Unveiling Encoder-Free Vision-Language Models

Haiwen Diao[1,2]* Yufeng Cui[2]* Xiaotong Li[3,2] Yueze Wang[2] Huchuan Lu[1]† Xinlong Wang[2]†

[1]Dalian University of Technology [2]Beijing Academy of Artificial Intelligence [3]Peking University

diaohw@mail.dlut.edu.cn, yfcui@baai.ac.cn, lixiaotong@stu.pku.edu.cn
yzwang@baai.ac.cn, lhchuan@dlut.edu.cn, wangxinlong@baai.ac.cn

Code & Models: https://github.com/baaivision/EVE

Abstract

Existing vision-language models (VLMs) mostly rely on vision encoders to extract visual features followed by large language models (LLMs) for visual-language tasks. However, the vision encoders set a strong inductive bias in abstracting visual representation, *e.g.*, resolution, aspect ratio, and semantic priors, which could impede the flexibility and efficiency of the VLMs. Training pure VLMs that accept the seamless vision and language inputs, *i.e.*, without vision encoders, remains challenging and rarely explored. Empirical observations reveal that direct training without encoders results in slow convergence and large performance gaps. In this work, we bridge the gap between encoder-based and encoder-free models, and present a simple yet effective training recipe towards pure VLMs. Specifically, we unveil the key aspects of training encoder-free VLMs efficiently via thorough experiments: (1) Bridging vision-language representation inside one unified decoder; (2) Enhancing visual recognition capability via extra supervision. With these strategies, we launch **EVE**, an encoder-free vision-language model that can be trained and forwarded efficiently. Notably, solely utilizing 35M publicly accessible data, **EVE** can impressively rival the encoder-based VLMs of similar capacities across multiple vision-language benchmarks. It significantly outperforms the counterpart Fuyu-8B [3] with mysterious training procedures and undisclosed training data. We believe that **EVE** provides a transparent and efficient route for developing pure decoder-only architecture across modalities.

1 Introduction

Recently, significant advancements in Large Language Models (LLMs) have catalyzed the emergence of Vision-Language Models (VLMs), showcasing powerful visual perception and cognition capability in visual question answering, image captioning, world knowledge understanding, *etc*. Typically, vision encoders (*e.g.*, CLIP [61] and EVA [22]) focus on extracting highly-compressed visual-semantic representations, succeeded by adaptable language models (*e.g.*, LLaMA [75] and Vicuna [10]) to handle vision-language alignments and instruction-following requirements. Nevertheless, these encoder-based VLMs have several potential drawbacks as shown in Figure 1:

(1) Image Resolution / Aspect Ratio. Existing LVMs are pre-trained with square and fixed-size images. However, this restriction forces VLMs to resize, pad, or partition images of varying shapes [48, 84, 51, 20], resulting in large layout distortion, fragmented connection between image slices, and extra computational burden [81], especially in terms of high-resolution images.

(2) Deployment Overhead. Generally, LVMs and LLMs are executed successively. The growing scale of LVMs [8, 73, 13] severely undermines computational efficiency in real-world deployment, especially when high-resolution images are divided and processed through LVMs multiple times.

---

*Equal contribution. † Correspondence to *lhchuan@dlut.edu.cn, wangxinlong@baai.ac.cn*.
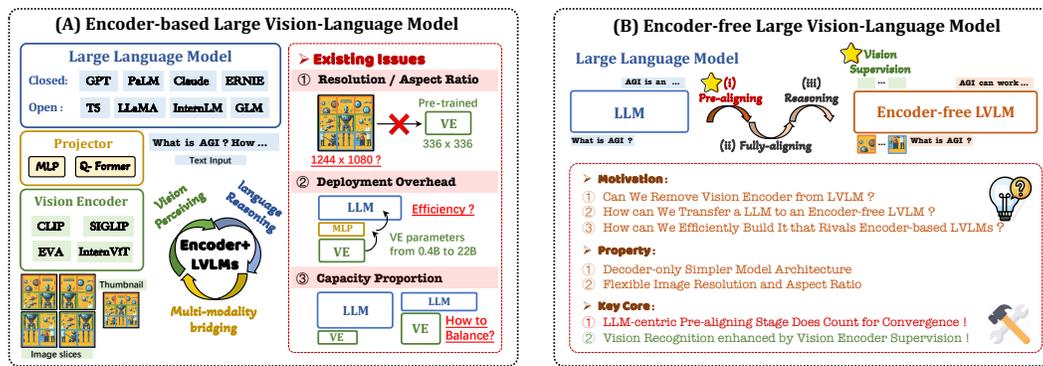
Figure 1: Overview of large (A) encoder-based and (B) encoder-free vision-language models. **Encoder-based VLMs** contain vision encoders (VE) and large language models (LLM), with a projector as a vision-language bridge, while **encoder-free VLMs** exclude vision encoders and handle vision perception and linguistic instruction simultaneously with one unified architecture.

(3) Model Capacity between LVMs and LLMs. Existing LVMs and LLMs are compartmentalized during the pre-training stage, raising uncertainty about how to match their capacities and capabilities. As the scale of LLMs rises from 1.3B to more than 540B, how to strike corresponding vision encoders to maximize their respective abilities remains tricky and elusive.

Given the above considerations, we wonder that: *Is it possible to bypass the constraints of vision encoders and integrate perception and reasoning capabilities into a single unified architecture?* Nevertheless, we empirically discover that even with enormous training data and large model sizes, this conceive remains highly challenging, suffering from greatly slow convergence and large performance gaps compared to encoder-based VLMs (*e.g.*, Fuyu-8B [3] vs. LLaVA-1.5 [49]). The essential problems of constructing encoder-free VLMs from scratch stem from:

(1) Representation Unity and Emergent Ability. Due to lacking high-quality image-text data, directly training encoder-free VLMs from scratch is impractical and expensive to learn abundant knowledge and universal representations across modalities. In contrast, the industry has capitalized on the wealth of language data and continuously updated various LLMs [75, 60, 5] with impressive capabilities. Consequently, we position LLMs as a central pivot and strive to compel LLMs per se to develop visual perception while preserving original linguistic proficiency. The process is not a light breeze. We observe that before scaling up pre-trained data, vision-language pre-aligning from an LLM-centric perspective does count. This dramatically prevents model collapse and optimization interference, leading to rapid yet stable model convergence and great performance gains.

(2) Visual Recognition Capability. Contrastive learning [61, 62], masked image modeling [45, 86] and auto-regressive generation [21, 16, 19] essentially attempt to prompt visual backbones to produce highly compressed holistic semantics and frequently neglect fine-grained visual clues [14, 15, 17, 18]. On the contrary, we transmit visual signals almost losslessly into encoder-free VLMs. This operation can allow VLMs to autonomously acquire the necessary visual-semantic information. By aligning with patch features captured from vision backbones and text labels predicted by encoder-based VLMs, we implicitly embed visual encoders with input-restricted patterns into encoder-free VLMs for comprehensive visual perception and cognition. This also sidesteps the expensive re-training process of visual encoders for arbitrary image shapes inside encoder-based VLMs.

From the above perspective, we launch **EVE-7B**, an encoder-free VLM evolved from Vicuna-7B [10] and trained with two 8-A100 (40G) nodes in ~9 days. The encoder-free property naturally supports high-resolution images with arbitrary aspect ratios. Notably, solely utilizing 35M publicly accessible data, we efficiently construct a decoder-only VLM that can impressively rival the encoder-based VLMs of similar capacities across multiple vision-language benchmarks. More importantly, we are pioneering a transparent, efficient, and practical route for subsequent VLM research. Our current version significantly outperforms the counterpart Fuyu-8B [3], which relies on mysterious training procedures and undisclosed training data. Beyond facilitating the transition from LLMs to VLMs, EVE holds promising potential for developing scalable and efficient training strategies for encoder-free multi-modality models, requiring fewer training data and device resources.

