**51.4**	**12.8**	44.5	**85.0**	56.1	50.5	**53.1**	65.8	54.6	61.0	**31.0**	**45.9**	51.7
	VMamba-B	224x224	42.6	64.2	53.2	48.0	12.1	43.8	80.8	54.0	47.4	50.4	42.1	31.4	36.2	15.2	25.2	30.0
		256x256	46.3	65.1	54.3	49.9	12.4	**44.9**	82.3	55.7	51.2	52.0	51.2	39.1	44.5	18.8	32.2	37.2
		512x512	48.6	66.8	55.3	49.2	12.5	44.4	83.8	56.2	49.5	52.2	56.7	44.4	50.8	26.4	36.3	42.9
Detection	ViT-S/16	224x224	45.9	60.8	50.3	45.2	11.3	44.0	80.9	53.2	51.3	49.6	57.2	42.5	49.6	16.7	33.8	39.9
		256x256	46.7	61.6	50.9	41.9	11.7	43.9	80.2	52.6	48.7	49.0	60.4	46.4	54.8	17.4	36.4	43.1
		512x512	52.3	64.7	53.3	47.2	11.5	44.2	84.7	57.6	51.3	51.8	71.7	58.2	65.1	26.6	43.9	53.1
	ViT-B/16	224x224	46.1	61.8	51.0	47.1	11.8	44.0	81.4	55.0	48.9	50.1	56.2	42.6	50.4	16.4	32.8	39.7
		256x256	48.1	63.0	51.6	46.8	11.6	44.4	81.5	55.1	49.5	50.4	62.8	49.5	55.9	17.5	35.6	44.3
		512x512	52.9	65.8	54.2	45.4	11.7	44.4	84.6	56.3	52.2	51.8	73.4	60.5	67.5	27.6	43.6	54.5
	VMamba-T	224x224	52.0	64.8	53.7	42.3	11.8	44.7	83.4	57.7	54.9	51.7	73.1	58.8	67.4	22.5	41.2	52.6
		256x256	52.8	65.4	53.7	46.4	11.7	44.1	83.9	57.1	53.9	52.0	73.9	60.2	68.1	25.8	42.3	54.1
		512x512	54.7	**67.2**	55.0	46.9	11.7	44.0	86.8	**59.4**	54.5	53.2	76.2	**62.9**	69.5	31.8	45.9	57.3
	VMamba-S	224x224	52.9	64.9	53.6	45.5	11.7	**44.9**	84.3	56.6	53.9	51.9	74.4	59.3	69.3	26.9	41.8	54.4
		256x256	53.6	65.4	54.0	**49.9**	11.8	44.3	85.4	56.9	**56.0**	53.0	75.1	59.6	68.9	26.8	43.4	54.7
		512x512	**55.8**	67.0	**55.1**	48.6	**12.3**	44.7	88.2	58.8	55.5	**53.8**	**78.0**	62.5	**72.2**	**35.1**	**47.0**	**59.0**
	VMamba-B	224x224	52.2	64.7	53.5	46.5	11.8	43.9	84.7	56.8	53.0	51.8	73.1	57.7	67.5	24.8	41.1	52.8
		256x256	52.6	65.0	53.8	43.4	11.8	44.4	85.4	57.2	53.9	51.9	73.8	58.5	68.1	27.6	40.7	53.7
		512x512	55.3	66.3	**55.1**	49.1	12.1	44.4	**88.6**	**59.4**	54.9	53.7	76.8	60.9	71.6	34.2	45.4	57.8
Segmentation	ViT-S/16	224x224	41.6	59.8	50.6	46.9	11.5	44.6	75.3	52.0	45.2	48.2	44.4	34.0	37.7	11.0	28.5	31.1
		256x256	42.8	60.4	51.5	46.8	11.3	43.8	77.8	51.5	47.5	48.8	47.1	35.8	40.9	12.1	30.7	33.3
		512x512	47.0	62.8	53.3	48.2	11.6	44.0	80.4	53.0	47.4	50.1	57.3	45.6	50.6	18.4	38.9	42.2
