

UC Berkeley

UC Berkeley Electronic Theses and Dissertations

Title

Vision Models That See 10 Billion Pixels at Once

Permalink

<https://escholarship.org/uc/item/6842j5qr>

ISBN

9798189272141

Author

Shi, Baifeng

Publication Date

2026-05-01

Peer reviewed|Thesis/dissertation

Vision Models That See 10 Billion Pixels at Once

by

Baifeng Shi

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Computer Science

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Trevor Darrell, Chair

Professor Jitendra Malik

Professor Alexei A. Efros

Dr. Pavlo Molchanov

Spring 2026

Vision Models That See 10 Billion Pixels at Once

Copyright 2026
by
Baifeng Shi

Abstract

Vision Models That See 10 Billion Pixels at Once

by

Baifeng Shi

Doctor of Philosophy in Computer Science

University of California, Berkeley

Professor Trevor Darrell, Chair

Modern vision models face a fundamental scaling challenge: real-world images and videos contain far more visual information than current models can process since they treat every pixel equally and encode all of them uniformly. High-resolution images contain small text, distant objects, and fine local details, while long videos multiply this burden across time. This thesis studies efficient methods and designs to scale vision models to extremely high spatiotemporal fidelity, inspired by the selective nature of human vision where high-fidelity processing is allocated unevenly across space, scale, and time.

The first part of the thesis studies whether scaling vision models to higher fidelity is necessary. I show that Scaling on scales (S^2), a naive approach of scaling up visual resolution by running frozen vision models on tiles of larger images, can already match or exceed the benefit of scaling model size across classification, dense prediction, multimodal language-model benchmarks, and robotic manipulation. The second part explores how to scale vision pre-training to extremely high resolution. I introduce PS3, a scale-selective vision-language pre-training method that enables pre-training on 4K-resolution images with near-constant cost by contrasting between local high-resolution regions and captions. The resulting PS3 and VILA-HD model improves high-resolution multimodal language-model performance, especially on the self-collected 4KPro benchmark that evaluates visual question answering that truly requires 4K perception. The third part discusses how to scale not only spatial, but also temporal fidelity for video models. I introduce AutoGaze, an autoregressive gazing module that removes redundant video patches before passing to expensive vision encoders or vision-language models, enabling efficient understanding of long, high-resolution videos, improving the performance on various video understanding benchmarks, especially on HLVID, the self-collected high-resolution long-form video understanding benchmark. Together, these contributions demonstrate the way of efficient visual scaling to extremely high spatiotemporal fidelity and achieving human-level visual understanding of fine-grained details across spatial and temporal dimensions.

To my family, friends, and mentors.

Contents

Contents	ii
List of Figures	iv
List of Tables	x
1 Introduction	1
2 When Do We Not Need Larger Vision Models?	5
2.1 Introduction	5
2.2 Related Work	6
2.3 The Power of Scaling on Scales	7
2.4 The (Non)Necessity of Scaling Model Size	14
2.5 Discussion	19
3 Scaling Vision Pre-Training to 4K Resolution	20
3.1 Introduction	20
3.2 PS3: Vision Pre-Training at 4K Resolution	23
3.3 VILA-HD: Enabling High-Resolution MLLM with PS3	28
3.4 Scaling Properties of PS3	31
3.5 4KPro: Benchmarking PS3 at 4K Resolution	35
3.6 Comparing PS3 and VILA-HD to State of the Arts	38
3.7 Ablations and Analysis	44
3.8 Related Work	48
3.9 Current Limitations and Future Directions	50
3.10 Conclusion	52
4 Attend Before Attention: Efficient and Scalable Video Understanding via Autoregressive Gazing	53
4.1 Introduction	53
4.2 Related Work	56
4.3 AutoGaze for Efficient Video Understanding	56
4.4 Experiments	60

4.5 Conclusion	67
5 Conclusion	69
Bibliography	70

List of Figures

1.1	The information pyramid of human vision. Anderson and Van Essen describe visual processing as a sequence of information bottlenecks: retinal images contain more than 10^{10} bits/sec, the optic nerve transmits roughly 3×10^6 bits/sec, attentive scrutiny admits less than 10^4 bits/sec, and long-term memory stores only about 1–10 bits/sec [3]. The reduction from retinal images to attentive scrutiny alone is on the order of $10^6\times$, illustrating the compression required for efficient visual intelligence.	2
1.2	Two bottlenecks in human visual information processing. The acuity bottleneck arises because recognizability decreases with eccentricity, so humans move their gaze to bring important regions onto the high-resolution fovea. The attention bottleneck remains even at a fixed gaze location: attention can still be allocated to different parts of the scene, including peripheral regions, for detailed processing.	3
2.1	S²-Wrapper is a simple mechanism that extends any pre-trained vision model to multiple image scales in a parameter-free manner. Taking ViT-B as an example, S ² -Wrapper first interpolates the input image to different scales (<i>e.g.</i> , 224^2 and 448^2) and splits each into several sub-images of the same size as the default input size ($448^2 \rightarrow 4 \times 224^2$). For each scale, all sub-images are fed into the same model and the outputs (<i>e.g.</i> , 4×16^2) are merged into feature map of the whole image (32^2). Feature maps of different scales are average-pooled to the original spatial size (16^2) and concatenated together. The final multi-scale feature has the same spatial shape as single-scale feature while having higher channel dimension (<i>e.g.</i> , 1536 <i>vs.</i> 768).	7

2.2	Comparison of S^2 scaling and model size scaling on three models (ViT, DINOv2, and OpenCLIP) and three tasks (ImageNet classification, semantic segmentation, and depth estimation). For each model and each task, we test base, large, and huge/giant models for model size scaling (plotted in gray curve). For S^2 scaling (plotted in green curve), we test three sets of scales from single-scale (1x) to multi-scale (up to 3x), and we adjust each set of scale so that it matches the GFLOPs of the respective model size. Note that for specific models and tasks, we test S^2 scaling on both base and large models (plotted in light green and dark green curves separately). We can see that in (a), (d), (e), (f), (g), and (i), the base model with S^2 scaling already achieves comparable or better performances than larger models with similar GFLOPs and much smaller model size. For (b), (h), S^2 scaling from the large model is comparable with the giant model, again with similar GFLOPs and fewer parameters. The only failure case is (c), where S^2 scaling on either base or large models does not compete with model size scaling.	9
2.3	Comparison of S^2 scaling and model size scaling on MLLM. S^2 scaling has comparable or better scaling curve than model size scaling on all three types of benchmarks. Using large image scales consistently gives better performance while using larger model can degrade model performance in certain cases. . . .	11
2.4	LLaVA-1.5 with S^2 scaling is able to recognize extremely fine-grained details in an image , <i>e.g.</i> , the color of a water bottle which lives in only 23×64 pixels of a 2250×1500 image.	12
2.5	S^2 vs. model size scaling on cube picking task. S^2 scaling on base-size model improves the success rate by about 20%.	13
2.6	Which model size should we scale up image scales on? The answer varies for different pre-trained models. For ViT and OpenCLIP, S^2 scaling from base or large model gives similar performances under computation budget beyond the huge-size model while the former performs better under similar GFLOPS as the large-size model. For DINOv2, S^2 scaling from the large size model has better performance than scaling from base size, especially under the same level of computation budget as the giant-size model.	14
2.7	Types of samples that ViT-L improves the most but ViT-B-S^2 does not. (a) Rare cases. These samples clearly belong to the class but are hard to classify due to the rare appearance (<i>e.g.</i> , sculptures of television and flute). (b) Ambiguous cases. These samples have ambiguous labels. For example, the lotion could also be soap dispenser due to their high similarity, or the label could be either airship or traffic light when these two objects co-exist.	16

- 3.1 **Left:** Regular vision models such as SigLIP processes images at a low resolution (*e.g.*, 378×378 pixels), which is not enough for many daily tasks such as spotting the stop sign while driving. In contrast, PS3 is able to both encode low-res features and efficiently process high-res information of 4K-resolution images via top-down patch selection, *i.e.*, selectively processing relevant patches based on any text prompt. **Top Right:** SigLIP is pre-trained by contrasting global vision features and global captions, which is costly for high-resolution images. PS3 is pre-trained with additional contrast between local high-res features with local captions, enabling pre-training at 4K resolution with $79\times$ less cost than SigLIP. **Bottom Right:** VILA-HD with PS3 as the vision encoder is able to select high-res regions to process based on the user prompt. VILA-HD outperforms state-of-the-art MLLMs such as Qwen2.5-VL [12] by 7.5% on the proposed 4KPro benchmark while achieving $1.67\times$ speedup. 22
- 3.2 **Curation of bounding boxes and captions of salient regions in the pre-training data.** For each high-resolution image, we segment all the masks, detect salient regions with small or dense masks, and use an MLLM to generate captions about the local regions. 24
- 3.3 **Pre-training data example.** Each instance contains an image with resolution up to 4K, bounding boxes of the salient regions in the image, and captions about details in the regions such as text or small objects. 24
- 3.4 **Model architecture of PS3.** The model consists of 3 stages. In Stage 1, the model encodes global low-resolution features. In Stage 2, based on the low-resolution features as well as auxiliary high-resolution features extracted by a light-weight encoder, the model selects local regions that are either relevant to a text prompt (top-down selection) or salient by themselves (bottom-up selection). In Stage 3, the model processes multi-scale high-res patches from the selected regions with the same encoder from Stage 1. KV cache from the low-res tokens in Stage 1 is added to the self-attention layers to provide a global context for local high-res encoding. 25
- 3.5 **Pre-training algorithm of PS3.** (a) During training, PS3 extracts the high-res features from the labeled local regions and contrasts them with embeddings of the local captions. To maintain the low-res feature quality, we also mix pairs of low-res features and global caption embedding in each batch. Both high-res and low-res features are extracted in the same way as Figure 3.4. (b) The top-down patch selection score is supervised by ground-truth score map generated from the bounding box corresponding to the local caption. (c) The supervision for bottom-up selection is similar to top-down selection, except that the ground-truth selection score is generated from all the labeled bounding boxes of the image. . . 27

3.6	Qualitative examples of patch selection. <i>Left:</i> PS3 is pre-trained to perform bottom-up selection based on image saliency (denoted by $\emptyset$) or top-down selection based on local captions. The selection process is detailed in Figure 3.4 and Section 3.2. <i>Middle & Right:</i> We fine-tune PS3 with MLLM to select patches based on questions about local regions (Figure 3.7 and Section 3.3).	27
3.7	Model design of VILA-HD. For any input image and text prompt, VILA-HD first extracts the low-res image features using PS3 and sends them along with the text tokens to the LLM. The last-layer embedding of the last token is used to select high-res patches in PS3, whose features are then extracted by PS3, added with additional positional embedding, and sent to the LLM. Although PS3 only processes at most 2560 high-res patches at a time, one can run the patch selection and high-res feature extraction for N times (N can be an arbitrary number) to encode up to $2560 \times N$ high-res patches.	29
3.8	Additional fine-tuning data for MLLMs with PS3. <i>Left:</i> To fine-tune top-down patch selection, we generate data with pairs of high-res image, question about a local region, and the bounding box of the region. This is generated by taking the PS3 pre-training data and retargeting the local captions into questions using LLaMA-3.1. <i>Right:</i> To align the PS3 high-res features to the LLM text space, it requires fine-tuning data that contains QA pairs on high-res images. We generate this by taking regular low-res image QA data and pasting the image onto a large-size background to get the new high-res image while the question and answer are inherited.	30
3.9	Scaling properties of PS3 on VILA-HD. (Left) Overall results. We report average performance of the MLLM on seven benchmarks under different maximum input resolution. The size of each data point indicates the number of high-res vision tokens input to the LLM. (a) When selecting all high-res patches for MLLM, the performance of PS3 scales better with the resolution than the baselines without high-resolution pre-training. (b) PS3 is able to process higher resolution and improve performance while selecting a fixed number of high-res patches for MLLM. (c) Within the same resolution, PS3 is able to trade compute for performance by selecting more high-res patches. (d) PS3 can select more high-res patches at test time even if its selects a fixed number of high-res patches during MLLM training.	32
3.10	Image resolution and MRR of different benchmarks. Existing benchmarks contain high-res images but the resolution required to answer the questions (MRR) is mostly under 1K. In contrast, 4KPro contains questions only solvable at 4K resolution.	35
3.11	Examples from 4KPro and comparison of different models. Each example corresponds to one out of four categories (Autonomous Vehicle, Household, Gaming, and UI Understanding) and each question can only be answered without ambiguity under 4K resolution. VILA-PS3 improves the accuracy over the state-of-the-art MLLMs such as GPT-4o and Qwen2-VL.	36

3.12	Scaling properties of PS3 on 4KPro. PS3 shows consistently improved performance by scaling to 4K resolution and achieves further improvements by scaling up number of high-res tokens at test time, greatly outperforming the baselines.	37
3.13	Trade-off between image scales for different benchmarks. <i>Select @ 756/1512</i> are the percentage of selected patches at 756 and 1512 scales at test time, respectively. PS3 can flexibly adjust token selection ratios at different image scales to achieve the best performance for different downstream tasks.	47
3.14	PCA visualization of visual features. The baselines, S ² and AnyRes, have either noisy or blurry features at 4K resolution, while PS3 shows extremely fine-grained features that highlight details such as small texts on the banners. .	49
4.1	AutoGaze reduces the computational cost of video understanding, scaling MLLMs to long, high-res videos. (<i>Left</i>) Existing MLLMs either process all pixels or prune tokens only in their LLMs, leaving ViTs the bottleneck. In contrast, AutoGaze eliminates redundant patches by up to 100× <i>before</i> ViTs, accelerating ViTs/MLLMs by up to 19×. (<i>Right</i>) This enables scaling to 1K-frame, 4K-res videos, achieving superior performance on HLVID, our new long, high-res video benchmark, over MLLMs limited to short or low-res videos.	54
4.2	What is AutoGaze paying attention to? For each example, we show the original video, multi-scale gazed patches, and reconstructed video. Note that we only show gazing on two scales to save space while AutoGaze actually uses four. In general, AutoGaze can 1) focus on moving objects while removing redundancy in static regions (a-e), 2) adapt to scene changes by selecting more patches (c, e), and 3) distribute attention with different granularity based on detailedness (c, f). This allows AutoGaze to select a small ratio of patches (gazing ratio) without much information loss, as reflected by the reconstruction quality.	55
4.3	Architecture and training pipeline of AutoGaze. (<i>Left & Middle</i>) Given a video, AutoGaze processes each frame and autoregressively decodes indices of multi-scale patches based on the history of frames and selected patches. Once it believes the previously-gazed patches are sufficient to reconstruct the current frame, it automatically stops gazing and moves to the next frame. (<i>Right</i>) AutoGaze is trained in two stages: next-token-prediction pre-training on collected gazing sequences, and RL post-training with reconstruction reward.	57
4.4	AutoGaze targets patches with higher optical flow. (<i>Left</i>) AutoGaze uses coarser scales to capture higher optical flow. (<i>Right</i>) Across all scales, AutoGaze more frequently selects patches with higher optical flow. Error bars represent SEM.	60
4.5	Gazing scale correlates with patch detail. (<i>Left</i>) At finer scales, AutoGaze selects more detailed patches (measured as Laplacian variance). (<i>Right</i>) With increasing detail, AutoGaze uses finer scales ($\rho = .12, p < 0.001$). Sample patches with Laplacian variances are shown below the x-axis. Error bars represent SEM.	61

4.6	Generalizability of AutoGaze to OOD videos. <i>(a)</i> We show model behavior on videos with OOD semantics, including a CCTV clip (left), robot grasping demo (middle), and a video with object swapping (right). In each example, AutoGaze still robustly tracks the changing parts despite the unseen semantics, object categories, and unexpected changes. <i>(b)</i> We show AutoGaze output on the same video with different style transfer. AutoGaze consistently tracks the falling person regardless of visual style, texture and global illumination.	61
4.7	What gazing ratio do we need for different video types? <i>(Left)</i> There is a trade-off between gazing ratio and reconstruction loss: videos with higher FPS or resolution need lower gazing ratio to reach the same reconstruction quality. <i>(Right)</i> Gazing ratios required to reach a loss of 0.7 for videos with different FPS and resolution. 30-FPS, 4K-res videos only need $\sim 1\%$ patches.	62
4.8	Efficiency gain on ViTs and MLLMs with AutoGaze. We benchmark the ViT and MLLM latency of encoding one second of video with varying FPS and resolution. AutoGaze can select different numbers of patches to vary latency depending on user needs. When using the gazing ratio required for a reconstruction loss of 0.7, AutoGaze reduces the ViT and MLLM latency by up to $19\times$ and $10\times$	63
4.9	Scaling MLLMs to more video tokens. AutoGaze enables scaling to longer videos and higher resolution, while the baseline runs out of memory beyond 256 frames. Performance boosts are especially clear with HLVideo, where high-resolution video processing is required.	63
4.10	Comparing to baseline gazing methods. AutoGaze can use lower gazing ratios to reach the same reconstruction loss, compared to other heuristics-based approaches.	64

List of Tables

2.1	Results on MLLM. We evaluate three types of benchmarks: visual detail understanding (V* [227]), VQA benchmarks (VQAv2 [84], TextVQA [178], VizWiz [88]), and MLLM benchmarks (MMMU [253], MathVista [135], MMBench [127], SEED-Bench [112], MM-Vet [250]). Notably, S ² significantly improves the detailed understanding capability on V* benchmark, outperforming commercial models such as GPT-4V.	13
2.2	Reconstructing representation of larger models from representation of regular or multi-scale smaller models. We test three classes of models (ViT, OpenCLIP, and MAE), and for each class we test base, multi-scale base (Base-S ²), and huge or giant model. We report results on both training and test set of ImageNet-1k, and for each we report the reconstruction loss, the amount of information reconstructed, and the percentage of information reconstructed compared to huge or giant model.	18
2.3	Training loss on instance memorization and image classification. A base model with S ² scaling has similar memorization and classification losses, which implies it has at least the same level of model capacity as a large model.	19
2.4	Pre-training with S². Applying S ² on a already pre-trained model has sub-optimal performance, while pre-training with S ² makes smaller models better.	19
3.1	Full results of scaling properties of PS3 on VILA-HD. # <i>HR Token</i> is the number of high-res tokens input to the MLLM. <i>Select (Train)</i> is the percentage of high-res patches PS3 selects when training the MLLMs and <i>Select (Test)</i> is the selection ratio at test time. Note the number of HR tokens is usually 1/4 of the number of selected patches due to the 2×2 downsampling connector in NVILA [130]. [†] # <i>HR tokens</i> for AnyRes depends on the input resolution and we report the maximum number of tokens.	33
3.2	Comparing VILA-HD-8B-PS3-1.5K-SigLIP2 and VILA-HD-8B-PS3-4K-SigLIP2 to state-of-the-art MLLMs on 4KPro. VILA-HD outperforms Qwen2-VL which has the best performance among existing MLLMs while having a lower latency. [†] We limit the maximum number of pixels to 4,000,000 to maintain a reasonable latency.	37

3.3	Comparison to state-of-the-art vision encoders. We train an MLLM on top of each vision encoder with the same recipe and compare the performance on various types of benchmarks including high-resolution VQA, VQA over text, documents, and charts, and general VQA. We compare PS3-1.5K-SigLIP2 and PS3_{Lang}-1.5K-SigLIP2 (abbreviated as PS3 and PS3 _{Lang} here) with previous state-of-the-art vision encoders [22, 41, 76, 160, 198]. For a fair comparison, we use similar ViT sizes (300M - 400M) for every encoder and use S^2 to scale all the vision encoders (except for PS3 which natively supports high resolution) to a similar high resolution round 1.5K. We compare the vision encoders with and without LLM alignment separately.	39
3.4	Comparing VILA-HD to state-of-the-art MLLMs on high-resolution benchmarks. Here VILA-HD-8B-4K refer to VILA-HD-8B-PS3-4K-SigLIP2 . Bold and underlined results mean 1st and 2nd best performance for each task. VILA-HD-8B-4K achieves top-2 performance in 4 of the 6 benchmarks. [†] For most of the models, the resolution and number of tokens are dependent on the maximum number of tiles to use at test time, and we set the maximum number of tiles such that the number of tokens are similar to VILA-HD-8B-4K when it is selecting 18% patches for a fair comparison.	41
3.5	Comparing VILA-HD to state-of-the-art MLLMs on other common benchmarks. Here VILA-HD-8B-1.5K and VILA-HD-8B-4K refer to VILA-HD-8B-PS3-1.5K-SigLIP2 and VILA-HD-8B-PS3-4K-SigLIP2 , respectively. <i>Res.</i> is the maximum resolution each model supports. Some models (<i>e.g.</i> , Qwen2-VL, InternVL2) can accept input images of different aspect ratios, for which the resolution is calculated as square root of the maximum number of pixels the model can take in. <i>Select</i> is the high-res patch selection ratio of PS3 at test time. <i>#Token</i> is the total number of visual tokens fed into LLM under the maximum input resolution. VILA-HD-8B-4K achieves state-of-the-art performance on high-resolution benchmarks such as V*Bench and 4KPro.	42
3.6	Comparing PS3 to state-of-the-art token pruning methods. PS3 has consistently lower ViT latency and achieve better performances than previous methods. PS3 is also the only method that can handle 4K resolution images. . .	43
3.7	Generalizability of PS3 pre-training to state-of-the-art vision encoders. <i>#Param of ViT</i> is the number of parameters of the ViT backbone. <i>Max Res.</i> is the maximum resolution each model processes in MLLM. <i>Max #Tok</i> is the maximum number of vision tokens in MLLM. All the PS3 models select all the high-res patches.	44
3.8	Ablation of PS3 pre-training, model, and MLLM designs. Δ is the change of the average performance on the seven benchmarks after adding the design. . .	46
3.9	Ablation of top-down and bottom-up patch selection. <i>Select (Train)</i> and <i>Select (Test)</i> are the percentage of high-res patches PS3 selects at training and test time. <i>Recall</i> is the recall rate of how many patches in the ground-truth regions are selected at test time.	46

3.10	Ablation of MLLM SFT data for high-resolution feature alignment. The high-resolution SFT data (<i>HR Data</i>) generally improves high-resolution perception, especially on natural images.	48
4.1	Comparison to state-of-the-art MLLMs. NVILA-8B-Video with AutoGaze is scaled to 1K-frame, 4K-resolution videos, achieving competitive performance on general and long video benchmarks and state-of-the-art result on HLVID. . .	65
4.2	AutoGaze vs. token reduction baselines. We compare to spatial / temporal / spatiotemporal (S-/T-/ST-) token reduction approaches that are either prompt-agnostic (PA) or prompt-dependent (PD). All methods select 6.25% visual tokens on average.	66
4.3	Ablation on AutoGaze training pipeline. Both NTP pre-training and RL post-training helps with the performance.	67
4.4	Ablations of AutoGaze model designs.	67

Acknowledgments

I would like to express my deepest gratitude to my family, friends, advisors, mentors, and collaborators for their support and guidance throughout my Ph.D. journey. You are the people who made all of this possible.

First, I would like to thank my advisor, Trevor. He was my internship host when I was still an undergraduate, and later became my Ph.D. advisor. Over the six years we have worked together, he has given me so much guidance on my research and my career that I hardly know where to begin. If I had to name the most valuable thing he has given me, it would be the freedom to do research. He has always supported the things I wanted to work on, even when the ideas were uncertain and risky. Because of Trevor, I had the privilege of exploring some of the wildest ideas and most ambitious topics I could imagine. You made my Ph.D. journey truly worthwhile.

I have also been fortunate to have many brilliant mentors. Xin was my first mentor when I joined Berkeley. She had just graduated from the same lab, and we had such a wonderful time working together. We worked on three papers together, from niche visual neuroscience ideas to scaling vision models, and she supported me through some of the toughest deadlines in the first two years of my Ph.D.. Beyond research, we also shared many happy memories. That month on the keys outside winter Miami, which somehow still felt like summer, is still one of the most unforgettable vacations I have ever had.

Around the summer of 2023, I began working on robotics projects as well, thanks to a random conversation with Trevor that secretly pivoted my research as I look back now. That was when I met Ilija, another important mentor in my Ph.D. journey. He is one of the rare researchers I know who is both extraordinarily creative and incredibly thorough. When we met, he had already done a great deal of amazing work, yet he remained humble, patient, and generous with me, even though I had no robotics background at all. We had a lot of fun, and also grinded very hard, on the RPT project that summer. Later, we collaborated on the humanoid locomotion project, which became one of the loveliest projects I worked on and deeply enjoyed. Ilija has taught me so much and has always been a role model for me, from polishing the smallest details in research to imagining the grand blueprint of how to train generalist robots (and many of his bets later turned out to be right as the field caught up). He has also shaped how I think about the philosophy of research, from how to create real impact to the difficulty and necessity of integrity in many situations. I will always be grateful for what he has passed on to me.

Later in 2024, I joined NVIDIA for an internship that, to my great surprise, lasted two full years and became one of the most valuable experiences of my Ph.D. That was where I met Danny and Pavlo, two mentors whom I now feel incredibly lucky to have had. Danny can always find the optimal way to solve a problem, whether in research or beyond it. Whenever I faced a research problem and did not know how to move forward, he could immediately give advice that was clear, practical, and deeply insightful. Pavlo is one of the very few researchers I know who has become senior and manages a large team while still following the latest research advances closely. I have always been amazed by his sharp intuition for deep

learning research and his delicate sense for experiment design. Danny and Pavlo gave me so much guidance in my research, and my internship would not have been nearly as fruitful without them.

During the last semester of my Ph.D., I was fortunate to work with many brilliant researchers at Physical Intelligence. Danny, Karl, Yunhao, Haroun, Xinyu, Ashwin, Suraj, Allen, Sergey, Chelsea, and many others: you showed me what world-class robotics research should look like. I am sure there are many more amazing things we will build together in the future.

I am also grateful for the guidance I received from many other senior advisors and mentors. Jitendra provided invaluable guidance on the robotics projects with Ilija, and I also learned a great deal from his robotics course. I always enjoyed Alyosha’s philosophical research conversations with us at BAIR. I was fortunate to be advised by Yadong, Qi, Jingdong, Huijuan, Judy, and Kate during my undergraduate research. I learned a lot from several courses I took, including Yi’s course on low-dimensional structure in high-dimensional data, Bruno’s introduction to visual neuroscience, and Stella and Jitendra’s computer vision course. I was also fortunate to collaborate with Neel, Yale, Ken, Adam, Koushil, Yao, Marco, Sifei, Song, Jan, and Shankar. Thank you all for your guidance.

Next, I would like to thank all my friends for the unforgettable memories we share. Dejjia was the person who brought me into the world of computer vision. Back in the summer of 2018, I could not possibly have known that one random conversation with him would change the trajectory of my life forever. Dinghuai, Tianyuan, Yilun, and Ziqi: the research seminar you organized back in college was a goldmine for me, and I feel so fortunate to have had you as people to discuss research with and to have fun with at conferences. The trip to DC with Dinghuai after ICML 2022 was especially memorable. I still remember the feeling of freedom the moment we sat in the Uber to DC, with Harry Styles’ “As It Was” playing in the background. I had so much fun playing Mahjong with Xin, Jenny, and Tianjun, skiing with Yinuo, Xingyu, and Jimmy, and hanging out with Henry, Dantong, and Yike. I am also fortunate to have so many friends at BAIR. Xiuyu, Tony, Xudong, Max, Boyi, and Jiaxin: I will miss all the lunches and dinners we shared, the nights we spent together in the lab, the deadlines we survived, and the silly jokes that made the hard days lighter. I have also been fortunate to collaborate with many of you: Stephanie, Tony, Jiaxin, Dantong, Roi, David, Sanjay, Giscard, Devin, Jerome, Yutong, Yuvan, Ani, Sarthak, Vikas, Rachel, Siyu, Maolin, Ziyang, Ken, Bike, and many more. Playing tennis with Stephanie, Justin, Xinyan, and Wenjie was always one of my best ways to find a little peace outside research. To all the NVIDIA folks, Hanrong, Xin, Shizhe, Mike, Greg, Daguang, Xiaolong, Qinghao, Han, Wei, Yukang, Yunhao, Ligeng, and Zhijian: I learned so much from each of you. To my external collaborators, Yulu, Dandan, Aaron, and David: we built cool things together, and I hope you enjoyed the process as much as I did. To all the labmates in Trevor’s lab, Jitendra’s lab, Alyosha’s lab, Angjoo’s lab, and other labs at BAIR: it was such a privilege to share the same roof with you.

Finally, I would like to thank the people closest to my heart. Mandy, thank you for all the companionship you have given me. Throughout the three years we have known each

other, we have shared so many precious memories, and you have always supported whatever I wanted to pursue. Mom and Dad, thank you for all the unconditional love and support you have given me. You are the ones who have always stood by my side, and none of this would have been possible without you.

Thank you all, every one of you. Live long and prosper!

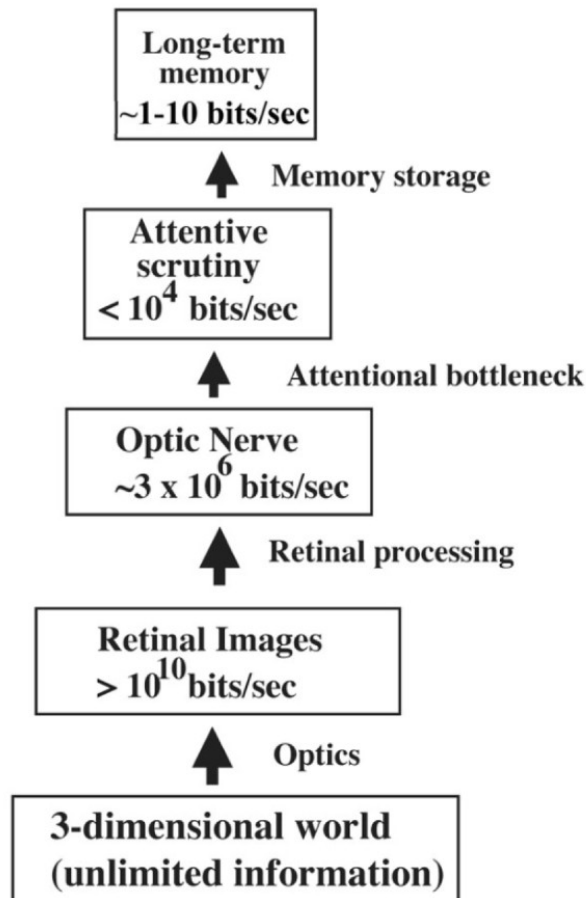
Chapter 1

Introduction

Human vision is remarkable in its capability, fidelity, and efficiency. It supports detailed recognition, navigation, manipulation, and memory across cluttered three-dimensional scenes. Especially, it preserves extremely high spatial resolution: at the center of gaze, human visual acuity is high enough that the visual field can be thought of as containing the equivalent of more than $20,000 \times 20,000$ pixels if the whole field were represented at foveal resolution. It also operates continuously over time, perceiving dynamic scenes at video rates of at least 30–60 frames per second. Despite the spatiotemporal fidelity, human vision is yet extremely efficient—it can easily process such high-fidelity visual signals in real time.

In contrast, modern vision models still have a hard time scaling to higher spatial and temporal resolution without severely compromising the efficiency. In fact, most of the state-of-the-art image and video encoders and multi-modal large language models are still limited to either low resolution (e.g., 448×448) or low-FPS (e.g., 1 or 2 FPS) [2, 8, 11, 147, 198]. This limits their ability to perceive fine-grained details in spatial (e.g., small text or distant objects) and temporal domain (e.g., highly-dynamic motions) which are crucial in everyday tasks.

Why can human vision be so efficient? Its efficiency comes from the fact that it does not process every incoming bit uniformly but aggressively compresses and selectively processes visual information that is salient or matters to the current task at hand [27, 268]. In fact, the information compression rate from raw signals at retina to the input signals to memory is over 1,000,000 times (Figure 1.1). This compression is achieved by two types of information bottleneck [199], as shown in Figure 1.2. The first is the acuity-based bottleneck: visual resolution is highest at the fovea and falls off rapidly in the periphery. The visual system therefore does not represent the entire scene at peak resolution at once, but instead only represents important regions at high resolution by moving one’s gaze at those regions. The second is the attention-based bottleneck: even after retinal and early visual compression, only a small subset of the available visual information is selected for detailed scrutiny, decision making, and memory. Eye movements and attention reveal this selection process, as observers actively sample task-relevant regions instead of uniformly processing every single bit of information.

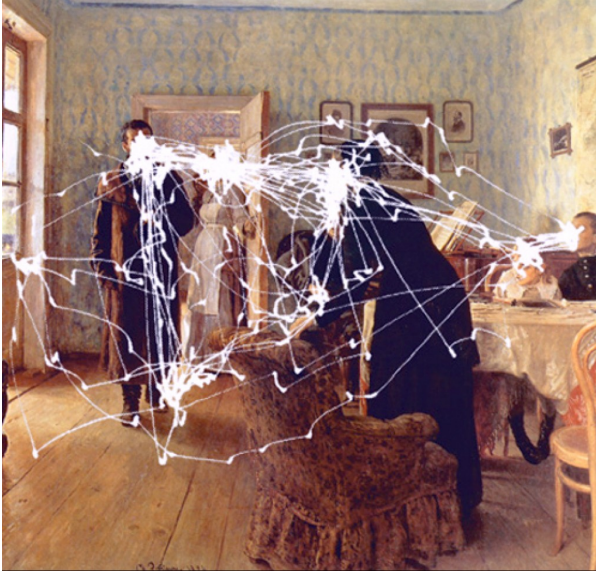


AN INFORMATION PYRAMID

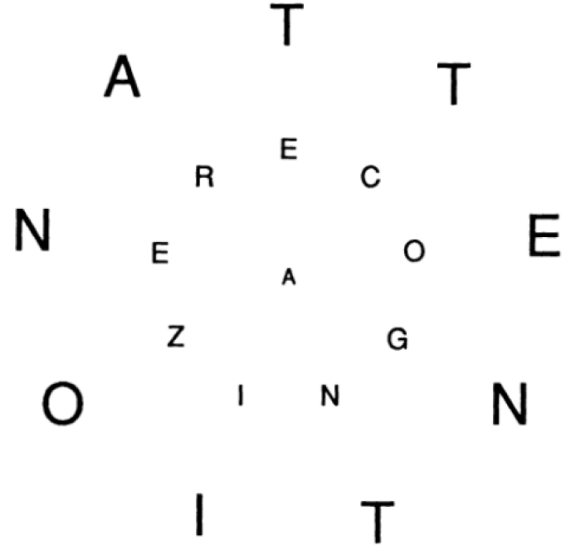
Figure 1.1: **The information pyramid of human vision.** Anderson and Van Essen describe visual processing as a sequence of information bottlenecks: retinal images contain more than 10^{10} bits/sec, the optic nerve transmits roughly 3×10^6 bits/sec, attentive scrutiny admits less than 10^4 bits/sec, and long-term memory stores only about 1–10 bits/sec [3]. The reduction from retinal images to attentive scrutiny alone is on the order of $10^6\times$, illustrating the compression required for efficient visual intelligence.

Can we borrow similar mechanisms to make vision models more efficient and scalable to higher spatiotemporal fidelity? This thesis focuses on acuity-based methods for improving the efficiency of vision models and making them more scalable to visual inputs with higher spatiotemporal fidelity. Specifically, we present several pieces of work that discuss the approach and benefit of scaling spatiotemporal resolution for vision models.

The first work asks whether scaling spatial resolution is actually necessary. It introduces



(a) The acuity bottleneck: recognizability changes sharply with eccentricity, reflecting the nonuniform spatial resolution of human vision. Humans constantly move their eye gaze to land on the most important spatial regions at their fovea and process them at high resolution.



(b) The attention bottleneck: even after the eye gazes at a fixed position, one can still dispatch the attention to different areas in the scene. For example, in this figure, when the eye is fixated on the center letter “A”, one can still choose to attend to different letters on the periphery.