2 Related Work

2.1 Encoder-based Vision-Language Model

With the remarkable advancements in large language models (LLMs) [75, 60, 1, 24, 76] and large vision models (LVMs) [61, 22, 73, 8, 13], integrating LLMs with LVMs has become mainstream for building vision-language models (VLMs) effectively. Commercial VLMs extend the capabilities of their proprietary LLM to incorporate images, texts, audio, and videos, including Anthropic's Claude-3V series [1], StepFun's Step-1V [68], xAI's Grok-1.5V [79], Apple's MM1 [58], Google's Gemini series [74], and OpenAI's GPT-4V [83]. In terms of open-source VLMs, existing methods (*e.g.*, BLIP series [42, 43, 12], LLaVA series [50, 49, 51], Emu series [72, 70], Intern-VL [8, 9], and *etc*) have demonstrated promising performance by employing simple intermediate layers to bridge the gap between LVMs and LLMs. Recently, some studies [48, 49, 20, 28] have recognized the significance of input image resolution and aspect ratio for visual perception and cognition, particularly in the interpretation of document, chart, table, and infographic data. However, limited by pre-trained resolution, vision encoders are compelled to partition images into multiple slices or explore a dual-path architecture for low-resolution and high-resolution images respectively, resulting in significant image distortion, fragmented relationship between image slices, and additional computational consumption. For model deployment, some popular open-source libraries, *e.g.*, SGLang [91] or vLLM [39] already support inference acceleration for auto-regressive prediction of encoder-based VLMs. However, as the capacity of vision encoders scales up, the deployment efficiency of vision models would increasingly become a bottleneck for encoder-based VLMs. Meanwhile, how to match the capacities and capabilities between LVMs and LLMs for encoder-based VLMs remains a highly debatable problem with no definitive conclusion. Some studies [49, 51] highlight the notable benefits via substituting CLIP-ViT-B with stronger CLIP-ViT-L-336px in enhancing multimodal models alongside Vicuna-7B [10]. Conversely, other findings [65] indicate that larger vision encoders may not be necessary, as features of multi-scale smaller ones can approximate their performance. Moreover, recent state-of-the-art approaches [9, 70] exhibit significant performance improvements by introducing extremely LVMs. Based on these observations, we attempt to explore a pure decoder-only VLM excluding vision encoders and integrate vision-language understanding and reasoning capabilities into one unified architecture. This could effectively bypass the inherent problems inside encoder-based VLMs, including input constraints of pre-trained vision encoders, inefficiency issues of application deployment, and tricky capacity trade-offs between LVMs and LLMs.

2.2 Encoder-free Vision-Language Model

Fuyu-8B [3] stands out as a pure decoder-only network that processes image inputs without relying on an image encoder. Fuyu-8B's design naturally handles high-resolution images with arbitrary aspect ratios, because image patches are fed directly into the model through a simple linear projection layer. However, it demonstrates only average performance across vision-language benchmarks and lacks transparency in training strategies and data sources. This straightforward architecture has inspired further research [40, 31, 44], which focuses on developing powerful supervised instruction datasets to further enhance application capabilities. In response, we explore a practical and promising direction toward developing pure VLMs and breaking the obstacles between encoder-based and encoder-free VLMs. We reveal two crucial lessons: (1) Before scaling up pre-trained data, it is essential to prioritize vision-language pre-alignment from an LLM-centric perspective. This foundational step stabilizes the training process and alleviates optimization interference for integrating visual and linguistic information. (2) Enhancing image recognition capability via visual representation supervision and language conceptual alignment generates stronger visual representations for various vision-language tasks. With publicly available web data, our strategies efficiently accelerate model convergence and achieve performance that can surpass Fuyu-8B and match encoder-based VLMs.

3 Methodology

3.1 Model Architecture

Figure 2 illustrates the overall framework of our proposed EVE. Firstly, we initialize decoder-only EVE by Vicuna-7B [10] to acquire rich linguistic knowledge and strong instruction-following ability. Under the premise of encoder-free VLMs, we then construct a lightweight patch embedding layer and
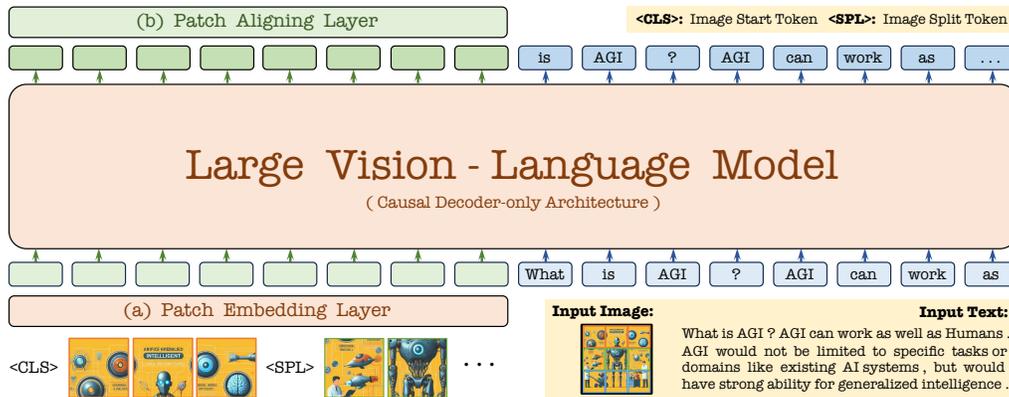
Figure 2: Overview of our proposed EVE. We start by encoding images using a concise (a) patch embedding layer, and then concatenate the patch and word features into a decoder-only network. Next, we facilitate image perception through visual representation supervision using a (b) patch aligning layer, and achieve linguistic conceptual alignment using a next-word prediction pretext task.
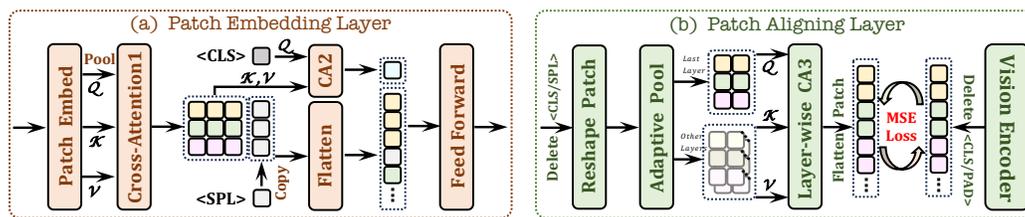

Figure 3: Overview of (a) patch embedding and (b) patch aligning layer. The former layer encodes images into patch features and employs cross-attention (CA1) within a limited receptive field to enhance representations. Meanwhile, a special <CLS> token provides a holistic view of each patch feature in the subsequent backbone. The latter layer removes all meaningless tokens and adjusts the size of patch features to align with the semantics of the same positional features in the vision encoder.

encode image and text inputs efficiently into the above network. After obtaining the network outputs, we attempt to align patch features with pair-wise ones from the vision encoder (VE) [61] through a hierarchical patch aligning layer. Meanwhile, EVE predicts next-word labels that are generated by multi-source encoder-based VLMs via Cross-Entropy (CE) loss.

Patch Embedding Layer (PEL). We introduce a succinct and trainable structure in Figure 3(a) to transmit images almost losslessly, rather than using deep encoders or tokenizers to compress image information into high-level semantic representations. Given an image with (H, W) resolution, we first apply a convolution layer to obtain a 2-D feature map with (h, w) sizes. To flexibly control computational overhead, we then implement an average pooling layer within each uncrossed feature slice. To further enhance these downsampled features, a Cross-Attention (CA1) layer is employed in a limited receptive field between each resulting feature and its corresponding slice. Besides, we employ a Cross-Attention (CA2) layer between a special <CLS> token and all patch features. The obtained feature serves as the starting symbol of the image and provides holistic information for patch features in the subsequent backbone. Considering the varying aspect ratios of image inputs, we insert a learnable newline <SPL> token at the end of each row of patch features. This helps the network understand the 2-D spatial structure and dependencies of the image. Finally, we flatten these features and pass them through a two-layer Feed Forward layer, which are then concatenated with text embeddings into one unified decoder-only architecture. Note that the aspect ratios of input images can be arbitrary, requiring no predefined sets, absolute position embedding, or partitioning operations to accommodate a pre-trained vision encoder.

Patch Aligning Layer (PAL). Besides coarse-grained text supervision, we further facilitate fine-grained representations by learning from the pre-trained vision encoder. Moreover, it is challenging to establish a shared space for aligning visual features with the vision encoder's output and mapping
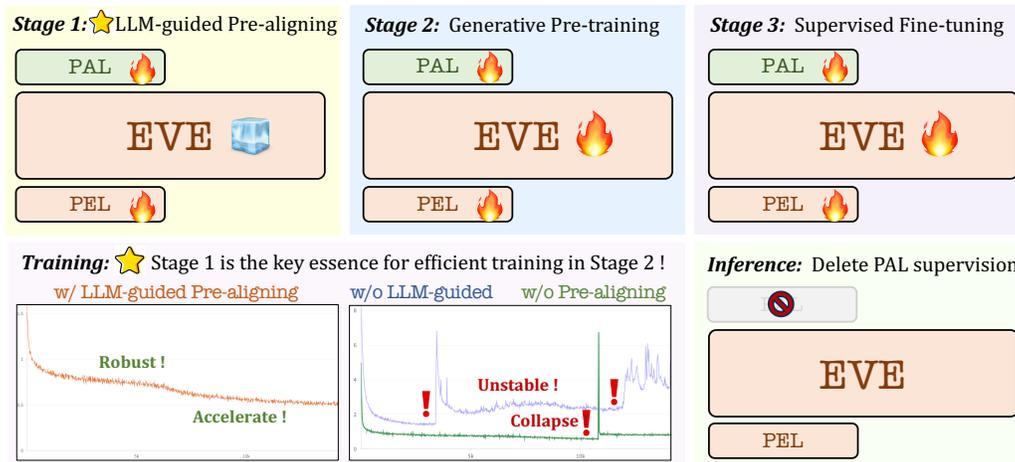
52548