	ViT-B/16	224x224	42.3	60.6	50.8	42.8	11.7	44.1	77.2	52.8	48.6	48.6	45.7	35.1	39.4	11.1	30.4	32.3
		256x256	43.8	61.4	51.6	46.2	11.9	43.8	75.9	51.8	47.7	48.8	50.5	38.9	43.3	13.3	33.7	36.0
		512x512	48.2	64.2	53.7	**48.3**	12.1	43.9	80.8	53.0	46.5	50.3	59.9	48.4	53.8	20.9	41.2	44.8
	VMamba-T	224x224	45.2	62.3	53.0	47.9	11.9	**44.8**	80.6	52.8	48.5	50.2	52.3	40.3	46.4	12.9	34.0	37.2
		256x256	45.6	62.8	53.5	45.6	11.9	44.7	80.4	53.6	48.4	50.1	53.5	41.6	46.6	14.9	35.6	38.4
		512x512	49.6	65.4	55.3	40.9	11.8	44.2	83.6	**55.7**	49.4	50.8	62.6	51.1	57.2	24.0	**43.4**	47.7
	VMamba-S	224x224	45.2	63.1	53.7	40.9	11.8	**44.8**	81.2	54.5	**51.4**	50.2	51.9	39.8	45.1	16.2	33.6	37.3
		256x256	47.2	63.9	54.1	47.3	11.9	44.3	81.8	53.7	49.3	50.8	55.8	45.0	50.1	18.8	36.9	41.3
		512x512	**50.4**	**66.4**	55.7	47.4	11.9	44.4	84.0	53.9	46.6	**51.3**	**64.2**	**54.0**	**59.1**	24.6	43.2	**49.0**
	VMamba-B	224x224	45.2	63.2	53.7	48.1	12.2	44.1	82.0	53.4	50.0	50.9	50.7	39.3	43.8	15.3	31.8	36.2
		256x256	46.3	64.0	54.2	46.8	12.1	44.3	82.1	54.8	**51.4**	51.2	52.8	41.6	47.1	15.2	35.0	38.4
		512x512	49.7	66.1	**55.9**	40.2	**12.3**	44.5	**84.4**	53.6	46.5	50.4	64.0	52.5	58.7	**25.9**	41.8	48.6

Table 8: Per-benchmark results for the interface comparisons in Figure 4. For each frozen encoder, we compare a two-layer MLP with and without LayerNorm before the connector, and a three-layer MLP with LayerNorm before the connector. The two MLP configurations differ in both depth and hidden width.

					VQA									Grounding					
Pretraining	Visual Encoder	LN	Connector	Overall Avg	VQA-v2	GQA	VizWiz	TextVQA	TextVQA +OCR	POPE	TallyQA	CV-Bench^{2D}	Avg	RefCOCO	RefCOCO+	RefCOCOg	OCID-Ref	Flickr30k	Avg
Contrastive	ViT-B/16	w/o	2-layer MLP	46.6	65.0	53.4	49.0	12.4	44.4	80.0	56.2	52.9	51.6	52.4	42.4	45.0	17.4	35.4	38.5
		w/	2-layer MLP	47.4	64.9	53.2	44.2	12.3	44.8	80.5	56.2	53.1	51.1	55.6	45.2	48.2	20.2	37.4	41.3
		w/	3-layer MLP	46.9	64.9	53.4	45.3	12.3	44.6	79.4	56.2	53.2	51.1	53.7	43.2	47.4	19.2	36.8	40.1
	VMamba-B	w/o	2-layer MLP	36.3	60.9	50.3	**49.3**	12.4	44.2	76.3	52.7	50.4	49.6	25.3	14.3	17.1	3.1	15.8	15.2