Figure 1.2: Two bottlenecks in human visual information processing. The acuity bottleneck arises because recognizability decreases with eccentricity, so humans move their gaze to bring important regions onto the high-resolution fovea. The attention bottleneck remains even at a fixed gaze location: attention can still be allocated to different parts of the scene, including peripheral regions, for detailed processing.

a simple approach to scaling spatial resolution called scaling on scales, or S^2 , which runs a frozen smaller vision model over multiple image scales by applying it to tiles of larger images and combining the resulting representations. This approach shows that scaling the spatial resolution of smaller models can match or exceed the performance of larger models across various tasks (including classification, dense prediction, multimodal language-model benchmarks, and robotic manipulation), verifying the importance of scaling spatial resolution of vision models. The work also analyzes when scaling image scales is preferable to scaling model size, and whether smaller models with larger spatial resolution can learn the same representations as larger models.

The second work explores how to pre-train vision encoders at high resolution (e.g., 4K resolution). It introduces PS3, a scale-selective approach that extends CLIP-style vision-language pre-training to 4K-resolution images with near-constant cost. Rather than contrasting global image representations with global captions, PS3 learns from local high-resolution regions

paired with local detailed captions. This enables VILA-HD, a multimodal language model that can process high-resolution image details efficiently by selecting relevant regions based on a user prompt. We show that by scaling the pre-training resolution to 4K, the resulted model achieves significantly improved performance on VQA benchmarks that require high resolution, including 4KPro, a self-collected benchmark designed to evaluate visual question answering that truly requires 4K-resolution perception.

The third work extends selective visual processing from images to videos. It introduces AutoGaze, a lightweight module that autoregressively selects a minimal set of multi-scale video patches before they are processed by a vision transformer or multimodal language model. By removing redundant patches early, AutoGaze accelerates video understanding and enables models to scale along multiple axes of temporal fidelity: higher resolution, higher frame rate, and longer duration. We show that this effectively improves the performance on various video understanding benchmarks, especially on HLVID, a self-collected benchmark that requires long-form 4K video understanding.

Together, these work argues for a view of visual scaling in which computation is allocated adaptively rather than uniformly. The methods developed in this thesis show that efficient scaling to high-fidelity vision is possible by attending to where information lives before paying the full cost of processing it.

Chapter 2

When Do We Not Need Larger Vision Models?

Scaling up the size of vision models has been the *de facto* standard to obtain more powerful visual representations. In this work, we discuss the point beyond which larger vision models are *not* necessary. First, we demonstrate the power of **Scaling on Scales (S^2)**, whereby a pre-trained and frozen smaller vision model (*e.g.*, ViT-B or ViT-L), run over multiple image scales, can outperform larger models (*e.g.*, ViT-H or ViT-G) on classification, segmentation, depth estimation, Multimodal LLM (MLLM) benchmarks, and robotic manipulation. Notably, S^2 achieves state-of-the-art performance in detailed understanding of MLLM on the V^* benchmark, surpassing models such as GPT-4V. We examine the conditions under which S^2 is a preferred scaling approach compared to scaling on model size. While larger models have the advantage of better generalization on hard examples, we show that features of larger vision models can be well approximated by those of multi-scale smaller models. This suggests most, if not all, of the representations learned by current large pre-trained models can also be obtained from multi-scale smaller models. Our results show that a multi-scale smaller model has comparable learning capacity to a larger model, and pre-training smaller models with S^2 can match or even exceed the advantage of larger models. We release a Python package that can apply S^2 on any vision model with *one line of code*: https://github.com/bfshi/scaling_on_scales.

2.1 Introduction

Scaling up model size has been one of the key drivers of recent progress in various domains of artificial intelligence, including language modeling [24, 153, 197], image and video generation [23, 107, 159, 247], *etc.* Similarly, for visual understanding, larger models have consistently shown improvements across a wide range of downstream tasks given sufficient pre-training data [47, 147, 185, 255]. This trend has led to the pursuit of gigantic models with up to tens of billions of parameters as a default strategy for achieving more powerful

visual representations and enhanced performance on downstream tasks [47, 55, 66, 183].

In this work, we revisit the question: *Is a larger model always necessary for better visual understanding?* Instead of scaling up model size, we consider scaling on the dimension of image scales—which we call **Scaling on Scales (S^2)**. With S^2 , a pre-trained and frozen smaller vision model (*e.g.*, ViT-B or ViT-L) is run on multiple image scales to generate a multi-scale representation. We take a model pre-trained on single image scale (*e.g.*, 224^2), interpolate the image to multiple scales (*e.g.*, 224^2 , 448^2 , 672^2), extract features on each scale by splitting larger images into sub-images of the regular size (224^2) and processing each separately before pooling them and concatenating with features from the original representation (Figure 2.1).

Surprisingly, from evaluations on visual representations of various pre-trained models (*e.g.*, ViT [62], DINOv2 [147], OpenCLIP [47], MVP [156]), we show that smaller models with S^2 scaling consistently outperform larger models on classification, semantic segmentation, depth estimation, MLLM benchmarks, and robotic manipulation, with significantly fewer parameters ($0.28\times$ to $0.07\times$) and comparable GFLOPS. Remarkably, by scaling up image scale to 1008^2 , we achieve state-of-the-art performance in MLLM visual detail understanding on V^* benchmark [227], surpassing open-source and even commercial MLLMs like Gemini Pro [189] and GPT-4V [1].

We further examine conditions under which S^2 is a preferred scaling approach compared to model size scaling. We find that while smaller models with S^2 achieve better downstream performance than larger models in many scenarios, larger models can still exhibit superior generalization on hard examples. This prompts an investigation into whether smaller models can achieve the same level of generalization capability as larger ones. Surprisingly, we find that the features of larger models can be well approximated by multi-scale smaller models through a single linear transform, which means smaller models should have at least a similar learning capacity of their larger counterparts. We hypothesize that their weaker generalization stems from being pre-trained with single image scale only. Through experiments of ImageNet-21k pre-training on ViT, we show that pre-training with S^2 scaling improves the generalizability of smaller models, enabling them to match or even exceed the advantages of larger models.

2.2 Related Work

Multi-scale representation has been a common technique to recognize objects in a scale-invariant way since the era of feature engineering [54, 58, 134] and is later introduced into convolutional neural networks [121, 166, 193, 203] to extract features with both high-level semantics and low-level details. It has become a default test-time augmentation method for tasks such as detection and segmentation [51, 230], albeit at the cost of significantly slower inference speeds and typically limited image scales (up to $2\times$). Along with recent progress in vision transformers (ViT), variants of multi-scale ViTs [30, 67, 110, 239] as well as hierarchical ViTs [129, 168] have been proposed. However, these studies have not explored multi-scale representation as a general scaling approach as they usually design special architectures and are not applicable to common pre-trained vision models.

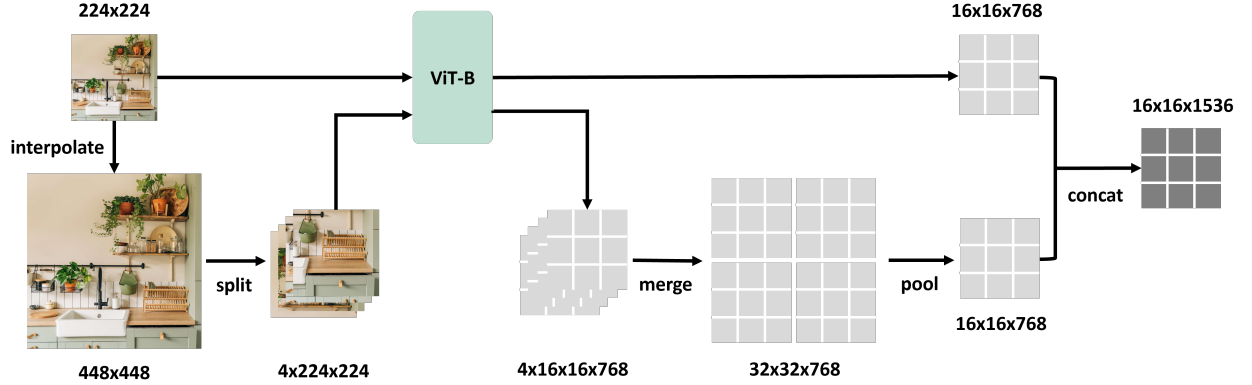


Figure 2.1: **S²-Wrapper** is a simple mechanism that extends any pre-trained vision model to multiple image scales in a parameter-free manner. Taking ViT-B as an example, S²-Wrapper first interpolates the input image to different scales (*e.g.*, 224^2 and 448^2) and splits each into several sub-images of the same size as the default input size ($448^2 \rightarrow 4 \times 224^2$). For each scale, all sub-images are fed into the same model and the outputs (*e.g.*, 4×16^2) are merged into feature map of the whole image (32^2). Feature maps of different scales are average-pooled to the original spatial size (16^2) and concatenated together. The final multi-scale feature has the same spatial shape as single-scale feature while having higher channel dimension (*e.g.*, 1536 *vs.* 768).

Scaling Vision Models. Training models with an increasing number of parameters has been the default approach to obtaining more powerful representations for visual pre-training [62, 90, 132, 147]. Previous research has studied how to optimally scale up vision models in terms of balancing model width, depth, and input resolution [16, 59, 185, 186, 223], although they are usually limited to convolutional networks or even specific architectures such as ResNet [90]. Recent work also explores model size scaling of vision transformers in various settings [13, 47, 55, 165, 255]. Others have incorporated high-resolution images into pre-training [70, 129, 132, 147], although the maximum resolution typically does not exceed 512^2 due to unbearable demands of computational resources. Hu *et al.* [95] study scaling on image scales through adjusting patch size for Masked Autoencoder (MAE) [92] where scaling is only applied on pre-training but not on downstream tasks.

2.3 The Power of Scaling on Scales

As an alternative to the conventional approach of scaling model size, we show the power of Scaling on Scales (S²), *i.e.*, keeping the same size of a pre-trained model while running it on more and more image scales. From case studies on image classification, semantic segmentation, depth estimation, Multimodal LLMs, as well as robotic manipulation, we observe that S² scaling on a smaller vision model (*e.g.*, ViT-B or ViT-L) often gives comparable or better

performance than larger models (*e.g.*, ViT-H or ViT-G), suggesting S^2 is a competitive scaling approach. In the following, we first introduce S^2 -Wrapper, a mechanism that extends any pre-trained frozen vision model to multiple image scales without additional parameters (Section 2.3). We then compare S^2 scaling and model size scaling in Section 2.3 - 2.3.

Scaling Pre-Trained Vision Models to Multiple Image Scales

We introduce S^2 -Wrapper, a parameter-free mechanism to enable multi-scale feature extraction on any pre-trained vision model. Regular vision models are normally pre-trained at a single image scale (*e.g.*, 224^2). S^2 -Wrapper extends a pre-trained model to multiple image scales (*e.g.*, 224^2 , 448^2) by splitting different scales of images to the same size as seen in pre-training. Specifically, given the image at 224^2 and 448^2 scales, S^2 -Wrapper first divides the 448^2 image into four 224^2 sub-images, which along with the original 224^2 image are fed to the same pre-trained model. The features of four sub-images are merged back to the large feature map of the 448^2 image, which is then average-pooled to the same size as the feature map of 224^2 image. Output is the concatenation of feature maps across scales. The whole process is illustrated in Figure 2.1. Note that instead of directly using the 448^2 resolution image, we obtain the 448^2 image by interpolating the 224^2 image. This is to make sure no additional high-resolution information is introduced so we can make a fair comparison with model size scaling which never sees the high-resolution image. For practitioners, directly using the high-resolution image is recommended.

There are several key designs that make S^2 -Wrapper efficient, effective, and easy to scale: (i) splitting the large image into small sub-images, instead of directly running on the whole large image, avoids quadratic computation complexity in self-attention and prevents performance degradation caused by position embedding interpolation [21], (ii) processing individual sub-images instead of using window attention allows using a pre-trained model that does not support window attention and avoids training additional parameters (*e.g.*, relative position embedding) from scratch, (iii) interpolating the large feature map into the regular size makes sure the number of output tokens stays the same, preventing computational overhead in downstream applications such as MLLMs. Note that we do not claim the novelty of extracting multi-scale features. Instead, we only choose the simplest algorithm design and study its scaling property.

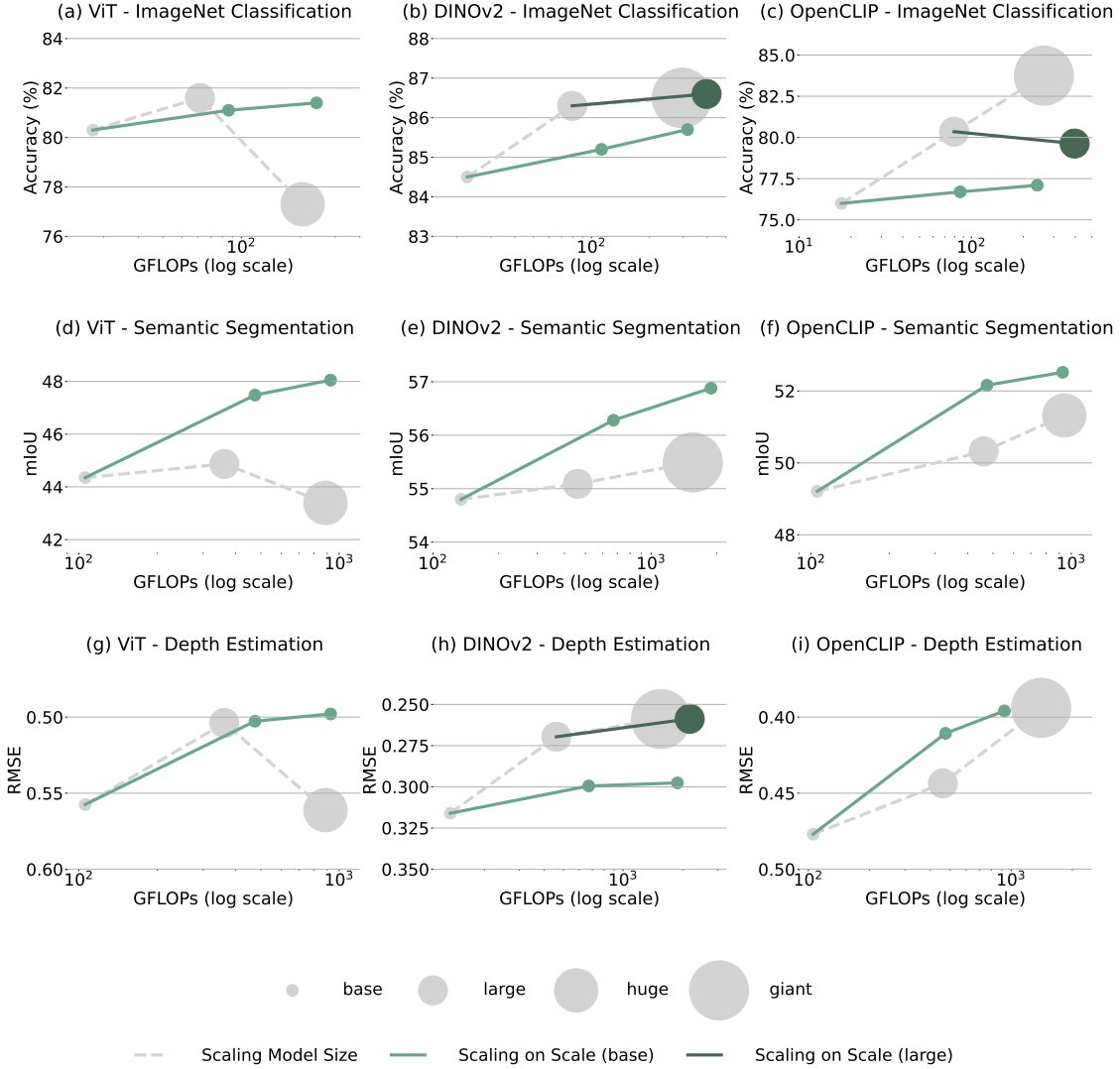


Figure 2.2: **Comparison of S^2 scaling and model size scaling** on three models (ViT, DINOv2, and OpenCLIP) and three tasks (ImageNet classification, semantic segmentation, and depth estimation). For each model and each task, we test base, large, and huge/giant models for model size scaling (plotted in gray curve). For S^2 scaling (plotted in green curve), we test three sets of scales from single-scale (1x) to multi-scale (up to 3x), and we adjust each set of scale so that it matches the GFLOPs of the respective model size. Note that for specific models and tasks, we test S^2 scaling on both base and large models (plotted in light green and dark green curves separately). We can see that in (a), (d), (e), (f), (g), and (i), the base model with S^2 scaling already achieves comparable or better performances than larger models with similar GFLOPs and much smaller model size. For (b), (h), S^2 scaling from the large model is comparable with the giant model, again with similar GFLOPs and fewer parameters. The only failure case is (c), where S^2 scaling on either base or large models does not compete with model size scaling.

Scaling on Image Scales Can Beat Scaling on Model Size

S^2 -Wrapper enables S^2 scaling, *i.e.*, keeping the same size of a pre-trained model while getting more and more powerful features by running on more and more image scales. Here we compare the scaling curve of S^2 to the regular approach of scaling up model size and show that S^2 scaling is a competitive, and in some cases, preferred scaling approach. To get a holistic analysis of two scaling approaches, we test their scaling curves on three representative tasks (image classification, semantic segmentation, and depth estimation) which correspond to the three dimensions of vision model capability [138], as well as on MLLMs and robotic manipulation which reflect the comprehensive ability of visual understanding.

Case study: image classification, semantic segmentation, and depth estimation.

We use ImageNet [167], ADE20k [269], and NYUv2 [177] datasets for each task, respectively. We test on three families of pre-trained models (ViT [62], DINOv2 [147], and OpenCLIP [47]), spanning pre-training with different datasets (ImageNet-21k, LVD-142M, LAION-2B) and different pre-training objectives (supervised, unsupervised, and weakly-supervised). To see if the same observation holds for convolutional networks, we also test on ConvNeXt [132]. To fairly evaluate the representation learned from pre-training, we freeze the backbone and only train the task-specific head for all experiments. We use a single linear layer, Mask2former [45], and VPD depth decoder [267] as decoder heads for three tasks, respectively. For model size scaling, we test the performance of base, large, and huge or giant size of each model on each task. For S^2 scaling, we test three sets of scales including (1x), (1x, 2x), (1x, 2x, 3x). For example, for ViT on ImageNet classification, we use three sets of scales: (224^2) , $(224^2, 448^2)$, and $(224^2, 448^2, 672^2)$, which have the comparable GFLOPs as ViT-B, ViT-L, and ViT-H, respectively. Note that the scales for specific models and tasks are adjusted to match the GFLOPS of respective model sizes.

The scaling curves are shown in Figure 2.2. We can see that in six out of nine cases ((a), (d), (e), (f), (g), (i)), S^2 scaling from base models gives a better scaling curve than model size scaling, outperforming large or giant models with similar GFLOPs and much fewer parameters. In two cases ((b) and (h)), S^2 scaling from base models has less competitive results than large models, but S^2 scaling from large models performs comparatively with giant models. The only failure case is (c) where both base and large models with S^2 scaling fail to compete with the giant model. Note that ViT-H is worse than ViT-L on all three tasks possibly due to the sub-optimal pre-training recipe [180]. We observe that S^2 scaling has more advantages on dense prediction tasks such as segmentation and depth estimation, which matches the intuition that multi-scale features can offer better detailed understanding which is especially required by these tasks. For image classification, S^2 scaling is sometimes worse than model size scaling (*e.g.*, multi-scale DINOv2-B *vs.* DINOv2-L). We hypothesize this is due to the weak generalizability of the base model feature because we observe that the multi-scale base model has a lower training loss than the large model despite the worse performance, which indicates overfitting. In Section 2.4 we show that this can be fixed by pre-training with S^2 scaling as well.

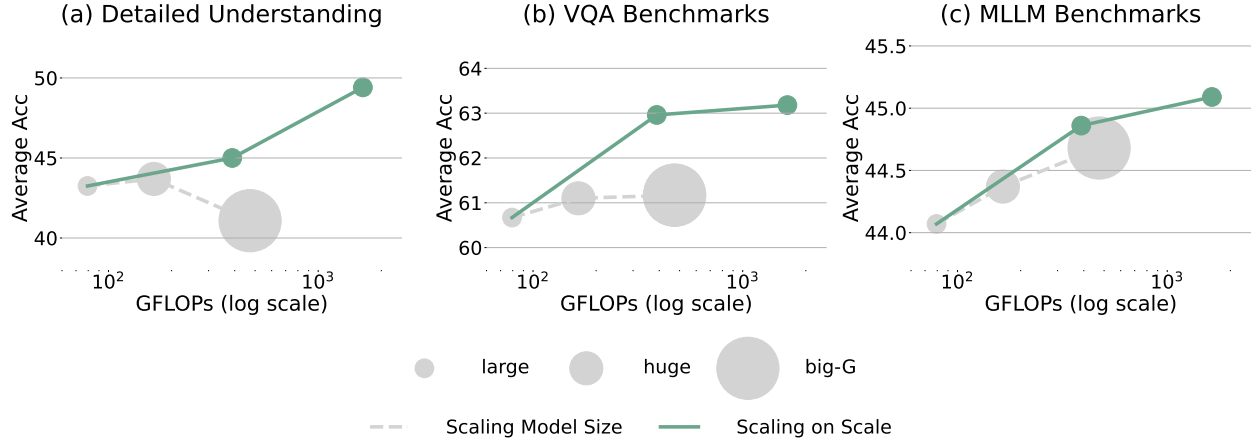


Figure 2.3: **Comparison of S^2 scaling and model size scaling on MLLM.** S^2 scaling has comparable or better scaling curve than model size scaling on all three types of benchmarks. Using large image scales consistently gives better performance while using larger model can degrade model performance in certain cases.

Case study: Multimodal LLMs. We compare S^2 scaling and model size scaling on MLLMs. We use a LLaVA [123]-style model where LLM is a Vicuna-7B [48] and the vision backbone is OpenCLIP. We keep the same LLM and only change the vision backbone. For model size scaling, we test vision model sizes of large, huge, and big-G. For S^2 scaling, we keep the large-size model and test scales of (224^2) , $(224^2, 448^2)$, and $(224^2, 448^2, 896^2)$. For all experiments, we keep the vision backbone frozen and only train a projector layer between the vision feature and LLM input space as well as a LoRA [94] on LLM. We follow the same training recipe as in LLaVA-1.5 [122]. We evaluate three types of benchmarks: (i) visual detail understanding (V^* [227]), (ii) VQA benchmarks (VQAv2 [84], TextVQA [178], VizWiz [88]), and (iii) MLLM benchmarks (MMMU [253], MathVista [135], MMBench [127], SEED-Bench [112], MM-Vet [250]).

A comparison of the two scaling approaches is shown in Figure 2.3. We report the average accuracy on each type of benchmarks. We can see that on all three types of benchmarks, S^2 scaling on large-size models performs better than larger models, using similar GFLOPs and much smaller model sizes. Especially, scaling to 896^2 improves the accuracy of detailed understanding by about 6%. On all benchmarks, larger image scales consistently improve performance while bigger models sometimes fail to improve or even hurt performance. These results suggest S^2 is a preferable scaling approach for vision understanding in MLLMs as well.

We also observe that LLaVA-1.5, when equipped with S^2 scaling, is already competitive or better than state-of-the-art open-source and even commercial MLLMs. Results are shown in Table 2.1. Here we use OpenAI CLIP [154] as the vision model for fair comparison. On visual detail understanding, LLaVA-1.5 with S^2 scaling outperforms all other open-source MLLMs as well as commercial models such as Gemini Pro and GPT-4V. This is credited to the highly fine-grained features we are able to extract by scaling image resolution to



Q: What is the color of the water bottle?

GPT-4V:
The water
bottle on the
ground is **blue**.

LLaVA-1.5:
The color of
the water
bottle is **blue**.

LLaVA-1.5-S²:
The color of
the water
bottle is **red**.



Q: What is the color of the cart?

GPT-4V:
The color of
the cart in the
image is **green**.

LLaVA-1.5:
The color of
the cart is **gray**.

LLaVA-1.5-S²:
The color of
the cart is
green.

Figure 2.4: LLaVA-1.5 with S² scaling is able to recognize extremely fine-grained details in an image, *e.g.*, the color of a water bottle which lives in only 23×64 pixels of a 2250×1500 image.

1008². A qualitative example is shown in Figure 2.4. We can see that LLaVA-1.5 with S² is able to recognize an extremely small object that only takes 23×64 pixels in a 2250×1500 image and correctly answer the question about it. In the meantime, both GPT-4V and LLaVA-1.5 fail to give the correct answer. On VQA and MLLM benchmarks, S² consistently improves the model performance as well, especially on benchmarks such as TextVQA which requires understanding of the fine details. Note that the improvement on certain MLLM benchmarks such as MathVista is not as significant as others, which is probably because these benchmarks require strong mathematical or reasoning capabilities which are not achievable by only improving vision but require stronger LLMs as well. In contrast to previous experiments, here we directly use the high-resolution image instead of interpolating from the low-resolution image in order to compare with the state of the arts. Note that despite the large image scale, we keep the same number of image tokens as baseline LLaVA-1.5 since we interpolate the feature map of the large-scale images to the same size as that of the original image (see Section 2.3). This makes sure the context length (and thus the computational cost) of LLM does not increase when using larger image scales, allowing us to use much higher resolution than the baselines.

Table 2.1: **Results on MLLM.** We evaluate three types of benchmarks: visual detail understanding (V^* [227]), VQA benchmarks (VQAv2 [84], TextVQA [178], VizWiz [88]), and MLLM benchmarks (MMMU [253], MathVista [135], MMBench [127], SEED-Bench [112], MM-Vet [250]). Notably, S^2 significantly improves the detailed understanding capability on V^* benchmark, outperforming commercial models such as GPT-4V.

Model	Res.	#Tok	Visual Detail		VQA Benchmarks			MLLM Benchmarks				
			V _{Att} [*]	V _{Spa} [*]	VQA ^{v2}	VQA ^T	Viz	MMMU	Math	MMB	SEED	MMVet
<i>Commercial or proprietary models</i>												
GPT-4V [1]	-	-	51.3	60.5	77.2	78.0	-	56.8	49.9	75.8	71.6	67.6
Gemini Pro [189]	-	-	40.9	59.2	71.2	74.6	-	47.9	45.2	73.6	70.7	64.3
Qwen-VL-Plus [192]	-	-	-	-	-	78.9	-	45.2	43.3	-	-	-
<i>Open-source models</i>												
InstructBLIP-7B [53]	224	-	25.2	47.4	-	50.1	34.5	-	-	36.0	-	26.2
QwenVL-7B [10]	448	1024	-	-	78.8	63.8	35.2	-	-	38.2	-	-
QwenVL-Chat-7B [10]	448	1024	-	-	78.2	61.5	38.9	-	-	60.6	-	-
CogVLM-Chat [208]	490	1225	-	-	82.3	70.4	-	41.1	34.5	77.6	72.5	51.1
LLaVA-1.5-7B [122]	336	576	43.5	56.6	78.5	58.2	50.0	36.2	25.2	64.3	65.7	30.5
LLaVA-1.5-7B-S ²	1008	576	51.3	61.8	80.0	61.0	50.1	37.7	25.3	66.2	67.9	32.4
LLaVA-1.5-13B [122]	336	576	41.7	55.3	80.0	61.3	53.6	36.4	27.6	67.8	68.2	35.4
LLaVA-1.5-13B-S ²	1008	576	50.4	63.2	80.9	63.1	56.0	37.4	27.8	67.9	68.9	36.4

Case study: robotic manipulation. We compare S^2 and model size scaling on a robotic manipulation task of cube picking. The task requires controlling a robot arm to pick up a cube on the table. We train a vision-based end-to-end policy on 120 demos using behavior cloning, and evaluate the success rate of picking on 16 randomly chosen cube positions, following the setting in [155]. We use MVP [156] as the pre-trained vision encoder to extract visual features which are fed to the policy. To compare S^2 and model size scaling, we evaluate base and large models with single scale of (224^2), as well as a multi-scale base model with scales of (224^2 , 448^2). Results are shown in Figure 2.5. Scaling from base to large model improves the success rate by about 6%, while scaling to larger image scales improves the success rate by about 20%. This demonstrates the advantage of S^2 over model size scaling on robotic manipulation tasks as well.

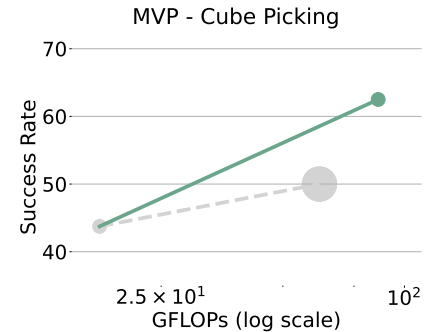


Figure 2.5: **S^2 vs. model size scaling on cube picking task.** S^2 scaling on base-size model improves the success rate by about 20%.

The Sweet Spot Between Model Size Scaling and S^2 Scaling

While S^2 scaling outperforms model size scaling on a wide range of downstream tasks, a natural question arises: on which model size should we perform S^2 scaling? We show that

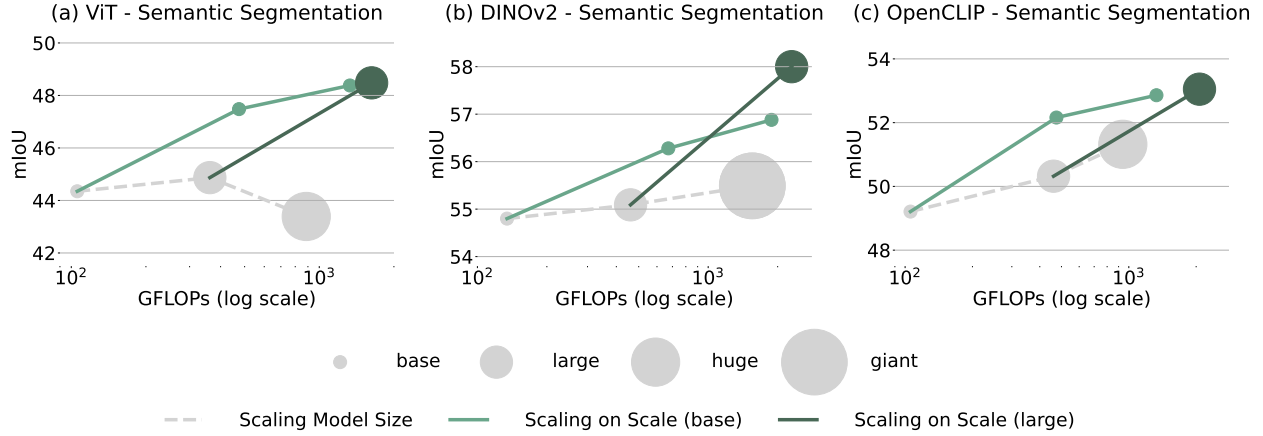


Figure 2.6: **Which model size should we scale up image scales on?** The answer varies for different pre-trained models. For ViT and OpenCLIP, S^2 scaling from base or large model gives similar performances under computation budget beyond the huge-size model while the former performs better under similar GFLOPs as the large-size model. For DINOv2, S^2 scaling from the large size model has better performance than scaling from base size, especially under the same level of computation budget as the giant-size model.

it depends on different pre-trained models. For certain models, S^2 scaling from a large-size model gives an even better scaling curve when S^2 scaling from base model already beats larger models. As an example, we compare S^2 scaling from base and large models on semantic segmentation for ViT, DINOv2, and OpenCLIP. Results are shown in Figure 2.6. We can see that for ViT and OpenCLIP, S^2 scaling from base models is better than from large models when the amount of computation is less than that of the huge-size models. These two curves eventually converge after going beyond the GFLOPs of the huge models. This means S^2 scaling from large models has no significant benefit than from base models. On the other hand, for DINOv2 we observe a clear advantage for S^2 scaling from the large model. When reaching the same level of GFLOPs as the giant-size model, S^2 scaling from the large model beats S^2 scaling from the base model by about 1 mIoU. These results indicate the optimal balancing between model size scaling and S^2 scaling varies for different models.

2.4 The (Non)Necessity of Scaling Model Size

Results from Section 2.3 suggest S^2 is a preferred scaling approach than model size scaling for various downstream scenarios. Nevertheless, larger vision models seem still necessary in certain cases (such as Figure 2.2(c)) where S^2 scaling cannot compete with model size scaling. In the following, we first study the advantage of larger models and show they usually generalize better on rare or hard instances than multi-scale smaller models (Section 2.4). Then, we explore if smaller models with S^2 scaling can achieve the same capability. We find

that features of larger models can be well approximated by features of multi-scale smaller models, which means smaller models can learn what larger models learn to a large extent (Section 2.4). Based on this observation, we verify that multi-scale smaller models have similar capacity as larger models, and pre-training with S^2 scaling endows smaller models with similar or better generalization capability than larger models (Section 2.4).

Larger Models Generalize Better on Hard Examples

We use image classification as a testbed to understand the advantage of larger models. We conduct a qualitative analysis of what kinds of images are recognized better by a larger model but not by using larger image scales. Specifically, we find samples in ImageNet that a larger model (ViT-L) improves the most over a smaller model (ViT-B) but a multi-scale model (ViT-B- S^2) fails to improve, as shown in Figure 2.7. For each sample, we also find two easy samples (which two models both recognize correctly) from the same class as a comparison. We can see that there are mainly two types of images that larger models have advantages on. The first type is rare samples. For example, a television or a flute but in the form of a sculpture instead of regular ones (Figure 2.7(a)). Larger models have larger capacity to learn to classify these rare examples during pre-training. The second type (Figure 2.7(b)) is ambiguous examples, where the object can belong to either category (*e.g.*, lotion and soap dispenser), or there are two categories co-existing in the same image and both labels should be correct (*e.g.*, airship and traffic light). In this case, despite multiple correct labels, the large model is able to remember the label presented in the dataset during pre-training. While the second type is due to the flawed labeling process of ImageNet which makes it an unfair comparison and does not imply any disadvantage of multi-scale smaller models [17, 144], the first type indicates larger model can generalize better on rare or hard cases.

Can Smaller Models Learn What Larger Models Learn?

Is the advantage of larger models due to some unique representation they have learned that smaller models cannot learn? We design experiments to study how much of the representation of larger models is also learned by multi-scale smaller models. Surprisingly, our preliminary results suggest that *most, if not all, of the representation of larger models is also learned by multi-scale smaller models.*

To quantify how much of the representation of a larger model (*e.g.*, ViT-L) is also learned by a multi-scale smaller model (*e.g.*, ViT-B- S^2), we adopt a reconstruction-based evaluation, *i.e.*, we train a linear transform to reconstruct the representation of a larger model from that of a multi-scale smaller model. Intuitively, low reconstruction loss means the representation of larger model can be equivalently learned by the multi-scale smaller model (through a linear transform) to a large extent. More formally, the reconstruction loss reflects the mutual information between two sets of representations. If we use MSE loss for reconstruction, the mutual information equals $I = -\log(l/l_0)$, where l is the reconstruction loss and l_0 is the loss of vanilla reconstruction where the large model representation is reconstructed by a dummy

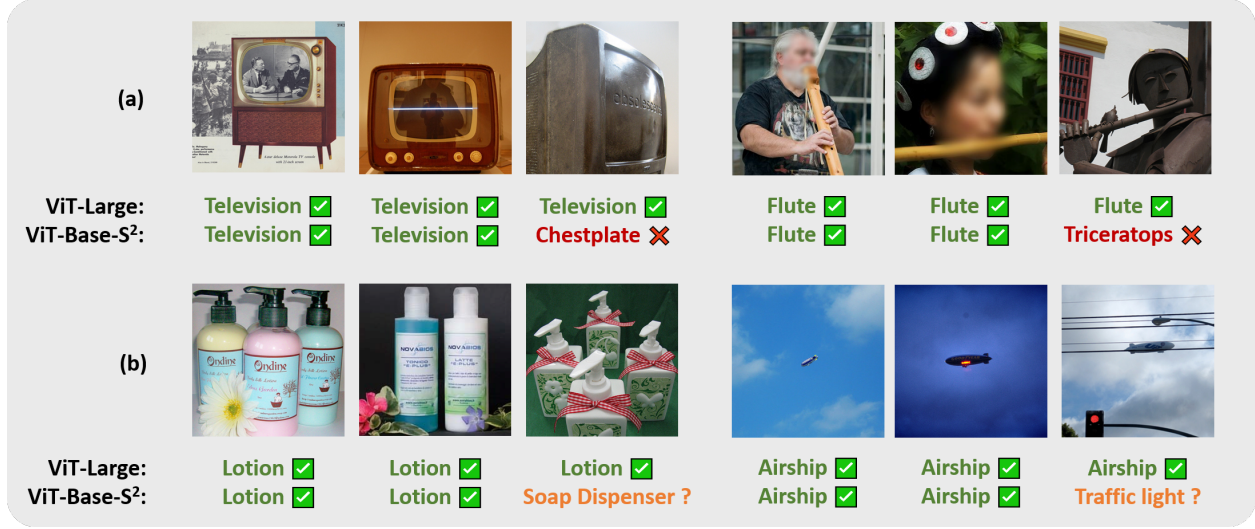


Figure 2.7: **Types of samples that ViT-L improves the most but ViT-B-S² does not.** (a) **Rare cases.** These samples clearly belong to the class but are hard to classify due to the rare appearance (*e.g.*, sculptures of television and flute). (b) **Ambiguous cases.** These samples have ambiguous labels. For example, the lotion could also be soap dispenser due to their high similarity, or the label could be either airship or traffic light when these two objects co-exist.

vector. This quantifies how much information in the larger model representation is also contained in the multi-scale smaller model. We use a linear transform for reconstruction to (i) account for operations that keep the representation equivalence (*e.g.*, channel permutation), (ii) measure the information that is useful for downstream tasks considering the task decoders are usually light-weight modules such as a single linear layer [236].

