
HaploVL: A Single-Transformer Baseline for Multi-Modal Understanding

Rui Yang¹ Lin Song^{2 *+} Yicheng Xiao³ Runhui Huang¹ Yixiao Ge² Ying Shan² Hengshuang Zhao^{1 *}

Abstract

Recent advancements in large language models (LLMs) have significantly propelled the development of large multi-modal models (LMMs), highlighting the potential for general and intelligent assistants. However, most LMMs model visual and textual modalities separately, leading to recent efforts to develop native LMMs using a single transformer. Despite the promise, these native models are resource-intensive and often exhibit performance gaps compared to their compositional counterparts. To alleviate this issue, we propose a simple yet efficient method to construct a baseline for the native and end-to-end large multi-modal model in a single transformer. First, we propose a new early-fusion LMM that can fuse multi-modal inputs in the early stage and respond to visual instructions in an auto-regressive manner. Second, we devise an efficient training recipe for the proposed model, which harnesses the prior knowledge of the pre-trained models, addressing both the performance limitations and the challenge of resource consumption. The proposed model demonstrates superior performance compared to other LMMs using one transformer and significantly narrows the performance gap with compositional LMMs.

1. Introduction

Large language models (LLMs) (OpenAI, 2023; Dubey et al., 2024; Yang et al., 2024a) have recently made significant strides in the realm of artificial intelligence. This progress has substantially accelerated the development of large multi-modal models (LMMs), which include both proprietary commercial models (Achiam et al., 2023; Team et al., 2024) and open-source models (Dai et al., 2023; Zhu et al., 2023a; Liu et al., 2024a). These models facilitate

¹The University of Hong Kong ²ARC Lab, Tencent PCG ³Tsinghua University. Team leader: Lin Song <ronnysong@tencent.com>. Correspondence to: Lin Song <ronnysong@tencent.com>, Hengshuang Zhao <hszhao@cs.hku.hk>.

Project page: <https://github.com/Tencent/HaploVLM>.

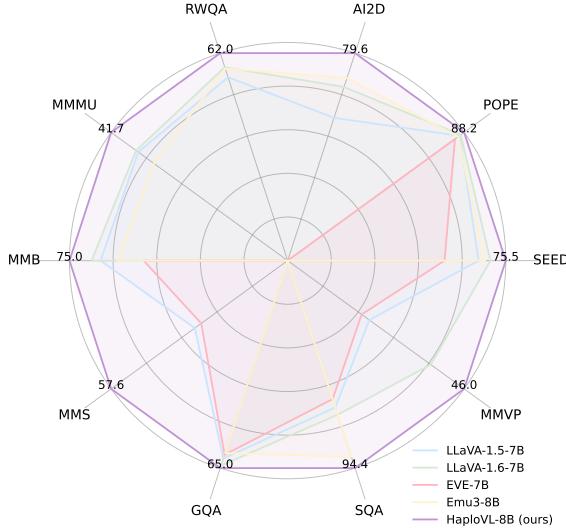


Figure 1: Performance comparison with single-transformer models on multi-modal understanding benchmarks. Our HaploVL demonstrates superiority over other counterparts.

complex vision-language dialogues and interactions. The majority of open-source models (Liu et al., 2024c; Dai et al., 2023) leverage one or more separate vision components to model the visual modality, thus equipping LLMs with visual understanding and reasoning capabilities. For instance, the LLaVA series (Liu et al., 2024c;a) directly harnesses the pretrained CLIP vision encoder (Radford et al., 2021) to extract high-level vision embeddings and uses a projector to connect these embeddings with LLMs.

Since language is human-generated signals that have already been abstracted (He et al., 2022), the text embeddings produced by the word embedding layer contain semantic information and are high-level. Therefore, it is reasonable to combine text embeddings with vision embeddings from a pre-trained vision encoder, as both types of embeddings are semantic. However, the off-the-shelf vision encoder (Radford et al., 2021) tends to produce highly compressed global semantics and neglect fine-grained visual information. Thus, it may fail to extract effective visual cues required by the text, leading to difficulties for LMMs in handling fine-grained tasks (Tong et al., 2024).