		w/	2-layer MLP	**50.7**	**66.8**	55.3	46.6	12.5	**44.9**	82.5	55.6	53.3	52.2	**62.8**	**51.7**	**57.2**	**26.2**	**44.2**	**48.4**
		w/	3-layer MLP	49.5	**66.8**	**55.6**	45.2	**12.8**	44.6	**83.0**	**58.0**	**54.1**	**52.5**	59.0	48.1	53.9	22.2	40.3	44.7
Classification	ViT-S/16	w/o	2-layer MLP	37.6	59.9	50.5	46.7	12.3	44.0	77.8	49.0	45.8	48.2	32.3	21.8	24.8	5.1	19.6	20.7
		w/	2-layer MLP	43.0	62.3	52.3	46.8	12.2	44.4	78.2	51.8	44.7	49.1	46.6	35.8	39.8	13.2	30.7	33.2
		w/	3-layer MLP	42.6	62.3	52.3	43.1	12.3	44.3	79.4	51.8	45.8	48.9	45.5	34.7	39.9	13.3	29.8	32.6
	ViT-B/16	w/o	2-layer MLP	35.5	59.6	49.2	45.6	12.1	44.5	75.9	50.6	44.1	47.7	26.7	15.4	19.1	4.1	15.1	16.1
		w/	2-layer MLP	43.5	62.6	51.9	46.2	12.2	**44.9**	79.5	52.8	48.8	49.8	46.3	34.7	40.9	13.6	30.6	33.2
		w/	3-layer MLP	42.9	62.9	52.0	45.6	12.3	43.7	79.2	52.0	47.9	49.4	45.3	34.6	40.2	11.5	30.9	32.5
	VMamba-T	w/o	2-layer MLP	47.6	64.4	54.0	46.8	12.0	44.2	81.1	54.9	49.0	50.8	57.2	46.1	50.6	20.6	37.8	42.5
		w/	2-layer MLP	47.8	65.0	54.0	45.0	12.2	44.6	81.7	54.6	50.4	50.9	**58.2**	46.6	51.7	20.2	37.8	42.9
		w/	3-layer MLP	**48.4**	65.2	53.9	44.9	12.2	44.8	81.2	56.3	50.9	51.2	**58.2**	**47.3**	**51.8**	22.0	**39.8**	**43.8**
	VMamba-S	w/o	2-layer MLP	34.5	60.2	49.8	46.0	11.8	44.7	75.5	49.7	42.8	47.6	23.2	12.9	15.9	3.7	12.3	13.6
		w/	2-layer MLP	47.6	65.2	54.1	45.0	12.2	44.1	**82.2**	55.5	50.0	51.0	56.5	44.3	49.9	23.1	36.8	42.1
		w/	3-layer MLP	47.8	**65.5**	**54.8**	47.4	**12.4**	43.8	81.7	55.9	**51.5**	**51.6**	54.6	43.9	49.2	**24.4**	36.9	41.8
	VMamba-B	w/o	2-layer MLP	34.4	59.6	49.6	46.3	12.0	43.9	74.9	50.0	48.4	48.1	21.6	12.3	15.2	2.7	11.2	12.6
		w/	2-layer MLP	42.6	64.2	53.2	48.0	12.1	43.8	80.8	54.0	47.4	50.4	42.1	31.4	36.2	15.2	25.2	30.0
		w/	3-layer MLP	44.4	64.5	53.8	**48.8**	**12.4**	44.0	81.8	**56.4**	51.3	**51.6**	45.4	34.0	39.3	16.8	28.4	32.8
Detection	ViT-S/16	w/o	2-layer MLP	36.5	59.4	49.2	41.5	11.4	44.0	80.4	52.3	45.1	47.9	29.9	17.3	22.4	4.3	17.7	18.3
		w/	2-layer MLP	52.3	64.7	53.3	47.2	11.5	44.2	84.7	57.6	51.3	51.8	71.7	58.2	65.1	26.6	43.9	53.1
		w/	3-layer MLP	52.4	64.5	54.0	45.8	11.7	43.8	84.7	56.6	49.7	51.4	72.6	58.0	66.1	27.3	45.7	53.9