Moreover, in practice we find the reconstruction loss is usually nowhere near zero. We hypothesize this is because part of the feature is *non-reconstructable* by nature, *i.e.*, feature that is not relevant to the pre-training task and is learned due to randomness in weight initialization, optimization dynamics, *etc.*, thus cannot be reconstructed from another model’s feature. To this end, we use an even larger (*e.g.*, ViT-G) model to reconstruct the large model features as a comparison. Its reconstruction loss and corresponding mutual information are denoted by l^* and $I^* = -\log(l^*/l_0)$. If we assume that, when pre-trained on the same task and the same dataset, any task-relevant feature learned by a smaller model can also be learned by a larger model, then all the useful features in a large-size model should be reconstructable by a huge or giant model as well. This means I^* , the amount of information reconstructed from a huge or giant model, should serve as an *upper bound* of I . We empirically find this is indeed the case (see below). Therefore, we use the reconstruction ratio I/I^* to measure how much representation in a larger model is also learned by a multi-scale smaller model.

We evaluate three classes of models: (i) ViT [62] pre-trained on ImageNet-21k, (ii) OpenCLIP [47] pre-trained on LAION-2B, and (iii) MAE [92] pre-trained on ImageNet-1k.

Reconstruction loss is averaged over all output tokens and is evaluated on ImageNet-1k. Results are shown in Table 2.2. Compared to base models, we observe that multi-scale base models consistently have lower loss and reconstructs more information of the large model representation (*e.g.*, 0.521 *vs.* 0.440 for ViT). More interestingly, we find that the amount of information reconstructed from a multi-scale base model is usually close to that of a huge or giant model, although sometimes slightly lower but never exceeding by a large margin. For example, while OpenCLIP-Base reconstructs 92.7% of the information, the multi-scale base model can reconstruct 99.9%. For other models, the reconstruction ratio of Base-S² model is usually close to 100% while never exceeding by more than 0.5%. This implies (i) huge/giant models are indeed a valid upper bound of feature reconstruction, and (ii) most part of the feature of larger models is also learned by multi-scale smaller models. The only exception is when we reconstruct OpenCLIP-Huge feature, the reconstruction ratio is 88.9%. Although it's not near 100%, it is still significantly better than the base-size model which means at least a large part of the huge model feature is still multi-scale feature. These results imply smaller models with S² scaling should have at least a similar level of capacity to learn what larger models learn. On the other hand, we also notice that there exists a gap between train and test set, *i.e.*, the reconstruction ratio on test set can be lower than train set (*e.g.* 96.3% *vs.* 99.9% on OpenCLIP-L). We hypothesize this is because we only apply multi-scale after pre-training and the base model feature pre-trained on single image scale only has weaker generalizability.

Pre-Training With S² Makes Smaller Models Better

Given that most of the representation larger models have learned is also learned by multi-scale smaller models, we conjecture smaller models with S² scaling have at least similar capacity as larger models. Since larger capacity allows memorizing more rare and atypical instances during pre-training when given sufficient data and thus improves generalization error [15, 46, 74, 75, 136], we further speculate smaller models can achieve similar or even better generalizability than larger models if pre-trained with S² scaling as well. We verify these in the following.

Multi-scale smaller models have similar capacity as larger models. To measure the model capacity, we use two surrogate metrics: (i) memorization capability, and (ii) training loss on a specific task. For memorization capability, given a dataset (*e.g.*, ImageNet), we regard each image as a separate category and train the model to classify individual images, which requires the model to memorize every single image. The classification loss reflects how well each instance is memorized and thus the model capacity [258]. We adopt the training pipeline from [231]. For training loss, we report classification loss on the training set of ImageNet-1k for DINOv2 and OpenCLIP. Lower loss means the model fits the training data better, which implies a larger model capacity. Results are shown in Table 2.3. For instance memorization, we can see that ViT-B with S² scaling (224² and 448²) has a similar loss as ViT-L. For ImageNet classification, ViT-B-S² has a similar training loss as ViT-L for

Table 2.2: **Reconstructing representation of larger models from representation of regular or multi-scale smaller models.** We test three classes of models (ViT, OpenCLIP, and MAE), and for each class we test base, multi-scale base (Base-S²), and huge or giant model. We report results on both training and test set of ImageNet-1k, and for each we report the reconstruction loss, the amount of information reconstructed, and the percentage of information reconstructed compared to huge or giant model.

Model Class	Target	Source	Train Set			Test Set		
			Loss	Info	Ratio (%)	Loss	Info	Ratio (%)
ViT	Large	Base	0.1100	0.440	82.9%	0.0994	0.524	87.6%
		Base-S ²	0.1040	0.521	98.1%	0.0942	0.601	100.5%
		Huge	0.1033	0.531	100%	0.0944	0.598	100%
MAE	Large	Base	0.0013	7.460	97.3%	0.0010	7.840	96.0%
		Base-S ²	0.0011	7.694	100.3%	0.0009	7.972	97.6%
		Huge	0.001	7.669	100%	0.0008	8.169	100%
OpenCLIP	Large	Base	0.3693	1.495	92.7%	0.3413	1.723	90.7%
		Base-S ²	0.3408	1.611	99.9%	0.3170	1.830	96.3%
		Giant	0.3402	1.613	100%	0.3022	1.900	100%
OpenCLIP	Huge	Base	0.3926	1.407	83.2%	0.4231	1.413	80.8%
		Base-S ²	0.3670	1.504	88.9%	0.3970	1.505	86.0%
		Giant	0.3221	1.692	100%	0.3354	1.749	100%

OpenCLIP, and an even lower loss for DINOv2. These results suggest that multi-scale smaller models have at least comparable model capacity as larger models.

Pre-training with S² makes smaller models better. We evaluate ImageNet classification of a base model scaled with S² either during pre-training or after pre-training. We pre-train the model on ImageNet-21k, using either ViT image classification or DINOv2 as the pre-training objective. We compare models with or without S² during pre-training with single-scale base and large models. Results are shown in Table 2.4. We can see that when the base models are trained with single image scale and only scaled to multiple image scales after pre-training, they have sub-optimal performances compared to the large models, which aligns with our observation in Section 2.3. However, when adding S² scaling into pre-training, the multi-scale base model is able to outperform the large model on ViT. For DINOv2, the base model pre-trained with S² achieves a performance that is significantly improved over the base model pre-trained without S², and is more comparable to the large model. Although it still slightly falls behind the large model, pre-training a large model with S² potentially can give a better scaling curve. These observations confirm our speculation that smaller models pre-trained with S² can match the advantage of larger models.

Table 2.3: **Training loss on instance memorization and image classification.** A base model with S^2 scaling has similar memorization and classification losses, which implies it has at least the same level of model capacity as a large model.

Model	Mem. Loss	Cls. Loss (DINOv2)	Cls. Loss (OpenCLIP)
Base	1.223	3.855	4.396
Large	1.206	3.350	3.735
Base- S^2	1.206	2.921	3.754

Table 2.4: **Pre-training with S^2 .** Applying S^2 on a already pre-trained model has sub-optimal performance, while pre-training with S^2 makes smaller models better.

Model	Pre-train w/ S^2	Acc. (ViT)	Acc. (DINOv2)
Base		80.3	77.6
Large		81.6	81.9
Base- S^2	X	81.1	78.4
Base- S^2	✓	82.4	80.4

2.5 Discussion

In this work, we ask the question *is a larger model always necessary for better visual understanding?* We find that scaling on the dimension of image scales—which we call Scaling on Scales (S^2)—instead of model size usually obtains better performance on a wide range of downstream tasks. We further show that smaller models with S^2 can learn most of what larger models learn, and pre-training smaller models with S^2 can match the advantage of larger models and even perform better. S^2 has a few implications for future work, including **(i) scale-selective processing**, *i.e.*, not every scale at every position in an image contains equally useful features, and depending on image content and high-level task, it is much more efficient to select certain scales to process for each region, which resembles the bottom-up and top-down selection mechanism in human visual attention [100, 172, 268], **(ii) parallel processing of single image**, *i.e.*, in contrast with regular ViT where the whole image is processed together at once, the fact that each sub-image is processed independently in S^2 enables parallel processing of different sub-images for a single image, which is especially helpful for scenarios where latency on processing single large images is critical [263].

Chapter 3

Scaling Vision Pre-Training to 4K Resolution

High-resolution perception of visual details is crucial for daily tasks. Current vision pre-training, however, is still limited to low resolutions (*e.g.*, 378×378 pixels) due to the quadratic cost of processing larger images. We introduce **PS3** that scales CLIP-style vision pre-training to 4K resolution with a *near-constant* cost. Instead of contrastive learning on global image representation, PS3 is pre-trained by selectively processing local regions and contrasting them with local detailed captions, enabling high-resolution representation learning with greatly reduced computational overhead. The pre-trained PS3 is able to both encode the global image at low resolution and selectively process local high-resolution regions based on their saliency or relevance to a text prompt. When applying PS3 to multi-modal LLMs (MLLMs), the resulting model, named **VILA-HD**, significantly improves high-resolution visual perception compared to baselines without high-resolution vision pre-training such as AnyRes and S² while using up to $4.3 \times$ fewer tokens. PS3 also unlocks appealing scaling properties of VILA-HD, including scaling up resolution for free and scaling up test-time compute for better performance. Compared to state of the arts, PS3 and VILA-HD outperform previous vision encoders (*e.g.*, SigLIP2 and Perception Encoder) and MLLMs (*e.g.*, NVILA and Qwen2.5-VL) respectively across multiple benchmarks and achieve better efficiency than latest token pruning approaches. Finally, we find current benchmarks do not require 4K-resolution perception, which motivates us to propose **4KPro**, a new benchmark of image QA at 4K resolution, on which VILA-HD outperforms all previous MLLMs, including a 16.1% improvement over GPT-4o and a 7.5% improvement and $1.67 \times$ speedup over Qwen2.5-VL. Project page: <https://nvlabs.github.io/PS3>.

3.1 Introduction

Vision models with large-scale pre-training [62, 92, 147, 154] have been the workhorses for both fundamental vision tasks [106, 240] and numerous downstream applications [105, 156, 159].

Notably, CLIP-style vision pre-training (*i.e.*, vision-language contrastive learning) such as CLIP [154] and SigLIP [256] have driven significant advancements in multi-modal large language models (MLLMs) by providing general-purpose language-aligned visual understanding in real-world tasks [111, 123, 130].

However, modern vision models including CLIP and SigLIP have one defect: they are *pre-trained with low resolution only*. Visual perception at high resolution (*e.g.*, 4K resolution) is essential in many real-world scenarios such as spotting the stop sign while driving (Figure 3.1(Left)). On the other hand, SigLIP, for example, is only pre-trained with a maximum resolution of 378×378 [256], making it incapable of perceiving visual details and thus unsuitable for assisting humans in everyday tasks. Existing methods propose to run pre-trained vision models at higher resolution in a training-free manner for downstream tasks [42, 124, 175]. However, this prevents the model from leveraging large-scale pre-training data to learn high-quality high-resolution perception, resulting in suboptimal performance [175].

What blocks the current vision pre-training from scaling to higher resolution? The computational cost. The compute spent by the vision model grows quadratically for CNNs and quartically for ViTs with increasing image resolution, making it even infeasible to pre-train over 1K resolution [147, 256].

In this work, we introduce **Pre-training with Scale-Selective Scaling**, or **PS3**, that scales CLIP-style pre-training to *4K resolution* with a *near-constant cost*. The key insight is that, instead of contrasting between global images and captions for the whole high-res image, it suffices to contrast between local regions and local captions to learn detailed feature extraction in high-resolution images. For example, in Figure 3.1(Left, Top Right), to learn to recognize the text on the stop sign, the model only needs to extract the high-resolution feature around the local region of the text and align it with the detailed description about the region. This is analogous to top-down selection mechanism in human vision [27, 268], *i.e.*, one usually focuses on a small portion of the scene that is relevant to the high-level task (*e.g.*, spotting the stop sign). In this way, the model enjoys greatly reduced computational cost by being scale-selective, *i.e.*, selectively processing a small region at fine-grained scale. By disentangling the region size from the image resolution, we are able to scale PS3 pre-training to 4K resolution with a near-constant cost, reducing the pre-training compute by $79\times$ compared to global contrastive learning of SigLIP (Figure 3.1(Top Right)).

The success of PS3 pre-training hinges on addressing three challenges: *data*, *model*, and *algorithm*. First, since the low-resolution image-text pairs used for CLIP pre-training is not suitable for PS3 pre-training, we collect 75M images with up to 4K resolution and build an automatic pipeline to curate 282M pairs of detailed captions and bounding boxes of salient local regions in the images. Second, we design a vision model that can not only extract low-resolution global features, but also select local patches based on image saliency or text queries and process high-resolution details of the patches. Third, we design an algorithm that pre-trains high-resolution perception through contrastive loss between local regions and local captions and pre-trains patch selection with supervision from the curated bounding boxes.

We show PS3 enables high-quality and efficient high-resolution perception in multi-modal

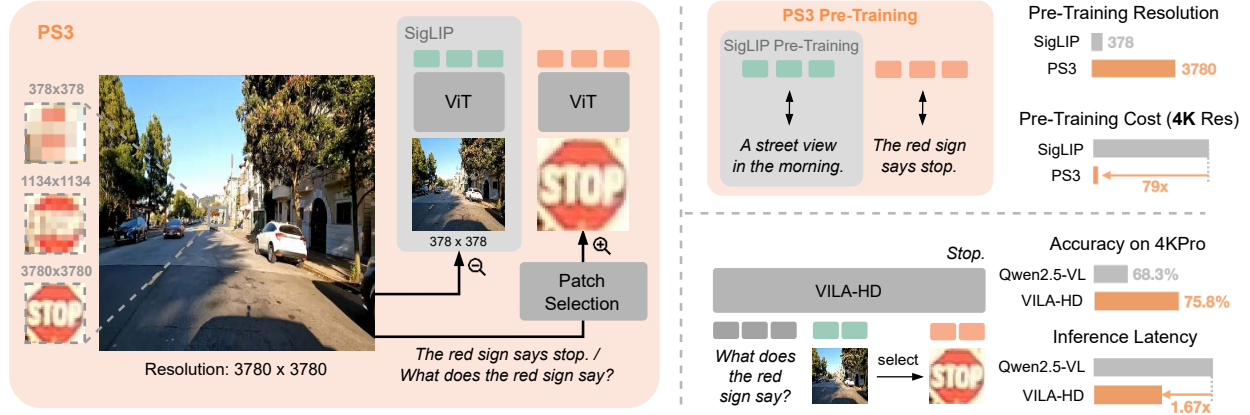


Figure 3.1: **Left:** Regular vision models such as SigLIP processes images at a low resolution (e.g., 378×378 pixels), which is not enough for many daily tasks such as spotting the stop sign while driving. In contrast, PS3 is able to both encode low-res features and efficiently process high-res information of 4K-resolution images via top-down patch selection, *i.e.*, selectively processing relevant patches based on any text prompt. **Top Right:** SigLIP is pre-trained by contrasting global vision features and global captions, which is costly for high-resolution images. PS3 is pre-trained with additional contrast between local high-res features with local captions, enabling pre-training at 4K resolution with $79\times$ less cost than SigLIP. **Bottom Right:** VILA-HD with PS3 as the vision encoder is able to select high-res regions to process based on the user prompt. VILA-HD outperforms state-of-the-art MLLMs such as Qwen2.5-VL [12] by 7.5% on the proposed 4KPro benchmark while achieving $1.67\times$ speedup.

LLMs (MLLMs). Specifically, we train a modern MLLM [130] using pre-trained PS3 as the vision encoder. The resulting MLLM, named **VILA-HD**, is capable of capturing the global image at low resolution and extracting high-resolution details in the local regions selected based on the user prompt. Evaluated on seven benchmarks that require high-res perception, VILA-HD significantly improves the performance over baselines that use either the original low-res SigLIP or approaches such as S^2 [175] and AnyRes [42, 124] that scale up the resolution of SigLIP without high-resolution vision pre-training, while using $4.3\times$ fewer tokens compared to the AnyRes baseline. Additionally, PS3 unlocks several intriguing scaling properties of VILA-HD, for example, scaling up the resolution without extra cost by selecting a constant number of high-res patches and trading more compute for higher performance at test time by selecting larger high-res regions. Comparing to the state of the arts, PS3 outperforms previous vision encoders such as SigLIP2 [198] and Perception Encoder [22] on 23 benchmarks under a controlled setting, and with a more advanced MLLM training recipe, VILA-HD is able to surpass state-of-the-art MLLMs such as NVILA [130] and Qwen2.5-VL [12] on various benchmarks and achieve superior efficiency and performance over latest token pruning approaches [20, 34, 241].

Despite the superior performance of PS3, we find most existing benchmarks do not actually

require 4K resolution. Therefore, we introduce **4KPro**, a benchmark that evaluates visual perception at 4K resolution in four professional use cases including autonomous vehicle, household, gaming, and UI understanding. For each category, 4KPro contains image QA pairs where each question can only be answered under 4K resolution. On 4KPro, PS3 shows a significant improvement of 15% over S² and AnyRes baselines and achieves state-of-the-art results compared to both proprietary and open-sourced MLLMs including GPT-4o [98] and Qwen2.5-VL [12] in terms of both performance and efficiency (Figure 3.1(Bottom Right)).

3.2 PS3: Vision Pre-Training at 4K Resolution

Based on the paradigm of contrastive language-image pre-training (CLIP) [154] which optimizes a contrastive loss between global images and global captions, we propose PS3 which instead optimizes the contrast loss between local regions and detailed captions about the regions (Figure 3.1(Upper Right)). In this way, the model efficiently learns language-aligned detailed representation by being scale-selective, *i.e.*, only processing the selected regions at a fine-grained scale. This detaches the computational cost from the global image size, allowing us to scale up to ultra-high image resolution during pre-training by controlling the size of the local regions.

The scale-selective pre-training requires a redesign of data, model, and algorithm. We first collect 75M high-resolution images with 282M pairs of bounding boxes and captions of salient local regions (Section 3.2). We then design the architecture of PS3 that can both encode low-resolution global images and select local high-resolution patches to process based on image saliency or their relevance to a text prompt (Section 3.2). We finally pre-train PS3 jointly with localized contrastive loss for high-res perception and box supervision for patch selection (Section 3.2).

Pre-Training Data of PS3

To learn fine-grained perception in high-res images through contrastive loss between local regions and captions, we need to collect high-res images together with bounding boxes and captions of local regions in each image. We need to make sure the local regions contain rich details in order for the model to learn fine-grained representation. In this work, we collect 75M high-res images with 282M pairs of bounding boxes and captions for both natural images and document images. An example of the pre-training data is shown in Figure 3.3. The curation of each component of the data is detailed below.

High-resolution images. We collect 75M images with 1K - 4K resolution. These include 38M natural images from DataComp [80] and SA-1B [106] and 37M document images from IDL [19] and PDFa [143].

Local captions and bounding boxes of salient regions for natural images. We propose a pipeline of first detecting salient regions and then generating local captions (Figure 3.2). For saliency detection, inspired by recent work on segmenting everything in an

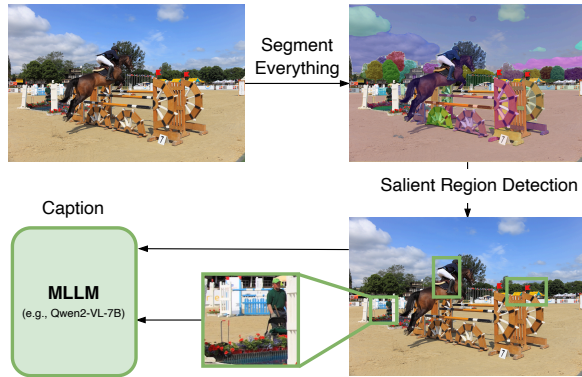


Figure 3.2: **Curation of bounding boxes and captions of salient regions in the pre-training data.** For each high-resolution image, we segment all the masks, detect salient regions with small or dense masks, and use an MLLM to generate captions about the local regions.



Figure 3.3: **Pre-training data example.** Each instance contains an image with resolution up to 4K, bounding boxes of the salient regions in the image, and captions about details in the regions such as text or small objects.

image [106, 213, 266], we first use EfficientViT-SAM [266] to generate segmentation masks of the whole image, and then select local regions containing small or dense masks as salient regions. The intuition is that a local region should contain small or cluttered objects in order to have rich details. For local captions, we use an off-the-shelf MLLM (*e.g.*, Qwen2-VL [206]) as a captioner. Specifically, we zoom in and crop the local region, send it along with the global image to the MLLM, and let it describe the local region based on the global context. This results in 3 - 4 pairs of local captions and bounding boxes per image and 134M pairs in total, with an average box size around 400×400 .

Local captions and bounding boxes for documents. Both IDL [19] and PDF4 [143] provide bounding boxes and OCR results of sentences or words in each PDF. Therefore, we simply sample from these bounding boxes and use the corresponding OCR results as the local captions. We generate 148M pairs of boxes and captions with an average box size around 50×400 .

Global captions. Our pre-training also uses global captions (see Section 3.2). We use the same MLLM captioner to generate global captions for natural images. For document images we do not use any global captions.

Model Design of PS3

We design the model such that given a high-res image, it can 1) extract low-res global features, 2) select local regions based on saliency or their relevance to an input text prompt, and 3) extract high-res features of the selected regions. The whole model can be divided into three

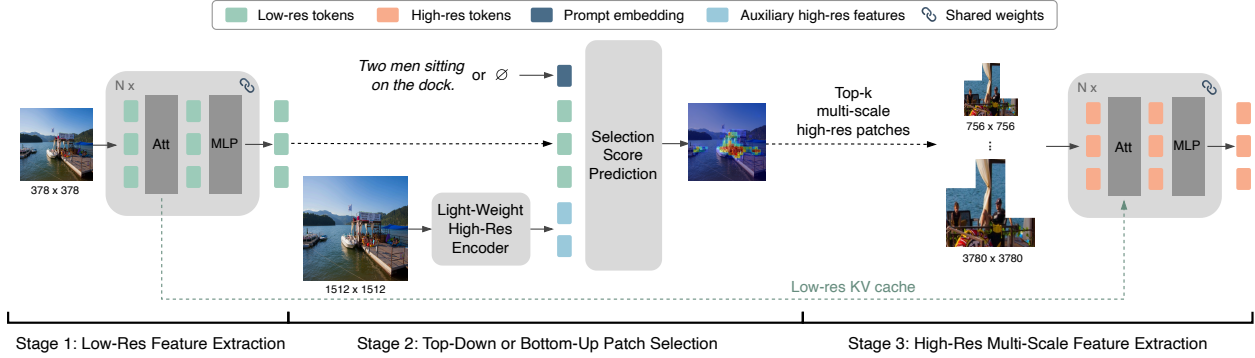


Figure 3.4: **Model architecture of PS3.** The model consists of 3 stages. In Stage 1, the model encodes global low-resolution features. In Stage 2, based on the low-resolution features as well as auxiliary high-resolution features extracted by a light-weight encoder, the model selects local regions that are either relevant to a text prompt (top-down selection) or salient by themselves (bottom-up selection). In Stage 3, the model processes multi-scale high-res patches from the selected regions with the same encoder from Stage 1. KV cache from the low-res tokens in Stage 1 is added to the self-attention layers to provide a global context for local high-res encoding.

stages corresponding to these three capabilities respectively (Figure 3.4). The design of each stage is detailed below.

Stage 1: Low-res feature extraction. We use the same vision transformer (ViT) architecture as SigLIP-SO400M [256] to extract low-res features. The image is resized to 378×378 which corresponds to 27×27 output tokens.

Stage 2: Top-down or bottom-up patch selection. In this stage, the model selects important regions either based on their relevance to a text prompt (*i.e.*, top-down selection) or based on the saliency of the region itself (*i.e.*, bottom-up selection) [27, 268]. See Figure 3.6(Left) for examples of such selection. To achieve this, the model predicts a selection score for each spatial position of the image by calculating the cosine similarity between the low-res visual features (from Stage 1) and the embedding of the prompt. The prompt embedding is either the text embedding in the case of top-down selection or a constant learnable vector in the case of bottom-up selection, following [172]. The text embedding comes from the text encoder in our contrastive pre-training.

The selection score is calculated with low-res features only, making it infeasible to locate fine-grained details. To alleviate this issue, we predict additional high-res selection score following the same process but with auxiliary high-res features extracted by a light-weight encoder. The light-weight encoder is a ConvNeXt [132] model with only 3 blocks and extracts features at 1512 resolution. The high-res and low-res selection scores are then interpolated to the same size and averaged as the final score.

Stage 3: High-res multi-scale feature extraction. Stage 3 consists of a few key steps. **1) Selecting top- k multi-scale high-res patches.** The model first resizes the high-res

image to a set of pre-defined scales and patchifies each. For example, we use three scales of 756×756 , 1512×1512 , and 3780×3780 for a maximum resolution of 4K. Each is then patchified to 54×54 , 108×108 , and 270×270 patches, respectively. The selection score from Stage 2 is also interpolated into the same each size. Then for each scale, top- k patches with the highest score are selected. Note that k can vary for different scales. During pre-training, we set k for each scale to be proportional to the total number of patches at that scale, *e.g.*, k equals to 80, 320, 2000 for scales of 756×756 , 1512×1512 , and 3780×3780 . This ensures the exact same image region is selected for each scale. However, the user can flexibly decide which scale to select more patches by setting different k for different scales, which we find benefits different downstream tasks (Section 3.7).

2) Scale-aware positional embedding. After embedding each selected patch with the same patch embedding module as in Stage 1, we then add positional embedding for each token by interpolating the original low-res positional embedding and selecting the positional embeddings that correspond to the selected patches. On top of that, we add a scale-specific positional embedding to tokens from each scale such that the ViT is able to discriminate tokens from the same spatial position but different scales.

3) High-res feature extraction with low-res KV cache. The selected tokens from different scales are gathered and simultaneously processed by the same ViT as in Stage 1. To make the local high-res features aware of the global visual context, we augment the keys and values in the self-attention layers with the keys and values from the corresponding layer in the low-res feature extraction stage, similar to the KV cache in modern LLMs [153]. The effect of each step above is studied in Section 3.7.

Pre-Training Algorithm of PS3

PS3 is pre-trained to jointly learn 1) detailed visual representation through localized contrastive loss and 2) top-down and bottom-up patch selection from box supervision.

Learning high-res visual representation. As illustrated in Figure 3.5(a), given the paired data of high-res images and detailed local captions, we use PS3 to extract the high-res features of the local regions that are relevant to the local captions as described in Section 3.2, extract text embedding of the local captions using a text encoder, and optimize a contrastive loss between the high-res visual features and the text embeddings. The total number of selected high-res patches for each image is limited to 2560 during pre-training for efficiency, while one can choose to select more tokens for downstream applications (Section 3.3). We use the same sigmoid contrastive loss as in SigLIP [256]. Both the ViT backbone in PS3 and the text encoder are initialized with the pre-trained SigLIP.

There are several key designs in the contrastive pre-training. The effect of each design is studied in Section 3.7.

1) Using ground-truth selection score for patch selection. Normally PS3 selects patches based on the local caption. However, in pre-training, to avoid inaccuracy in the selection score predicted by the model which may lead to selecting irrelevant regions, we use selection score generated from the ground-truth bounding box. This is similar to Teacher Forcing [224] in training recurrent neural networks, where the ground truth of an intermediate prediction is given in order to better train the consequent prediction of the

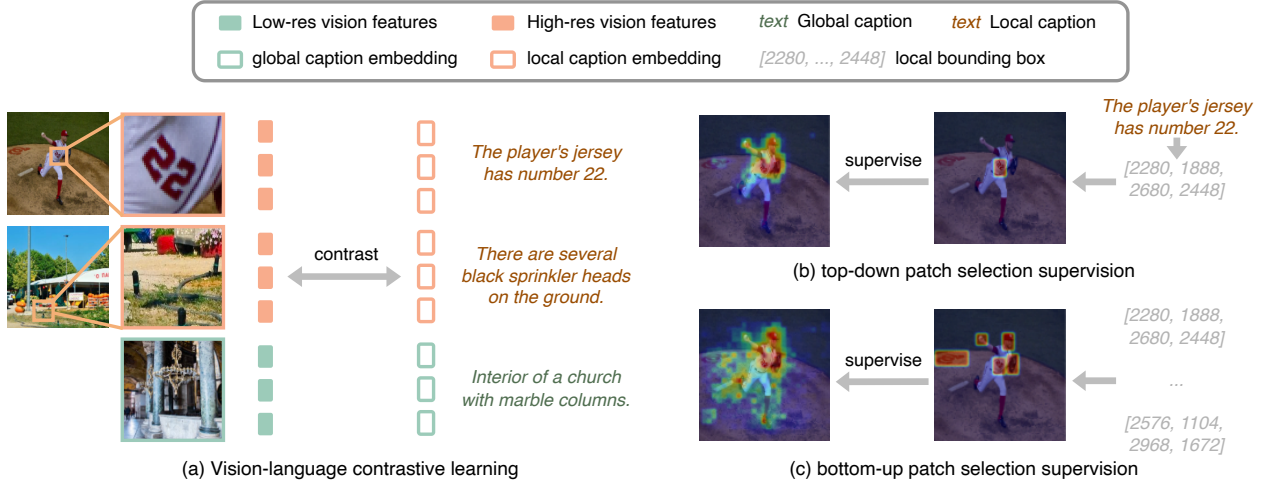


Figure 3.5: **Pre-training algorithm of PS3.** (a) During training, PS3 extracts the high-res features from the labeled local regions and contrasts them with embeddings of the local captions. To maintain the low-res feature quality, we also mix pairs of low-res features and global caption embedding in each batch. Both high-res and low-res features are extracted in the same way as Figure 3.4. (b) The top-down patch selection score is supervised by ground-truth score map generated from the bounding box corresponding to the local caption. (c) The supervision for bottom-up selection is similar to top-down selection, except that the ground-truth selection score is generated from all the labeled bounding boxes of the image.

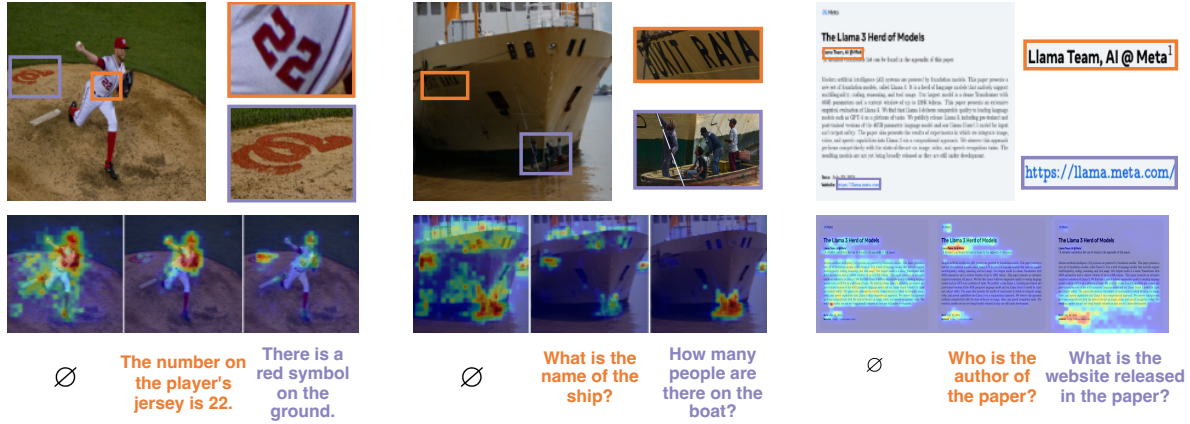


Figure 3.6: **Qualitative examples of patch selection.** *Left:* PS3 is pre-trained to perform bottom-up selection based on image saliency (denoted by $\emptyset$) or top-down selection based on local captions. The selection process is detailed in Figure 3.4 and Section 3.2. *Middle & Right:* We fine-tune PS3 with MLLM to select patches based on questions about local regions (Figure 3.7 and Section 3.3).

model. The ground truth selection score is generated by setting the score inside the box to 1 and others to 0. **2) Pooling only tokens in the ground-truth boxes.** SigLIP uses attention pooling to compress all the output tokens into one for contrastive loss. When a box contains fewer patches than the pre-set k , the model will select patches outside the box as well. Pooling both tokens inside and outside the box results in aligning irrelevant visual features to the text embedding in contrastive loss. To avoid this, we constrain the attention pooling to only tokens inside the box. **3) Mixing global and local contrast.** We empirically find that optimizing contrastive loss only between local regions and captions can degrade the quality of global low-res representations. To this end, we mix global and local contrast, *i.e.*, each batch contains both pairs of global low-resolution features and global caption embeddings and pairs of local high-resolution features and local caption embeddings. **4) Avoiding intra-image contrast.** Since we have multiple local boxes and captions for each high-res image, there is a chance that one batch contains multiple local regions from the same image. It can be problematic to contrast between different regions of the same image if those regions are visually similar [32]. We make sure each image only appears once in a batch to avoid intra-image contrast. The effect of each design is studied in Section 3.7.

Learning top-down and bottom-up patch selection. As illustrated in Figure 3.5(b), we use box supervision to learn top-down and bottom-up patch selection. During pre-training, the model predicts both the bottom-up selection score (*i.e.*, without any text prompt) and the top-down selection score based on the local caption for each image. To learn top-down patch selection, we supervise the selection score with the ground-truth bounding box that corresponds to the local caption. Specifically, we treat it as a binary semantic segmentation problem, where the predicted likelihood of binary classification is the selection score at each position and the ground truth label is 1 inside the ground-truth box and 0 outside. A position-wise cross entropy loss as well as a DICE loss [181] is then optimized between the selection score map and the ground-truth map. For bottom-up selection, since we already label the bounding boxes of salient regions in each image, we directly use these boxes to generate ground truth segmentation map by setting the label of a position to 1 as long as it belongs to one of the boxes. We then supervise the bottom-up selection score using the same loss functions as above. Figure 3.6(Left) shows examples of the learned top-down and bottom-up patch selection.

3.3 VILA-HD: Enabling High-Resolution MLLM with PS3

We apply PS3 to MLLMs to enhance their high-resolution perception capability. Specifically, we propose VILA-HD, an MLLM with PS3 as the vision encoder that shows superior performance and efficiency in processing images of up to 4K resolution. In the following we introduce the model design (Section 3.3) and training data (Section 3.3) of VILA-HD.