Figure 4: Overview of training procedure with three successive stages. We perform initial vision-language alignment guided by a frozen LLM in Stage 1, and then update the entire backbone for Stage 2 and Stage 3. We empirically find that Stage 1 is quite crucial to avoid collapse and accelerate convergence, thereby enhancing training efficiency. Notably, PAL is removed during inference.

text features to language vocabulary simultaneously. Hence, we explore a hierarchical aggregation strategy in Figure 3(b) to integrate intermediate features across l layers from entire L layers (Interval $= \frac{L}{l}$). Specifically for the vision encoder, we discard meaningless <CLS>/<PAD> tokens from its final output and record the 2-D shape of valid patch fields. For EVE, we first exclude <CLS>/<SPL> tokens from intermediate features of selected layers, and reshape the sequential features back into their original 2-D shape, each of which aligns with the previously recorded shape from the vision encoder via an adaptive pooling layer. We then implement a layer-wise Cross-Attention (CA3) function, using tokens from the last layer as the Query and corresponding positional tokens from other layers as the Key and Value. We normalize the token features obtained from multi-layer aggregation to better match the normalized one from vision encoder one-to-one, utilizing Mean Squared Error (MSE) loss. Such an operation "implicitly" compresses a vision encoder with absolute position embedding (small resolution, fixed aspect ratio) into the decoder-only EVE framework (flexible resolution, arbitrary aspect ratio), enhancing the visual perception ignored by overly simplistic captions.

3.2 Training Procedure

In Figure 4, the entire training process is categorized into three successive stages, where we train EVE with publicly available image data captioned by existing VLMs, diverse QA data, and multi-modality dialogue datasets. Notably, after finishing training, we remove PAL supervision during inference.

LLM-guided Pre-training. The primary goal of this stage is to establish an initial connection between vision and language modalities, thereby driving LLM to understand the visual concepts and entities in the images from scratch. We introduce publicly available web-scale data in Table 1, including image-only: SA-1B [36], OpenImages [38]; and image-text: LAION [63]. We remove noisy text captions and reproduce 33M high-quality descriptions via Emu2 (17B) and LLaVA-1.5 (13B) as EVE-cap$_{33M}$. Here, only patch embedding and aligning layers are trainable to align with frozen Vicuna-7B [10] for better initializing an encoder-free VLM. Considering that these layers have limited parameter capacities and capabilities, we only adopt 16M of 33M image-text data (EVE-cap$_{16/33M}$) in this stage. We minimize CE loss with text labels and patch-wise MSE loss between EVE and vision encoder. We empirically discover that Stage 1 does count for efficient training, as it prevents collapse and accelerates convergence throughout the entire process.

Generative Pre-training. In this phase, we explore in-depth pretraining strategies by unfreezing patch embedding and aligning layers, and the full LLM modules for comprehending vision-language contents. Here, we make use of all 33M image-text pairs (EVE-cap$_{33M}$), and keep both text CE loss and image MSE loss as training objectives. Nevertheless, we discover that though multi-modality performance gradually increases, language capability suffers from a significant downtrend. To strike

Table 1: Details of pre-training datasets. We only extract 16/33M data for Stage 1, and utilize the full 33M data for Stage 2

Dataset	Captioner	#Samples
SA-1B [36]	LLaVA-1.5 (13B)	11M
OpenImages [38]	Emu2 (17B)	7M
LAION [63]	Emu2 (17B)	15M
EVE-cap$_{33M}$	Total	33M

Table 2: Details of fine-tuning datasets. We only adopt LLaVA-mix-665K for standard EVE-7B, and further train EVE (HD) with high resolution and all datasets

Task	Dataset	#Samples
OCR-VQA	AI2D [33], Synthdog [34], DVQA [32] ChartQA [57], DocVQA [11]	270K
Multi-Task	Vision-Flan [82], Bunny-695K [27] LLaVA-mix-665K [49]	1.5M

a compromise between enhancing vision-language capacities and maintaining linguistic competency, we opt for a relatively lower learning rate for stable optimization. Alternatively, involving language-only data remains a prospective avenue for further exploration [53]. Stage 2 serves as a catalyst, driving EVE per se to develop vision-language encoding and aligning capabilities.

Supervised Fine-tuning. After obtaining a well-aligned state, EVE is expected to possess further capabilities for following linguistic instructions and learning dialogue patterns. The entire architecture is updated using the same loss functions in Stage 2. Here, we utilize LLaVA-mix-665K [49] (LLaVA-mix$_{665K}$) with both VL and pure NLP dialogues as supervised fine-tuning (SFT) data to obtain the standard version of EVE-7B. Besides, we also attempt to enlarge the limitation of maximum resolution *only* in the SFT stage. To bridge the resolution gap between pre-training and fine-tuning stages, we further involve 1.2M SFT conversation data, as listed in Table 2 (including AI2D [33], DocVQA [11], and *etc*) to develop a high-resolution version, dubbed EVE-7B (HD).

4 Experiments

4.1 Training Settings

Data Preparation. (1) Pre-training Datasets. We train EVE using 33M publicly accessible samples from SA-1B [36], OpenImages [38], and LAION [63]. Following pre-processing pipelines [61, 46], we retain only samples with image resolutions higher than 448×448. In particular, the text data from LAION [63] are noisy and simplistic, lacking linguistic complexity. Besides, it suffers from duplication problems [78], which heavily hinder image diversity and quality. To extract more representative images, we apply K-means clustering technology on image features extracted by EVA-CLIP [71]. This process results in 50,000 clusters, from which we selected the 300 images closest to each cluster center, yielding a carefully curated subset of 15M image samples from LAION [63]. Subsequently, we utilize Emu2 (17B) and LLaVA-1.5 (13B) to regenerate high-quality image descriptions for the above three datasets and eliminate image-text samples with repetitive text or incomplete sentences. (2) Supervised Fine-tuning Datasets. We import LLaVA-mix-665K [49] for the standard EVE-7B, and collect a blended set of AI2D [33], Synthdog [34], DVQA [32], ChartQA [57], DocVQA [11], Vision-Flan [82], and Bunny-695K [27] for high-resolution version.

Implementation Details. EVE-7B is developed from Vicuna-7B [10] for vision-language domains. To control complexity, we limit the longest image edge to 672 for EVE-7B and 1344 for EVE-7B (HD), maintaining image aspect ratios unless otherwise specified. For visual supervision, we use CLIP-ViT-L/14 [61] as visual encoder and follow LLaVA-1.5 [49] protocols to add padding pixels and resize images to 336×336. Besides, the head is 8 for all cross-attention layers. The stride of the convolution layer and average pooling layer in PEL are 14 and 2, while the interval factor in PAL is 4. The patch features from EVE and VE are normalized by ℓ_2-norm before MSE loss.

We adopt the AdamW optimizer [35], warm-up strategy, and cosine scheduler for training EVE. The maximum learning rates for Stage 1, 2, 3 are 4×10^{-4}, 4×10^{-5}, 2×10^{-5}, while the number of batch size and training samples are 512, 512, 128 and 16M, 33M, 665K for EVE-7B, which spends ~9 days using two 8-A100 (40G) nodes. Notably, we only implement a high-resolution training strategy in supervised fine-tuning stage to obtain EVE-7B (HD) by involving an extra 1.2M SFT data.

4.2 Main Results

We evaluate EVE on a series of public visual-language benchmarks, including academic-task-oriented benchmarks (VQA-v2 [25], GQA [29], VizWiz [26], and TextVQA [67]), hallucination benchmarks

Table 3: **Comparison with state-of-the-art VLMs on vision-language benchmarks.** #Samples: the number of samples in the pretraining/supervised fine-tuning stage. AR: image aspect ratio. HD: high image resolution. We evaluate VLMs on VQA^{v2}: VQA-v2 [25]; GQA [29]; VizWiz [26]; SQA^I: ScienceQA-IMG [54]; VQA^T: TextVQA [67]; POPE [47]; MME [23]; MMB: MMBench [52]; SEED: SEED-Bench [41]; MM-Vet [89]. †Includes in-house data that is not publicly accessible