To address this issue, we propose an early-fusion LMM named HaploVL. Our model fuses the vision and text em-

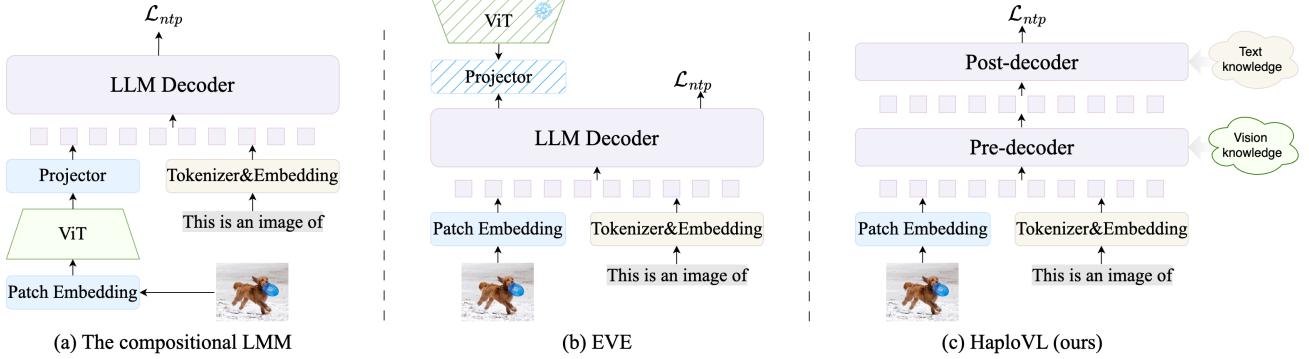


Figure 2: Architecture comparison with the compositional LMM (Liu et al., 2024a), EVE (Diao et al., 2024). In our HaploVL, the pre-decoder dynamically extracts vision cues based on the input text, and the post-decoder further fuses the multi-modal embeddings. Our model inherits the prior knowledge from vision and language models, thus requiring less data than EVE.

Method	Data (M)	Early fusion	Seamntic align	SEED	MMStar _{FP}	MMVP
LLaVA-1.5-7B	1.2	X	✓	66.1	24.0	21.3
EVE-7B	35.0	✓	X	54.3	24.2	19.3
HaploVL-7B	1.2	✓	✓	67.5	28.9	24.7

Table 1: Comparison with compositional LLaVA (Liu et al., 2024a) and unified EVE (Diao et al., 2024) on multi-modal benchmarks: SEED-Bench (Li et al., 2023a), the fine-grained split of MMStar (Chen et al., 2024a), and MMVP (Tong et al., 2024).

beddings at an early stage, enabling text embeddings to autonomously acquire the necessary vision cues. Specifically, HaploVL uses a lightweight patch embedding layer, *a single linear layer*, to embed visual input and a text embedding layer to process textual inputs. Subsequently, the transformer backbone extracts the necessary vision information based on the text input and generates language responses according to the resulting fused representations.

Some recent studies (Bavishi et al., 2023; Team, 2024; Diao et al., 2024) also fall under the category of early-fusion LMMs and have endeavored to develop a unified multi-modal transformer with a concise inference process. For example, Fuyu (Bavishi et al., 2023) directly utilizes a simple linear layer instead of a vision encoder to embed the input image and leaves the mixed modality sequence to the subsequent transformer. EVE (Diao et al., 2024) aims to replicate Fuyu by being distilled from a fixed vision encoder, thus reducing the training data. However, it forces alignment between a large language model (7B) and a small ViT (300M) without allowing the LMM to learn from high-level vision features. Therefore, there is a significant performance gap between it and compositional LMMs on vision-language benchmarks, despite using 35M training data.