	ViT-B/16	w/o	2-layer MLP	48.4	62.8	52.2	47.0	11.4	44.6	83.1	53.6	48.8	50.4	63.7	47.0	56.8	20.8	37.7	45.2
		w/	2-layer MLP	52.9	65.8	54.2	45.4	11.7	44.4	84.6	56.3	52.2	51.8	73.4	60.5	67.5	27.6	43.6	54.5
		w/	3-layer MLP	52.4	64.9	54.0	44.8	11.7	44.6	85.1	55.3	51.6	51.5	72.4	58.5	66.1	28.2	43.5	53.8
	VMamba-T	w/o	2-layer MLP	51.1	65.0	53.7	45.4	11.8	44.3	85.7	56.8	52.9	52.0	67.7	53.1	61.4	24.7	41.7	49.7
		w/	2-layer MLP	54.7	67.2	55.0	46.9	11.7	44.0	86.8	59.4	54.5	53.2	76.2	62.9	69.5	31.8	45.9	57.3
		w/	3-layer MLP	55.7	**67.6**	55.5	42.0	11.9	44.4	87.2	61.1	53.4	52.9	**78.6**	**64.7**	**73.0**	36.1	**48.2**	**60.1**
	VMamba-S	w/o	2-layer MLP	38.1	61.3	50.2	44.0	12.0	**44.7**	82.4	56.5	50.3	50.2	31.9	18.2	23.0	4.2	16.2	18.7
		w/	2-layer MLP	55.8	67.0	55.1	48.6	**12.3**	**44.7**	88.2	58.8	55.5	53.8	78.0	62.5	72.2	35.1	47.0	59.0
		w/	3-layer MLP	**56.7**	66.7	**55.8**	48.8	12.2	44.3	**88.8**	**63.1**	**58.1**	**54.7**	78.3	62.6	72.1	**37.5**	**48.2**	59.8
	VMamba-B	w/o	2-layer MLP	50.3	63.8	53.6	48.6	11.7	44.1	87.7	56.3	52.0	52.2	66.1	49.4	60.0	23.0	38.1	47.3
		w/	2-layer MLP	55.3	66.3	55.1	**49.1**	12.1	44.4	88.6	59.4	54.9	53.7	76.8	60.9	71.6	34.2	45.4	57.8
		w/	3-layer MLP	56.1	66.2	55.4	47.4	12.2	44.1	**88.8**	61.7	56.4	54.0	78.5	62.3	72.8	36.2	47.7	59.5
Segmentation	ViT-S/16	w/o	2-layer MLP	43.6	61.1	52.1	47.9	11.5	44.1	79.7	46.8	43.8	48.4	50.7	38.7	43.6	13.2	33.2	35.9
		w/	2-layer MLP	47.0	62.8	53.3	48.2	11.6	44.0	80.4	53.0	47.4	50.1	57.3	45.6	50.6	18.4	38.9	42.2
		w/	3-layer MLP	48.1	63.1	53.3	**50.1**	11.9	43.8	79.9	55.1	**52.4**	51.2	58.2	45.6	52.3	19.7	39.4	43.0
	ViT-B/16	w/o	2-layer MLP	38.4	60.4	50.4	44.1	11.7	43.8	77.2	48.1	43.3	47.4	36.8	25.1	28.9	5.3	23.7	24.0
		w/	2-layer MLP	48.2	64.2	53.7	48.3	12.1	43.9	80.8	53.0	46.5	50.3	59.9	48.4	53.8	20.9	41.2	44.8
		w/	3-layer MLP	49.9	64.7	53.9	49.4	12.0	44.3	81.1	54.4	48.0	51.0	62.9	51.4	57.1	24.6	**45.3**	48.3
	VMamba-T	w/o	2-layer MLP	47.0	64.4	54.6	44.4	11.7	43.9	82.5	53.0	49.2	50.5	55.0	44.2	50.2	18.8	38.4	41.3
		w/	2-layer MLP	49.6	65.4	55.3	40.9	11.8	44.2	83.6	55.7	49.4	50.8	62.6	51.1	57.2	24.0	43.4	47.7