Method	LLM	#Samples	AR	VQA^{v2}	GQA	VizWiz	SQA^I	VQA^T	POPE	MME	MMB	SEED	MM-Vet
Encoder-based Vision-Language Models													
InstructBLIP	Vicuna-7B	129M/1.2M	Fix	–	49.2	34.5	60.5	50.1	–	–	36.0	53.4	26.2
IDEFICS-9B	LLaMA-7B	353M/1M	Fix	50.9	38.4	35.5	–	25.9	–	–	48.2	–	–
QwenVL	Qwen-7B	1.4B†/50M†	Fix	78.8	59.3	35.2	67.1	63.8	–	–	38.2	56.3	–
QwenVL-Chat	Qwen-7B	1.4B†/50M†	Fix	78.2	57.5	38.9	68.2	61.5	–	1487.5	60.6	58.2	–
LLaVA-1.5	Vicuna-7B	558K/665K	Fix	78.5	62.0	50.0	66.8	58.2	85.9	1510.7	64.3	58.6	30.5
InternVL-Chat	Vicuna-7B	4.98B/665K	Fix	79.3	62.9	52.5	–	57.0	86.4	1525.1	–	–	–
mPLUG-Owl2	LLaMA2-7B	400M/1.2M	Fix	79.4	56.1	54.5	68.7	58.2	86.2	1450.2	64.5	57.8	36.5
LVIS-4V	Vicuna-7B	558K/885K	Fix	79.6	62.6	51.8	68.3	58.7	86.0	1528.2	66.2	60.6	31.5
ShareGPT4V	Vicuna-7B	1.2M/665K	Fix	80.6	63.3	57.2	68.4	60.4	–	1567.4	68.8	–	37.6
Monkey (HD)	Qwen-7B	NA/1.44M	Enum	80.3	60.7	61.2	69.4	–	67.6	–	–	–	–
LLaVA-1.6 (HD)	Vicuna-7B	558K/790K	Enum	81.8	64.2	57.6	70.1	64.9	86.5	1519.3	67.4	64.7	43.9
Encoder-free Vision-Language Models													
EVE-7B	Vicuna-7B	33M/665K	Any	75.4	60.8	41.8	63.0	51.9	83.6	1217.3	49.5	54.3	25.6
Fuyu-8B (HD)	Persimmon-8B	–†/–†	Any	74.2	–	–	–	–	74.1	728.6	10.7	–	21.4
EVE-7B (HD)	Vicuna-7B	33M/1.8M	Any	78.6	62.6	51.1	64.9	56.8	85.0	1305.7	52.3	56.8	25.7

(POPE [47]), open-world multi-modal understanding benchmarks (MME [23], MMBench [52], SEED-Bench [41], and MM-Vet [89]), scientific problem benchmarks (ScienceQA-IMG [54]).

(1) In Table 3, EVE demonstrates superior performance compared to Fuyu-8B [3], an encoder-free counterpart, across various vision-language benchmarks, despite its smaller size. The incorporation of diverse SFT datasets and larger image sizes in EVE (HD) significantly enhances its image recognition capabilities. This highlights how our adaptable image processing approach benefits from higher resolution inputs and more comprehensive instructional data. (2) EVE (HD) shows the competitive performance when compared to encoder-based VLMs, *e.g.*, InternVL-Chat [8], mPLUG-Owl2 [85], LVIS-4V [77], ShareGPT4V [7], and Monkey [48], without requiring additional visual encoders for complex image encoding. Notably, EVE (HD) outperforms several VLMs (InstructBLIP [12], IDEFICS-9B [30], QwenVL-Chat [2]), and performs on par with the well-regarded LLaVA-1.5 [49]. Notably, EVE faces challenges in accurately responding to specific instructions like option letters and binary questions. Besides, extensive training steps with only vision-language data have somewhat diminished its language competency and instruction-following capability. Consequently, the performance of EVE-7B on MME, MMB, SQA, and MM-Vet benchmarks is relatively subpar.

The fact that EVE-7B can match encoder-based VLMs using a simpler architecture and publicly available million-scale data is quite encouraging. This discovery suggests that encoder-free VLMs, in virtue of appropriate training strategies and high-quality image-text data, can efficiently achieve performance on par with or even surpass that of encoder-based VLMs. This opens up promising avenues for addressing the challenges typically associated with encoder-based VLMs, such as inflexible input modes, inefficient deployment, and inconsistent capacity across modalities.

4.3 Ablation Studies

Configurations of patch embedding and aligning layer. To validate the effectiveness of the proposed PEL and PAL, we conduct experiments with different configurations. $EVE_{0.5M}$, EVE_{4M}, and EVE_{8M} represent models trained by LLaVA-pretrain-558K [49] (LLaVA-cap$_{558K}$), and subsets of 4M and 8M from the overall 33M EVE pre-training datasets (EVE-cap$_{4/33M}$, EVE-cap$_{8/33M}$) in Stages 1-2. All models use LLaVA-mix-665K [49] (LLaVA-mix$_{665K}$) in Stage 3. Besides, MSE_{NP} means aligning features between the current EVE output and the next tokens from the vision encoder. From Table 4, (1) we observe that removing any cross-attention layer in PEL results in a marginal performance degradation; (2) Pairwise patch alignments perform slightly better than next-patch prediction, possibly because the vision-specific layer is shallow, causing the network to prefer the current patch encoding over predicting the next patch in an auto-regressive manner; (3) Whether using small-scale or large-scale training data, visual supervision via PAL effectively enhances fine-grained image representations, thereby facilitating visual perception and accelerating model convergence.

Table 4: Configurations of PEL and PAL. $EVE_{0.5M}$, EVE_{4M}, EVE_{8M} utilize LLaVA-cap_{558K}, EVE-$cap_{4/33M}$, EVE-$cap_{8/33M}$ in Stage 1-2 with LLaVA-mix_{665K} in Stage 3. MSE_{NP} denotes the next-patch alignment

Model	VQA^{v2}	GQA	MMB	SEED
$EVE_{0.5M}$	58.5	49.8	35.3	39.3
- PEL CA1	58.2	49.4	34.4	40.1
- PEL CA2	58.4	49.8	35.8	39.3
+ MSE_{NP}	58.0	50.0	34.8	39.6
- PAL	55.6	47.5	34.5	37.8
EVE_{4M}	69.4	56.5	42.0	48.7
- PAL	66.4	55.3	41.2	47.5
EVE_{8M}	71.2	58.9	44.0	50.3
- PAL	69.4	57.3	42.7	49.2

Table 5: Configurations of training procedure. The top half all use LLaVA-mix_{665K} for fine-tuning. The bottom half all use EVE-cap_{33M} for pre-training

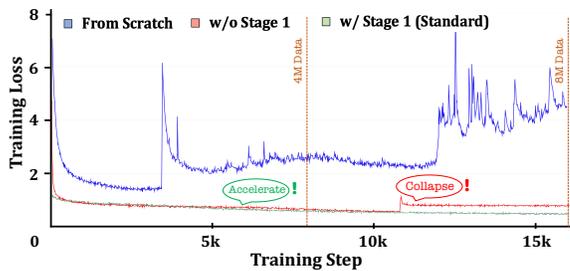| Model	VQA^{v2}	GQA	MMB	SEED
EVE w/o Stage 1				
+ Stage 2 (4M) ↑	64.6	54.1	40.6	45.4
+ Stage 2 (8M) ↓	50.2	42.5	26.8	36.2
EVE w/ Stage 1	59.3	51.0	36.5	39.7
+ Stage 2 (4M) ↑	69.4	56.5	42.0	48.7
+ Stage 2 (8M) ↑	71.2	58.9	44.0	50.3
EVE w/ Stage 1-2				
+ Stage 3 (665K) ↑	75.4	60.8	49.5	54.3
+ HD ↑	77.5	62.6	47.7	55.2
+ Stage 3 (1.8M) ↑	76.7	61.8	51.2	54.8
+ HD ↑	78.6	62.6	52.3	56.8

Figure 5: Training loss in Stage 2 using various strategies. Optimization remains unstable and prone to collapse, despite searching learning rate from 2e-5 to 1e-3.
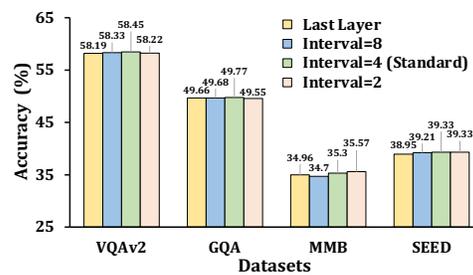

Figure 6: Configurations of interval factor. We observe that only the last vision output is insufficient and interval = 4 works the best.

Stage 1: LLM-guided pre-aligning stage does count for efficient training. In Table 5, we display the influence of our training strategies, including pre-training and supervised fine-tuning stages. We discover that (1) without LLM-guided pre-aligning between vision and language in Stage 1, the entire pre-training process becomes unstable and suffers from sudden collapse. This results in an unusual situation where performance rapidly decreases as data volume increases beyond a certain point, as shown in Figure 5. We assume that the randomly initialized PEL introduces significant noise and negative interference into network optimization for severely turning LLM into VLM, exacerbated by accumulation problems. Hence, Stage 1 stabilizes the training process and seamlessly transitions into Stage 2, emphasizing its indispensable role in efficient training to construct an encoder-free VLM. This stage prevents collapses and expedites convergence, guaranteeing a smooth and robust training route. (2) Increasing image resolutions and incorporating diverse instruction data in Stage 3 bring extreme performance gains after obtaining a well-aligned state across modalities in Stage 1-2. The high-resolution property greatly enhances fine-grained visual recognition, particularly for charts, tables, OCR information, and high-definition samples. Moreover, due to the extensive pre-training data, there is a pressing need for more supervised fine-tuning data to improve inherent linguistic proficiency, commonsense knowledge, and instruction-following patterns forgotten by EVE. (3) Training VLMs from scratch in Figure 5 often leads to instability and poses significant challenges for model optimization. Therefore, LLMs serve as an effective initialization for developing VLMs.