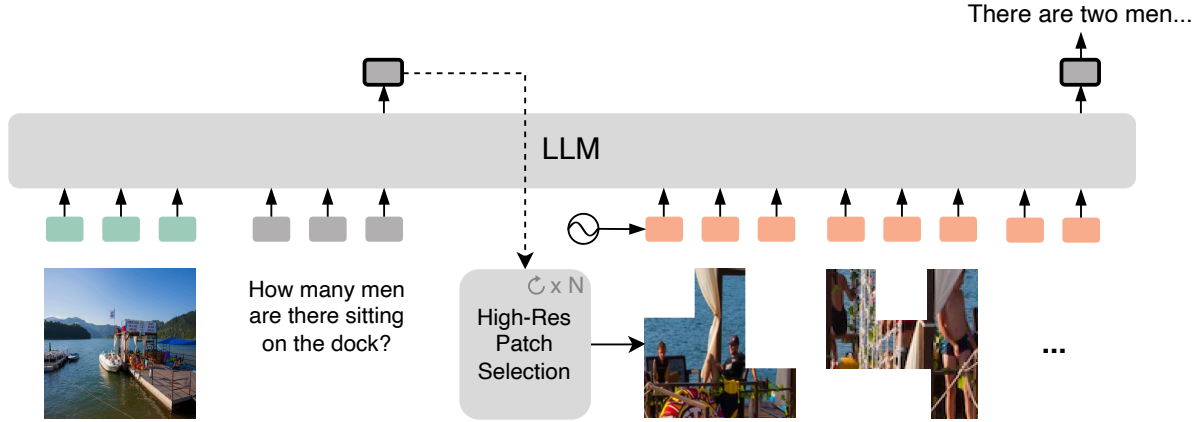


Figure 3.7: **Model design of VILA-HD.** For any input image and text prompt, VILA-HD first extracts the low-res image features using PS3 and sends them along with the text tokens to the LLM. The last-layer embedding of the last token is used to select high-res patches in PS3, whose features are then extracted by PS3, added with additional positional embedding, and sent to the LLM. Although PS3 only processes at most 2560 high-res patches at a time, one can run the patch selection and high-res feature extraction for N times (N can be an arbitrary number) to encode up to $2560 \times N$ high-res patches.

Building VILA-HD with PS3

VILA-HD’s model architecture is based on NVILA [130]. NVILA is a LLaVA-style [123] MLLM with SigLIP [256] as the vision encoder, and we replace the vision encoder with PS3 to build VILA-HD. The model design of VILA-HD is illustrated in Figure 3.7. We first extract the global low-res features following Stage 1 of PS3 and send them along with the text tokens to LLM. We then select high-res patches in either a bottom-up or a top-down way. Bottom-up selection is exactly the same as in pre-training. For top-down selection, since we need to select regions that can help answer the user’s question, instead of using the text embedding from the SigLIP text encoder as the prompt, we use the latent embedding of the last token in the user’s text input from the last layer of LLM as the prompt embedding. This is inspired by LISA [108] which uses the same embedding for reasoning segmentation. Finally, the selected high-res patches are encoded by Stage 3 of PS3 and sent to LLM after the text tokens, from which the following text generation resumes. We also add an additional positional embedding to the high-res features before sending them to LLM such that LLM is aware of the spatial positions of the selected patches, similar to the scale-aware positional embedding Stage 3 of PS3. Note that while the number of selected high-res patches is limited to 2560 during pre-training, one can select an arbitrary number of patches when applying to MLLMs by running patch selection and high-res feature extraction for multiple times. For example, to select 3840 high-res patches, one can first select the top 2560 patches to process in Stage 3, and then select another top 1280 patches among the rest of the unselected patches and process them in Stage 3. The multiple runs of Stage 3 are independent from each other.

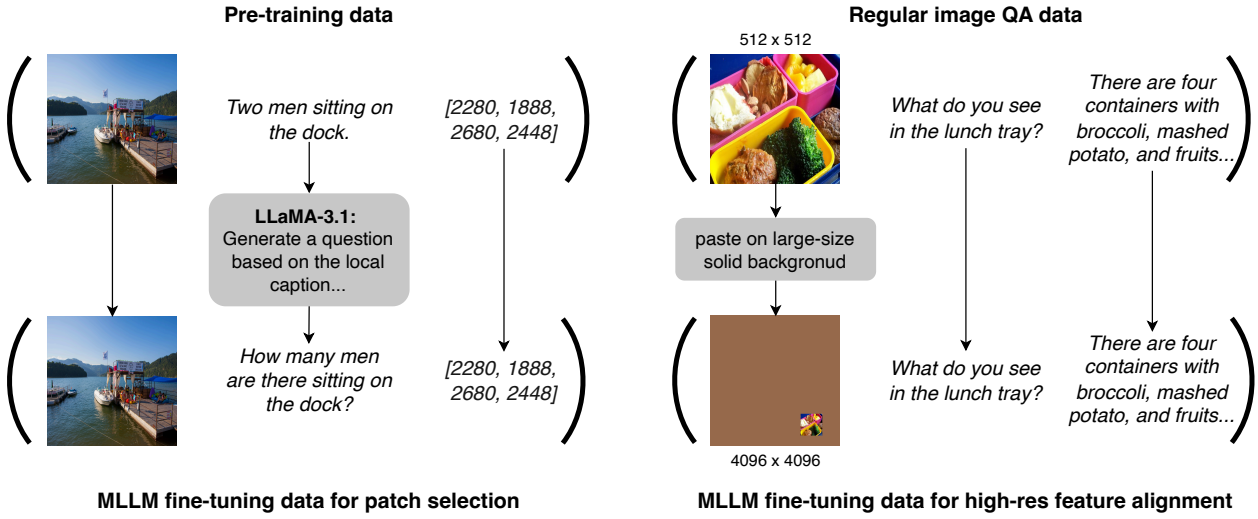


Figure 3.8: **Additional fine-tuning data for MLLMs with PS3.** *Left:* To fine-tune top-down patch selection, we generate data with pairs of high-res image, question about a local region, and the bounding box of the region. This is generated by taking the PS3 pre-training data and retargeting the local captions into questions using LLaMA-3.1. *Right:* To align the PS3 high-res features to the LLM text space, it requires fine-tuning data that contains QA pairs on high-res images. We generate this by taking regular low-res image QA data and pasting the image onto a large-size background to get the new high-res image while the question and answer are inherited.

Then all the high-res tokens are gathered together and sent to LLM.

Fine-Tuning Data of VILA-HD

Except for regular fine-tuning data for training MLLMs in previous work [111, 130], we collect additional data for 1) learning top-down patch selection in VILA-HD and 2) aligning the high-res features from PS3 to the text space of VILA-HD. Each type of data is introduced below and Figure 3.8 illustrates the curation of the data.

Fine-tuning data for patch selection. When training VILA-HD, we jointly fine-tune the top-down patch selection since it uses different prompt embeddings compared to pre-training. For this purpose, we collect data of high-res images paired with *questions* about local regions as well as their bounding boxes. As shown in Figure 3.8(a) This is achieved by directly sampling images and bounding boxes from the pre-training data (Section 3.2) and then automatically generate questions based on the local captions using LLaMA-3.1 [63]. We generate 500k data samples from the natural images and document images in the pre-training dataset respectively. To check the quality of the generated data, we randomly sample 500 examples and manually check whether the generated questions correctly corresponds to the original captions and local regions, and find 95.8% of the questions are accurate. During

training, we run VILA-HD with PS3 on this data to perform top-down patch selection based on the questions, and then supervise the patch selection following the same objective as in Section 3.2. Note that this data does not have answers to the questions so we do not optimize the next-token-prediction loss on this data. See Figure 3.6(Middle & Right) for visualization of the learned patch selection.

Fine-tuning data for high-res vision feature alignment. Since most of the existing fine-tuning data for MLLMs only contains low-resolution images, the high-res vision features from PS3, especially the features extracted at 4K resolution, are not properly aligned to the LLM’s text space if they are only trained on low-resolution data. To solve this problem, we collect 225k VQA data at up to 4K resolution. The data curation pipeline is shown in Figure 3.8(b). Specifically, we take existing low-res VQA data, paste the images onto a solid-color background image with larger size such as 4096×4096 , and use the new synthesized image with the original question as the high-res VQA data. By training on this data, VILA-HD is forced to utilize the high-res features from PS3 to answer the question because the informative region in the synthesized image is small, *e.g.*, a 256×256 region in a 4K image. During training, we mix this data with other fine-tuning data. We find this simple approach works reasonably well in our empirical studies (Section 3.7).

3.4 Scaling Properties of PS3

In this section, we evaluate how well the performance of PS3 scales with the pre-training resolution. Specifically, we pre-train PS3 at different resolutions and compare the performance of VILA-HD with the PS3 encoders as well as baseline vision encoders using a fixed MLLM training recipe. We pre-train PS3 at three resolutions of 756, 1512, and 3780, where for each resolution we extract multi-scale high-res features at scales of (756), (756, 1512), and (756, 1512, 3780), respectively. We show four types of scaling: **1) Whole-image resolution scaling.** PS3 selects all the high-res patches at each resolution for VILA-HD. This is to compare the high-res feature quality of PS3 with the baselines that also process all high-res patches (see *Experiment settings* below). **2) Constant-cost scaling.** PS3 selects a constant number of high-res patches when scaling the resolution for VILA-HD. This evaluates if performance scales “for free”, *i.e.*, by maintaining a constant MLLM training and inference cost (note that the pre-training cost is already near-constant). **3) Constant-resolution scaling.** At the same resolution, PS3 selects increasingly more high-res patches for VILA-HD. This evaluates if we gain benefits from selecting more patches in downstream without changing the pre-training and fine-tuning resolution. **4) Test-time scaling.** Similar to 3), but we increase the number of high-res patches at *test* time. The scaling curves are shown in Figure 3.9 and more detailed quantitative results are reported in Table 3.1.

Experiment settings. PS3 is initialized with SigLIP-SO400M [256] before pre-training. VILA-HD is trained using a subset of the training data in NVILA. For evaluation, we report average accuracy on seven resolution-sensitive benchmarks: TextVQA [178], ChartQA [140], DocVQA [141], InfoVQA [142], OCRBench [128], V*Bench [228], and RealWorldQA [232].

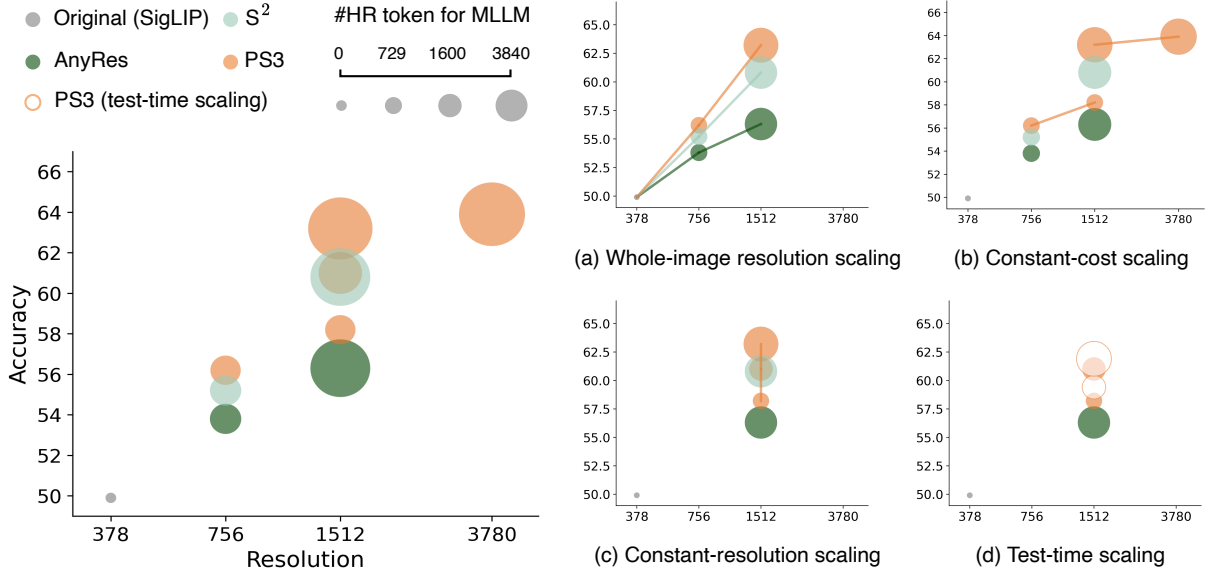


Figure 3.9: **Scaling properties of PS3 on VILA-HD.** (Left) Overall results. We report average performance of the MLLM on seven benchmarks under different maximum input resolution. The size of each data point indicates the number of high-res vision tokens input to the LLM. (a) When selecting all high-res patches for MLLM, the performance of PS3 scales better with the resolution than the baselines without high-resolution pre-training. (b) PS3 is able to process higher resolution and improve performance while selecting a fixed number of high-res patches for MLLM. (c) Within the same resolution, PS3 is able to trade compute for performance by selecting more high-res patches. (d) PS3 can select more high-res patches at test time even if it selects a fixed number of high-res patches during MLLM training.

We compare PS3 to the original SigLIP as well as two baselines, AnyRes [42, 124] and S^2 [175], that run SigLIP at larger resolution in a training-free way by splitting large images into smaller tiles. For AnyRes baselines, we set the maximum number of tiles to 4 for 756 resolution and 9 for 1512 resolution. For S^2 baselines, instead of resizing the feature maps from different scales to the smallest scale as in the original algorithm, we resize them to the largest scale to get the best performance. For PS3, the number of selected patches for multiple image scales at test time might vary between benchmarks to obtain the best performance (Section 3.7).

Whole-Image Resolution Scaling

We first scale up the resolution of PS3 and make it select all the high-res patches at each resolution for VILA-HD. Results are shown in Figure 3.9(a). We can see that the effect of scaling up resolution is significant, where PS3 at 1512 resolution improves 14.2% over SigLIP baseline. Compared to AnyRes and S^2 , PS3 shows consistent improvements across

Table 3.1: **Full results of scaling properties of PS3 on VILA-HD.** $\#HR$ Token is the number of high-res tokens input to the MLLM. *Select (Train)* is the percentage of high-res patches PS3 selects when training the MLLMs and *Select (Test)* is the selection ratio at test time. Note the number of HR tokens is usually 1/4 of the number of selected patches due to the 2×2 downsampling connector in NVILA [130]. $^\dagger \#HR$ tokens for AnyRes depends on the input resolution and we report the maximum number of tokens.

Vision Encoder	Max Res.	$\#HR$ Token	Select (Train)	Select (Test)	Text VQA	Chart QA	Doc VQA	Info VQA	OCR Bench	V* Bench	Real World	Avg
SigLIP [256]	378	0	-	-	62.3	56.6	51.9	30.7	387	51.8	57.1	49.9
AnyRes [124]	756	784 †	-	-	65.3	58.0	60.6	32.7	416	59.2	59.1	53.8
S ² [175]	756	729	-	-	65.9	65.5	63.0	32.3	471	53.1	59.6	55.2
PS3	756	320	44%	44%	66.7	62.8	62.6	33.1	460	56.3	61.7	55.6
	756	729	100%	100%	66.8	63.5	64.6	33.9	462	56.5	61.7	56.2
AnyRes [124]	1512	3136 †	-	-	67.4	58.4	67.9	34.1	468	60.2	59.0	56.3
S ² [175]	1512	2916	-	-	66.1	71.0	78.3	41.1	526	55.2	61.0	60.8
PS3	1512	729	20%	20%	67.3	64.7	66.5	34.8	505	60.7	62.6	58.2
	1512	1600	20%	44%	67.7	65.9	70.7	35.7	515	62.0	62.6	59.4
	1512	1600	44%	44%	68.4	68.0	74.5	37.3	509	63.1	65.0	61.0
	1512	3645	44%	100%	68.4	68.0	76.5	39.4	522	66.7	62.0	61.9
	1512	3645	100%	100%	69.3	71.1	79.4	41.3	534	64.0	63.8	63.2
AnyRes [124]	3780	19600 †	-	-	OOM							
S ² [175]	3780	18225	-	-	OOM							
PS3	3780	3840	18%	18%	69.8	70.9	79.1	40.5	543	67.8	64.7	63.9

different resolution while using a similar number of high-res tokens. For example, at 1512 resolution, PS3 improves by 2.4% over S² and 6.9% over AnyRes which is commonly used by modern MLLMs [42, 111]. Since all the methods are processing the whole high-res image, the advantage of PS3 mainly comes from the improved high-res feature quality which is brought by our high-res pre-training. Note that we cannot scale up to 4K resolution while selecting all the high-res patches due to the drastic computational cost, although it is achievable with constant-cost scaling in Section 3.4.

Constant-Cost Scaling

Scaling up resolution comes at a cost of quadratically increasing number of tokens in Section 3.4. However, for PS3, higher resolution is still beneficial even when selecting a constant number of patches (Figure 3.9(b)). For example, using 729 (20%) high-res tokens with 1512 resolution improves the accuracy by 2% over using 729 (100%) tokens with 756 resolution (Table 3.1). This is because at 756 resolution, not all patches are relevant to the user’s questions, and by scaling up the resolution to 1512, even it selects the same number of patches, there are fewer irrelevant 756-resolution patches but more relevant 1512-resolution patches being selected, thanks to our top-down selection mechanism. Comparing to AnyRes, PS3 achieves 58.2% accuracy at resolution of 1512 through constant-cost scaling, improving over AnyRes by 2%

accuracy while using $5\times$ fewer tokens. Notably, while neither AnyRes or S^2 can scale to 4K resolution, PS3 is able to scale up to 4K resolution with a constant training and inference cost, where PS3 achieves 63.9% accuracy, improving over S^2 at 1512-resolution by 3.1% while using a similar number of tokens. Specifically, the accuracy on V*Bench improves from 64.0% to 67.8% by scaling to 4K resolution, which shows that 4K resolution is important for such detailed visual perception. We observe an even larger advantage for tasks that require 4K resolution (Section 3.5).

Constant-Resolution Scaling

Pre-trained at a fixed resolution, PS3 can flexibly select different number of patches for VILA-HD to trade compute for performance. As shown by Figure 3.9(c), selecting more patches at 1512 resolution consistently improves performance. By increasing the number of high-res tokens from 729 (20%) to 1600 (44%), PS3 is able to outperform S^2 with only half the number of high-res tokens (Table 3.1). Further increasing to 3645 (100%) tokens boosts the performance by another 2.2%. We observe that the improvement is especially significant for tasks that has dense visual information and requires processing most parts of the image such as document understanding in DocVQA. On the other hand, tasks such as TextVQA and RealWorldQA which only require understanding of local regions only have minor performance drops (2.0% and 1.2%, respectively) when we select only 20% of the high-res patches. In practice, PS3 allows one to select different number of high-res patches based on the computational budget and the requirements for high-res information density in different tasks.

Test-Time Scaling

Constant-resolution scaling is still valid at test time, *i.e.*, we select a fixed number of high-res patches during training but select more at test time. As shown in Figure 3.9(d), at 1512 resolution, we can train with 20% high-res patches and test with 44% patches, which improves the accuracy by 1.2%. Similarly, training with 40% patches but testing with 100% patches gives an improvement of 0.9%. Note that scaling at test time still performs worse than training time, which is possibly because 1) the MLLM receives lower-quality supervision if seeing fewer patches at training time, and 2) the LLM is not adapted to longer context length during training.

Key Finding 1: Scaling up the pre-training resolution of PS3 significantly improves performance of downstream MLLM over baselines without high-res pre-training. This is achievable with near-constant pre-training cost and even with no extra downstream training and inference cost by selecting a fixed number of high-res patches. PS3 can also flexibly select different number of patches to trade training or test-time compute for better performance.

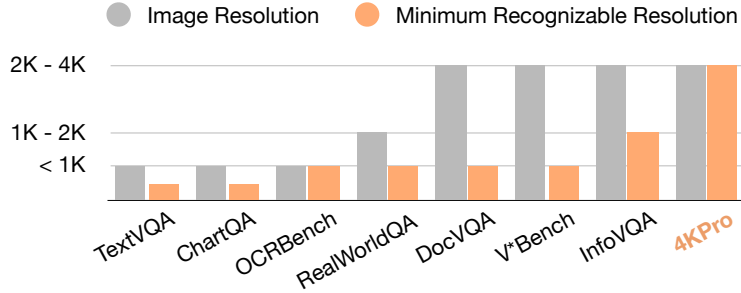


Figure 3.10: **Image resolution and MRR of different benchmarks.** Existing benchmarks contain high-res images but the resolution required to answer the questions (MRR) is mostly under 1K. In contrast, 4KPro contains questions only solvable at 4K resolution.

3.5 4KPro: Benchmarking PS3 at 4K Resolution

Despite the superior performance of PS3 and VILA-HD on existing benchmarks as shown in the previous section, we find those benchmarks do not actually require high resolution visual perception, especially 4K-resolution perception, even though some of them contain high-resolution images. To verify this, we examine the minimum recognizable resolution (MRR) of the existing benchmarks, *i.e.*, the minimum resolution required to answer the questions. We calculate the MRR by randomly sampling examples from each benchmark, manually checking the minimum resolution (4K, 2K, or 1K) under which the visual details are clear enough to answer the question for each example, and averaging the minimum resolutions of different samples for each benchmark. As shown in Figure 3.10, we can see for all previous benchmarks their MRR is lower than the average image resolutions, which means these benchmarks don’t require visual perception at the same resolution as the images. Specifically, even though benchmarks like DocVQA and V*Bench contain images at 4K resolution, the MRR is mostly around 1K. InfoVQA has the highest MRR of 2K, although it is solely focused on infographic understanding.

To effectively evaluate 4K-resolution perception in real-world tasks, we introduce 4KPro, an image QA benchmark with MRR of 4K from four professional use cases including Autonomous Vehicle, Household, Gaming, and UI Understanding. Each QA pair is in the form of multi-choice problem with four options. Examples of 4KPro are shown in Figure 3.11, where we can see that all the questions cannot be accurately answered without 4K resolution.

Experiment settings. The training setting is the same as Section 3.4 where we pre-train PS3 at resolutions of 756, 1512, and 3780, train VILA-HD with each PS3 encoder and evaluate on 4KPro. When comparing with state-of-the-art MLLMs, we use SigLIP2-SO400M [198] as initialization during PS3 pre-training and evaluate VILA-HD with PS3 pre-trained at 1512 and 3780 resolutions, which we denote by VILA-HD-8B-PS3-1.5K-SigLIP2 and VILA-HD-8B-PS3-4K-SigLIP2, and we use the full training recipe in NVILA-8B [130].

Main Results

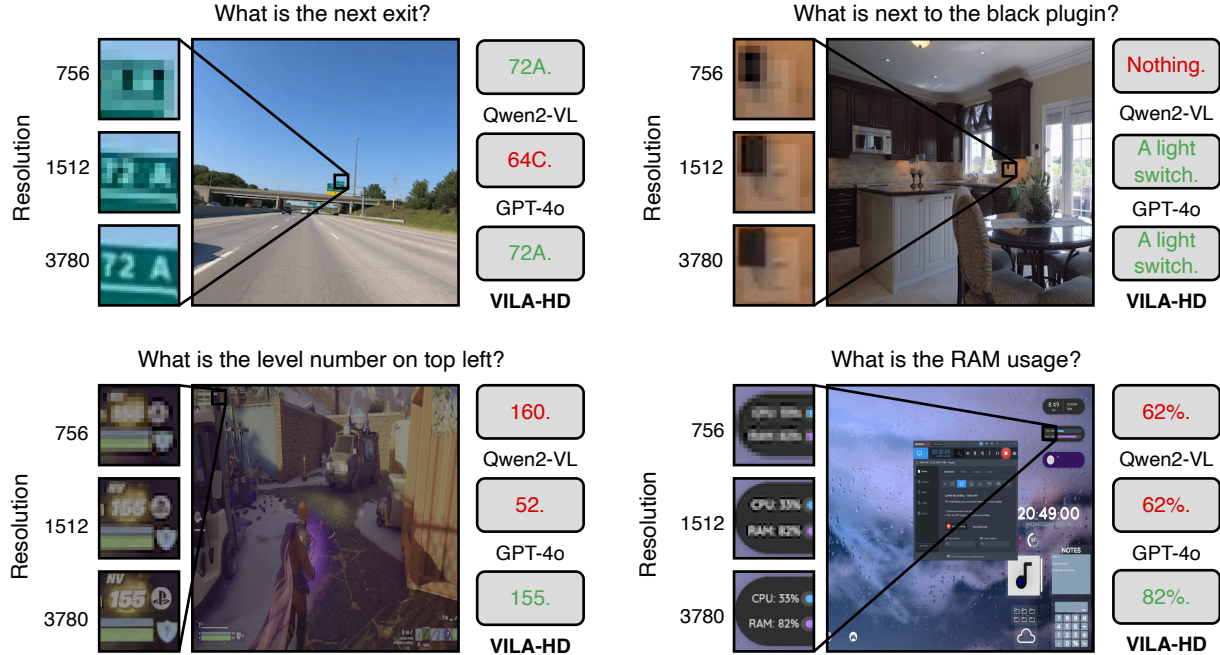


Figure 3.11: **Examples from 4KPro and comparison of different models.** Each example corresponds to one out of four categories (Autonomous Vehicle, Household, Gaming, and UI Understanding) and each question can only be answered without ambiguity under 4K resolution. VILA-PS3 improves the accuracy over the state-of-the-art MLLMs such as GPT-4o and Qwen2-VL.

Scaling properties of PS3. We evaluate the performance of VILA-HD when scaling up the resolution of PS3. Results are shown in Figure 3.12. We can see PS3 outperforms other baselines at resolution of 756 and 1512, although by a relatively small margin, which implies 4KPro is generally unsolvable with resolution lower than 4K. However, while it is infeasible to scale to 4K resolution for the baselines, PS3 enables 4K resolution perception by selecting the same number of high-res patches as 1512 resolution, using the same constant-cost scaling scheme as in Section 3.4. This improves the performance by 4.8%. Taking a step further, we can double the number of high-res patches at test time to boost the performance by another 8.2%. This achieves an improvement of $\sim 15\%$ over AnyRes at 1512 resolution, which is more significant than the improvements we get by scaling to 4K resolution in Section 3.4, indicating that 4K-resolution pre-training is most helpful for downstream tasks requiring ultra high-res perception. On the other direction, we can also shrink the number of patches by $3\times$, achieving 48.4% accuracy which is 3.2% higher than AnyRes at 1512 resolution while using $2.5\times$ fewer tokens.

Figure 3.12: **Scaling properties of PS3 on 4KPro.** PS3 shows consistently improved performance by scaling to 4K resolution and achieves further improvements by scaling up number of high-res tokens at test time, greatly outperforming the baselines.

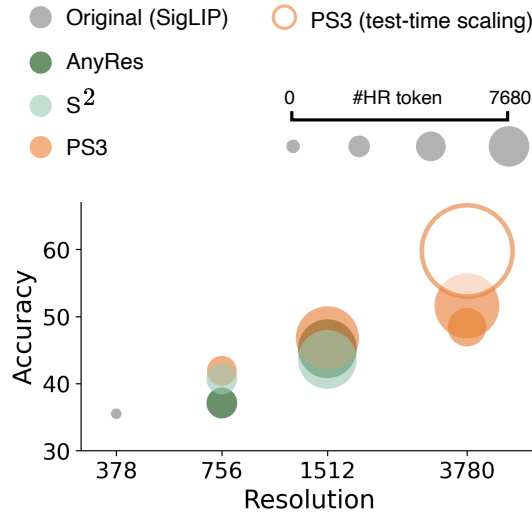


Table 3.2: **Comparing VILA-HD-8B-PS3-1.5K-SigLIP2 and VILA-HD-8B-PS3-4K-SigLIP2 to state-of-the-art MLLMs on 4KPro.** VILA-HD outperforms Qwen2-VL which has the best performance among existing MLLMs while having a lower latency. [†]We limit the maximum number of pixels to 4,000,000 to maintain a reasonable latency.

Model	Select	Latency	Acc.
GPT-4o [98]	-	-	59.7
Claude 3.5 Sonnet [5]	-	-	29.0
Gemini-1.5-Pro [190]	-	-	59.7
NVILA-8B [130]	-	0.82s	58.1
Cambrian-1-8B [194]	-	2.78s	50.0
InternVL2-8B [191]	-	1.65s	58.1
IXC-2.5-7B [260]	-	2.11s	32.3
LLaVA-OneVision-7B [111]	-	1.75s	67.7
InternVL3-8B [271]	-	2.78s	61.3
Qwen2-VL-7B-Instruct [†] [206]	-	3.61s	71.0
Qwen2.5-VL-7B-Instruct [†] [12]	-	2.98s	68.3
VILA-HD-1.5K	33%	0.46s	59.7
VILA-HD-1.5K	100%	1.20s	67.7
VILA-HD-4K	18%	1.22s	71.0
VILA-HD-4K	35%	1.78s	75.8

Comparison to state of the arts. We compare the performance of VILA-HD-8B-PS3-1.5K-SigLIP2 and VILA-HD-8B-PS3-4K-SigLIP2 with other proprietary or open-source MLLMs (Table 3.2), where we abbreviated the two models as VILA-HD-1.5K and VILA-HD-4K. All the proprietary MLLMs do not perform well, with the best accuracy at 59.7% for GPT-4o and Gemini-1.5-Pro. We find part of the errors for proprietary MLLMs are caused by erroneous instruction following or refusing to answer the question when unable to recognize the details. For open-source models, most achieve accuracy around 50% - 60%. Qwen2-VL-7B-Instruct performs the best, reaching an accuracy of 71.0%, although at a cost of larger latency than other models due to processing the full high-resolution images in its vision encoder. VILA-HD-4K, on the other hand, achieves 75.8% accuracy at 3780 resolution when selecting 35% patches, which improves over both the proprietary models and the state-of-the-art open-source model (Qwen2-VL) while having a lower latency thanks to the top-down patch selection mechanism. VILA-HD-4K also significantly improves over VILA-HD-1.5K by 8.1%. Note that the performance under 1.5K resolution is still better than random guess (25%) even though the visual details are not clear, which is probably because the blurry details still provide some clues for an informed guess. By selecting fewer patches (*e.g.*, 18%), VILA-HD-4K still maintains superior performance of 71.0% while enjoying only 1/3 of the latency comparing to Qwen2-VL. We also show qualitative results in Figure 3.11. We can see that compared

to state-of-the-art MLLMs, VILA-HD with PS3 is able to accurately recognize extremely fine-grained details under 4K resolution, *e.g.*, tiny text on highway signs or monitor screens and small light switch on a distant wall.

Key Finding 2: Most of previous benchmarks do not require perception at 4K resolution despite the images at 4K resolution, and on 4KPro benchmark that does require 4K resolution perception, VILA-HD with PS3 significantly improves over previous MLLMs in both accuracy and efficiency.

3.6 Comparing PS3 and VILA-HD to State of the Arts

In this section, we compare PS3 and VILA-HD with other state-of-the-art models. Specifically, we compare the visual representation quality of PS3 with state-of-the-art vision encoders [22, 41, 76, 160, 198], compare the performance of VILA-HD with state-of-the-art MLLMs [5, 60, 98, 111, 119, 130, 190, 191, 194, 206, 259, 260], compare the efficiency of patch selection in PS3 and VILA-HD with other MLLM token pruning methods [20, 34, 241], and also verify the generalizability and effectiveness of PS3 pre-training pipeline on different state-of-the-art vision encoders [160, 256].

Comparing PS3 to State-Of-The-Art Vision Encoders

Experiment settings. To compare with state-of-the-art vision encoders, we first pre-train the PS3 encoder at 1512 resolution using SigLIP2-SO400M [198] as the initialization, which we call PS3-1.5K-SigLIP2. Then we evaluate PS3-1.5K-SigLIP2 as well as each baseline vision encoder by training an MLLM on top of it using the same training recipe as in Section 3.4 and testing it on various MLLM benchmarks. We do not compare the 4K-resolution PS3 here because we are not able to train MLLM with any baseline encoder at 4K resolution due to OOM issue and thus cannot build a fair comparison between them and PS3-4K-SigLIP2. We compare PS3-1.5K-SigLIP2 with state-of-the-art vision encoders including SigLIP2-SO400M [198], C-RADIOv2-L [160], PE_{Core}-L [22], and AIMv2-L [76]. Additionally, we follow the previous literature [22] and present PS3_{Lang}-SigLIP2-1.5K which is the PS3 model that is further aligned with LLMs by co-training with VILA-HD (Section 3.6). Specifically, after training VILA-HD, we take out the vision encoder part as an independent vision encoder. We compare the LLM-aligned version of PS3 with other LLM-aligned vision encoders such as PE_{Lang}-L [22] and InternViT2.5-300M [41]. For a fair comparison, we train MLLMs with each vision encoder at roughly the same resolution around 1.5K to 1.7K. Since every vision encoder except for PS3 is pre-trained at low resolutions, we use S^2 to scale them to high resolution when training MLLMs. We evaluate on three types of benchmarks, including high-resolution VQA, VQA related to texts, documents, and charts, and general VQA.

Table 3.3: **Comparison to state-of-the-art vision encoders.** We train an MLLM on top of each vision encoder with the same recipe and compare the performance on various types of benchmarks including high-resolution VQA, VQA over text, documents, and charts, and general VQA. We compare **PS3-1.5K-SigLIP2** and **PS3_{Lang}-1.5K-SigLIP2** (abbreviated as PS3 and PS3_{Lang} here) with previous state-of-the-art vision encoders [22, 41, 76, 160, 198]. For a fair comparison, we use similar ViT sizes (300M - 400M) for every encoder and use S² to scale all the vision encoders (except for PS3 which natively supports high resolution) to a similar high resolution round 1.5K. We compare the vision encoders with and without LLM alignment separately.

Benchmark	<i>w/o LLM alignment</i>				<i>w/ LLM alignment</i>			
	SigLIP2	C-RADIOv2	PE _{Core}	AIMv2	PS3	InternViT2.5	PE _{Lang}	PS3 _{Lang}
<i>High-Resolution VQA</i>								
4KPro	37.1	32.3	46.8	43.5	50.0	43.5	45.2	53.2
HRBench _{4K} [211]	55.0	51.8	56.1	51.5	58.5	54.6	52.0	62.9
HRBench _{8K} [211]	49.9	44.8	50.3	47.6	52.5	47.3	49.9	50.5
MME-RealWorld [264]	43.7	40.7	44.1	43.4	42.1	44.9	43.5	40.4
V* [228]	58.8	59.7	64.9	63.1	64.2	57.4	61.1	62.7
<i>Avg</i>	48.9	45.9	52.4	49.8	53.5	49.5	50.3	53.9
<i>Text / Doc / Chart</i>								
TextVQA [178]	71.5	68.1	72.0	72.6	71.9	70.2	72.3	73.6
ChartQA [140]	71.3	72.3	73.8	72.3	74.0	75.8	76.4	78.0
CharXiV _{dq} [220]	40.7	40.3	39.3	41.1	41.3	42.7	45.6	41.8
CharXiV _{rq} [220]	15.4	16.2	21.3	15.6	18.2	18.2	24.6	22.9
DocVQA [141]	79.1	82.5	84.5	81.8	82.9	86.0	87.8	87.2
InfoVQA [142]	39.3	40.4	44.2	42.2	44.2	45.9	48.2	48.7
OCRBench [128]	566	542	608	590	558	647	581	659
SEEDBench-2-Plus [113]	60.0	61.3	62.7	61.8	63.6	62.8	64.1	65.1
<i>Avg</i>	<i>54.2</i>	<i>54.4</i>	<i>57.3</i>	<i>55.8</i>	<i>56.5</i>	<i>58.3</i>	<i>59.6</i>	<i>60.4</i>
<i>General VQA</i>								
BLINK [79]	44.2	45.4	44.1	44.3	44.0	43.8	43.2	43.4
MME [77]	1952	1983	1897	1962	1995	2029	2000	2031
MMBench _{en_dev} [127]	75.2	74.1	76.4	75.9	76.0	73.2	73.9	76.5
MMBench _{cn_dev} [127]	74.3	73.3	75.4	74.6	75.3	70.8	72.1	74.1
MMBench-v1.1 _{en_dev} [127]	73.0	72.0	74.6	73.9	73.6	70.5	72.1	74.5
MMBench-v1.1 _{cn_dev} [127]	72.8	71.4	74.1	73.1	73.5	68.1	70.0	73.2
MMMU-Pro _{standard} [254]	27.3	26.8	28.2	28.0	27.2	28.1	27.9	29.3
MMMU-Pro _{vision} [254]	12.3	12.9	12.1	13.0	12.3	13.5	14.2	14.6
MMStar [35]	51.1	51.3	51.9	50.1	52.9	50.4	53.7	53.9
MMVet [251]	44.7	43.7	44.1	43.9	46.3	42.2	44.1	47.1
RealWorldQA [232]	60.3	62.1	61.6	60.9	64.6	59.0	61.4	61.7
<i>Avg</i>	<i>55.0</i>	<i>54.9</i>	<i>55.5</i>	<i>55.3</i>	<i>56.1</i>	<i>53.8</i>	<i>54.9</i>	<i>55.5</i>

Results. Table 3.3 shows the full results. For vision encoders without LLM alignment, PE_{Core} is the most competitive encoder between the baselines and PS3-SigLIP2-1.5K is able to obtain better performance for high-resolution VQA and general VQA. Especially, PS3 improves the performance by 2% - 4% on high-res benchmarks such as 4KPro and HRBench, indicating the effectiveness of high-res pre-training of PS3. Note that PS3 is worse than PE_{Core} on MME-RealWorld, which is possibly because the benchmark requires capabilities such as counting that are beyond simple high-resolution perception which is the main pre-training objective of PS3. For benchmarks with texts, documents, and charts, PS3 is competitive with the state-of-the-art encoder while achieving the best results on benchmarks such as ChartQA and CharXiV_{dq}, which is partially credited to the document images in the pre-training data. However, the PS3 pre-training data is not specifically designed for text recognition in natural images, as reflected by the lower performance on OCRBench and TextVQA, leaving huge room for improvement in the future. For general VQA, PS3 also achieves the best performance on average, showing significant improvements on benchmarks such as MMVet and RealWorldQA. On the other hand, PS3_{Lang}-SigLIP2-1.5K achieves significantly better results compared to other encoders with or without LLM alignment, rendering it the state-of-the-art vision encoder across the benchmarks we evaluate here. Compared to PE_{Lang} , PS3_{Lang} dominates the performance on most of the benchmarks, including 8.0% and 10.9% improvements on highres benchmarks of 4KPro and HRBench_{4K}. It is also worth noting that if we compare the PS3 encoders with and without LLM alignment, the improvements are mostly on the text-related benchmarks, which implies the current MLLM training is still language-centric and fails to improve the vision-centric capabilities of vision encoders.