		w/	3-layer MLP	**51.9**	65.7	55.0	49.1	11.9	44.3	83.5	56.6	52.2	**52.3**	66.8	55.4	**61.7**	**28.1**	44.6	**51.3**
	VMamba-S	w/o	2-layer MLP	35.5	59.6	49.3	49.2	11.9	44.0	78.1	51.4	43.7	48.4	26.0	14.0	17.3	2.3	15.0	14.9
		w/	2-layer MLP	50.4	**66.4**	55.7	47.4	11.9	44.4	84.0	53.9	46.6	51.3	64.2	54.0	59.1	24.6	43.2	49.0
		w/	3-layer MLP	51.6	**66.4**	55.9	44.8	12.0	43.9	84.3	**57.8**	49.0	51.8	**67.3**	**55.9**	61.5	27.1	44.6	**51.3**
	VMamba-B	w/o	2-layer MLP	34.8	59.5	49.8	47.5	11.8	43.9	78.1	46.4	44.4	47.7	25.5	13.6	16.3	2.0	13.0	14.1
		w/	2-layer MLP	49.7	66.1	55.9	40.2	12.3	**44.5**	84.4	53.6	46.5	50.4	64.0	52.5	58.7	25.9	41.8	48.6
		w/	3-layer MLP	50.8	66.3	**56.1**	46.0	**12.4**	44.3	**84.6**	55.8	51.7	52.2	64.0	51.2	58.2	27.0	42.5	48.6

Table 9: Per-benchmark results for the connector comparison in Figure 2 (right). The CC12M-pretrained ViT-B/16 and VMamba-B use Vicuna-7B and LayerNorm before the connector. We compare MLP and attention-based connectors.

			VQA										Grounding					
Connector	Visual Encoder	Overall Avg	VQA-v2	GQA	VizWiz	TextVQA	TextVQA +OCR	POPE	TallyQA	CV-Bench^{2D}	Avg	RefCOCO	RefCOCO+	RefCOCOg	OCID-Ref	Flickr30k	Avg
2-layer MLP	ViT-B/16	47.4	64.9	53.2	44.2	12.3	44.8	80.5	56.2	53.1	51.1	55.6	45.2	48.2	20.2	37.4	41.3
	VMamba-B	**50.7**	**66.8**	55.3	**46.6**	12.5	**44.9**	82.5	55.6	53.3	52.2	**62.8**	**51.7**	**57.2**	**26.2**	**44.2**	**48.4**
3-layer MLP	ViT-B/16	46.9	64.9	53.4	45.3	12.3	44.6	79.4	56.2	53.2	51.1	53.7	43.2	47.4	19.2	36.8	40.1
	VMamba-B	49.5	**66.8**	**55.6**	45.2	**12.8**	44.6	**83.0**	**58.0**	**54.1**	**52.5**	59.0	48.1	53.9	22.2	40.3	44.7
6-layer MLP	ViT-B/16	30.5	50.7	43.5	39.8	10.7	42.2	67.4	45.6	39.6	42.4	19.4	10.9	12.7	2.3	12.1	11.5
	VMamba-B	35.5	59.8	49.7	43.2	12.2	43.9	80.3	52.4	46.3	48.5	23.7	13.4	17.3	4.5	14.9	14.8
Cross	ViT-B/16	35.7	60.4	49.3	41.8	11.7	44.2	74.8	53.2	49.6	48.1	26.8	15.0	17.3	4.7	15.9	16.0
	VMamba-B	36.2	61.4	50.4	44.7	12.4	44.2	76.8	54.0	50.9	49.4	25.4	14.5	18.0	4.0	14.6	15.3
Self-Attn	ViT-B/16	36.2	60.8	49.3	46.2	11.8	43.6	78.2	53.1	48.3	48.9	26.2	15.1	17.6	4.4	15.8	15.8
	VMamba-B	35.5	61.6	50.1	42.5	12.1	44.6	77.1	53.1	52.4	49.2	23.4	13.4	15.0	3.2	13.5	13.7