Merging patch features across layers outperforms the last layer output. Figure 6 illustrates the impact of varying interval factors for selecting cross-layer features in PAL. Optimal performance emerges when the interval factor is set to 4, yielding peak scores across VQA-v2, GQA, and SEED benchmarks. Our findings solidify the superiority of multi-layer aggregation over singularly relying on the last layer output. That is because a competitive interplay unfolds within the dynamic mapping space between vision and text, resulting in projection coupling and mutual interference within the last layer of LLM. Furthermore, hierarchical integration for vision features empowers multi-level representations, facilitating adaptive alignments with abundant characteristics from the vision encoder.
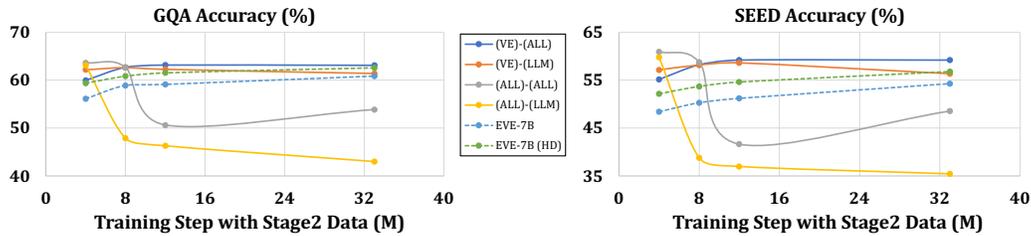
Figure 7: Scaling performance on GQA and SEED using LLaVA-1.5 as an encoder-based baseline. We first train its projector in Stage 1 with EVE-cap$_{16/33M}$. Here, (VE)-(LLM) indicates training Vision Encoder in Stage 2 and LLM in Stage 3, where we train the projector across all stages.

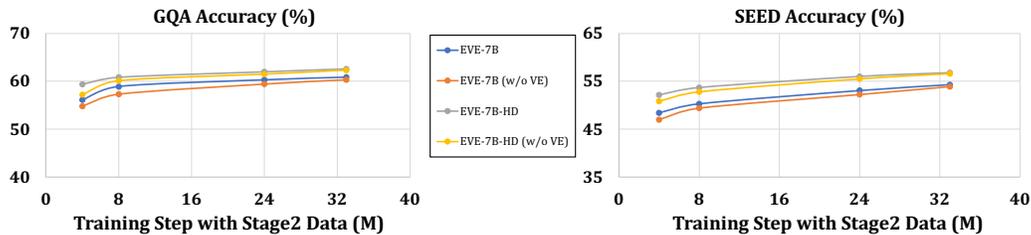Figure 8: Scaling performance on GQA and SEED w/ or w/o vision encoder supervision (PAL).

EVE consistently improves with more pre-training data, nearing encoder-based performance.
In Figure 7, we first train LLaVA-1.5's projector via EVE-cap$_{16/33M}$ in Stage 1, and then attempt various training strategies in Stage 2-3 following [85, 2, 8, 9, 53, 7]. For example, (VE)-(ALL) denotes training vision encoder in Stage 2 and all model parameters in Stage 3. We discover that: (1) Using synthetic captions, encoder-based model often suffers from collapse [88] due to simplified language structure and lack of knowledge details. Only the (VE)-(ALL) strategy avoids this issue and outperforms the baseline by freezing LLM weights during pre-training to preserve language proficiency and world knowledge. (2) EVE shows consistent performance gains with scaled-up data, gradually approaching the encoder-based model's performance. This may be because encoding and aligning vision and text in a unified network is more challenging and requires more training data, making encoder-free models less prone to overfitting compared to encoder-based ones. (3) For the vision encoder (CLIP) within the PAL of EVE, we discover that unfreezing it at all stages results in model collapse, which does not yield any improvements and instead increases computational costs.

Vision supervision aids in early convergence but becomes less critical during large data scale-up.
Vision encoder supervision improves training efficiency, especially with a relatively limited data scale compared to the billion scale of CLIP image-text pairs. Actually, this component can be simplified with sufficient data resources as shown in Figure 8. Our findings indicate that its influence diminishes over large data scales, and by the 24M mark, the difference in performance with or without this supervision is about 0.3-0.8%. This may be because large amounts of high-quality and detailed captions greatly enhance the understanding of visual information, thus gradually reducing the need for visual supervision. We emphasize that vision supervision (PAL) is not the crucial factor for training stability and scaling efficiency. Even with vision supervision, performance without LLM-guided pre-aligning in Stage 1 rapidly decreases as data volume increases beyond a certain point.

Efficient deployment and lower latency. Table 6 demonstrates several inference metrics, including floating point operations per second (FLOPs) and time delay. By eliminating the deep pre-trained vision encoder, EVE significantly accelerates the image encoding process, achieving an order of magnitude speed improvement over its counterparts. The CA1 component in PEL serves to control computational complexity. Even with a minimal downsampling rate of 2, EVE (HD) surpasses LLaVA-1.6 in both efficient deployment and reduced inference

Table 6: Model FLOPs and inference latency

Model	Vision Part		LLM Part	
	FLOPs(G)	Time(s)	FLOPs(T)	Time(s)
LLaVA-1.5	372	0.033	15.2	0.48
EVE-7B	42	0.003	15.2	0.48
LLaVA-1.6 (HD)	1860	0.13	76.1	2.07
EVE-7B (HD)	170	0.013	60.8	1.52

delay. By optimizing FLOPs and minimizing inference latency, EVE ensures rapid and efficient processing. This leads to a more responsive and resource-efficient system, positioning encoder-free VLMs as superior to encoder-based VLMs in practical applications.

5 Limitation and Discussion

EVE demonstrates a desired and powerful encoder-free decoder-only architecture that essentially solves various issues of existing encoder-based VLMs. Despite its promising results, a key limitation is the performance gap that still exists between EVE and state-of-the-art encoder-based VLMs. Additionally, due to time constraints, several questions and considerations remain:

(1) Further performance gain. We empirically find that training solely with vision-language data significantly reduces language capability (SQA score drops from 65.3% to 63.0%) while gradually improving multimodal performance. This indicates catastrophic forgetting of linguistic knowledge within the LLM. To address this, we suggest merging language and multimodal data appropriately or employing mixture-of-experts (MoE) strategies to mitigate vision-language interference.

(2) Encoder-free expectation. We validate that encoder-free VLMs can rival encoder-based VLMs but require larger training samples. How about the performance under the same LLM capacity and training data? We are working on the assumption that scaling up LLM capacity and training data would enable encoder-free VLMs to match or even surpass encoder-based VLMs, due to the nearly lossless image inputs that bypass pre-processing by vision encoders.

(3) Multi-modality inspiration. As a bonus, we gain inspiration for integrating additional modalities (*e.g.*, audio, video, thermal, depth, *etc*.). The core idea is to pre-align these modalities with a frozen LLM from an LLM-centric perspective before introducing large-scale unified training, supervised by corresponding single-modality encoders and linguistic conceptual alignments.

6 Conclusion

In this paper, we introduce EVE, a simple yet effective encoder-free model with a decoder-only architecture, designed to revolutionize vision-language understanding. Rather than merely presenting another VLM, we aim to unveil the under-explored bottlenecks towards constructing an encoder-free VLM. Our key findings reveal that: (1) An LLM-centric perspective, where vision and language modalities are pre-aligned, delivers robust, efficient, and consistent improvements as the pre-trained data scale up. (2) Incorporating fine-grained alignment with pre-trained vision encoders and linguistic conceptual supervision significantly enhances vision recognition, resulting in faster convergence and substantial performance gains. Remarkably, EVE performs on par with mainstream encoder-based VLMs solely relying on 35M publicly available data, and dramatically outperforms the counterpart Fuyu-8B, trained by undisclosed strategies and private datasets. As a pioneer, EVE provides a transparent and efficient roadmap for future VLM development, essentially tackling the challenges of multi-modal input processing, deployment efficiency, and model capacity across modalities. In the future, we are devoted to scaling up the model capacity with high-quality training data to explore the limits of encoder-free VLMs. Besides, we are working on translating the single-modality LLMs to large multimodal models with more modalities through our proposed training strategies.

Acknowledgments

References

[1] AI Anthropic. The claude 3 model family: Opus, sonnet, haiku. *Claude-3 Model Card*, 2024.