Key Finding 3: Compared to previous vision encoders with low-resolution pre-training only, PS3 achieves state-of-the-art results on various MLLM benchmarks, especially the ones that require high-resolution. Aligning PS3 with LLMs further boosts its performance on text-related benchmarks.

Comparing VILA-HD to State-Of-The-Art MLLMs

Experiment settings. To compare with state-of-the-art MLLMs, we train VILA-HD using the full recipe in NVILA-8B [130] and compare its performance with state-of-the-art MLLMs. We train VILA-HD with two PS3 encoders, PS3-1.5K-SigLIP2 and PS3-4K-SigLIP2, and denote them by VILA-HD-8B-PS3-1.5K-SigLIP2 and VILA-HD-8B-PS3-4K-SigLIP2, respectively. We evaluate on several high-resolution VQA benchmarks (4KPro, HRBench [211], MME-RealWorld [264], and V* [228]) and other general VQA benchmarks (TextVQA [178], ChartQA [140], DocVQA [141], InfoVQA [142], MathVista [135], MMBench [127], MMMU-Pro [254], OCRBench [128], and RealWorldQA [232]). For some benchmarks, we balance the patch selection ratio across different scales such that it reaches the optimal performance at test time.

Results. We first show the results on high-res benchmarks in Table 3.4. We compare with

Table 3.4: **Comparing VILA-HD to state-of-the-art MLLMs on high-resolution benchmarks. Here VILA-HD-8B-4K refer to VILA-HD-8B-PS3-4K-SigLIP2.** Bold and underlined results mean 1st and 2nd best performance for each task. VILA-HD-8B-4K achieves top-2 performance in 4 of the 6 benchmarks. [†]For most of the models, the resolution and number of tokens are dependent on the maximum number of tiles to use at test time, and we set the maximum number of tiles such that the number of tokens are similar to VILA-HD-8B-4K when it is selecting 18% patches for a fair comparison.

	Res. [†]	#Token [†]	4KPro	HRBench _{4K}	HRBench _{8K}	MME-RW	V [*] _{att}	V [*] _{pos}
NVILA-8B [130]	1552	3072	58.1	59.0	55.5	52.0	68.7	65.8
PLM-8B [49]	1735	4096	59.7	<u>71.3</u>	61.1	<u>63.1</u>	72.2	68.4
InternVL2.5-8B [41]	1735	4096	66.1	65.5	57.4	58.9	75.8	<u>76.5</u>
InternVL3-8B [271]	1735	4096	56.5	67.5	62.5	<u>60.7</u>	72.2	<u>75.0</u>
Qwen2-VL-7B [206]	1735	3840	<u>69.4</u>	66.9	62.1	58.1	<u>76.5</u>	67.1
Qwen2.5-VL-7B [12]	1735	3840	66.1	65.6	<u>64.1</u>	57.1	<u>78.3</u>	71.1
VILA-HD-8B-4K	3780	4036	<u>71.0</u>	<u>67.8</u>	<u>65.3</u>	50.9	<u>76.5</u>	71.1

state-of-the-art MLLMs under approximately the same number of vision tokens. We can see VILA-HD-8B-4K can process higher resolution with similar number of tokens, thanks to the top-down patch selection mechanism. As a result, VILA-HD achieves the best results on 4KPro and HRBench_{8K} with improvements of 4.9% and 2.2% respectively on 4KPro and HRBench_{8K} over Qwen2.5-VL-7B [12]. On HRBench_{4K} and V^{*}_{att}, VILA-HD also achieves second-to-best results. Notably, VILA-HD achieves 7.3% improvement on average over NVILA-8B, which showcases the advantage of high-res vision pre-training considering it is the only difference between the two models. The only benchmark where VILA-HD fails to perform well is MME-RealWorld, which is probably because the benchmark requires more complicated capabilities such as counting which is beyond the scope of PS3 pre-training which is focused on basic capabilities of high-res perception and recognition.

We show the results on other general benchmarks in Table 3.5, VILA-HD shows competitive performance compared to state-of-the-art MLLMs such as NVILA and Qwen2-VL and achieves the best results in 6 out of 11 benchmarks. Specifically, VILA-HD-1.5K achieves the best results on benchmarks that have the MRR around 512-1K including MathVista and MMMU-Pro, and VILA-HD-4K obtains state-of-the-art performance on benchmarks that require more detailed understanding (MRR between 1K and 4K) such as V^{*}Bench and 4KPro, surpassing NVILA despite using the same recipe and less data, showing the effect of PS3 pre-training. VILA-HD-4K also outperforms all proprietary MLLMs on these high-MRR benchmarks, for example, improving by 13.0% and 11.3% over Gemini-1.5-Pro on V^{*}Bench and 4KPro. Note that this is achieved by selecting only 18% of the high-res patches, showing both the efficacy and efficiency of our model. On the other hand, VILA-HD has slightly worse results than NVILA and Qwen2-VL on OCR-related benchmarks including ChartQA, DocVQA, TextVQA, InfoVQA, and OCRBench. This is probably because the pre-training data of PS3 is not specifically optimized for OCR and text recognition and adding such data in pre-training can potentially solve the problem. We further show that PS3 can achieve even

Table 3.5: **Comparing VILA-HD to state-of-the-art MLLMs on other common benchmarks.** Here VILA-HD-8B-1.5K and VILA-HD-8B-4K refer to **VILA-HD-8B-PS3-1.5K-SigLIP2** and **VILA-HD-8B-PS3-4K-SigLIP2**, respectively. *Res.* is the maximum resolution each model supports. Some models (*e.g.*, Qwen2-VL, InternVL2) can accept input images of different aspect ratios, for which the resolution is calculated as square root of the maximum number of pixels the model can take in. *Select* is the high-res patch selection ratio of PS3 at test time. *#Token* is the total number of visual tokens fed into LLM under the maximum input resolution. VILA-HD-8B-4K achieves state-of-the-art performance on high-resolution benchmarks such as V*Bench and 4KPro.

	Res.	Select	#Tok	ChartQA (test)	DocVQA (test)	InfoVQA (test)	MathVista (testmini)	MMBench (en-dev)	MMU- Pro (standard)	OCRBench (test)	V*Bench (test)	RealWorld (test)	TextVQA (val)	4KPro (test)
<i>Proprietary</i>														
GPT-4o [98]	-	-	-	85.7	92.8	-	63.8	-	54.0	736	53.7	58.6	-	59.7
Claude 3.5 Sonnet [5]	-	-	-	90.8	95.2	49.7	67.7	-	55.0	788	23.0	59.9	-	29.0
Gemini-1.5-Pro [190]	-	-	-	87.2	93.1	81.0	63.9	-	49.4	754	60.3	70.4	78.7	59.7
<i>Open-source</i>														
VILA-1.5-8B [119]	336	-	576	52.7	40.6	25.9	36.7	68.9	-	-	-	52.7	68.5	33.9
Cambrian-1-8B [194]	1024	-	-	73.3	77.8	-	49.0	75.9	-	624	59.2	64.2	71.7	50.0
MM1.5-7B [259]	2016	-	1440	78.6	88.1	59.5	47.6	-	-	635	-	62.5	76.5	-
NVILA-8B [130]	1552	-	3072	86.1	93.7	70.7	65.4	87.6	33.6	794	67.2	66.4	80.1	58.1
LLaVA-OV-7B [111]	2304	-	7252	80.0	87.5	68.8	63.2	80.8	29.5	-	69.2	66.3	-	67.7
IXC2-4KHD [60]	2479	-	7920	81.0	90.0	68.6	57.8	80.2	-	675	-	-	77.2	42.8
IXC-2.5-7B [260]	2743	-	10000	82.2	90.9	70.0	59.6	82.2	-	690	45.6	67.8	78.2	32.3
InternVL2-8B [191]	2833	-	10496	83.3	91.6	74.8	58.3	81.7	32.5	794	65.8	64.4	77.4	58.1
Qwen2-VL-7B [206]	3584	-	16384	83.0	94.5	76.5	58.2	-	-	866	71.0	70.1	84.3	71.0
VILA-HD-8B-1.5K	1512	33%	1411	84.4	89.5	62.3	66.2	92.0	35.0	795	66.5	69.3	79.5	59.7
	1512	67%	2626	85.8	92.8	68.0	65.8	92.3	35.2	817	68.0	69.7	80.3	67.7
	1512	100%	3841	85.4	93.1	70.0	66.6	92.4	35.0	819	68.5	70.8	80.4	67.7
VILA-HD-8B-4K	3780	6%	1476	83.6	89.0	59.7	65.8	91.7	34.9	791	66.5	68.2	79.9	56.5
	3780	12%	2756	84.7	92.3	66.3	66.2	92.4	35.2	794	69.7	69.5	80.4	59.7
	3780	18%	4036	84.7	92.7	68.2	66.6	92.6	34.5	802	73.3	69.9	80.5	71.0

higher efficiency with only minor performance degradation on most benchmarks. Specifically, by selecting only 6% patches for PS3, it maintains competitive performance of VILA-HD-4K while only using 1476 tokens for a maximum resolution of 4K which is less than 1/10 of #token of Qwen2-VL. We can see the performance stays similar compared to 18% patch selection for benchmarks such as TextVQA and RealWorldQA, and only has minor drops for CharQA and V*Bench. Tasks like DocVQA require denser visual information and thus have larger performance drop.

Comparing PS3 to State-Of-The-Art Token Pruning Methods

Experiment settings. We compare the efficiency of PS3 on high-res images with state-of-the-art token pruning methods for MLLMs including ToMe [20], FastV [34], and VisionZip [241]. We test the models in both 1512 and 3780 resolution and try different token selection ratio for each resolution. All the baselines use PS3 as the vision encoder for a fair comparison

Table 3.6: **Comparing PS3 to state-of-the-art token pruning methods.** PS3 has consistently lower ViT latency and achieve better performances than previous methods. PS3 is also the only method that can handle 4K resolution images.

Method	Select (Test)	ViT Latency	LLM Latency	Text VQA	Chart QA	Doc VQA	Info VQA	OCR Bench	V* Bench	Real World	Avg
<i>1512 Resolution</i>											
Full	100%	0.286s	0.375s	78.6	84.1	92.2	68.1	787	67.9	69.8	77.1
ToMe [20]	50%	0.286s	0.260s	74.1	70.2	59.7	47.3	622	66.8	67.2	63.9
FastV [34]	50%	0.286s	0.264s	78.2	81.2	90.0	60.4	769	66.2	69.0	74.6
VisionZip [241]	50%	0.286s	0.260s	75.2	77.2	79.8	55.7	722	64.0	67.1	70.2
PS3	50%	0.167s	0.260s	77.7	83.4	89.8	60.8	774	67.9	69.1	75.2
ToMe [20]	25%	0.286s	0.180s	72.5	65.5	51.7	42.8	61.1	62.2	63.4	59.9
FastV [34]	25%	0.286s	0.185s	76.1	66.3	78.1	49.5	651	64.6	65.2	66.6
VisionZip [241]	25%	0.286s	0.180s	74.6	76.0	72.8	51.5	694	62.7	64.6	67.4
PS3	25%	0.096s	0.180s	76.8	80.4	84.4	54.6	738	65.7	67.8	71.9
<i>3780 Resolution</i>											
Full	100%	1.812s	OOM	-	-	-	-	-	-	-	-
ToMe [20]	20%	1.812s	OOM	-	-	-	-	-	-	-	-
FastV [34]	20%	1.812s	OOM	-	-	-	-	-	-	-	-
VisionZip [241]	20%	1.812s	OOM	-	-	-	-	-	-	-	-
PS3	20%	0.417s	0.383s	77.8	83.9	91.6	65.0	773	72.8	70.1	76.9

but instead of using the PS3 patch selection module, they use PS3 to process the whole image and prune the tokens in their own way. The PS3 encoder is pre-trained on top of SigLIP-SO400M as in Section 3.4.

Results. Full results are shown in Table 3.6. First of all, we can see that PS3 achieves similar efficiency on the LLM backbone when pruning the same number of tokens, while significantly reduces the ViT latency (*e.g.*, 0.286s $\rightarrow$ 0.096s when selecting 25% patches at 1512 resolution). This is because PS3 reduces the tokens in both ViT and LLM while other methods only prune the tokens in LLM. This advantage is especially important for 4K-resolution images where the ViT latency dominates the LLM latency. For the same reason, the previous methods all run out of memory on 4K resolution images while PS3 is able to process 4K resolution efficiently. In the meantime, PS3 achieves superior results over all previous methods under the same selection ratio. This is due to several possible reasons. For example, the previous methods are heuristic-based, either merging visual tokens based on their similarities or pruning tokens based on the attention score in the vision encoder or LLM backbone, while PS3 adopts a learning-based approach to select important or relevant visual tokens, which is more accurate than heuristic-based methods given enough data. On the other hand, unlike previous methods such as ToMe and VisionZip that are prompt-agnostic, PS3 is able to select patches based on the user prompt which can be more accurate.

Key Finding 4: PS3’s learning-based top-down patch selection achieves better performance and efficiency than heuristic-based token pruning methods.

Table 3.7: **Generalizability of PS3 pre-training to state-of-the-art vision encoders.** $\#Param$ of ViT is the number of parameters of the ViT backbone. $Max Res.$ is the maximum resolution each model processes in MLLM. $Max \#Tok$ is the maximum number of vision tokens in MLLM. All the PS3 models select all the high-res patches.

Vision Encoder	$\#Param$ of ViT	Max Res.	Max $\#Tok$	Text VQA	Chart QA	Doc VQA	Info VQA	OCR Bench	V* Bench	Real World	Avg
SigLIP-SO400M [256]	400M	378	196	62.3	56.6	51.9	30.7	387	51.8	57.1	<i>49.9</i>
+ AnyRes [124]	400M	1512	3332	67.4	58.4	67.9	34.1	468	60.2	59.0	<i>56.3</i>
+ S ² [175]	400M	1512	2916	66.1	71.0	78.3	41.1	526	55.2	61.0	<i>60.8</i>
PS3-SigLIP-SO400M	400M	1512	3841	69.3	71.1	79.4	41.3	534	64.0	63.8	63.2
C-RADIOv2-L [160]	320M	384	144	65.0	58.8	53.1	30.9	405	51.5	57.5	<i>51.0</i>
+ AnyRes [124]	320M	1536	2448	68.1	62.8	70.0	35.8	497	65.9	62.8	<i>59.3</i>
+ S ² [175]	320M	1536	2304	68.1	72.3	82.5	40.4	542	59.7	62.1	<i>62.8</i>
PS3-C-RADIOv2-L	320M	1536	3024	68.4	72.6	83.2	43.4	569	68.2	61.5	64.9

Generalizability of PS3 Pre-Training to State-Of-The-Art Vision Encoders

We verify the generalizability of PS3 pre-training pipeline by pre-training PS3 on top of several state-of-the-art vision encoders and compare with the same vision encoders without high-res pre-training. Specifically, we take the state-of-the-art language-aligned vision model, SigLIP-SO400M [256], and agglomerative vision model, C-RADIO-v2-L [160], as the baselines. We take their pre-trained model and continue pre-training on high-res images using PS3 pre-training pipeline to get PS3-SigLIP-SO400M and PS3-C-RADIO-v2-L. We compare them with the original models as well as their AnyRes [124] and S² [175] variants. Table 3.7 shows the results. We can see that for both the base models, the S² variant achieves the best performance compared to AnyRes variant as well as the original version, while PS3 encoders pre-trained on top of each base model are able to achieve better results, showing that PS3 is a general high-res pre-training pipeline that can be applied to any base vision encoder. We also observe that C-RADIO baseline consistently outperforms SigLIP baseline, and PS3-C-RADIO-v2-L is able to inherit that advantage and outperforms PS3-SigLIP.

3.7 Ablations and Analysis

Key Designs in Pre-Training Algorithm and Model Architecture

We conduct ablation studies on the pre-training algorithm designs (Section 3.2), PS3 model designs (Section 3.2), and MLLM model design (Section 3.3). We follow the same experimental setting as in Section 3.4 and use PS3 with max resolution of 1512 and 100% patch selection as the baseline. For the ablation of each design, we report the improvement of the baseline model on the average accuracy of the seven benchmarks compared to the model without the design. Results are shown in Table 3.8. Overall all the designs are helpful. Among

the pre-training algorithm designs, we can see that it is crucial to select the patches in the ground-truth boxes during patch selection and pool only their corresponding tokens before calculating the contrastive loss. This is because otherwise the model will select irrelevant patches and contrast them with the local caption, leading to noisy pre-training and degrading the performance by over 8%. We also find avoiding contrast between local regions in the same image improves the performance by 5.3% which aligns with the observation in previous work [32]. The model architecture designs also help with the performance. Notably, the low-res KV cache allows the model to see the global context while extracting the local high-res features, which significantly improves the performance by 8.8%. Additionally, extracting features at multiple scales (*e.g.*, 756 and 1512) performs better than only extracting features at the largest scale (*e.g.*, 1512 only) as in AnyRes. Adding scale-aware positional embeddings in PS3 and additional vision positional embeddings when feeding PS3 features to LLM also improve the performance by a small margin.

Key Finding 5: High-resolution pre-training with localized contrastive learning requires several key algorithm designs, such as contrasting within the ground-truth selection boxes and avoiding contrasting between different regions in the same image, and key model designs including low-res KV cache that makes high-res features aware of the global context.

Top-Down and Bottom-Up Patch Selection Matters

We compare the performance of using random, bottom-up, and top-down patch selection, as shown in Table 3.9. We report benchmark accuracy as well as the patch recall rate which evaluates how many patches out of the ground-truth region are selected at test time. This is evaluated on a test set split from the data used to train patch selection in MLLM. We can see that both bottom-up and top-down selection significantly improves the recall and the accuracy over random selection. For example, when selecting 44% patches, bottom-up selection improves the recall rate by 43.7% and the accuracy by 7.4% over random selection. Top-down selection further improves the recall rate by 3.8% and the accuracy by 1.3%. Interestingly, patch selection not only affects test time but training as well. When we train MLLM with 44% patches, even if we select all patches at test time, top-down selection is still better than the other two. The reason is likely that top-down selection provides more informative visual context for MLLM during training, which leads to less noisy training dynamics. Note that this model has performance comparable to training with 100% patches, which means that training with top-down patch selection improves training efficiency without hurting the performance too much.

Key Finding 6: Top-down and bottom-up patch selection provides more relevant visual information to MLLM, which does not only improve performance at test time but also improve the model optimization during training.

Table 3.8: **Ablation of PS3 pre-training, model, and MLLM designs.** Δ is the change of the average performance on the seven benchmarks after adding the design.

Training and Model Design Choices	Δ
<i>Pre-training algorithm designs (Section 3.2)</i>	
- using ground truth selection score	+5.1
- pooling only tokens in ground-truth boxes	+3.7
- mixing global and local contrast	+1.0
- w/o intra-image contrast	+5.3
<i>PS3 model designs (Section 3.2)</i>	
- multi-scale feature extraction	+1.0
- scale-aware pos. emb.	+0.8
- low-res KV cache	+8.8
<i>MLLM model design (Section 3.3)</i>	
- additional vision pos. emb.	+0.8

Table 3.9: **Ablation of top-down and bottom-up patch selection.** *Select (Train)* and *Select (Test)* are the percentage of high-res patches PS3 selects at training and test time. *Recall* is the recall rate of how many patches in the ground-truth regions are selected at test time.

Patch Selection	Select (Train)	Select (Test)	Recall (Test)	Avg. Acc.
Random	44%	44%	43.7%	52.3
Bottom-up	44%	44%	87.4%	59.7 (+7.4)
Top-down	44%	44%	91.2%	61.0 (+8.7)
Random	44%	100%	100%	56.5
Bottom-up	44%	100%	100%	61.1 (+4.6)
Top-down	44%	100%	100%	61.9 (+5.4)
Top-down	100%	100%	100%	63.2

Trade-Off Between Different Image Scales

We find the optimal balance of patch selection between image scales varies for different tasks (Figure 3.13). For example, when using in total 729 (20%) high-res tokens, V*Bench performance peaks when selecting no patches (0%) at 756 scale and only patches (25%) at 1512 scale, while it performs the best on DocVQA when selecting 67% 756-scale patches and only 8% 1512-scale patches. Note that selecting more patches at 756 scale covers more regions of the image because one 756-scale patch represents larger area in the original image than a 1512-scale patch, but lose more details at the 1512 scale. This indicates V*Bench needs smaller coverage of image regions but requires more high-res information at 1512 resolution

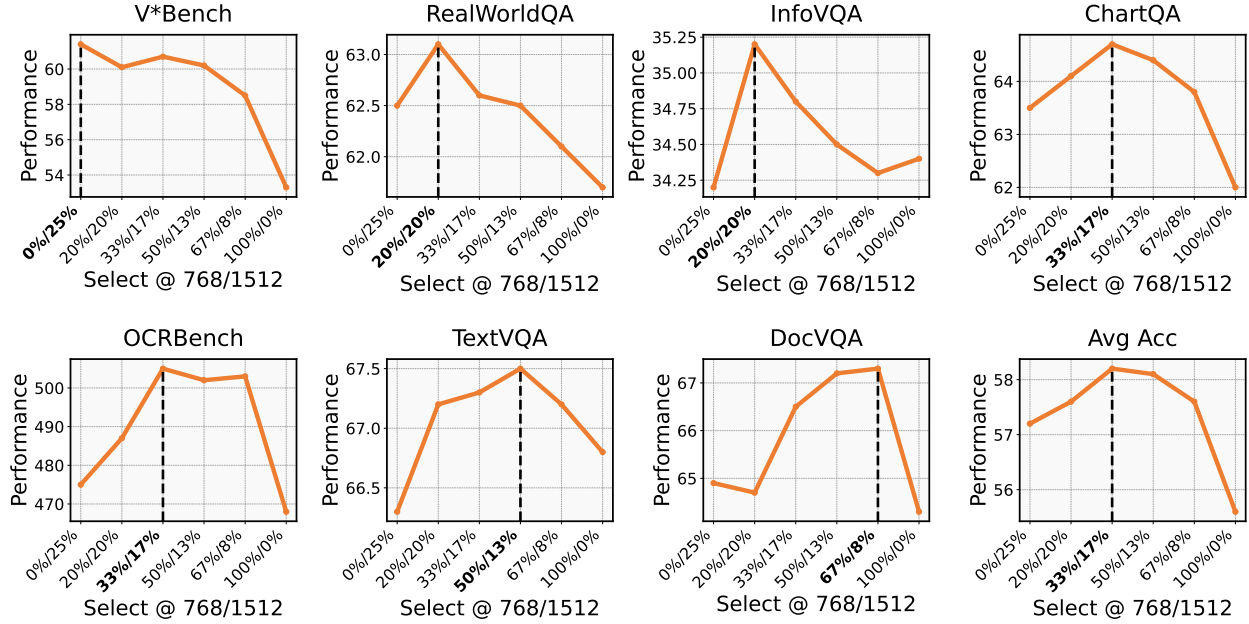


Figure 3.13: **Trade-off between image scales for different benchmarks.** *Select @ 756/1512* are the percentage of selected patches at 756 and 1512 scales at test time, respectively. PS3 can flexibly adjust token selection ratios at different image scales to achieve the best performance for different downstream tasks.

to perform well. DocVQA, on the other hand, needs more coverage of the image, which is probably because one usually needs to read the whole document to answer the questions. We also observe degraded performance on all benchmarks when selecting only tokens at 756 scale, which is because not all tokens at 756 scales are relevant to the question while the relevant information from 1512 scale is completely lost in this way.

Key Finding 7: The optimal balance of patch selection between different image scales varies for different tasks. Selecting from smaller scales covers more image regions, which is better for tasks with denser visual information. Selecting from larger scales covers less region but captures more detailed information, which is favored by tasks that only require localized understanding.

SFT Data for High-Resolution Feature Alignment

We study the effect of high-resolution SFT data for MLLMs using PS3 as the vision encoder. Commonly-used image QA data normally contains only low-res images or lacks questions about details in high-res images, from which it is hard for MLLM to learn to utilize the high-res vision features. We hypothesize the high-res QA data in Section 3.3, though generated

Table 3.10: **Ablation of MLLM SFT data for high-resolution feature alignment.** The high-resolution SFT data (*HR Data*) generally improves high-resolution perception, especially on natural images.

Vision Encoder	Max Res.	HR Data	Text VQA	Chart QA	Doc VQA	Info VQA	OCR Bench	V* Bench	Real World	<i>Avg</i>
PS3	1512	✗	68.8	71.2	79.6	39.6	535	60.9	62.9	<i>62.4</i>
PS3	1512	✓	69.3 (+0.5)	71.1 (-0.1)	79.4 (-0.2)	41.3 (+1.7)	534 (-0.1)	64.0 (+3.1)	63.8 (+0.9)	<i>63.2</i> (+0.8)

in a naive way, can help align the high-res visual features to the text space of LLM, thus improving the high-res perception capability. We conduct an ablation study in Table 3.10. We find that the high-res SFT data in general helps with the performance on resolution-sensitive benchmarks. The improvements are especially significant on natural images, *e.g.*, a 3.1% improvement on V*Bench. On the other hand, the performance on ChartQA and DocVQA does not change, which is probably because the current high-res SFT data does not contain document images.

Visualization of PS3 Visual Representations

We visualize the visual features in PS3 using principal component analysis (PCA) and compare with the baselines of S² and AnyRes that does not pre-train vision models at high resolutions. Results are shown in Figure 3.14. We visualize the features of SigLIP-S² and SigLIP-AnyRes at 3780×3780 resolution and the features from three different scales (756×756, 1512×1512, and 3780×3780) for PS3. We can see the feature map of SigLIP-S² is blurry compared to other models because S² concatenates both low-res and high-res features and the PCA visualization is partially dominated by the low-res components. The AnyRes feature map, which is equivalent to S² but only with the high-res features, shows noisy patterns. One possible reason is AnyRes splits the high-res image into small tiles and individually processes each, which means the features lack global context and are inconsistent when running PCA on the whole feature map. On the other hand, PS3 shows sharp and fine-grained features at 4K resolution, *e.g.*, at the edges of objects. We also observe PS3 features from different scales tend to group pixels at different scales. For example, the features at 756 and 1512 scales group large-scale objects or things such as the dock and the sky, while the features at 3780 resolution usually group small-scale patterns such as the texts on the banners.

3.8 Related Work

Scaling up vision pre-training. The enormous efforts from the past few years in both supervised pre-training [43, 47, 55, 62, 76, 90, 132, 154, 255, 256] and self-supervised pre-

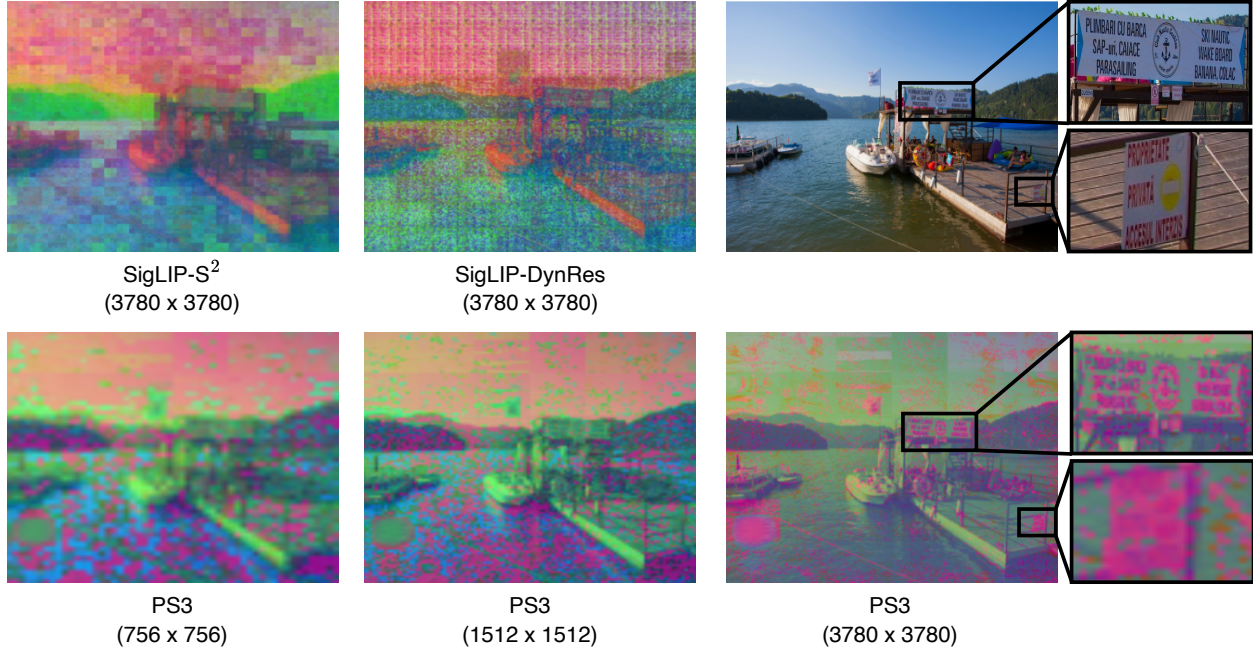


Figure 3.14: **PCA visualization of visual features.** The baselines, S² and AnyRes, have either noisy or blurry features at 4K resolution, while PS3 shows extremely fine-grained features that highlight details such as small texts on the banners.

training [7, 26, 36, 37, 66, 85, 91, 92, 147] have successfully scaled up the size of vision model to billions of parameters and pre-training data to billions of images, which is proven effective in numerous downstream tasks [104, 105, 156, 159, 229, 240]. However, all these vision models are pre-trained at a low resolution (*e.g.*, 384×384) and unable to scale up to higher pre-training resolution due to the costly processing of full images. Previous work has studied the importance of scaling up input resolution in vision pre-training [16, 59, 185, 186] and attempted at incorporating high-resolution pre-training data [70, 129, 132, 147]. However, they are mostly limited to tasks such as ImageNet classification [167] that are less sensitive to input resolution by nature, and none of these work has tried scaling pre-training to ultra-high resolution (*e.g.*, over 1K resolution) which proves to be important for various real-world applications including MLLMs [60, 175, 191, 206]. RADIO [93, 160] uses a maximum pre-training resolution of 1024×1024 although it does not explore CLIP-style pre-training and is pre-trained by distilling from other pre-trained vision encoders and thus is limited by the resolution of the teacher encoders. Another thread of work tries to improve fine-grained visual perception via localized contrastive learning [18, 32, 184] but none of these work scales up the pre-training resolution with the proposed approaches. In this work we scale up vision pre-training to 4K resolution and show the significant improvement on different MLLM benchmarks.

High-resolution visual processing in MLLMs. The success of LLMs [10, 145, 146, 197] has spurred the progresses of MLLMs [42, 111, 120, 124, 125, 176, 206] with evolving

recipe [69, 130, 194, 221] and functionality [29, 44, 87, 96]. Recent work shows the importance of high-resolution perception in MLLMs [42, 116, 124, 175, 235] for tasks that require fine-grained detail perception. Different approaches to high-resolution image processing for MLLMs have been proposed, including splitting the high-res image into smaller tiles and processing each tile separately with a pre-trained low-resolution vision model [42, 118, 124, 175], using multiple vision models to encode an image at different resolution [116, 137, 176, 194], and directly using one single encoder to process high-res images [200, 206]. However, most of these work does not scale up image resolution during the vision pre-training stage and directly processes high-res images with vision encoders pre-trained on low-res images, which is shown to be suboptimal [175]. Furthermore, all the previous methods exhaustively process every pixel in the high-res images without any saliency-based or prompt-aware region selection, leading to extreme inefficiency and rendering them unable to process ultra-high resolution such as 4K resolution. In contrast, PS3 scales up to 4K resolution during the pre-training stage while using a top-down patch selection mechanism that significantly improves the efficiency.

Token pruning for vision encoders and MLLMs. Redundancy in visual information often leads to inefficiency encoding in vision models and MLLMs. Previous work has explored improving the efficiency of vision encoders by pruning redundant vision tokens [20, 38, 161, 201, 246], although the models are all optimized for a specific task such as image classification and cannot adapt to general VQA tasks where token dropping should be based on the input prompt. AbSViT [172] learns to focus on different image regions based on any text prompt but it does not prune irrelevant tokens and is more computationally expensive. Other work has explored adaptively focusing on important regions for fine-grained image recognition [218, 219] although is still limited to low resolution. With the recent progress of MLLMs, visual token pruning methods for MLLMs have been proposed [34, 233, 241, 265] which removes redundant tokens based on either the self-attention in ViT or the text-image cross attention inside LLM. However, these methods still need to process the whole image in the ViT, resulting in large ViT latency especial for high-res images and making it unable to scale up to 4K resolution. SEAL [228] processes high-res images in a search-and-focus style although the searching process is guided by LLM which adds on huge latency. PS3, on the other hand, processes 4K resolution images with significantly improved efficiency and demonstrates superior performance over previous methods.

3.9 Current Limitations and Future Directions

Current Limitations

Data. Designs of the pre-training data are not extensively studied in this work due to limit of resources: **1) Quality of high-res images.** It is not enough to have images that are just high-res. They should contain rich details as well so the model has the incentive to learn fine-grained representations. This is supported by a preliminary experiment where we

compare the effect of SA-1B and DataComp data in pre-training. Specifically, we separately remove the SA-1B and DataComp data from pre-training and find that removing SA-1B has larger impact on the performance despite its smaller size than DataComp. We hypothesize this is because the images in SA-1B usually contains dense or small objects while DataComp images usually have large or even single object in spite of the high resolution. In this work, we do not add any filtering on the richness of visual details in the images, which is necessary for future improvement. **2) Quality of local bounding boxes of salient regions.** In this work, we detect salient regions that contain dense or small SAM masks. This heuristic can be improved, for example, by running OCR model to detect regions with texts for text-heavy downstream applications or by filtering out SAM masks that only contain textures but not meaningful objects. **3) Quality of local captions.** We use off-the-shelf MLLMs to generate local captions without any data cleaning or filtering. Future improvements can include borrowing the data filtering techniques from LLM synthetic data generation or using visual fact checkers [82] to ensure accuracy of the generated captions. Furthermore, the quality of high-res SFT data for MLLM also needs improving. Currently we vanillaly generate high-res QA data from low-res data, which does not reflect the real distribution of high-res images in the wild. Manual or automatic curation of QA pairs on natural high-res images is better.