[2] Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, Jianxin Ma, Rui Men, Xingzhang Ren, Xuancheng Ren, Chuanqi Tan, Sinan Tan, Jianhong Tu, Peng Wang, Shijie Wang, Wei Wang, Shengguang Wu, Benfeng Xu, Jin Xu, An Yang, Hao Yang, Jian Yang, Shusheng Yang, Yang Yao, Bowen Yu, Hongyi Yuan, Zheng Yuan, Jianwei Zhang, Xingxuan Zhang, Yichang Zhang, Zhenru Zhang, Chang Zhou, Jingren Zhou, Xiaohuan Zhou, and Tianhang Zhu. Qwen technical report. *arXiv: 2309.16609*, 2023.

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

[4] Shikha Bordia and Samuel R. Bowman. Identifying and reducing gender bias in word-level language models. In *NAACL-HLT*, pages 7–15, 2019.

[5] Zheng Cai, Maosong Cao, Haojiong Chen, Kai Chen, Keyu Chen, Xin Chen, Xun Chen, Zehui Chen, Zhi Chen, Pei Chu, Xiaoyi Dong, Haodong Duan, Qi Fan, Zhaoye Fei, Yang Gao, Jiaye Ge, Chenya Gu, Yuzhe Gu, Tao Gui, Aijia Guo, Qipeng Guo, Conghui He, Yingfan Hu, Ting Huang, Tao Jiang, Penglong Jiao, Zhenjiang Jin, Zhikai Lei, Jiaxing Li, Jingwen Li, Linyang Li, Shuaibin Li, Wei Li, Yining Li, Hongwei Liu, Jiangning Liu, Jiawei Hong, Kaiwen Liu, Kuikun Liu, Xiaoran Liu, Chengqi Lv, Haijun Lv, Kai Lv, Li Ma, Runyuan Ma, Zerun Ma, Wenchang Ning, Linke Ouyang, Jiantao Qiu, Yuan Qu, Fukai Shang, Yunfan Shao, Demin Song, Zifan Song, Zhihao Sui, Peng Sun, Yu Sun, Huanze Tang, Bin Wang, Guoteng Wang, Jiaqi Wang, Jiayu Wang, Rui Wang, Yudong Wang, Ziyi Wang, Xingjian Wei, Qizhen Weng, Fan Wu, Yingtong Xiong, and et al. Internlm2 technical report. *arXiv: 2403.17297*, 2024.

[6] Nicholas Carlini, Florian Tramèr, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom B. Brown, Dawn Song, Úlfar Erlingsson, Alina Oprea, and Colin Raffel. Extracting training data from large language models. In *USENIX Security Symposium*, pages 2633–2650, 2021.

[7] Lin Chen, Jisong Li, Xiaoyi Dong, Pan Zhang, Conghui He, Jiaqi Wang, Feng Zhao, and Dahua Lin. Sharegpt4v: Improving large multi-modal models with better captions. *arXiv: 2311.12793*, 2023.

[8] Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, Bin Li, Ping Luo, Tong Lu, Yu Qiao, and Jifeng Dai. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. *arXiv: 2312.14238*, 2023.

[9] 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:2404.16821*, 2024.

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

[11] Christopher Clark and Matt Gardner. Simple and effective multi-paragraph reading comprehension. In *ACL*, pages 845–855, 2018.

[12] Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven C. H. Hoi. Instructblip: Towards general-purpose vision-language models with instruction tuning. In *NeurIPS*, 2023.

[13] Mostafa Dehghani, Josip Djolonga, Basil Mustafa, Piotr Padlewski, Jonathan Heek, Justin Gilmer, Andreas Peter Steiner, Mathilde Caron, Robert Geirhos, Ibrahim Alabdulmohsin, Rodolphe Jenatton, Lucas Beyer, Michael Tschannen, Anurag Arnab, Xiao Wang, Carlos Riquelme Ruiz, Matthias Minderer, Joan Puigcerver, Utku Evci, Manoj Kumar, Sjoerd van Steenkiste, Gamaleldin Fathy Elsayed, Aravindh Mahendran, Fisher Yu, Avital Oliver, Fantine Huot, Jasmijn Bastings, Mark Collier, Alexey A. Gritsenko,

https://doi.org/10.52202/079017-1665

Vighnesh Birodkar, Cristina Nader Vasconcelos, Yi Tay, Thomas Mensink, Alexander Kolesnikov, Filip Pavetic, Dustin Tran, Thomas Kipf, Mario Lucic, Xiaohua Zhai, Daniel Keysers, Jeremiah J. Harmsen, and Neil Houlsby. Scaling vision transformers to 22 billion parameters. In *ICML*, volume 202, pages 7480–7512, 2023.

[14] Haiwen Diao, Ying Zhang, Lin Ma, and Huchuan Lu. Similarity reasoning and filtration for image-text matching. In *AAAI*, pages 1218–1226, 2021.

[15] Haiwen Diao, Ying Zhang, Wei Liu, Xiang Ruan, and Huchuan Lu. Plug-and-play regulators for image-text matching. *TIP*, 32:2322–2334, 2023.

[16] Haiwen Diao, Bo Wan, Ying Zhang, Xu Jia, Huchuan Lu, and Long Chen. Unipt: Universal parallel tuning for transfer learning with efficient parameter and memory. In *CVPR*, 2024.

[17] Haiwen Diao, Ying Zhang, Shang Gao, Xiang Ruan, and Huchuan Lu. Deep boosting learning: A brand-new cooperative approach for image-text matching. *TIP*, 2024.

[18] Haiwen Diao, Ying Zhang, Shang Gao, Jiawen Zhu, Long Chen, and Huchuan Lu. Gssf: Generalized structural sparse function for deep cross-modal metric learning. *arXiv preprint arXiv:2410.15266*, 2024.

[19] Haiwen Diao, Bo Wan, Xu Jia, Yunzhi Zhuge, Ying Zhang, Huchuan Lu, and Long Chen. Sherl: Synthesizing high accuracy and efficient memory for resource-limited transfer learning. In *ECCV*, pages 75–95, 2025.

[20] Xiaoyi Dong, Pan Zhang, Yuhang Zang, Yuhang Cao, Bin Wang, Linke Ouyang, Songyang Zhang, Haodong Duan, Wenwei Zhang, Yining Li, Hang Yan, Yang Gao, Zhe Chen, Xinyue Zhang, Wei Li, Jingwen Li, Wenhai Wang, Kai Chen, Conghui He, Xingcheng Zhang, Jifeng Dai, Yu Qiao, Dahua Lin, and Jiaqi Wang. Internlm-xcomposer2-4khd: A pioneering large vision-language model handling resolutions from 336 pixels to 4k HD. *arXiv: 2404.06512*, 2024.

[21] 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.

[22] 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 *CVPR*, pages 19358–19369, 2023.

[23] Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Zhenyu Qiu, Wei Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, and Rongrong Ji. MME: A comprehensive evaluation benchmark for multimodal large language models. *arXiv: 2306.13394*, 2023.

[24] Google. Google bard. https://bard.google.com/, 2023.

[25] 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 *CVPR*, pages 6325–6334, 2017.

[26] 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 *CVPR*, pages 3608–3617, 2018.

[27] Muyang He, Yexin Liu, Boya Wu, Jianhao Yuan, Yueze Wang, Tiejun Huang, and Bo Zhao. Efficient multimodal learning from data-centric perspective. *arXiv: 2402.11530*, 2024.

[28] Wenyi Hong, Weihan Wang, Qingsong Lv, Jiazheng Xu, Wenmeng Yu, Junhui Ji, Yan Wang, Zihan Wang, Yuxiao Dong, Ming Ding, et al. Cogagent: A visual language model for gui agents. *arXiv:2312.08914*, 2023.

[29] Drew A. Hudson and Christopher D. Manning. GQA: A new dataset for real-world visual reasoning and compositional question answering. In *CVPR*, pages 6700–6709, 2019.

[30] IDEFICS Research Team. Introducing idefics: An open reproduction of state-of-the-art visual language model. https://huggingface.co/blog/idefics, 2023.

[31] Dongfu Jiang, Xuan He, Huaye Zeng, Con Wei, Max Ku, Qian Liu, and Wenhu Chen. Mantis: Interleaved multi-image instruction tuning. *arXiv:2405.01483*, 2024.

[32] Kushal Kafle, Brian L. Price, Scott Cohen, and Christopher Kanan. DVQA: understanding data visualizations via question answering. In *CVPR*, pages 5648–5656, 2018.

[33] Aniruddha Kembhavi, Mike Salvato, Eric Kolve, Minjoon Seo, Hannaneh Hajishirzi, and Ali Farhadi. A diagram is worth a dozen images. In *ECCV*, pages 235–251, 2016.

[34] Geewook Kim, Teakgyu Hong, Moonbin Yim, JeongYeon Nam, Jinyoung Park, Jinyeong Yim, Wonseok Hwang, Sangdoo Yun, Dongyoon Han, and Seunghyun Park. Ocr-free document understanding transformer. In *ECCV*, 2022.