Training. In the current model, patch selection is supervised by labeled bounding boxes of salient regions. This can have several drawbacks. For example, since we first generate boxes and then generate captions from the boxes, it is possible the caption also corresponds to other regions outside the box but we treat those regions as negative when training patch selection. On the other hand, the current patch selection only select important regions and process all the scales in those regions, while it is usually the case that not all the scales need to be processed. Learning to select which scale to process is hard to be supervised because of lack of labeled data. All these issues can be addressed by using, for example, reinforcement learning instead of supervised learning where the model learns to find an optimal patch selection strategy itself in order to maximize the downstream performance.

Model. There are several defects in the current model design. First, when we run Stage 2 and Stage 3 of PS3 for multiple times, each group of high-res patches only have the context of the low-res patches but not high-res patches from other groups. This might affect PS3’s ability to model long-range correlation of high-res details. On the other hand, when selecting high-res patches, the selected patches can be scattered around the whole image and sparse within a local region. The sparsity of information affects the quality of local features. This is remedied by adding a soft locality constraint on patch selection, *e.g.*, the model is encouraged to select more centralized and clustered patches. We verify this by adding gaussian smoothing on the selection score map such that patches around any high-score patch also have high scores and each group of selected patches is more clustered. We find this improves the performance when selecting all the patches, supporting our hypothesis. Nevertheless, this degrades the performance when selecting only parts of the patches, which is possibly because the selection score of salient objects is smoothed out if they are small.

Future Directions

Beyond vision-language pre-training. While PS3 follows the paradigm of CLIP-style vision-language pre-training, the same idea can be applied to any vision pre-training paradigm such as self-supervised pre-training. Taking DINOv2 [147] as an example, the original pre-training objective is to improve the representation consistency of two different views of the image, and using the similar idea of localized processing, one can scale DINOv2 pre-training to 4K resolution by applying the DINOv2 objective to local regions. Furthermore, despite the recent progress on vision expert aggregation [93, 116, 126, 160, 176, 195], it is still an open question of how to combine vision-language and self-supervised pre-training by either joint or sequential training and how it affects performances of MLLM and other downstream tasks. Ideally, pre-training on 4K resolution images with both types of objectives can help the model learn both verbalizable and nonverbalizable features, for example, action-related features for robotic tasks.

Beyond images. PS3 processes 4K-resolution images by removing the redundant patches. Videos, especially high-res, high-fps, and long-form videos, contain much more redundancy and thus can benefit more from the similar idea of patch selection. Furthermore, learning such a patch selection mechanism can force the model to better grasp the world knowledge behind the videos such as physics and dynamics. Although video pre-training still faces lots of problems such as lack of video data and high-quality captions, such bottom-up or top-down spatiotemporal patch selection is a promising way to scale up video pre-training.

3.10 Conclusion

We propose PS3 that scales up CLIP-style vision pre-training to 4K resolution with a near-constant cost. PS3 learns high-resolution perception through local-global region-caption contrast. It encodes low-resolution global image and selectively processes only informative high-resolution regions. PS3 enables VILA-HD, a high-res MLLM whose performance scales with the PS3 pre-training resolution and surpasses state-of-the-art MLLMs with better efficiency. We also introduce 4KPro, a benchmark that evaluates visual perception at 4K resolution, on which VILA-HD outperforms state-of-the-art MLLMs.

Chapter 4

Attend Before Attention: Efficient and Scalable Video Understanding via Autoregressive Gazing

Multi-modal large language models (MLLMs) have advanced general-purpose video understanding but struggle with long, high-resolution videos—they process every pixel equally in their vision transformers (ViTs) or LLMs despite significant spatiotemporal redundancy. We introduce **AutoGaze**, a lightweight module that removes redundant patches before processed by a ViT or an MLLM. Trained with next-token prediction and reinforcement learning, AutoGaze autoregressively selects a minimal set of multi-scale patches that can reconstruct the video within a user-specified error threshold, eliminating redundancy while preserving information. Empirically, AutoGaze reduces visual tokens by $4\times$ - $100\times$ and accelerates ViTs and MLLMs by up to $19\times$, scaling MLLMs to 1K-frame 4K-resolution videos and achieving superior results on video benchmarks (e.g., 67.0% on VideoMME). Furthermore, we introduce **HLVid**: the first high-resolution, long-form video QA benchmark with 5-minute 4K-resolution videos, where an MLLM scaled with AutoGaze improves over the baseline by 10.1% and outperforms the previous best MLLM by 4.5%. Project page: <https://autogaze.github.io/>.

4.1 Introduction

When observing a moving scene, humans don’t process every detail equally. Our eyes dart around to moving objects, capture fine details, and skip over static backgrounds, efficiently understanding scenes by selectively attending to informative regions [4, 100, 101, 164]. This allows us to process high-FPS, high-resolution video streams in real time. In contrast, modern video understanding models (e.g., multi-modal large language models (MLLMs) [12, 98, 111, 131, 190]) still process every pixel in every frame equally, wasting computation due to spatiotemporal redundancy in videos [31, 109, 182, 222, 225]. For example, in Fig. 4.2 (top-

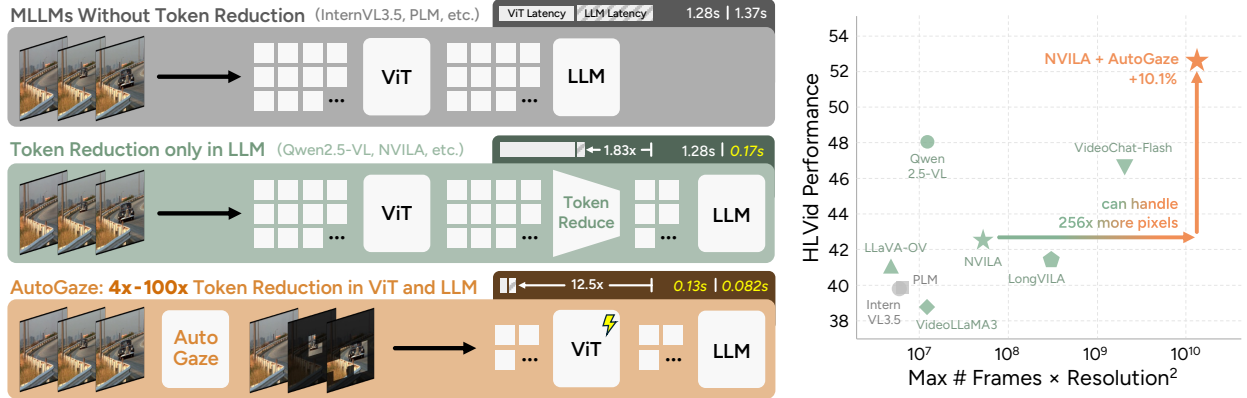


Figure 4.1: **AutoGaze reduces the computational cost of video understanding, scaling MLLMs to long, high-res videos.** (Left) Existing MLLMs either process all pixels or prune tokens only in their LLMs, leaving ViTs the bottleneck. In contrast, AutoGaze eliminates redundant patches by up to 100× *before* ViTs, accelerating ViTs/MLLMs by up to 19×. (Right) This enables scaling to 1K-frame, 4K-res videos, achieving superior performance on HLVID, our new long, high-res video benchmark, over MLLMs limited to short or low-res videos.

left), the static background only needs to be viewed once. Thus, these models cannot scale to *long-form* and *high-resolution* videos crucial for real-world applications [52, 56, 81, 163, 244] due to computational cost.

Recent work attempts to reduce video redundancy in MLLMs, but typically prunes tokens only in the LLM while the vision transformer (ViT) still processes all pixels, creating a huge efficiency bottleneck that prevents scaling to longer, higher-resolution videos [115, 170, 171, 187, 243] (Fig. 4.1). Moreover, these methods either rely on heuristics such as attention scores which underperforms learned approaches [174] or involves heavy search and reasoning that adds overhead and further limits scalability [207, 212, 245, 249].

To this end, we propose **AutoGaze**, a 3M-parameter lightweight model that attends to informative patches and removes redundant ones *before* a ViT. Specifically, AutoGaze perceives each frame and autoregressively selects a minimal set of *multi-scale* patches which, along with the selected patches from previous frames, can reconstruct the current frame within a user-specified reconstruction loss threshold. This model, pre-trained with next-token-prediction on a curated dataset of gazing sequences and post-trained with RL on reconstruction rewards, learns to focus only on newly emerged content while ignoring repeated information, and use multi-scale patches to cover broad areas coarsely, zooming in on fine details where needed. For example, Fig. 4.2 shows AutoGaze removing redundant patches in static regions and selecting coarser scales in low-detail areas. By only processing the selected multi-scale patches, both ViTs and LLMs are substantially sped up, unlocking efficient processing of long, high-resolution videos (Fig. 4.1).

Empirically, AutoGaze reduces the number of patches by 4×-100× for videos with different

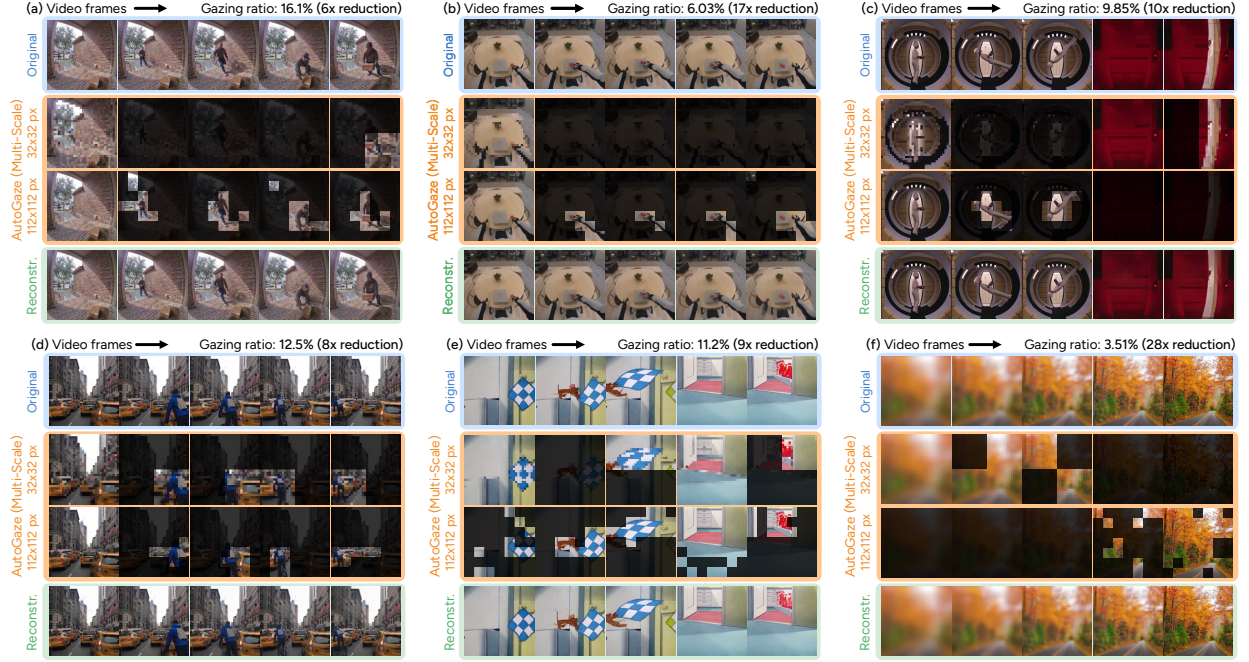


Figure 4.2: **What is AutoGaze paying attention to?** For each example, we show the original video, multi-scale gazed patches, and reconstructed video. Note that we only show gazing on two scales to save space while AutoGaze actually uses four. In general, AutoGaze can **1)** focus on moving objects while removing redundancy in static regions (a-e), **2)** adapt to scene changes by selecting more patches (c, e), and **3)** distribute attention with different granularity based on detailedness (c, f). This allows AutoGaze to select a small ratio of patches (gazing ratio) without much information loss, as reflected by the reconstruction quality.

FPS and resolution (*e.g.*, 1% patches for 30-FPS 4K-resolution videos) while maintaining downstream MLLM performance. This leads to up to $19\times$ and $10\times$ speedup for ViTs and MLLMs. Leveraging this efficiency, we scale an MLLM (NVILA [131]) to 1K-frame 4K-resolution videos, demonstrating consistent improvements on various benchmark (*e.g.*, 67.0% on VideoMME [78]) and outperforming strong MLLMs such as Qwen2.5-VL [12]. We also show that AutoGaze generalizes to videos with out-of-distribution styles and semantics.

Furthermore, noticing that existing benchmarks only focus on long videos but not high resolution [78, 139, 209, 226, 270], we introduce **HLVid**, the first high-resolution, long-form video QA benchmark, to stress-test AutoGaze’s scalability. It consists of 268 QAs about details in up to 5-minute, 4K-resolution videos, requiring visual perception at 1K - 2K resolution to solve. We show that scaling an MLLM [131] to 1K frames and 4K resolution via AutoGaze significantly improves its performance from 42.5% to 52.6%, outperforming the previous best MLLM [115] by 4.5% (Fig. 4.1).

4.2 Related Work

Video understanding and Long-Context MLLMs. Classical video understanding has long been driven by supervised or self-supervised video encoders including 3D-ConvNets and early transformers [6, 28, 71, 72], and pre-training algorithms such as masked auto-encoding [8, 14, 73, 196, 205], predictive coding [89, 158, 202], and large-scale vision-language pre-training [22, 214–216, 234, 237]. Recent MLLMs have extended these encoders to general-purpose video QA and captioning [12, 98, 111, 131, 190, 257]. However, these models usually operate on short, low-resolution clips due to costs of scaling to higher spatiotemporal resolution. While new long-video benchmarks [139, 226, 270] and models [39, 40, 115, 150, 261] emphasize extended temporal understanding, they remain limited to low resolutions due to inefficient whole-video processing, leaving a gap for methods and benchmarks that support *both* thousand-frame context and 4K-resolution detail under realistic compute constraints.

Token Reduction and Compression. A rapidly growing line of work has targeted ViT and MLLM efficiency by reducing input tokens. Spatial methods [20, 25, 114, 149, 161, 174, 242, 252] compress tokens or select informative patches based on attention scores or task relevance. Temporal methods reduce frame redundancy via sub-sampling [204], segment-level pooling [68, 162], or learned keyframe selection [188, 272]; spatiotemporal schemes such as STORM [102], FastVID [170], LongVU [171], and VideoChat-Flash [115], either simply pool tokens or use the ViT features to prune or aggregate tokens. However, all of these models only prune tokens inside the model or between the ViT and LLM, leaving part of the model still processing the full video at high cost. In contrast, AutoGaze removes redundant patches *before* the ViT, significantly improving efficiency. Other works on adaptive tokenization lean where to allocate tokens rather than using a fixed uniform grid [9, 64, 65, 238, 248]. However, their large tokenizer adds additional computational overhead and the tokenization is not adaptive to pre-trained ViTs.

4.3 AutoGaze for Efficient Video Understanding

Given a video, AutoGaze selects a minimal set of patches (i.e., “gazing”) which can reconstruct the video within a reconstruction loss threshold. Formally, for a T -frame video $\mathbf{X}^{1:T}$ where $\mathbf{X}^t$ is the t -th frame and each frame contains V patches, AutoGaze outputs a set of patch indices:

$$\text{AutoGaze} : \mathbf{X}^{1:T} \rightarrow p_{1:N^1}^1, \dots, p_{1:N^T}^T, \quad (4.1)$$

where $p_k^t \in \{1, \dots, V\}$ is the index of the k -th patch selected at frame t , and N^t is the number of selected patches (or “gazing length”) at frame t .

To select the minimal set satisfying the threshold, AutoGaze is able to select patches that minimize reconstruction loss under *any* $N^{1:T}$ and find the smallest $N^{1:T}$ satisfying the threshold. Formally, given any $N^{1:T}$, AutoGaze can predict patch indices $p_{1:N^1}^1, \dots, p_{1:N^T}^T$

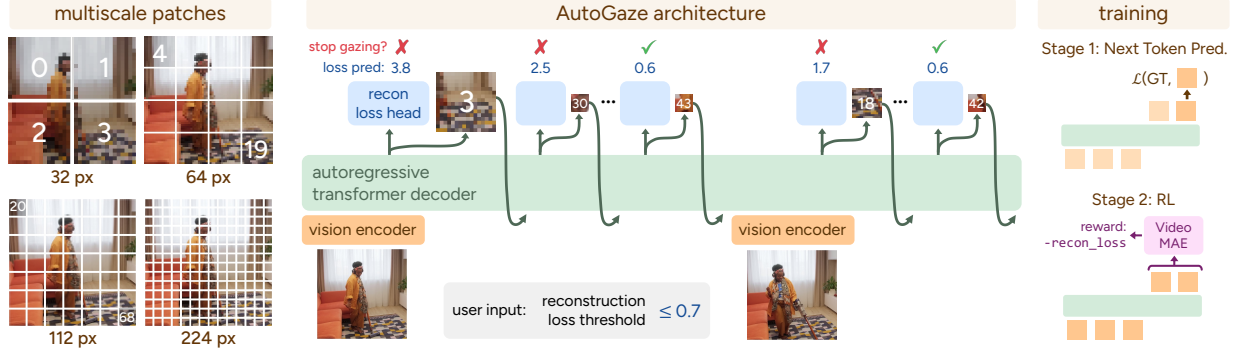


Figure 4.3: **Architecture and training pipeline of AutoGaze.** (Left & Middle) Given a video, AutoGaze processes each frame and autoregressively decodes indices of multi-scale patches based on the history of frames and selected patches. Once it believes the previously-gazed patches are sufficient to reconstruct the current frame, it automatically stops gazing and moves to the next frame. (Right) AutoGaze is trained in two stages: next-token-prediction pre-training on collected gazing sequences, and RL post-training with reconstruction reward.

that optimize

$$\min_{p_1^1, \dots, p_{N^T}^T} L(\mathbf{X}^{1:T}, \text{Recon}(\mathbf{X}^1[p_1^1], \dots, \mathbf{X}^T[p_{N^T}^T])), \quad (4.2)$$

where $\mathbf{X}^t[p_k^t]$ is the p_k^t -th patch in frame t , $\text{Recon}(\cdot)$ is a model that reconstructs the original video from the gazed patches, and $L(\cdot, \cdot)$ is a distance function between the original and the reconstructed videos. We instantiate $\text{Recon}(\cdot)$ as a custom VideoMAE [196] with block-causal attention, and $L(\cdot, \cdot)$ as a weighted sum of pixel reconstruction loss and perceptual loss [103, 262]. At the same time, AutoGaze can identify the smallest $N^{1:T}$ that satisfies $L^*(N^{1:T}) < \epsilon$ where $L^*(N^{1:T})$ is the optimal reconstruction loss under gazing lengths $N^{1:T}$ (Eq. 4.2) and ϵ is a user-specified loss threshold.

To achieve this, we build AutoGaze to **autoregressively select patch indices** that optimize reconstruction loss for any gazing length, while **automatically deciding the smallest gazing length** by predicting reconstruction loss on the fly and stopping once it falls below the threshold. Below, we introduce model design (Sec. 4.3), training pipeline (Sec. 4.3), how to apply it to videos of any duration and resolution, and integrate it into any ViT (Sec. 4.3), and a new benchmark to stress-test scalability (Sec. 4.3).

Model Design

Fig. 4.3 (Middle) illustrates AutoGaze’s lightweight design: a convolutional encoder and autoregressive transformer decoder, totaling 3M parameters.

Autoregressive gazing. Given a video, AutoGaze interleaves frame encoding and patch gazing. It starts by encoding the first frame with the convolutional encoder, passing the features to the decoder, and autoregressively decoding patch indices. The decoding process

mirrors LLMs except the vocabulary contains only patch indices $\{1, \dots, V\}$ instead of words. Next, AutoGaze encodes the second frame and decodes its patch indices based on *the features of both frames and the gazed patch indices of the first frame*. This lets the model avoid redundant patches by referring to frame and gazing history. The process repeats for subsequent frames.

Automatically deciding the gazing length. To identify the smallest N^t satisfying the reconstruction loss threshold, we add a head on the decoder that, when decoding every p_k^t , predicts the loss of reconstructing frame t from the patches gazed up to that step, *i.e.*, $\{p_1^1, \dots, p_k^t\}$. Once the predicted loss falls below the threshold, it stops gazing for that frame.

Multi-scale gazing. Considering that not all regions need full resolution (*e.g.*, solid-colored regions can be stored losslessly in low resolution), AutoGaze supports multi-scale gazing. The decoder’s vocabulary includes patches from multiple scales (Fig. 4.3 (Left)), letting the decoder select different scales for regions with different level of detail, reducing patches while preserving reconstruction quality (Sec. 4.4). This also requires the downstream ViT to accept multi-scale patches as input, which we detail in Sec. 4.3.

Multi-token prediction. We adopt multi-token prediction [83] by using multiple heads to output multiple patch indices and corresponding reconstruction losses at once, speeding up gazing with little performance loss (Sec. 4.4).

Training Pipeline

AutoGaze is trained to decode patch indices that minimize reconstruction loss at any gazing length and predict reconstruction loss at each step for automatic stopping. Inspired by modern LLM training [1, 86, 148, 152], we train AutoGaze in two stages (Fig. 4.3 (Right)). First, we pre-train with next-token prediction (NTP) on videos paired with ground-truth gazing sequences that are collected via greedy search to approximately minimize reconstruction loss. Next, since the pre-trained gazing quality is bounded by the sub-optimal gazing data, we further post-train AutoGaze using RL with reconstruction reward to discover gazing sequences with lower reconstruction loss. We also train reconstruction loss prediction in both stages to enable automatic stopping.

Pre-training with next-token-prediction (NTP). Given a dataset with pairs of video $\mathbf{X}^{1:T}$, gazing sequences $\{\tilde{p}_1^1, \dots, \tilde{p}_{N^T}^T\}$ that approximately minimize reconstruction loss under random gazing length $N^{1:T}$, and $\{\tilde{l}_1^1, \dots, \tilde{l}_{N^T}^T\}$ where $\tilde{l}_k^t$ records reconstruction loss of frame t after gazing at $\tilde{p}_k^t$, we pre-train AutoGaze with NTP cross-entropy loss

$$L_{NTP} = - \sum_{t=1}^T \sum_{k=1}^{N^t} \log \pi_{\theta}(\tilde{p}_k^t \mid \mathbf{X}^{1:t}, \tilde{p}_{1:N^1}^1, \dots, \tilde{p}_{1:k-1}^t), \quad (4.3)$$

where π_{θ} is the model and $\pi_{\theta}(\tilde{p}_k^t \mid \mathbf{X}^{1:t}, \tilde{p}_{1:N^1}^1, \dots, \tilde{p}_{1:k-1}^t)$ is the probability of decoding $\tilde{p}_k^t$ based on previous frames and gazing. We also supervise reconstruction loss prediction with an ℓ_2 loss using $\{\tilde{l}_1^1, \dots, \tilde{l}_{N^T}^T\}$. AutoGaze thus learns sub-optimal gazing at different gazing length and learns to predict reconstruction loss at each decoding step.

Post-training with RL. Since the pre-training data only contains sub-optimal gazing, we further improve AutoGaze with RL post-training, using a simplified, on-policy GRPO [133, 169] algorithm with reconstruction loss as reward:

$$L_{GRPO} = - \sum_{t=1}^T \sum_{k=1}^{N^t} \frac{\pi_{\theta}(p_k^t)}{\pi_{\theta_{detached}}(p_k^t)} A_k^t, \quad (4.4)$$

where $\pi_{\theta}(p_k^t)$ is short for the decoding probability of patch index p_k^t as in Eq. 4.3, $\pi_{\theta_{detached}}$ is π_{θ} without gradient, and advantage A_k^t is the return G_k^t normalized within the group of GRPO where $G_k^t = \sum_{\tau=t}^T \gamma^{N^t-k+\sum_{s=t+1}^{\tau} N^s} \cdot (-l_{N^{\tau}}^{\tau})$, i.e., sum of negative reconstruction loss of future frames discounted by γ . Additionally, we supervise reconstruction loss prediction at the last patch of each frame (i.e., $l_{N^t}^t$) using the actual reconstruction loss at frame t .

Training data curation. The pretraining pipeline above uses raw videos and paired gazing sequences. We first collect 800K videos spanning egocentric, exocentric, natural, and text-rich videos. Each video is sampled at 16 frames and 224x224. We then collect gazing sequences that approximately minimize reconstruction loss for 250K videos using greedy search. Specifically, we start from the first patch of the first frame and exhaustively find which patch gives the lowest reconstruction loss. We repeat this until reaching the first frame’s gazing length, then proceed to the second frame and so on. We also record reconstruction loss at each step to supervise loss prediction.

Downstream Usage of AutoGaze

Inference on videos with any resolution and duration. Despite being trained on 16-frame 224×224 videos, AutoGaze processes videos of any resolution and duration without additional training. Inspired by any-resolution MLLMs [42, 124, 173], we split the video into 16×224×224 spatiotemporal tiles, run AutoGaze on each tile, and merge the gazed positions back together, allowing AutoGaze to scale to 1K-frame and 4K-resolution videos (Sec. 4.4).

Integrating AutoGaze into ViTs and MLLMs. Current MLLMs typically encode each full frame using an image ViT [12, 131, 210]. To integrate AutoGaze, we make two changes. First, we allow ViTs to take multi-scale patch input by interpolating each frame and positional embeddings to different scales, running patch embedding on each scale separately, and then feeding embedded tokens from all scales to the ViT. Second, we repurpose image ViTs into video ViTs by letting them process tokens from all 16 frames in the same sequence. With these changes, AutoGaze selects multi-scale patches for a video, encodes them with a ViT, and the encoded tokens can be fed into MLLMs as usual.

HLVid: A High-Res, Long Video Benchmark

Although AutoGaze enables efficient understanding of long, high-resolution videos, benchmarks to evaluate this capability are still missing—current benchmarks [139, 179, 226] only focus on video length. We propose HLVid, the first long-form, high-resolution video QA

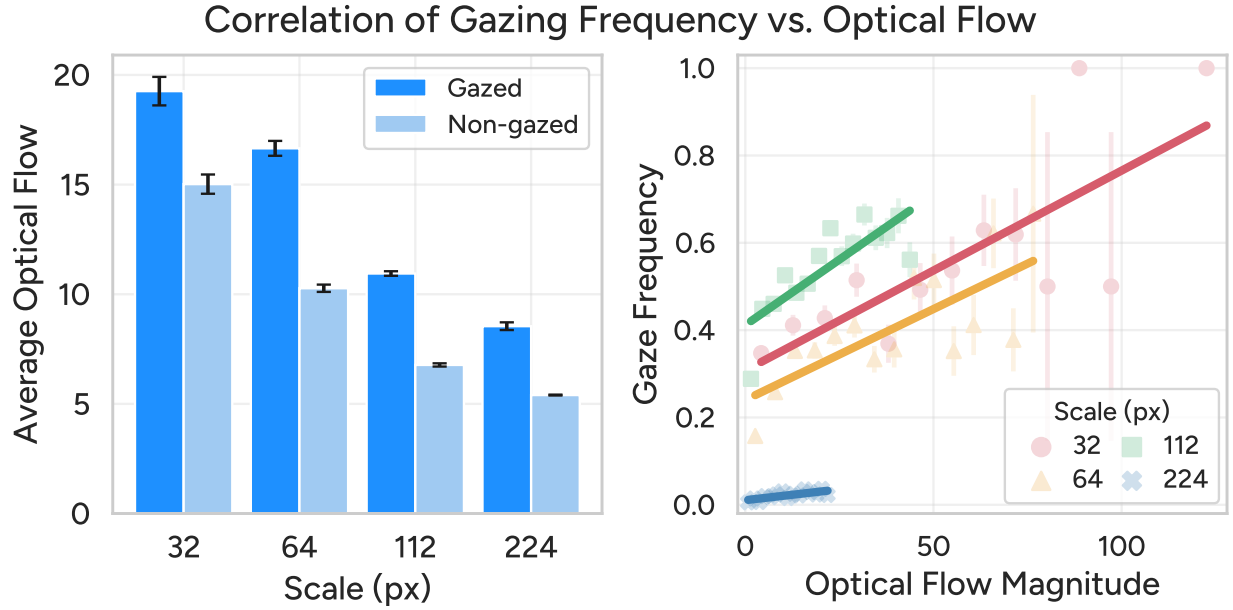


Figure 4.4: **AutoGaze targets patches with higher optical flow.** (Left) AutoGaze uses coarser scales to capture higher optical flow. (Right) Across all scales, AutoGaze more frequently selects patches with higher optical flow. Error bars represent SEM.

benchmark featuring 268 QA pairs on up to 5-minute, 4K-resolution videos. Each question is manually reviewed to ensure high resolution is required. We find that an MLLM scaled to 1K frames and 4K resolution via AutoGaze achieves significant improvement and unlocks state-of-the-art performance on HLVID (Sec. 4.4).

4.4 Experiments

We evaluate AutoGaze’s behavior, efficiency, and performance. Sec. 4.4 examines which patches AutoGaze selects or ignores and tests its generalization to unseen video styles and semantics. Sec. 4.4 measures its efficiency gains for ViTs and MLLMs. Leveraging the efficiency, Sec. 4.4 shows that AutoGaze enables higher-resolution and longer video processing in MLLMs with improved performance. Sec. 4.4 compares AutoGaze against gazing and MLLM token-reduction baselines, and Sec. 4.4 ablates training and modeling choices. We use SigLIP2-SO400M [198] and NVILA-8B-Video [131] as the ViT and MLLM by default.

What is AutoGaze paying attention to?

AutoGaze’s efficiency comes from selecting only a small fraction of patches — but does it make principled decisions about *which* patches to select and at *what* scale? We examine the

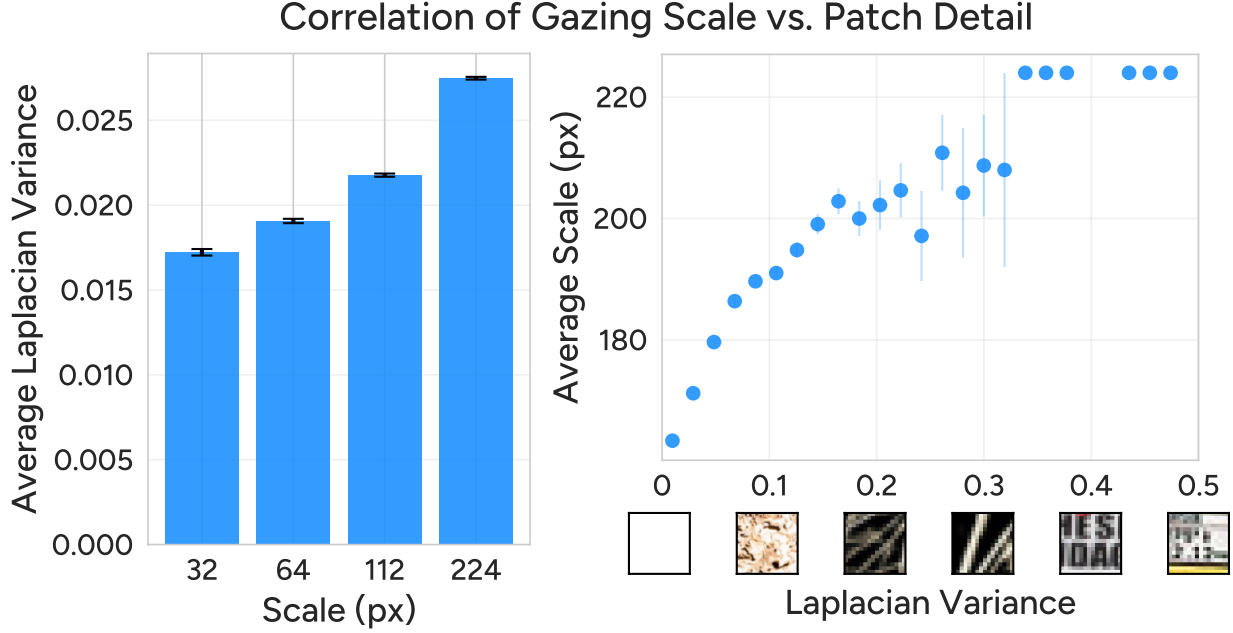


Figure 4.5: **Gazing scale correlates with patch detail.** (Left) At finer scales, AutoGaze selects more detailed patches (measured as Laplacian variance). (Right) With increasing detail, AutoGaze uses finer scales ($\rho = .12, p < 0.001$). Sample patches with Laplacian variances are shown below the x-axis. Error bars represent SEM.



Figure 4.6: **Generalizability of AutoGaze to OOD videos.** (a) We show model behavior on videos with OOD semantics, including a CCTV clip (left), robot grasping demo (middle), and a video with object swapping (right). In each example, AutoGaze still robustly tracks the changing parts despite the unseen semantics, object categories, and unexpected changes. (b) We show AutoGaze output on the same video with different style transfer. AutoGaze consistently tracks the falling person regardless of visual style, texture and global illumination.

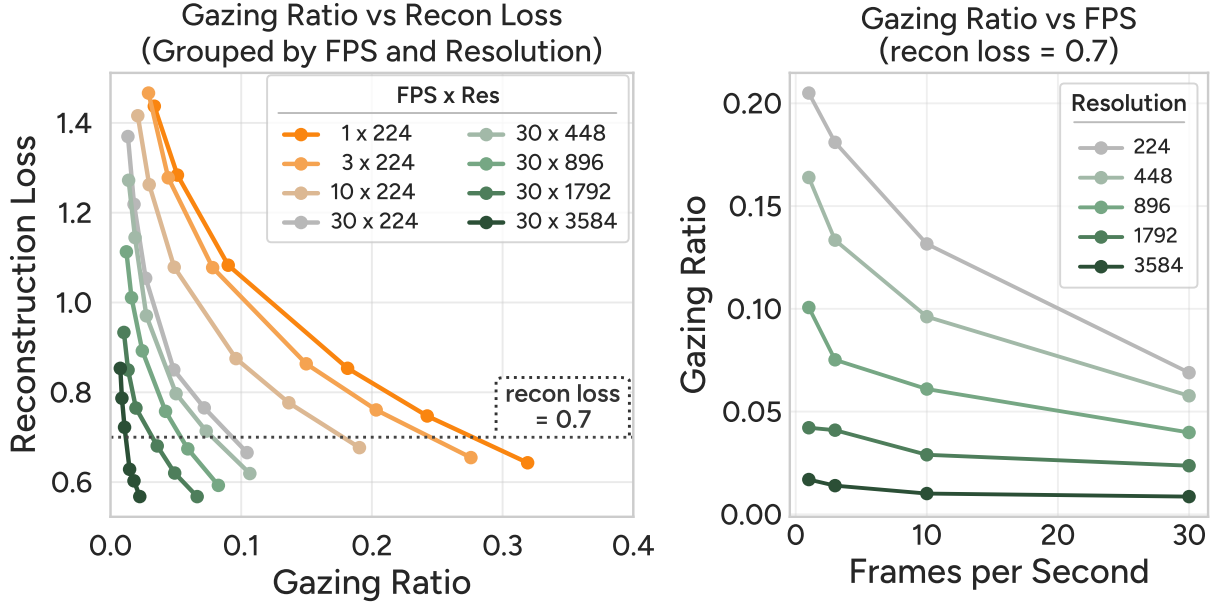


Figure 4.7: **What gazing ratio do we need for different video types?** (Left) There is a trade-off between gazing ratio and reconstruction loss: videos with higher FPS or resolution need lower gazing ratio to reach the same reconstruction quality. (Right) Gazing ratios required to reach a loss of 0.7 for videos with different FPS and resolution. 30-FPS, 4K-res videos only need $\sim 1\%$ patches.

factors that influence the behavior of AutoGaze and its generalization to videos with unseen styles and semantics.

AutoGaze gazes more at moving patches. Motion is a primary source of new information across video frames, and thus should intuitively be selected (examples are shown in Fig. 4.2). As illustrated in Fig. 4.4, AutoGaze does indeed prioritize motion: tested on pairs of videos and flow data from FlyingChairs [61, 99], we find that across all scales, it more frequently selects patches with higher optical flow.

AutoGaze uses finer scales for more detailed patches. Regions with different detailedness should be represented with different scales, as illustrated in Fig. 4.2. To verify this, we measure the relationship between gazing scale and patch detail by convolving 2,000 ImageNet images [57] with a Laplacian kernel and computing variance over each patch (higher values indicate more detail). Fig. 4.5 (left) shows that at finer scales, AutoGaze tends to select more detailed patches. Fig. 4.5 (right) confirms that AutoGaze gazes at higher resolutions to capture fine detail.

AutoGaze generalizes to OOD videos. We test whether AutoGaze transfers beyond its training distribution to unseen semantics and styles, as shown in Fig. 4.6. First, we show that AutoGaze behavior holds in unconventional scenarios including a CCTV footage, a

robot video, and a video [157] that constantly swaps its foreground object between human, gorilla, and humanoid robot (created with Luma’s Ray2 Flash). In each example, AutoGaze successfully tracks changing regions despite the novel semantics or unexpected changes. Next, we test on unseen styles by style-transferring a video with TokenFlow [151] to vary texture and global illumination. Across styles, AutoGaze maintains consistent gazing patterns, continuing to track the falling subject.

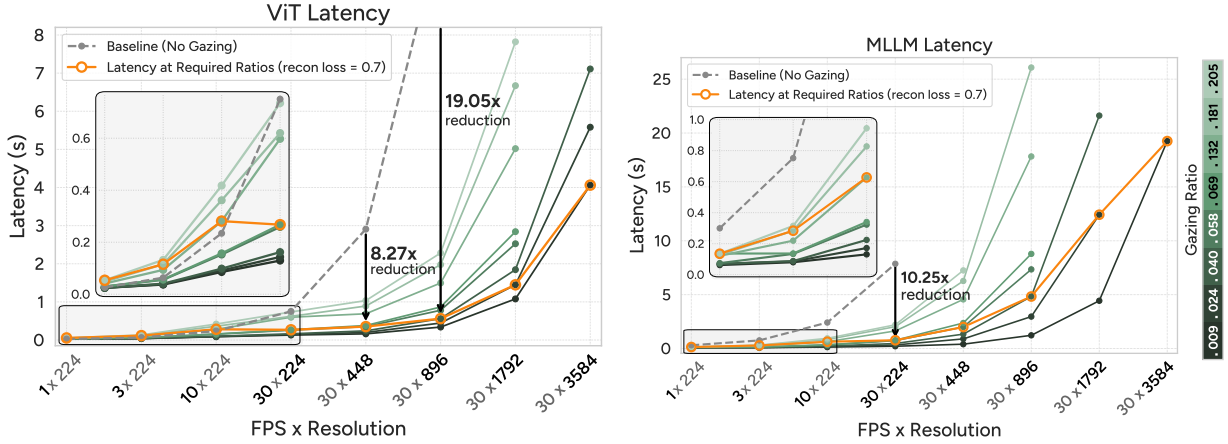


Figure 4.8: **Efficiency gain on ViTs and MLLMs with AutoGaze.** We benchmark the ViT and MLLM latency of encoding one second of video with varying FPS and resolution. AutoGaze can select different numbers of patches to vary latency depending on user needs. When using the gazing ratio required for a reconstruction loss of 0.7, AutoGaze reduces the ViT and MLLM latency by up to 19 $\times$ and 10 $\times$.

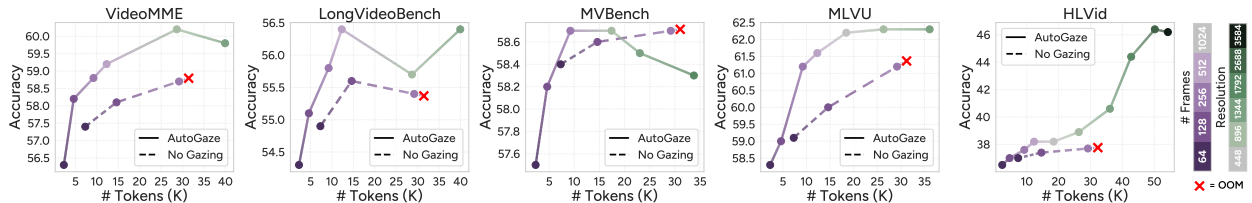


Figure 4.9: **Scaling MLLMs to more video tokens.** AutoGaze enables scaling to longer videos and higher resolution, while the baseline runs out of memory beyond 256 frames. Performance boosts are especially clear with HLVID, where high-resolution video processing is required.

Efficiency of ViT and MLLM with AutoGaze

We now study how efficient ViTs and MLLMs can be by selecting fewer patches via AutoGaze. To answer this question, we first analyze the number of patches required to represent a video

with AutoGaze, and then benchmark the latency of ViT and MLLM when only the selected patches are processed.

How many patches do we need to represent a video? The number of patches needed depends on both the required reconstruction loss and the level of redundancy (e.g., different FPS and resolution) in the video. We first pinpoint the reconstruction loss that leads to minimal performance drop in downstream MLLMs, and find that a threshold of 0.7 usually leads to less than 0.5% performance degradation across benchmarks. Next, we analyze how many patches are needed to represent videos with varying FPS and resolutions in order to achieve a reconstruction loss of 0.7. Fig. 4.7 (Left) shows the reconstruction loss for different gazing ratio and different FPS and resolution. We can see the gazing ratio required for a certain loss decreases with higher FPS and resolution. Fig. 4.7 (Right) shows complete results of gazing ratios required to reach a loss of 0.7 for different videos. Usually a video can be represented with $4\times$ - $100\times$ fewer patches. Specifically, only $\sim 1\%$ patches are needed for 30-FPS, 4K videos.

How much faster are ViTs and MLLMs with AutoGaze? With a target reconstruction loss of 0.7, we test wall-clock ViT and MLLM latency when processing one second of video. We use FP32 and disable flash attention. We report aggregated latency of AutoGaze and ViT / MLLM, and compare to the baseline without gazing in Fig. 4.8. The ViT baseline runs out of memory around 30 FPS and 896 resolution, and the MLLM baseline can only encode 30 FPS and 224 resolution. In contrast, AutoGaze helps efficiently process videos with lower gazing ratios. When using the gazing ratio required for a reconstruction loss of 0.7, it achieves up to $19\times$ and $10\times$ speedup for ViTs and MLLMs respectively.

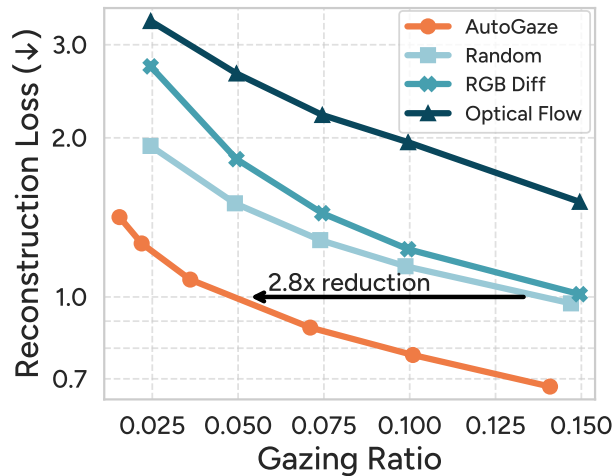


Figure 4.10: **Comparing to baseline gazing methods.** AutoGaze can use lower gazing ratios to reach the same reconstruction loss, compared to other heuristics-based approaches.

Table 4.1: **Comparison to state-of-the-art MLLMs.** NVILA-8B-Video with AutoGaze is scaled to 1K-frame, 4K-resolution videos, achieving competitive performance on general and long video benchmarks and state-of-the-art result on HLVID.

Models	Max #F	Max Res.	general video				long video			high-res & long
			VideoMME (w/o sub)	VideoMME (w/ sub)	MVBench (test)	NExT-QA (mc)	L-VidBench (val)	EgoSchema (test)	MLVU (m-avg)	HLVID (test)
Gemini 1.5-Pro [190]	-	-	75.0	81.3	60.5	-	64.0	71.2	-	-
Gemini 2.5 Flash-Lite [50]	-	-	65.0	-	-	-	-	-	69.3	52.2
GPT-4o [98]	-	-	71.9	77.2	64.6	-	66.7	72.2	64.6	49.3
LLaVA-OV-8B [111]	32	384	58.2	61.5	56.7	79.4	56.5	60.1	64.7	41.1
LongVILA-7B [39]	2048	384	60.1	65.1	67.1	80.7	57.1	-	-	41.4
LongVILA-R1-7B [40]	8192	448	65.1	71.1	-	81.5	58.0	-	-	42.2
Apollo-7B [273]	2FPS	384	61.3	63.3	-	-	58.5	-	70.9	-
VideoLLaMA3-7B [257]	180	384	66.2	70.3	69.7	84.5	59.8	63.3	73.0	38.8
VideoChat-Flash [115]	10000	448	65.3	69.7	74.0	-	64.7	-	74.7	46.6
InternVL3.5-8B [210]	64	448	66.0	68.6	72.1	-	62.1	-	70.2	39.9
Qwen2.5-VL-7B [12]	48	896	65.1	71.6	69.6	-	56.0	65.0	70.2	48.1
NVILA-8B-Video [131]	256	448	64.2	70.0	68.1	82.2	57.7	-	70.1	42.5
+ AutoGaze	1024	3584	67.0	71.8	69.7	82.8	61.0	66.9	71.6	52.6
(vs. NVILA-8B-Video)	($\times 4$)	($\times 8$)	(+2.8)	(+1.8)	(+1.6)	(+0.6)	(+3.3)	-	(+1.5)	(+10.1)

Scaling MLLMs with AutoGaze

Leveraging AutoGaze’s efficiency, we scale MLLMs to longer, higher-resolution videos and achieve state-of-the-art performance on video benchmarks.

Scaling properties. We compare performance and efficiency when scaling MLLMs at test time to longer and high-resolution videos with or without AutoGaze, and report results in Fig. 4.9. We first scale the number of frames, identify the best frame count for each benchmark, then scale resolution. Starting from 64 frames and 448 resolution, MLLM with AutoGaze has slightly worse performance than the baseline while using $\sim 4\times$ fewer tokens. This performance drop vanishes after scaling to 256 frames. When further scaling video duration and resolution, the baseline runs out of memory while AutoGaze enables scaling to 1K frames and 4K resolution with consistent improvements. Note that on some benchmarks, using too long or too high-resolution videos is detrimental, likely because those benchmarks require neither, while scaling to 4K resolution significantly improves performance on HLVID, verifying it does require high resolution.

Comparing to state-of-the-art MLLMs. We train NVILA-8B-Video [131] with AutoGaze at 256 frames and 896 resolution, scale up to 1K frames and 4K resolution at test time, and compare to existing MLLMs in Tab. 4.1. Using AutoGaze, it consistently improves over the base NVILA-8B-Video on all benchmarks. Notably, performance on HLVID improves by 10.1%, surpassing all open-source MLLMs like Qwen2.5-VL-7B [12] and proprietary models like GPT-4o [98] despite much smaller model size or training dataset. NVILA-8B-Video with AutoGaze also outperforms others on VideoMME, though falls short on MVBench and LongVideoBench likely due to differing training recipes.

Table 4.2: **AutoGaze vs. token reduction baselines.** We compare to spatial / temporal / spatiotemporal (S-/T-/ST-) token reduction approaches that are either prompt-agnostic (PA) or prompt-dependent (PD). All methods select 6.25% visual tokens on average.

Type	Method	ViT lat.	LLM lat.	V-MME (w/o sub)	L-Vid (val)
-	No Reduction	2.20s	1.42s	53.4	51.1
S-PA	S-Pool	2.20s	0.18s	51.5	47.2
	ToMe [20]	2.23s	0.11s	51.5	49.3
	VisionZip [242]	2.22s	0.15s	50.7	48.5
S-PD	FastV [33]	2.23s	0.38s	53.0	46.3
T-PA	T-Pool	2.20s	0.13s	52.2	50.0
T-PD	AKS [187]	3.27s	0.12s	50.8	49.5
ST-PA	ST-Pool	2.19s	0.13s	52.0	49.8
	STORM [102]	2.18s	0.15s	52.7	51.5
	FastVID [170]	2.34s	0.12s	52.4	50.3
	F-16 [117]	2.20s	0.18s	51.8	50.0
ST-PD	LongVU [171]	2.17s	0.12s	52.2	50.1
	PruneVID [97]	2.52s	0.15s	50.3	48.0
	VChat-Flash [115]	2.21s	0.15s	52.4	49.9
ST-PA	AutoGaze	0.55s	0.10s	52.3	50.3

Comparing to Token Pruning Baselines

Comparing to baseline gazing approaches. In Fig. 4.10, we compare AutoGaze to heuristic-based gazing approaches that select patches randomly (**Random Gaze**), with the largest RGB difference (**RGB-Diff Gaze**) or with the largest optical flow using SEARAFT [217] (**Optical-Flow Gaze**). We left-pad a blank image when computing differences for the first frame. AutoGaze greatly improves efficiency — for example, reaching reconstruction loss 1.0 with 5% patches versus 15% for Random Gaze. We find RGB-Diff Gaze and Optical-Flow Gaze to be worse than Random Gaze because they fixate on the first frame (an abrupt change from the padding) and ignore other frames, causing information loss.

Comparing to MLLM token reduction baselines. We compare AutoGaze to existing MLLM token reduction methods [20, 33, 97, 102, 115, 117, 170, 171, 187, 242] and three simple baselines: spatial, temporal, and spatiotemporal pooling (S-Pool, T-Pool, and ST-Pool). These approaches process the whole video in the ViT, then reduce visual tokens in the LLM. We use 128 frames and 6.25% selection ratio and report results in Tab. 4.2. The baseline without token reduction has high ViT and LLM latency, with ViT latency slightly higher than LLM latency despite its smaller size due to LLMs’ token shuffling [12, 131]. While baseline methods improve LLM latency by $3.7\times$ - $13.4\times$, the ViT latency remains unchanged. In contrast, AutoGaze significantly improves ViT latency by $4\times$ in addition to the LLM

latency improvement. Beyond speedups, both the token-reduction baselines and AutoGaze retain performance comparable to the no-reduction baseline.

Ablations

Training pipeline of AutoGaze. We compare the gazing ratio required to reach reconstruction loss 0.7 when ablating training stages. As shown in Tab. 4.3, with only pre- or post-training, efficiency improves over no training (pre-training contributes more). With both stages, we obtain the lowest gazing ratio, improving by $\sim 10\%$ over pre-training only.

Table 4.3: **Ablation on AutoGaze training pipeline.** Both NTP pre-training and RL post-training helps with the performance.

Pre-Train	Post-Train	Recon Loss	Gazing Ratio
✗	✗	0.7	0.263
✓	✗	0.7	0.102
✗	✓	0.7	0.209
✓	✓	0.7	0.094

Model designs of AutoGaze. We analyze the effect of multi-token prediction and multi-scale gazing on the model performance and efficiency (Tab. 4.4). We report the gazing ratio and corresponding latency required to reach reconstruction loss 0.7 for a 1-second video at 10 FPS and 224 resolution. We find decoding more tokens at a time leads to lower latency but higher gazing ratio, with decoding 10 tokens balancing both factors well. Additionally, multi-scale gazing reduces gazing ratio and improves efficiency by $2.3\times$.

Table 4.4: **Ablations of AutoGaze model designs.**

Multi-Token Pred.	Multi-Scale Gazing	Recon Loss	Gazing Ratio	Latency
1	✓	0.7	0.074	0.949s
5	✓	0.7	0.078	0.246s
10	✓	0.7	0.094	0.193s
20	✓	0.7	0.109	0.156s
10	✓	0.7	0.094	0.193s
10	✗	0.7	0.220	0.467s

4.5 Conclusion

We introduce AutoGaze, a lightweight framework that removes redundant video patches to improve ViT and MLLM efficiency. Trained via NTP on gazing sequences collected

through a greedy algorithm and RL with reconstruction reward, AutoGaze learns to select a minimal set of multi-scale patches that reconstructs the video within a user-specified threshold. Empirically, AutoGaze reduces visual tokens by $4\times$ – $100\times$ and accelerates ViTs and MLLMs by up to $19\times$ and $10\times$, enabling 1024-frame 4K-resolution video understanding and improving performance on video benchmarks. We further introduce HLVID, the first long-form (5-minute), high-resolution (4K) video QA benchmark, where an MLLM with AutoGaze outperforms previous SOTA model by 4.5%.

Chapter 5

Conclusion

This thesis develops efficient scaling methods for vision models by rethinking how visual computation should be allocated across scale, space, and time. The motivating observation is that high-resolution visual understanding is essential for many real-world tasks, but uniformly processing every pixel at maximum fidelity is computationally wasteful. Human vision offers a useful analogy: it maintains high acuity only where needed and relies on strong information bottlenecks to transform rich visual input into compact representations for attention, action, and memory.

Across the three paper chapters, this thesis explores several ways to bring this principle into modern vision systems. Scaling on scales shows that scaling spatial resolution for vision models is necessary and can provide similar level of benefits as scaling model sizes. PS3 shows that high-resolution image pre-training can be made efficient by contrasting local high-resolution regions with local captions rather than processing full 4K images globally. AutoGaze extends selective processing to videos by learning to remove redundant spatiotemporal patches before they enter expensive vision transformers or multimodal language models. Together, this thesis argues that efficient visual intelligence requires adaptive allocation of computation. The methods presented here show that by respecting the structure of visual information, vision models can scale to significantly higher spatiotemporal fidelity in an efficient way.

Bibliography

- [1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.
- [2] Xiang An, Yin Xie, Kaicheng Yang, Wenkang Zhang, Xiuwei Zhao, Zheng Cheng, Yirui Wang, Songcen Xu, Changrui Chen, Didi Zhu, et al. Llava-onevision-1.5: Fully open framework for democratized multimodal training. *arXiv preprint arXiv:2509.23661*, 2025.
- [3] Charles H. Anderson and David C. Van Essen. Directed visual attention and the dynamic control of information flow. In Laurent Itti, Geraint Rees, and John K. Tsotsos, editors, *Neurobiology of Attention*, pages 11–17. Elsevier, 2005.
- [4] John Robert Anderson and Jane Crawford. Cognitive psychology and its implications. 1995.
- [5] Anthropic. Claude 3.5 sonnet, 2024. URL <https://www.anthropic.com/news/3-5-models-and-computer-use>.
- [6] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6836–6846, 2021.
- [7] Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15619–15629, 2023.
- [8] Mido Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Matthew Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, Artem Zhohus, et al. V-jepa 2: Self-supervised video models enable understanding, prediction and planning. *arXiv preprint arXiv:2506.09985*, 2025.
- [9] Roman Bachmann, Jesse Allardice, David Mizrahi, Enrico Fini, Oğuzhan Fatih Kar, Elmira Amirloo, Alaaeldin El-Nouby, Amir Zamir, and Afshin Dehghan. Flextok: Re-

- sampling images into 1d token sequences of flexible length. In *Forty-second International Conference on Machine Learning*, 2025.
- [10] Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-vl: A frontier large vision-language model with versatile abilities. *arXiv preprint arXiv:2308.12966*, 2023.
 - [11] Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.
 - [12] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibor Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.
 - [13] Yutong Bai, Xinyang Geng, Karttikeya Mangalam, Amir Bar, Alan L Yuille, Trevor Darrell, Jitendra Malik, and Alexei A Efros. Sequential modeling enables scalable learning for large vision models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22861–22872, 2024.
 - [14] Adrien Bardes, Quentin Garrido, Jean Ponce, Xinlei Chen, Michael Rabbat, Yann LeCun, Mahmoud Assran, and Nicolas Ballas. Revisiting feature prediction for learning visual representations from video. *arXiv preprint arXiv:2404.08471*, 2024.
 - [15] Peter L Bartlett, Philip M Long, Gábor Lugosi, and Alexander Tsigler. Benign overfitting in linear regression. *Proceedings of the National Academy of Sciences*, 117(48):30063–30070, 2020.
 - [16] Irwan Bello, William Fedus, Xianzhi Du, Ekin Dogus Cubuk, Aravind Srinivas, Tsung-Yi Lin, Jonathon Shlens, and Barret Zoph. Revisiting resnets: Improved training and scaling strategies. *Advances in Neural Information Processing Systems*, 34:22614–22627, 2021.
 - [17] Lucas Beyer, Olivier J Hénaff, Alexander Kolesnikov, Xiaohua Zhai, and Aäron van den Oord. Are we done with imagenet? *arXiv preprint arXiv:2006.07159*, 2020.
 - [18] Ioana Bica, Anastasija Ilić, Matthias Bauer, Goker Erdogan, Matko Bošnjak, Christos Kaplanis, Alexey A Gritsenko, Matthias Minderer, Charles Blundell, Razvan Pascanu, et al. Improving fine-grained understanding in image-text pre-training. *arXiv preprint arXiv:2401.09865*, 2024.
 - [19] Ali Furkan Biten, Ruben Tito, Lluís Gomez, Ernest Valveny, and Dimosthenis Karatzas. Ocr-idl: Ocr annotations for industry document library dataset. In *European Conference on Computer Vision*, pages 241–252. Springer, 2022.

- [20] Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token merging: Your vit but faster. *arXiv preprint arXiv:2210.09461*, 2022.
- [21] Daniel Bolya, Chaitanya Ryali, Judy Hoffman, and Christoph Feichtenhofer. Window attention is bugged: How not to interpolate position embeddings. *arXiv preprint arXiv:2311.05613*, 2023.
- [22] Daniel Bolya, Po-Yao Huang, Peize Sun, Jang Hyun Cho, Andrea Madotto, Chen Wei, Tengyu Ma, Jiale Zhi, Jathushan Rajasegaran, Hanoona Rasheed, et al. Perception encoder: The best visual embeddings are not at the output of the network. *arXiv preprint arXiv:2504.13181*, 2025.
- [23] Tim Brooks, Bill Peebles, Connor Homes, Will DePue, Yufei Guo, Li Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, Clarence Ng, Ricky Wang, and Aditya Ramesh. Video generation models as world simulators. 2024. URL <https://openai.com/research/video-generation-models-as-world-simulators>.
- [24] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.
- [25] Qingqing Cao, Bhargavi Paranjape, and Hannaneh Hajishirzi. PuMer: Pruning and merging tokens for efficient vision language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 12890–12903, 2023.
- [26] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021.
- [27] Marisa Carrasco. Visual attention: The past 25 years. *Vision research*, 51(13):1484–1525, 2011.
- [28] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017.
- [29] Boyuan Chen, Zhuo Xu, Sean Kirmani, Brain Ichter, Dorsa Sadigh, Leonidas Guibas, and Fei Xia. Spatialvlm: Endowing vision-language models with spatial reasoning capabilities. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14455–14465, June 2024.

- [30] Chun-Fu Richard Chen, Quanfu Fan, and Rameswar Panda. Crossvit: Cross-attention multi-scale vision transformer for image classification. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 357–366, 2021.
- [31] Hao Chen, Bo He, Hanyu Wang, Yixuan Ren, Ser Nam Lim, and Abhinav Shrivastava. Nerv: Neural representations for videos. *Advances in Neural Information Processing Systems*, 34:21557–21568, 2021.
- [32] Hong-You Chen, Zhengfeng Lai, Haotian Zhang, Xinze Wang, Marcin Eichner, Keen You, Meng Cao, Bowen Zhang, Yinfei Yang, and Zhe Gan. Contrastive localized language-image pre-training. *arXiv preprint arXiv:2410.02746*, 2024.
- [33] Liang Chen, Haozhe Zhao, Tianyu Liu, Shuai Bai, Junyang Lin, Chang Zhou, and Baobao Chang. An image is worth 1/2 tokens after layer 2: Plug-and-play inference acceleration for large vision-language models. In *European Conference on Computer Vision*, pages 19–35. Springer, 2024.
- [34] Liang Chen, Haozhe Zhao, Tianyu Liu, Shuai Bai, Junyang Lin, Chang Zhou, and Baobao Chang. An image is worth 1/2 tokens after layer 2: Plug-and-play inference acceleration for large vision-language models. In *European Conference on Computer Vision*, pages 19–35. Springer, 2025.
- [35] Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Jiaqi Wang, Yu Qiao, Dahua Lin, et al. Are we on the right way for evaluating large vision-language models? *Advances in Neural Information Processing Systems*, 37: 27056–27087, 2024.
- [36] Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, and Ilya Sutskever. Generative pretraining from pixels. In *International conference on machine learning*, pages 1691–1703. PMLR, 2020.
- [37] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PmLR, 2020.
- [38] Xuanyao Chen, Zhijian Liu, Haotian Tang, Li Yi, Hang Zhao, and Song Han. Sparsevit: Revisiting activation sparsity for efficient high-resolution vision transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2061–2070, 2023.
- [39] Yukang Chen, Fuzhao Xue, Dacheng Li, Qinghao Hu, Ligeng Zhu, Xiuyu Li, Yunhao Fang, Haotian Tang, Shang Yang, Zhijian Liu, et al. Longvila: Scaling long-context visual language models for long videos. *arXiv preprint arXiv:2408.10188*, 2024.

- [40] Yukang Chen, Wei Huang, Baifeng Shi, Qinghao Hu, Hanrong Ye, Ligeng Zhu, Zhijian Liu, Pavlo Molchanov, Jan Kautz, Xiaojuan Qi, et al. Scaling rl to long videos. *arXiv preprint arXiv:2507.07966*, 2025.
- [41] Zhe Chen, Weiyun Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Erfei Cui, Jinguo Zhu, Shenglong Ye, Hao Tian, Zhaoyang Liu, et al. Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling. *arXiv preprint arXiv:2412.05271*, 2024.
- [42] Zhe Chen, Weiyun Wang, Hao Tian, Shenglong Ye, Zhangwei Gao, Erfei Cui, Wenwen Tong, Kongzhi Hu, Jiapeng Luo, Zheng Ma, et al. How far are we to gpt-4v? closing the gap to commercial multimodal models with open-source suites. *arXiv preprint arXiv:2404.16821*, 2024.
- [43] Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, et al. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24185–24198, 2024.
- [44] An-Chieh Cheng, Hongxu Yin, Yang Fu, Qiushan Guo, Ruihan Yang, Jan Kautz, Xiaolong Wang, and Sifei Liu. Spatialrgpt: Grounded spatial reasoning in vision language model. *arXiv preprint arXiv:2406.01584*, 2024.
- [45] Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1290–1299, 2022.
- [46] Chen Cheng, John Duchi, and Rohith Kuditipudi. Memorize to generalize: on the necessity of interpolation in high dimensional linear regression. In *Conference on Learning Theory*, pages 5528–5560. PMLR, 2022.
- [47] Mehdi Cherti, Romain Beaumont, Ross Wightman, Mitchell Wortsman, Gabriel Ilharco, Cade Gordon, Christoph Schuhmann, Ludwig Schmidt, and Jenia Jitsev. Reproducible scaling laws for contrastive language-image learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2818–2829, 2023.
- [48] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, et al. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. See <https://vicuna.lmsys.org> (accessed 14 April 2023), 2023.
- [49] Jang Hyun Cho, Andrea Madotto, Effrosyni Mavroudi, Triantafyllos Afouras, Tushar Nagarajan, Muhammad Maaz, Yale Song, Tengyu Ma, Shuming Hu, Suyog Jain, et al. Perceptionlm: Open-access data and models for detailed visual understanding. *arXiv preprint arXiv:2504.13180*, 2025.

- [50] Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025.
- [51] MMSegmentation Contributors. MMSegmentation: Openmmlab semantic segmentation toolbox and benchmark. <https://github.com/open-mmlab/mmdetection>, 2020.
- [52] Kellie Corona, Katie Osterdahl, Roderic Collins, and Anthony Hoogs. Meva: A large-scale multiview, multimodal video dataset for activity detection. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 1060–1068, 2021.
- [53] Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven Hoi. Instructblip: Towards general-purpose vision-language models with instruction tuning, 2023.
- [54] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR’05)*, volume 1, pages 886–893. Ieee, 2005.
- [55] Mostafa Dehghani, Josip Djolonga, Basil Mustafa, Piotr Padlewski, Jonathan Heek, Justin Gilmer, Andreas Peter Steiner, Mathilde Caron, Robert Geirhos, Ibrahim Alabdulmohsin, et al. Scaling vision transformers to 22 billion parameters. In *International Conference on Machine Learning*, pages 7480–7512. PMLR, 2023.
- [56] Adrien Deliege, Anthony Cioppa, Silvio Giancola, Meisam J Seikavandi, Jacob V Dueholm, Kamal Nasrollahi, Bernard Ghanem, Thomas B Moeslund, and Marc Van Droogenbroeck. Soccernet-v2: A dataset and benchmarks for holistic understanding of broadcast soccer videos. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4508–4519, 2021.
- [57] 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*, pages 248–255, 2009. doi: 10.1109/CVPR.2009.5206848.
- [58] Piotr Dollár, Ron Appel, Serge Belongie, and Pietro Perona. Fast feature pyramids for object detection. *IEEE transactions on pattern analysis and machine intelligence*, 36(8):1532–1545, 2014.
- [59] Piotr Dollár, Mannat Singh, and Ross Girshick. Fast and accurate model scaling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 924–932, 2021.

- [60] Xiaoyi Dong, Pan Zhang, Yuhang Zang, Yuhang Cao, Bin Wang, Linke Ouyang, Songyang Zhang, Haodong Duan, Wenwei Zhang, Yining Li, et al. Internlm-xcomposer2-4khd: A pioneering large vision-language model handling resolutions from 336 pixels to 4k hd. *arXiv preprint arXiv:2404.06512*, 2024.
- [61] A. Dosovitskiy, P. Fischer, E. Ilg, P. Häusser, C. Hazırbaş, V. Golkov, P. v.d. Smagt, D. Cremers, and T. Brox. FlowNet: Learning optical flow with convolutional networks. In *IEEE International Conference on Computer Vision (ICCV)*, 2015. URL <http://lmb.informatik.uni-freiburg.de/Publications/2015/DFIB15>.
- [62] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.
- [63] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.
- [64] Shivam Duggal, Phillip Isola, Antonio Torralba, and William T. Freeman. Adaptive length image tokenization via recurrent allocation. In *Scalable Optimization for Efficient and Adaptive Foundation Models (NeurIPS Workshop)*, 2024.
- [65] Shivam Duggal, Sanghyun Byun, William T Freeman, Antonio Torralba, and Phillip Isola. Single-pass adaptive image tokenization for minimum program search. *arXiv preprint arXiv:2507.07995*, 2025.
- [66] Alaaeldin El-Nouby, Michal Klein, Shuangfei Zhai, Miguel Angel Bautista, Alexander Toshev, Vaishaal Shankar, Joshua M Susskind, and Armand Joulin. Scalable pre-training of large autoregressive image models. *arXiv preprint arXiv:2401.08541*, 2024.
- [67] Haoqi Fan, Bo Xiong, Karttikeya Mangalam, Yanghao Li, Zhicheng Yan, Jitendra Malik, and Christoph Feichtenhofer. Multiscale vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6824–6835, 2021.
- [68] Haoqi Fan, Bo Xiong, Karttikeya Mangalam, Yanghao Li, Zhicheng Yan, Jitendra Malik, and Christoph Feichtenhofer. Multiscale vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6824–6835, 2021.
- [69] Yunhao Fang, Ligeng Zhu, Yao Lu, Yan Wang, Pavlo Molchanov, Jang Hyun Cho, Marco Pavone, Song Han, and Hongxu Yin. Vila2: Vila augmented vila. *arXiv preprint arXiv:2407.17453*, 2024.

- [70] Yuxin Fang, Wen Wang, Binhui Xie, Quan Sun, Ledell Wu, Xinggang Wang, Tiejun Huang, Xinlong Wang, and Yue Cao. Eva: Exploring the limits of masked visual representation learning at scale. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19358–19369, 2023.
- [71] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6202–6211, 2019.
- [72] Christoph Feichtenhofer, Haoqi Fan, Bo Xiong, Ross Girshick, and Kaiming He. A large-scale study on unsupervised spatiotemporal representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3299–3309, 2021.
- [73] Christoph Feichtenhofer, Yanghao Li, Kaiming He, et al. Masked autoencoders as spatiotemporal learners. *Advances in neural information processing systems*, 35:35946–35958, 2022.
- [74] Vitaly Feldman. Does learning require memorization? a short tale about a long tail. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, pages 954–959, 2020.
- [75] Vitaly Feldman and Chiyuan Zhang. What neural networks memorize and why: Discovering the long tail via influence estimation. *Advances in Neural Information Processing Systems*, 33:2881–2891, 2020.
- [76] Enrico Fini, Mustafa Shukor, Xiujuan Li, Philipp Dufter, Michal Klein, David Haldimann, Sai Aitharaju, Victor Guilherme Turrise da Costa, Louis Béthune, Zhe Gan, et al. Multimodal autoregressive pre-training of large vision encoders. *arXiv preprint arXiv:2411.14402*, 2024.
- [77] Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Zhenyu Qiu, Wei Lin, Jinrui Yang, Xiawu Zheng, et al. Mme: A comprehensive evaluation benchmark for multimodal large language models. *arXiv preprint arXiv:2306.13394*, 2023.
- [78] Chaoyou Fu, Yuhan Dai, Yongdong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, et al. Video-mme: The first-ever comprehensive evaluation benchmark of multi-modal llms in video analysis. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 24108–24118, 2025.
- [79] Xingyu Fu, Yushi Hu, Bangzheng Li, Yu Feng, Haoyu Wang, Xudong Lin, Dan Roth, Noah A Smith, Wei-Chiu Ma, and Ranjay Krishna. Blink: Multimodal large language models can see but not perceive. In *European Conference on Computer Vision*, pages 148–166. Springer, 2024.

- [80] Samir Yitzhak Gadre, Gabriel Ilharco, Alex Fang, Jonathan Hayase, Georgios Smyrnis, Thao Nguyen, Ryan Marten, Mitchell Wortsman, Dhruva Ghosh, Jieyu Zhang, et al. Datacomp: In search of the next generation of multimodal datasets. *Advances in Neural Information Processing Systems*, 36, 2024.
- [81] Ruiyuan Gao, Kai Chen, Bo Xiao, Lanqing Hong, Zhenguo Li, and Qiang Xu. Magicdrive-v2: High-resolution long video generation for autonomous driving with adaptive control. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 28135–28144, 2025.
- [82] Yunhao Ge, Xiaohui Zeng, Jacob Samuel Huffman, Tsung-Yi Lin, Ming-Yu Liu, and Yin Cui. Visual fact checker: Enabling high-fidelity detailed caption generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14033–14042, 2024.
- [83] Fabian Gloeckle, Badr Youbi Idrissi, Baptiste Rozière, David Lopez-Paz, and Gabriel Synnaeve. Better & faster large language models via multi-token prediction. *arXiv preprint arXiv:2404.19737*, 2024.
- [84] 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*, pages 6904–6913, 2017.
- [85] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284, 2020.
- [86] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.
- [87] Qiushan Guo, Shalini De Mello, Hongxu Yin, Wonmin Byeon, Ka Chun Cheung, Yizhou Yu, Ping Luo, and Sifei Liu. Regiongpt: Towards region understanding vision language model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13796–13806, 2024.
- [88] 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*, pages 3608–3617, 2018.
- [89] Tengda Han, Weidi Xie, and Andrew Zisserman. Video representation learning by dense predictive coding. In *Proceedings of the IEEE/CVF international conference on computer vision workshops*, pages 0–0, 2019.