[35] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

[36] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloé Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross B. Girshick. Segment anything. *arXiv: 2304.02643*, 2023.

[37] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A. Shamma, Michael S. Bernstein, and Li Fei-Fei. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *IJCV*, 123(1):32–73, 2017.

[38] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper R. R. Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Tom Duerig, and Vittorio Ferrari. The open images dataset V4: unified image classification, object detection, and visual relationship detection at scale. *arXiv: 1811.00982*, 2018.

[39] Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *SOSP*, pages 611–626, 2023.

[40] Bo Li, Peiyuan Zhang, Jingkang Yang, Yuanhan Zhang, Fanyi Pu, and Ziwei Liu. Otterhd: A high-resolution multi-modality model. *arXiv: 2311.04219*, 2023.

[41] Bohao Li, Rui Wang, Guangzhi Wang, Yuying Ge, Yixiao Ge, and Ying Shan. Seed-bench: Benchmarking multimodal llms with generative comprehension. *arXiv: 2307.16125*, 2023.

[42] Junnan Li, Dongxu Li, Caiming Xiong, and Steven C. H. Hoi. BLIP: bootstrapping language-image pre-training for unified vision-language understanding and generation. In *ICLR*, volume 162, pages 12888–12900, 2022.

[43] Junnan Li, Dongxu Li, Silvio Savarese, and Steven C. H. Hoi. BLIP-2: bootstrapping language-image pre-training with frozen image encoders and large language models. In *ICML*, volume 202, pages 19730–19742, 2023.

[44] Lin Li, Guikun Chen, Hanrong Shi, Jun Xiao, and Long Chen. A survey on multimodal benchmarks: In the era of large ai models, 2024. URL https://arxiv.org/abs/2409.18142.

[45] Xiaotong Li, Yixiao Ge, Kun Yi, Zixuan Hu, Ying Shan, and Ling-Yu Duan. mc-beit: Multi-choice discretization for image bert pre-training. In *ECCV*, pages 231–246, 2022.

[46] Xiaotong Li, Fan Zhang, Haiwen Diao, Yueze Wang, Xinlong Wang, and Ling-Yu Duan. Densefusion-1m: Merging vision experts for comprehensive multimodal perception. *arXiv preprint arXiv:2407.08303*, 2024.

[47] Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne Xin Zhao, and Ji-Rong Wen. Evaluating object hallucination in large vision-language models. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *EMNLP*, pages 292–305, 2023.

[48] 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. *arXiv: 2311.06607*, 2023.

[49] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. *arXiv: 2310.03744*, 2023.

[50] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *NeurIPS*, 2023.

[51] Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. Llava-next: Improved reasoning, ocr, and world knowledge, 2024. URL https://llava-vl.github.io/blog/2024-01-30-llava-next/.

[52] Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, Kai Chen, and Dahua Lin. Mmbench: Is your multi-modal model an all-around player? *arXiv: 2307.06281*, 2023.

[53] Haoyu Lu, Wen Liu, Bo Zhang, Bingxuan Wang, Kai Dong, Bo Liu, Jingxiang Sun, Tongzheng Ren, Zhuoshu Li, Hao Yang, Yaofeng Sun, Chengqi Deng, Hanwei Xu, Zhenda Xie, and Chong Ruan. Deepseek-vl: Towards real-world vision-language understanding. *arXiv: 2403.05525*, 2024.

[54] Pan Lu, Swaroop Mishra, Tony Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering. In *NeurIPS*, 2022.

[55] Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, Alan L. Yuille, and Kevin Murphy. Generation and comprehension of unambiguous object descriptions. In *CVPR*, pages 11–20, 2016.

[56] Kenneth Marino, Mohammad Rastegari, Ali Farhadi, and Roozbeh Mottaghi. Ok-vqa: A visual question answering benchmark requiring external knowledge. In *CVPR*, pages 3195–3204, 2019.

[57] Ahmed Masry, Xuan Long Do, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. Chartqa: A benchmark for question answering about charts with visual and logical reasoning. In *ACL*, pages 2263–2279, 2022.

[58] Brandon McKinzie, Zhe Gan, Jean-Philippe Fauconnier, Sam Dodge, Bowen Zhang, Philipp Dufter, Dhruti Shah, Xianzhi Du, Futang Peng, Floris Weers, Anton Belyi, Haotian Zhang, Karanjeet Singh, Doug Kang, Ankur Jain, Hongyu Hè, Max Schwarzer, Tom Gunter, Xiang Kong, Aonan Zhang, Jianyu Wang, Chong Wang, Nan Du, Tao Lei, Sam Wiseman, Guoli Yin, Mark Lee, Zirui Wang, Ruoming Pang, Peter Grasch, Alexander Toshev, and Yinfei Yang. MM1: methods, analysis & insights from multimodal LLM pre-training. *arXiv: 2403.09611*, 2024.

[59] Anand Mishra, Shashank Shekhar, Ajeet Kumar Singh, and Anirban Chakraborty. Ocr-vqa: Visual question answering by reading text in images. In *ICDAR*, pages 947–952, 2019.

[60] OpenAI. GPT-4 technical report. *arXiv: 2303.08774*, 2023.

[61] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *ICML*, volume 139, pages 8748–8763, 2021.

[62] Yongming Rao, Wenliang Zhao, Guangyi Chen, Yansong Tang, Zheng Zhu, Guan Huang, Jie Zhou, and Jiwen Lu. Denseclip: Language-guided dense prediction with context-aware prompting. In *CVPR*, pages 18061–18070, 2022.

[63] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *NeurIPS*, 35:25278–25294, 2022.

[64] Dustin Schwenk, Apoorv Khandelwal, Christopher Clark, Kenneth Marino, and Roozbeh Mottaghi. A-okvqa: A benchmark for visual question answering using world knowledge. In *ECCV*, pages 146–162, 2022.

[65] Baifeng Shi, Ziyang Wu, Maolin Mao, Xin Wang, and Trevor Darrell. When do we not need larger vision models? *arXiv: 2403.13043*, 2024.

[66] Oleksii Sidorov, Ronghang Hu, Marcus Rohrbach, and Amanpreet Singh. Textcaps: A dataset for image captioning with reading comprehension. In *ECCV*, volume 12347, pages 742–758, 2020.

[67] Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. Towards VQA models that can read. In *CVPR*, 2019.

[68] StepFun Research Team. Step-1v: A hundred billion parameter multimodal large model. https://platform.stepfun.com, 2024.

[69] Emma Strubell, Ananya Ganesh, and Andrew McCallum. Energy and policy considerations for deep learning in NLP. In *ACL*, pages 3645–3650, 2019.

[70] Quan Sun, Yufeng Cui, Xiaosong Zhang, Fan Zhang, Qiying Yu, Zhengxiong Luo, Yueze Wang, Yongming Rao, Jingjing Liu, Tiejun Huang, and Xinlong Wang. Generative multimodal models are in-context learners. *arXiv: 2312.13286*, 2023.

[71] Quan Sun, Yuxin Fang, Ledell Wu, Xinlong Wang, and Yue Cao. EVA-CLIP: improved training techniques for CLIP at scale. *arXiv: 2303.15389*, 2023.

[72] Quan Sun, Qiying Yu, Yufeng Cui, Fan Zhang, Xiaosong Zhang, Yueze Wang, Hongcheng Gao, Jingjing Liu, Tiejun Huang, and Xinlong Wang. Generative pretraining in multimodality. *arXiv: 2307.05222*, 2023.

[73] Quan Sun, Jinsheng Wang, Qiying Yu, Yufeng Cui, Fan Zhang, Xiaosong Zhang, and Xinlong Wang. EVA-CLIP-18B: scaling CLIP to 18 billion parameters. *arXiv: 2402.04252*, 2024.

[74] 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: 2312.11805*, 2023.

[75] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models. *arXiv: 2302.13971*, 2023.

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

[77] Junke Wang, Lingchen Meng, Zejia Weng, Bo He, Zuxuan Wu, and Yu-Gang Jiang. To see is to believe: Prompting GPT-4V for better visual instruction tuning. *arXiv: 2311.07574*, 2023.

[78] Ryan Webster, Julien Rabin, Loic Simon, and Frederic Jurie. On the de-duplication of laion-2b. *arXiv preprint arXiv:2303.12733*, 2023.

[79] X.ai. Grok-1.5 vision preview. `https://x.ai/blog/grok-1.5v`, 2024.

[80] Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, and Daxin Jiang. Wizardlm: Empowering large language models to follow complex instructions. *arXiv: 2304.12244*, 2023.

[81] 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: 2403.11703*, 2024.

[82] Zhiyang Xu, Trevor Ashby, Chao Feng, Rulin Shao, Ying Shen, Di Jin, Qifan Wang, and Lifu Huang. Vision-flan:scaling visual instruction tuning, 2023. URL `https://vision-flan.github.io/`.