- [90] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
- [91] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020.
- [92] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16000–16009, 2022.
- [93] Greg Heinrich, Mike Ranzinger, Yao Lu, Jan Kautz, Andrew Tao, Bryan Catanzaro, Pavlo Molchanov, et al. Radio amplified: Improved baselines for agglomerative vision foundation models. *arXiv preprint arXiv:2412.07679*, 2024.
- [94] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.
- [95] Ronghang Hu, Shoubhik Debnath, Saining Xie, and Xinlei Chen. Exploring long-sequence masked autoencoders. *arXiv preprint arXiv:2210.07224*, 2022.
- [96] De-An Huang, Shijia Liao, Subhashree Radhakrishnan, Hongxu Yin, Pavlo Molchanov, Zhiding Yu, and Jan Kautz. Lita: Language instructed temporal-localization assistant. *arXiv preprint arXiv:2403.19046*, 2024.
- [97] Xiaohu Huang, Hao Zhou, and Kai Han. Prunevid: Visual token pruning for efficient video large language models. *arXiv preprint arXiv:2412.16117*, 2024.
- [98] Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.
- [99] E. Ilg, T. Saikia, M. Keuper, and T. Brox. Occlusions, motion and depth boundaries with a generic network for disparity, optical flow or scene flow estimation. In *European Conference on Computer Vision (ECCV)*, 2018. URL <http://lmb.informatik.uni-freiburg.de/Publications/2018/ISKB18>.
- [100] Laurent Itti and Christof Koch. Computational modelling of visual attention. *Nature reviews neuroscience*, 2(3):194–203, 2001.
- [101] Laurent Itti, Christof Koch, and Ernst Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on pattern analysis and machine intelligence*, 20(11):1254–1259, 2002.

- [102] Jindong Jiang, Xiuyu Li, Zhijian Liu, Muyang Li, Guo Chen, Zhiqi Li, De-An Huang, Guilin Liu, Zhiding Yu, Kurt Keutzer, et al. Storm: Token-efficient long video understanding for multimodal llms. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5830–5841, 2025.
- [103] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, pages 694–711. Springer, 2016.
- [104] Shubham Juneja, Povilas Daniušis, and Virginijus Marcinkevičius. Visual place recognition pre-training for end-to-end trained autonomous driving agent. *IEEE access*, 11: 128421–128428, 2023.
- [105] Shubham Juneja, Povilas Daniušis, and Virginijus Marcinkevičius. Dino pre-training for vision-based end-to-end autonomous driving. *arXiv preprint arXiv:2407.10803*, 2024.
- [106] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4015–4026, 2023.
- [107] Dan Kondratyuk, Lijun Yu, Xiuye Gu, José Lezama, Jonathan Huang, Rachel Hornung, Hartwig Adam, Hassan Akbari, Yair Alon, Vighnesh Birodkar, et al. Videopoet: A large language model for zero-shot video generation. *arXiv preprint arXiv:2312.14125*, 2023.
- [108] Xin Lai, Zhuotao Tian, Yukang Chen, Yanwei Li, Yuhui Yuan, Shu Liu, and Jiaya Jia. Lisa: Reasoning segmentation via large language model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9579–9589, 2024.
- [109] Didier Le Gall. Mpeg: A video compression standard for multimedia applications. *Communications of the ACM*, 34(4):46–58, 1991.
- [110] Youngwan Lee, Jonghee Kim, Jeffrey Willette, and Sung Ju Hwang. Mpvit: Multi-path vision transformer for dense prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7287–7296, 2022.
- [111] Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Yanwei Li, Ziwei Liu, and Chunyuan Li. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*, 2024.
- [112] Bohao Li, Rui Wang, Guangzhi Wang, Yuying Ge, Yixiao Ge, and Ying Shan. Seed-bench: Benchmarking multimodal llms with generative comprehension. *arXiv preprint arXiv:2307.16125*, 2023.

- [113] Bohao Li, Yuying Ge, Yi Chen, Yixiao Ge, Ruimao Zhang, and Ying Shan. Seed-bench-2-plus: Benchmarking multimodal large language models with text-rich visual comprehension. *arXiv preprint arXiv:2404.16790*, 2024.
- [114] Ling Li, David Thorsley, and Joseph Hassoun. SaiT: Sparse vision transformers through adaptive token pruning. *CoRR*, abs/2210.05832, 2022.
- [115] Xinhao Li, Yi Wang, Jiashuo Yu, Xiangyu Zeng, Yuhang Zhu, Haian Huang, Jianfei Gao, Kunchang Li, Yinan He, Chenting Wang, et al. Videochat-flash: Hierarchical compression for long-context video modeling. *arXiv preprint arXiv:2501.00574*, 2024.
- [116] Yanwei Li, Yuechen Zhang, Chengyao Wang, Zhisheng Zhong, Yixin Chen, Ruihang Chu, Shaoteng Liu, and Jiaya Jia. Mini-gemini: Mining the potential of multi-modality vision language models. *arXiv preprint arXiv:2403.18814*, 2024.
- [117] Yixuan Li, Changli Tang, Jimin Zhuang, Yudong Yang, Guangzhi Sun, Wei Li, Zejun Ma, and Chao Zhang. Improving llm video understanding with 16 frames per second. *arXiv preprint arXiv:2503.13956*, 2025.
- [118] Zhang Li, Biao Yang, Qiang Liu, Zhiyin Ma, Shuo Zhang, Jingxu Yang, Yabo Sun, Yuliang Liu, and Xiang Bai. Monkey: Image resolution and text label are important things for large multi-modal models. In *proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 26763–26773, 2024.
- [119] Ji Lin, Hongxu Yin, Wei Ping, Yao Lu, Pavlo Molchanov, Andrew Tao, Huizi Mao, Jan Kautz, Mohammad Shoeybi, and Song Han. Vila: On pre-training for visual language models. *CVPR*, 2024.
- [120] Ji Lin, Hongxu Yin, Wei Ping, Pavlo Molchanov, Mohammad Shoeybi, and Song Han. Vila: On pre-training for visual language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26689–26699, 2024.
- [121] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017.
- [122] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. *arXiv preprint arXiv:2310.03744*, 2023.
- [123] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *arXiv preprint arXiv:2304.08485*, 2023.
- [124] Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. Llava-next: Improved reasoning, ocr, and world knowledge, January 2024. URL <https://llava-vl.github.io/blog/2024-01-30-llava-next/>.

- [125] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36, 2024.
- [126] Shikun Liu, Linxi Fan, Edward Johns, Zhiding Yu, Chaowei Xiao, and Anima Anandkumar. Prism: A vision-language model with an ensemble of experts. *arXiv preprint arXiv:2303.02506*, 3, 2023.
- [127] Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, et al. Mmbench: Is your multi-modal model an all-around player? *arXiv preprint arXiv:2307.06281*, 2023.
- [128] Yuliang Liu, Zhang Li, Biao Yang, Chunyuan Li, Xucheng Yin, Cheng-lin Liu, Lianwen Jin, and Xiang Bai. On the hidden mystery of ocr in large multimodal models. *arXiv preprint arXiv:2305.07895*, 2023.
- [129] 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 *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021.
- [130] Zhijian Liu, Ligeng Zhu, Baifeng Shi, Zhuoyang Zhang, Yuming Lou, Shang Yang, Haocheng Xi, Shiyi Cao, Yuxian Gu, Dacheng Li, et al. Nvila: Efficient frontier visual language models. *arXiv preprint arXiv:2412.04468*, 2024.
- [131] Zhijian Liu, Ligeng Zhu, Baifeng Shi, Zhuoyang Zhang, Yuming Lou, Shang Yang, Haocheng Xi, Shiyi Cao, Yuxian Gu, Dacheng Li, et al. Nvila: Efficient frontier visual language models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 4122–4134, 2025.
- [132] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11976–11986, 2022.
- [133] Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. Understanding r1-zero-like training: A critical perspective. *arXiv preprint arXiv:2503.20783*, 2025.
- [134] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60:91–110, 2004.
- [135] 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. *arXiv preprint arXiv:2310.02255*, 2023.

- [136] Michal Lukasik, Vaishnavh Nagarajan, Ankit Singh Rawat, Aditya Krishna Menon, and Sanjiv Kumar. What do larger image classifiers memorise? *arXiv preprint arXiv:2310.05337*, 2023.
- [137] 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. *arXiv preprint arXiv:2403.03003*, 2024.
- [138] Jitendra Malik, Pablo Arbeláez, Joao Carreira, Katerina Fragkiadaki, Ross Girshick, Georgia Gkioxari, Saurabh Gupta, Bharath Hariharan, Abhishek Kar, and Shubham Tulsiani. The three r’s of computer vision: Recognition, reconstruction and reorganization. *Pattern Recognition Letters*, 72:4–14, 2016.
- [139] Karttikeya Mangalam, Raiymbek Akshulakov, and Jitendra Malik. Egoschema: A diagnostic benchmark for very long-form video language understanding. *Advances in Neural Information Processing Systems*, 36:46212–46244, 2023.
- [140] 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. *arXiv preprint arXiv:2203.10244*, 2022.
- [141] Minesh Mathew, Dimosthenis Karatzas, and CV Jawahar. Docvqa: A dataset for vqa on document images. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 2200–2209, 2021.
- [142] Minesh Mathew, Viraj Bagal, Rubèn Tito, Dimosthenis Karatzas, Ernest Valveny, and CV Jawahar. Infographicvqa. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1697–1706, 2022.
- [143] Pablo Montalvo and Ross Wightman. <https://huggingface.co/datasets/pixparse/pdfa-eng-wds>.
- [144] Curtis G Northcutt, Anish Athalye, and Jonas Mueller. Pervasive label errors in test sets destabilize machine learning benchmarks. *arXiv preprint arXiv:2103.14749*, 2021.
- [145] OpenAI. GPT-4 technical report. Technical report, OpenAI, 2023. <https://arxiv.org/abs/2303.08774>.
- [146] OpenAI. ChatGPT: Optimizing language models for dialogue. <https://openai.com/blog/chatgpt>, 2023. Accessed: 2023.
- [147] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

- [148] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744, 2022.
- [149] Bowen Pan, Rameswar Panda, Yifan Jiang, Zhangyang Wang, Rogério Feris, and Aude Oliva. IA-RED²: Interpretability-aware redundancy reduction for vision transformers. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 24898–24911, 2021.
- [150] Ziqi Pang and Yu-Xiong Wang. Mr. video: "mapreduce" is the principle for long video understanding. *arXiv preprint arXiv:2504.16082*, 2025.
- [151] Liao Qu, Huichao Zhang, Yiheng Liu, Xu Wang, Yi Jiang, Yiming Gao, Hu Ye, Daniel K. Du, Zehuan Yuan, and Xinglong Wu. Tokenflow: Unified image tokenizer for multimodal understanding and generation, 2025. URL <https://arxiv.org/abs/2412.03069>.
- [152] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018.
- [153] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.
- [154] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.
- [155] Ilija Radosavovic, Baifeng Shi, Letian Fu, Ken Goldberg, Trevor Darrell, and Jitendra Malik. Robot learning with sensorimotor pre-training. *arXiv preprint arXiv:2306.10007*, 2023.
- [156] Ilija Radosavovic, Tete Xiao, Stephen James, Pieter Abbeel, Jitendra Malik, and Trevor Darrell. Real-world robot learning with masked visual pre-training. In *Conference on Robot Learning*, pages 416–426. PMLR, 2023.
- [157] Ilija Radosavovic, Bike Zhang, Baifeng Shi, Jathushan Rajasegaran, Sarthak Kamat, Trevor Darrell, Koushil Sreenath, and Jitendra Malik. Humanoid locomotion as next token prediction. *Advances in neural information processing systems*, 37:79307–79324, 2024.
- [158] Jathushan Rajasegaran, Ilija Radosavovic, Rahul Ravishankar, Yossi Gandelsman, Christoph Feichtenhofer, and Jitendra Malik. An empirical study of autoregressive pre-training from videos. *arXiv preprint arXiv:2501.05453*, 2025.

- [159] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 1(2):3, 2022.
- [160] Mike Ranzinger, Greg Heinrich, Jan Kautz, and Pavlo Molchanov. Am-radio: Agglomerative vision foundation model reduce all domains into one. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12490–12500, June 2024.
- [161] Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. DynamicViT: Efficient vision transformers with dynamic token sparsification. In *NeurIPS*, 2021.
- [162] Shuhuai Ren, Sishuo Chen, Shicheng Li, Xu Sun, and Lu Hou. TESTA: Temporal-spatial token aggregation for long-form video-language understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 932–947, 2023.
- [163] Weiming Ren, Huan Yang, Jie Min, Cong Wei, and Wenhui Chen. Vista: Enhancing long-duration and high-resolution video understanding by video spatiotemporal augmentation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 3804–3814, 2025.
- [164] Ronald A Rensink. The dynamic representation of scenes. *Visual cognition*, 7(1-3): 17–42, 2000.
- [165] Carlos Riquelme, Joan Puigcerver, Basil Mustafa, Maxim Neumann, Rodolphe Jenatton, André Susano Pinto, Daniel Keysers, and Neil Houlsby. Scaling vision with sparse mixture of experts. *Advances in Neural Information Processing Systems*, 34:8583–8595, 2021.
- [166] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pages 234–241. Springer, 2015.
- [167] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115: 211–252, 2015.
- [168] Chaitanya Ryali, Yuan-Ting Hu, Daniel Bolya, Chen Wei, Haoqi Fan, Po-Yao Huang, Vaibhav Aggarwal, Arkabandhu Chowdhury, Omid Poursaeed, Judy Hoffman, et al. Hiera: A hierarchical vision transformer without the bells-and-whistles. *arXiv preprint arXiv:2306.00989*, 2023.

- [169] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.
- [170] Leqi Shen, Guoqiang Gong, Tao He, Yifeng Zhang, Pengzhang Liu, Sicheng Zhao, and Guiguang Ding. Fastvid: Dynamic density pruning for fast video large language models. *arXiv preprint arXiv:2503.11187*, 2025.
- [171] Xiaoqian Shen, Yunyang Xiong, Changsheng Zhao, Lemeng Wu, Jun Chen, Chenchen Zhu, Zechun Liu, Fanyi Xiao, Balakrishnan Varadarajan, Florian Bordes, et al. Longvu: Spatiotemporal adaptive compression for long video-language understanding. *arXiv preprint arXiv:2410.17434*, 2024.
- [172] Baifeng Shi, Trevor Darrell, and Xin Wang. Top-down visual attention from analysis by synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2102–2112, 2023.
- [173] Baifeng Shi, Ziyang Wu, Maolin Mao, Xin Wang, and Trevor Darrell. When do we not need larger vision models? In *European Conference on Computer Vision*, pages 444–462. Springer, 2024.
- [174] Baifeng Shi, Boyi Li, Han Cai, Yao Lu, Sifei Liu, Marco Pavone, Jan Kautz, Song Han, Trevor Darrell, Pavlo Molchanov, et al. Scaling vision pre-training to 4k resolution. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 9631–9640, 2025.
- [175] Baifeng Shi, Ziyang Wu, Maolin Mao, Xin Wang, and Trevor Darrell. When do we not need larger vision models? In *European Conference on Computer Vision*, pages 444–462. Springer, 2025.
- [176] Min Shi, Fuxiao Liu, Shihao Wang, Shijia Liao, Subhashree Radhakrishnan, De-An Huang, Hongxu Yin, Karan Sapra, Yaser Yacoob, Humphrey Shi, et al. Eagle: Exploring the design space for multimodal llms with mixture of encoders. *arXiv preprint arXiv:2408.15998*, 2024.
- [177] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgb-d images. In *Computer Vision—ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7–13, 2012, Proceedings, Part V 12*, pages 746–760. Springer, 2012.
- [178] 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*, pages 8317–8326, 2019.

- [179] Enxin Song, Wenhao Chai, Guanhong Wang, Yucheng Zhang, Haoyang Zhou, Feiyang Wu, Haozhe Chi, Xun Guo, Tian Ye, Yanting Zhang, et al. Moviechat: From dense token to sparse memory for long video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18221–18232, 2024.
- [180] Andreas Steiner, Alexander Kolesnikov, Xiaohua Zhai, Ross Wightman, Jakob Uszkoreit, and Lucas Beyer. How to train your vit? data, augmentation, and regularization in vision transformers. *arXiv preprint arXiv:2106.10270*, 2021.
- [181] Carole H Sudre, Wenqi Li, Tom Vercauteren, Sebastien Ourselin, and M Jorge Cardoso. Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations. In *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support: Third International Workshop, DLMIA 2017, and 7th International Workshop, ML-CDS 2017, Held in Conjunction with MICCAI 2017, Québec City, QC, Canada, September 14, Proceedings 3*, pages 240–248. Springer, 2017.
- [182] Gary J Sullivan, Jens-Rainer Ohm, Woo-Jin Han, and Thomas Wiegand. Overview of the high efficiency video coding (hevc) standard. *IEEE Transactions on circuits and systems for video technology*, 22(12):1649–1668, 2012.
- [183] Quan Sun, Yuxin Fang, Ledell Wu, Xinlong Wang, and Yue Cao. Eva-clip: Improved training techniques for clip at scale. *arXiv preprint arXiv:2303.15389*, 2023.
- [184] Zeyi Sun, Ye Fang, Tong Wu, Pan Zhang, Yuhang Zang, Shu Kong, Yuanjun Xiong, Dahua Lin, and Jiaqi Wang. Alpha-clip: A clip model focusing on wherever you want. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13019–13029, 2024.
- [185] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR, 2019.
- [186] Mingxing Tan and Quoc Le. Efficientnetv2: Smaller models and faster training. In *International conference on machine learning*, pages 10096–10106. PMLR, 2021.
- [187] Xi Tang, Jihao Qiu, Lingxi Xie, Yunjie Tian, Jianbin Jiao, and Qixiang Ye. Adaptive keyframe sampling for long video understanding. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 29118–29128, 2025.
- [188] Xi Tang, Jihao Qiu, Lingxi Xie, Yunjie Tian, Jianbin Jiao, and Qixiang Ye. Adaptive keyframe sampling for long video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 29118–29127, 2025.

- [189] Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.
- [190] Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*, 2024.
- [191] OpenGVLab Team. Internvl2: Better than the best—expanding performance boundaries of open-source multimodal models with the progressive scaling strategy. <https://internvl.github.io/blog/2024-07-02-InternVL-2.0/>.
- [192] Qwen Team. Introducing qwen-vl, Jan 2024. URL <https://qwenlm.github.io/blog/qwen-vl/>.
- [193] Jonathan Tompson, Ross Goroshin, Arjun Jain, Yann LeCun, and Christoph Bregler. Efficient object localization using convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 648–656, 2015.
- [194] Shengbang Tong, Ellis Brown, Penghao Wu, Sanghyun Woo, Manoj Middepogu, Sai Charitha Akula, Jihan Yang, Shusheng Yang, Adithya Iyer, Xichen Pan, et al. Cambrian-1: A fully open, vision-centric exploration of multimodal llms. *arXiv preprint arXiv:2406.16860*, 2024.
- [195] Shengbang Tong, Zhuang Liu, Yuexiang Zhai, Yi Ma, Yann LeCun, and Saining Xie. Eyes wide shut? exploring the visual shortcomings of multimodal llms. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9568–9578, 2024.
- [196] Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training. *Advances in neural information processing systems*, 35:10078–10093, 2022.
- [197] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.
- [198] Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, et al. Siglip 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. *arXiv preprint arXiv:2502.14786*, 2025.

- [199] David Van Essen, Bruno A Olshausen, Clifford H Anderson, and JTL Gallant. Pattern recognition, attention, and information bottlenecks in the primate visual system. In *Visual information processing: From neurons to chips*, volume 1473, pages 17–28. SPIE, 1991.
- [200] Pavan Kumar Anasosalu Vasu, Fartash Faghri, Chun-Liang Li, Cem Koc, Nate True, Albert Antony, Gokul Santhanam, James Gabriel, Peter Grasch, Oncel Tuzel, et al. Fastvlm: Efficient vision encoding for vision language models. *arXiv preprint arXiv:2412.13303*, 2024.
- [201] Thomas Verelst and Tinne Tuytelaars. Dynamic convolutions: Exploiting spatial sparsity for faster inference. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2320–2329, 2020.
- [202] Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. Anticipating visual representations from unlabeled video. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 98–106, 2016.
- [203] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, et al. Deep high-resolution representation learning for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, 43(10):3349–3364, 2020.
- [204] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *European Conference on Computer Vision (ECCV)*, pages 20–36, 2016.
- [205] Limin Wang, Bingkun Huang, Zhiyu Zhao, Zhan Tong, Yinan He, Yi Wang, Yali Wang, and Yu Qiao. Videomae v2: Scaling video masked autoencoders with dual masking. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14549–14560, 2023.
- [206] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. Qwen2-vl: Enhancing vision-language model’s perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024.
- [207] Shihao Wang, Guo Chen, De-an Huang, Zhiqi Li, Minghan Li, Guilin Li, Jose M Alvarez, Lei Zhang, and Zhiding Yu. Videoitg: Multimodal video understanding with instructed temporal grounding. *arXiv preprint arXiv:2507.13353*, 2025.
- [208] Weihaan Wang, Qingsong Lv, Wenmeng Yu, Wenyi Hong, Ji Qi, Yan Wang, Junhui Ji, Zhuoyi Yang, Lei Zhao, Xixuan Song, et al. Cogvlm: Visual expert for pretrained language models. *arXiv preprint arXiv:2311.03079*, 2023.

- [209] Weihai Wang, Zehai He, Wenyi Hong, Yean Cheng, Xiaohan Zhang, Ji Qi, Ming Ding, Xiaotao Gu, Shiyu Huang, Bin Xu, et al. Lvbench: An extreme long video understanding benchmark. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22958–22967, 2025.
- [210] 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*, 2025.
- [211] Wenbin Wang, Liang Ding, Minyan Zeng, Xiabin Zhou, Li Shen, Yong Luo, Wei Yu, and Dacheng Tao. Divide, conquer and combine: A training-free framework for high-resolution image perception in multimodal large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 7907–7915, 2025.
- [212] Xiaohan Wang, Yuhui Zhang, Orr Zohar, and Serena Yeung-Levy. Videoagent: Long-form video understanding with large language model as agent. In *European Conference on Computer Vision*, pages 58–76. Springer, 2024.
- [213] Xudong Wang, Jingfeng Yang, and Trevor Darrell. Segment anything without supervision. *arXiv preprint arXiv:2406.20081*, 2024.
- [214] Yi Wang, Kunchang Li, Yizhuo Li, Yinan He, Bingkun Huang, Zhiyu Zhao, Hongjie Zhang, Jilan Xu, Yi Liu, Zun Wang, et al. Internvideo: General video foundation models via generative and discriminative learning. *arXiv preprint arXiv:2212.03191*, 2022.
- [215] Yi Wang, Yinan He, Yizhuo Li, Kunchang Li, Jiashuo Yu, Xin Ma, Xinhao Li, Guo Chen, Xinyuan Chen, Yaohui Wang, et al. Internvid: A large-scale video-text dataset for multimodal understanding and generation. *arXiv preprint arXiv:2307.06942*, 2023.
- [216] Yi Wang, Kunchang Li, Xinhao Li, Jiashuo Yu, Yinan He, Guo Chen, Baoqi Pei, Rongkun Zheng, Zun Wang, Yansong Shi, et al. Internvideo2: Scaling foundation models for multimodal video understanding. In *European Conference on Computer Vision*, pages 396–416. Springer, 2024.
- [217] Yihan Wang, Lahav Lipson, and Jia Deng. Sea-raft: Simple, efficient, accurate raft for optical flow. In *European Conference on Computer Vision*, pages 36–54. Springer, 2024.
- [218] Yulin Wang, Kangchen Lv, Rui Huang, Shiji Song, Le Yang, and Gao Huang. Glance and focus: a dynamic approach to reducing spatial redundancy in image classification. *Advances in Neural Information Processing Systems*, 33:2432–2444, 2020.
- [219] Yulin Wang, Rui Huang, Shiji Song, Zeyi Huang, and Gao Huang. Not all images are worth 16x16 words: Dynamic transformers for efficient image recognition. *Advances in neural information processing systems*, 34:11960–11973, 2021.

- [220] Zirui Wang, Mengzhou Xia, Luxi He, Howard Chen, Yitao Liu, Richard Zhu, Kaiqu Liang, Xindi Wu, Haotian Liu, Sadhika Malladi, et al. Charxiv: Charting gaps in realistic chart understanding in multimodal llms. *Advances in Neural Information Processing Systems*, 37:113569–113697, 2024.
- [221] Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*, 2021.
- [222] Thomas Wiegand, Gary J Sullivan, Gisle Bjontegaard, and Ajay Luthra. Overview of the h. 264/avc video coding standard. *IEEE Transactions on circuits and systems for video technology*, 13(7):560–576, 2003.
- [223] Ross Wightman, Hugo Touvron, and Hervé Jégou. Resnet strikes back: An improved training procedure in timm. *arXiv preprint arXiv:2110.00476*, 2021.
- [224] Ronald J Williams and David Zipser. A learning algorithm for continually running fully recurrent neural networks. *Neural computation*, 1(2):270–280, 1989.
- [225] Chao-Yuan Wu, Nayan Singhal, and Philipp Krahenbuhl. Video compression through image interpolation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 416–431, 2018.
- [226] Haoning Wu, Dongxu Li, Bei Chen, and Junnan Li. Longvideobench: A benchmark for long-context interleaved video-language understanding. *Advances in Neural Information Processing Systems*, 37:28828–28857, 2024.
- [227] Penghao Wu and Saining Xie. V*: Guided visual search as a core mechanism in multimodal llms. *arXiv preprint arXiv:2312.14135*, 2023.
- [228] Penghao Wu and Saining Xie. V?: Guided visual search as a core mechanism in multimodal llms. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13084–13094, 2024.
- [229] Penghao Wu, Li Chen, Hongyang Li, Xiaosong Jia, Junchi Yan, and Yu Qiao. Policy pre-training for autonomous driving via self-supervised geometric modeling. *arXiv preprint arXiv:2301.01006*, 2023.
- [230] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. <https://github.com/facebookresearch/detectron2>, 2019.
- [231] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3733–3742, 2018.
- [232] x.ai. <https://x.ai/blog/grok-1.5v>.

- [233] Long Xing, Qidong Huang, Xiaoyi Dong, Jiajie Lu, Pan Zhang, Yuhang Zang, Yuhang Cao, Conghui He, Jiaqi Wang, Feng Wu, et al. Pyramiddrop: Accelerating your large vision-language models via pyramid visual redundancy reduction. *arXiv preprint arXiv:2410.17247*, 2024.
- [234] Hu Xu, Gargi Ghosh, Po-Yao Huang, Dmytro Okhonko, Armen Aghajanyan, Florian Metze, Luke Zettlemoyer, and Christoph Feichtenhofer. Videoclip: Contrastive pre-training for zero-shot video-text understanding. *arXiv preprint arXiv:2109.14084*, 2021.
- [235] Ruyi Xu, Yuan Yao, Zonghao Guo, Junbo Cui, Zanlin Ni, Chunjiang Ge, Tat-Seng Chua, Zhiyuan Liu, Maosong Sun, and Gao Huang. Llava-uhd: an lmm perceiving any aspect ratio and high-resolution images. *arXiv preprint arXiv:2403.11703*, 2024.
- [236] Yilun Xu, Shengjia Zhao, Jiaming Song, Russell Stewart, and Stefano Ermon. A theory of usable information under computational constraints. *arXiv preprint arXiv:2002.10689*, 2020.
- [237] Shen Yan, Tao Zhu, Zirui Wang, Yuan Cao, Mi Zhang, Soham Ghosh, Yonghui Wu, and Jiahui Yu. Videococa: Video-text modeling with zero-shot transfer from contrastive captioners. *arXiv preprint arXiv:2212.04979*, 2022.
- [238] Wilson Yan, Volodymyr Mnih, Aleksandra Faust, Matei Zaharia, Pieter Abbeel, and Hao Liu. Elastictok: Adaptive tokenization for image and video. *arXiv preprint arXiv:2410.08368*, 2024.
- [239] Jianwei Yang, Chunyuan Li, Pengchuan Zhang, Xiyang Dai, Bin Xiao, Lu Yuan, and Jianfeng Gao. Focal attention for long-range interactions in vision transformers. *Advances in Neural Information Processing Systems*, 34:30008–30022, 2021.
- [240] Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything: Unleashing the power of large-scale unlabeled data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10371–10381, 2024.
- [241] Senqiao Yang, Yukang Chen, Zhuotao Tian, Chengyao Wang, Jingyao Li, Bei Yu, and Jiaya Jia. Visionzip: Longer is better but not necessary in vision language models. *arXiv preprint arXiv:2412.04467*, 2024.
- [242] Senqiao Yang, Yukang Chen, Zhuotao Tian, Chengyao Wang, Jingyao Li, Bei Yu, and Jiaya Jia. Visionzip: Longer is better but not necessary in vision language models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 19792–19802, 2025.

- [243] Hanrong Ye, Haotian Zhang, Erik Daxberger, Lin Chen, Zongyu Lin, Yanghao Li, Bowen Zhang, Haoxuan You, Dan Xu, Zhe Gan, et al. Mm-ego: Towards building egocentric multimodal llms for video qa. *arXiv preprint arXiv:2410.07177*, 2024.
- [244] Hanrong Ye, Chao-Han Huck Yang, Arushi Goel, Wei Huang, Ligeng Zhu, Yuanhang Su, Sean Lin, An-Chieh Cheng, Zhen Wan, Jinchuan Tian, et al. Omnivinci: Enhancing architecture and data for omni-modal understanding llm. *arXiv preprint arXiv:2510.15870*, 2025.
- [245] Jinhui Ye, Zihan Wang, Haosen Sun, Keshigeyan Chandrasegaran, Zane Durante, Cristobal Eyzaguirre, Yonatan Bisk, Juan Carlos Niebles, Ehsan Adeli, Li Fei-Fei, et al. Re-thinking temporal search for long-form video understanding. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 8579–8591, 2025.
- [246] Hongxu Yin, Arash Vahdat, Jose M Alvarez, Arun Mallya, Jan Kautz, and Pavlo Molchanov. A-vit: Adaptive tokens for efficient vision transformer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10809–10818, 2022.
- [247] Jiahui Yu, Yuanzhong Xu, Jing Yu Koh, Thang Luong, Gunjan Baid, Zirui Wang, Vijay Vasudevan, Alexander Ku, Yinfei Yang, Burcu Karagol Ayan, et al. Scaling autoregressive models for content-rich text-to-image generation. *arXiv preprint arXiv:2206.10789*, 2(3):5, 2022.
- [248] Qihang Yu, Mark Weber, Xueqing Deng, Xiaohui Shen, Daniel Cremers, and Liang-Chieh Chen. An image is worth 32 tokens for reconstruction and generation. In *Advances in Neural Information Processing Systems 37 (NeurIPS)*, 2024.
- [249] Sicheng Yu, Chengkai Jin, Huanyu Wang, Zhenghao Chen, Sheng Jin, Zhongrong Zuo, Xiaolei Xu, Zhenbang Sun, Bingni Zhang, Jiawei Wu, et al. Frame-voyager: Learning to query frames for video large language models. *arXiv preprint arXiv:2410.03226*, 2024.
- [250] Weihao Yu, Zhengyuan Yang, Linjie Li, Jianfeng Wang, Kevin Lin, Zicheng Liu, Xinchao Wang, and Lijuan Wang. Mm-vet: Evaluating large multimodal models for integrated capabilities. *arXiv preprint arXiv:2308.02490*, 2023.
- [251] Weihao Yu, Zhengyuan Yang, Linjie Li, Jianfeng Wang, Kevin Lin, Zicheng Liu, Xinchao Wang, and Lijuan Wang. Mm-vet: Evaluating large multimodal models for integrated capabilities. *arXiv preprint arXiv:2308.02490*, 2023.
- [252] Xin Yuan, Hongliang Fei, and Jinoo Baek. Efficient transformer adaptation with soft token merging. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3658–3668, 2024.

- [253] Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, et al. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. *arXiv preprint arXiv:2311.16502*, 2023.
- [254] Xiang Yue, Tianyu Zheng, Yuansheng Ni, Yubo Wang, Kai Zhang, Shengbang Tong, Yuxuan Sun, Botao Yu, Ge Zhang, Huan Sun, et al. Mmmu-pro: A more robust multi-discipline multimodal understanding benchmark. *arXiv preprint arXiv:2409.02813*, 2024.
- [255] Xiaohua Zhai, Alexander Kolesnikov, Neil Houlsby, and Lucas Beyer. Scaling vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12104–12113, 2022.
- [256] 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*, pages 11975–11986, 2023.
- [257] Boqiang Zhang, Kehan Li, Zesen Cheng, Zhiqiang Hu, Yuqian Yuan, Guanzheng Chen, Sicong Leng, Yuming Jiang, Hang Zhang, Xin Li, et al. Videollama 3: Frontier multimodal foundation models for image and video understanding. *arXiv preprint arXiv:2501.13106*, 2025.
- [258] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3):107–115, 2021.
- [259] Haotian Zhang, Mingfei Gao, Zhe Gan, Philipp Dufter, Nina Wenzel, Forrest Huang, Dhruvi Shah, Xianzhi Du, Bowen Zhang, Yanghao Li, et al. Mml. 5: Methods, analysis & insights from multimodal llm fine-tuning. *arXiv preprint arXiv:2409.20566*, 2024.
- [260] Pan Zhang, Xiaoyi Dong, Yuhang Zang, Yuhang Cao, Rui Qian, Lin Chen, Qipeng Guo, Haodong Duan, Bin Wang, Linke Ouyang, et al. Internlm-xcomposer-2.5: A versatile large vision language model supporting long-contextual input and output. *arXiv preprint arXiv:2407.03320*, 2024.
- [261] Peiyuan Zhang, Kaichen Zhang, Bo Li, Guangtao Zeng, Jingkan Yang, Yuanhan Zhang, Ziyue Wang, Haoran Tan, Chunyuan Li, and Ziwei Liu. Long context transfer from language to vision. *arXiv preprint arXiv:2406.16852*, 2024.
- [262] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018.

- [263] Wuyang Zhang, Zhezhi He, Luyang Liu, Zhenhua Jia, Yunxin Liu, Marco Gruteser, Dipankar Raychaudhuri, and Yanyong Zhang. Elf: accelerate high-resolution mobile deep vision with content-aware parallel offloading. In *Proceedings of the 27th Annual International Conference on Mobile Computing and Networking*, pages 201–214, 2021.
- [264] Yi-Fan Zhang, Huanyu Zhang, Haochen Tian, Chaoyou Fu, Shuangqing Zhang, Junfei Wu, Feng Li, Kun Wang, Qingsong Wen, Zhang Zhang, et al. Mme-realworld: Could your multimodal llm challenge high-resolution real-world scenarios that are difficult for humans? *arXiv preprint arXiv:2408.13257*, 2024.
- [265] Yuan Zhang, Chun-Kai Fan, Junpeng Ma, Wenzhao Zheng, Tao Huang, Kuan Cheng, Denis Gudovskiy, Tomoyuki Okuno, Yohei Nakata, Kurt Keutzer, et al. Sparsevlm: Visual token sparsification for efficient vision-language model inference. *arXiv preprint arXiv:2410.04417*, 2024.
- [266] Zhuoyang Zhang, Han Cai, and Song Han. Efficientvit-sam: Accelerated segment anything model without performance loss. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7859–7863, 2024.
- [267] Wenliang Zhao, Yongming Rao, Zuyan Liu, Benlin Liu, Jie Zhou, and Jiwen Lu. Unleashing text-to-image diffusion models for visual perception. *arXiv preprint arXiv:2303.02153*, 2023.
- [268] Li Zhaoping. *Understanding vision: theory, models, and data*. Oxford University Press (UK), 2014.
- [269] 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*, pages 633–641, 2017.
- [270] Junjie Zhou, Yan Shu, Bo Zhao, Boya Wu, Shitao Xiao, Xi Yang, Yongping Xiong, Bo Zhang, Tiejun Huang, and Zheng Liu. Mlvu: A comprehensive benchmark for multi-task long video understanding. *arXiv e-prints*, pages arXiv–2406, 2024.
- [271] Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Hao Tian, Yuchen Duan, Weijie Su, Jie Shao, et al. Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models. *arXiv preprint arXiv:2504.10479*, 2025.
- [272] Zirui Zhu, Hailun Xu, Yang Luo, Yong Liu, Kanchan Sarkar, Zhenheng Yang, and Yang You. FOCUS: Efficient keyframe selection for long video understanding. *CoRR*, abs/2510.27280, 2025.
- [273] Orr Zohar, Xiaohan Wang, Yann Dubois, Nikhil Mehta, Tong Xiao, Philippe Hansen-Estruch, Licheng Yu, Xiaofang Wang, Felix Juefei-Xu, Ning Zhang, et al. Apollo: An

exploration of video understanding in large multimodal models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 18891–18901, 2025.