[83] Zhengyuan Yang, Linjie Li, Kevin Lin, Jianfeng Wang, Chung-Ching Lin, Zicheng Liu, and Lijuan Wang. The dawn of lmms: Preliminary explorations with gpt-4v (ision). *arXiv: 2309.17421*, 9, 2023.

[84] Jiabo Ye, Anwen Hu, Haiyang Xu, Qinghao Ye, Ming Yan, Guohai Xu, Chenliang Li, Junfeng Tian, Qi Qian, Ji Zhang, et al. Ureader: Universal ocr-free visually-situated language understanding with multimodal large language model. *arXiv: 2310.05126*, 2023.

[85] Qinghao Ye, Haiyang Xu, Jiabo Ye, Ming Yan, Anwen Hu, Haowei Liu, Qi Qian, Ji Zhang, Fei Huang, and Jingren Zhou. mplug-owl2: Revolutionizing multi-modal large language model with modality collaboration. *arXiv: 2311.04257*, 2023.

[86] Kun Yi, Yixiao Ge, Xiaotong Li, Shusheng Yang, Dian Li, Jianping Wu, Ying Shan, and Xiaohu Qie. Masked image modeling with denoising contrast. In *ICLR*, 2023.

[87] Licheng Yu, Patrick Poirson, Shan Yang, Alexander C. Berg, and Tamara L. Berg. Modeling context in referring expressions. In *ECCV*, volume 9906, pages 69–85, 2016.

[88] Qiying Yu, Quan Sun, Xiaosong Zhang, Yufeng Cui, Fan Zhang, Yue Cao, Xinlong Wang, and Jingjing Liu. Capsfusion: Rethinking image-text data at scale. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14022–14032, 2024.

[89] 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: 2308.02490*, 2023.

[90] Bo Zhao, Boya Wu, and Tiejun Huang. SVIT: scaling up visual instruction tuning. *arXiv: 2307.04087*, 2023.

[91] Lianmin Zheng, Liangsheng Yin, Zhiqiang Xie, Jeff Huang, Chuyue Sun, Cody Hao Yu, Shiyi Cao, Christos Kozyrakis, Ion Stoica, Joseph E. Gonzalez, Clark W. Barrett, and Ying Sheng. Efficiently programming large language models using sglang. *arXiv: 2312.07104*, 2023.

A Appendix

A.1 Ethics Statement

We introduce EVE, a pioneering encoder-free VLM, along with efficient training strategies for developing an encoder-free VLM. It is important to address the ethical considerations surrounding large pre-trained models, such as potential bias or discrimination [6] and privacy concerns [4] in the extensive training data. Additionally, the computational cost and environmental impact have become significant topics of discussion [69]. To our knowledge, previous VLMs have not thoroughly investigated their impact on issues like bias or information leakage, which warrants further exploration. Unlike closed-source VLMs, we train EVE using publicly available data, thereby raising fewer ethical concerns. Furthermore, our approach reduces model deployment costs by lowering computational consumption and server resource requirements. Notably, our training experiments are conducted in a data center powered entirely by renewable energy sources.

A.2 Dataset Details

In Table 7, we provide a detailed description of SFT datasets, along with their sampling rates during Stage 3. The majority are derived from LLaVA-mix-665K [49] and Bunny-695K [27]. Besides, we compile a blended set from AI2D [33], Synthdog [34], DVQA [32], ChartQA [57], DocVQA [11] and Vision-Flan [82] to leverage the advantages of high-resolution inputs.

Table 7: **Supervised Fine-tuning Data Mixture**

Data	Sample Raito	Size	Task
LLaVA [49]	8.7%	158K	Conversation
SVIT-core-150K [90]	8.7%	158K	
ShareGPT [7]	2.2%	40K	Text-only
WizardLM-70K [80]	3.9%	70K	
VQAv2 [25]	9.1%	83K	General QA
GQA [29]	7.9%	72K	
OKVQA [56]	1.0%	9K	
A-OKVQA [64]	7.3%	66K	Knowledge QA
TextCaps [66]	2.4%	22K	OCR QA
Synthdog [34]	1.7%	30K	
OCRVQA [59]	8.8%	80K	
RefCOCO [87, 55]	5.3%	48K	Grounding
VG [37]	9.5%	86K	
Vision-Flan [82]	10.2%	186K	Multi-task
ChartQA [57]	1.0%	18K	Chart QA
DVQA [32]	11.0%	200K	
AI2D [33]	0.7%	12K	Science QA
DocQA [11]	0.6%	10K	Document
Total	100.0%	1.35M	

Table 8: **Hyperparameter settings**

Hyperparameter	Stage-1	Stage-2	Stage-3
Batch Size	512	512	128
Learning Rate (lr)	4e-4	4e-5	2e-5
LR Schedule	cosine decay		
LR Warmup Ratio	0.03	0.01	0.01
Weight Decay	0		
Epoch	1		
Optimizer	AdamW		
DeepSpeed stage	3	3	3

A.3 Training Details

We report the detailed training hyper-parameter settings of EVE during the three stages in Table 8.

A.4 Qualitative Results

The qualitative results are demonstrated for comparisons. We showcase the ability of EVE in the aspects of OCR, reasoning, and image captioning, respectively.

	Visual input example

User	Please describe image contents in detail.
LLaVA-1.5	The image features a man with an unusual appearance, standing in front of a mirror. He is wearing a purple shirt and appears to be a robot or a character from a comic. The man is taking a selfie with his cell phone, holding it up to his face while capturing the moment. In the scene, a potted plant is positioned nearby, adding a touch of greenery to the setting. Two other people can be seen in the background, one situated to the left and another towards the right, but they are not the main focus of the scene. Additionally, a bottle is visible at the far left of the image, potentially a decorative item or part of the room's design.
EVE	The image features a man taking a picture of himself in a bathroom mirror. He is standing in front of a sink, which is located on the left side of the room. The man is holding a cell phone in his hand, capturing the moment. The bathroom is decorated with a few potted plants, one of which is placed near the sink, and another one is located closer to the right side of the room. There is also a vase in the scene, positioned near the center of the room.
EVE (HD)	The image depicts a bathroom scene. A man is taking a selfie in the mirror, which is mounted on the wall. He is wearing a purple shirt and leopard print pants. He is holding a camera, capturing his reflection. The mirror is large and spans a significant portion of the wall. The bathroom features a white sink with a silver faucet, which is located on the left side of the image. The sink is positioned on a countertop. Above the sink, there is a mirror. The countertop is made of granite and has a white tile backsplash. To the right of the sink, there is a white tiled wall. A white tiled floor is visible in the lower part of the image. The bathroom also contains a plant with white flowers, which is placed in a vase. The plant is located near the sink and the mirror. The flowers are white and are arranged in a bouquet. There is also a poster on the wall, which is positioned to the right of the mirror. The poster is black and white and is quite large. It features a picture of a remote control and some text. The image also includes a white tile wall, which is located behind the man. The wall is clean and well-maintained. Overall, the image captures a moment in a bathroom, with a man taking a selfie in a mirror, a plant with white flowers, a sink, a mirror, a poster, and a tiled wall.

	Visual input example

User	Please describe image contents in detail.
LLaVA-1.5	The image depicts an unconventional scene of a man wearing a costume with a jaw, nose, and chin made of rocks. This unique character appears to be a humanoid figure made from stones, with a gold background that highlights the costume. The man's face, with its interesting rock-formed facial features, seems to be the focal point of the image.
EVE	The image features a man holding a wooden stick, possibly a staff or a walking stick, in front of a building. The man appears to be an older individual with a beard. He is standing in front of a wall, which could be a part of the building or a separate structure. The scene is set outdoors, with the man being the main focus of the image.
EVE (HD)	The image features a man with a long beard and a bald head. He is holding a wooden stick in his hand. The man's face is partially covered by a large yellow text that reads "Neanderthal". The man's face is also partially covered by a large yellow text that reads "Years 7 and 8". The man's face is also partially covered by a large yellow text that reads "Worst of the Day". The man's face is also partially covered by a large yellow text that reads "Friday 23 January 2015". The man's face is also partially covered by a large yellow text that reads "Years 7 and 8". The man's face is also partially covered by a large yellow text that reads "Neanderthal".

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main claims are made in abstract and introduction (Section 1).

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: This paper discusses the limitations in Section 5.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

Justification: This paper does not include theory assumptions or proofs.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: The training setting is listed in Section 4.1.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: The codes will be publicly available and the training data is described in Table 1 and Table 2.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: The training and testing details are listed in Section 3.2 and Section 4.1, respectively.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [No]

Justification: For fair comparsion, we follow the standard and commonly used setting for experiment evaluations.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)

- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: The computer resources are listed in Section 4.1.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

Answer: [Yes]

Justification: This research conforms with the NeurIPS Code of Ethics.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: The discussion is conducted in Appendix A.1.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [Yes]

Justification: The discussion is conducted in Appendix A.1.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: The original owners of the code, data, models are properly credited in this paper (Section 4.1).

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [NA]

Justification: The paper does not release new assets.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.