To this end, our HaploVL utilizes a pre-decoder to autonomously acquire the necessary vision cues according to text information, and a post-decoder to further process

the extracted high-level multi-modal embeddings. Since training such a model from scratch is very expensive, e.g., the energy consumption required to pre-train the Chameleon-30B (Team, 2024) is equivalent to what is needed to power a Tesla Model 3 to travel around the equator for about 225 times¹, we propose to leverage prior knowledge acquired from pre-trained models. This is because the pre-trained models have gained extensive knowledge by training on massive data, e.g., the CLIP vision encoder (Radford et al., 2021) obtained vision-based knowledge by seeing billions of images, and Llama (Dubey et al., 2024) gained text-based knowledge by seeing trillions of text tokens. Specifically, the pre-decoder inherits prior vision knowledge from a vision encoder while simultaneously processing text and vision modalities to perform modal expansion. Plus, the LLM retains its prior text knowledge and learns to take vision embeddings as a condition. In this way, we significantly reduce the required data and training costs in comparison to other early-fusion and single-transformer LMMs (Bavishi et al., 2023; Team, 2024; Diao et al., 2024; Wang et al., 2024b), and bridge the performance gap between unified and compositional LMMs. As shown in Table 1, HaploVL achieves a significant performance improvement over LLaVA and EVE (Diao et al., 2024) on fine-grained perception benchmarks (Chen et al., 2024a; Tong et al., 2024). This demonstrates promising potential for developing multi-modal models with a single transformer efficiently.

Our contributions can be summarized as follows:

- We develop a new early-fusion LMM with a single transformer that acquires the necessary vision cues in the early stage and generates language responses conditioned on fused multi-modal embeddings.
- We design an efficient training recipe for the proposed

¹Estimated by Chameleon-30B (Team, 2024)'s GPU hours, A100 GPU's power, and Tesla Model 3's power consumption.

model, which leverages the prior knowledge from pre-trained models. This approach not only reduces the need for large-scale data and computational resources but also bridges the performance gap between the unified and compositional LMMs.

2. Related Work

Encoder-decoder large multi-modal models as exemplified by LLaVA (Liu et al., 2024c), employ a pre-trained vision encoder like CLIP (Radford et al., 2021) to extract visual embeddings and an MLP layer to align the visual embeddings with large language models (LLMs). Then, these models with the “Encoder-MLP-LLM” configuration are fine-tuned on tailored instruction data to obtain the capability of image understanding and reasoning. Numerous innovations have sought to improve the performance of this method by utilizing more powerful vision encoders (Zhai et al., 2023; Chen et al., 2024b), expanding the input size to any resolution (Liu et al., 2024a), and synthesizing high-quality data (Li et al., 2024a; Chen et al., 2023). At the same time, inspired by this straightforward architecture, numerous studies have replaced the vision encoder with a domain-specific encoder to develop a modality-specific multi-modal model (Chu et al., 2023; Qi et al., 2024). Plus, others (Lu et al., 2022a; Zhan et al., 2024) integrate multiple modality-specific encoders with the language model to enable it to accommodate more additional modalities. However, a significant limitation of this method is the lengthy visual sequences. To alleviate this issue, BLIP-2 (Li et al., 2023b) develops a Q-former to replace the long visual features with a fixed number of learnable queries. This “Encoder-Q-former-LLM” configuration has been replicated by many studies (Zhu et al., 2023a; Dai et al., 2023; Li et al., 2024c).

Single-transformer multi-modal models aim to discard the vision encoder and merely allow the language model to process text embeddings and vision embeddings that are not fully compressed. Fuyu (Bavishi et al., 2023) utilizes a linear projector to patchify the raw image wherein the obtained low-level vision patch embeddings are treated as contiguous tokens. Compared with models with the “Encoder-MLP-LLM” configuration, Fuyu directly fuses low-level vision embeddings with text embeddings instead of high-level vision embeddings (hidden states of the vision encoder). Besides, Chameleon (Team, 2024) employs a VQ codebook (Van Den Oord et al., 2017) to discretize the image to a set of discrete visual tokens, akin to the process of the text tokenizer. Thus, the vision and text embedding can be extracted from the same embedding layer and processed by a decoder-only transformer. Emu3 (Wang et al., 2024b) has extended this streamlined pipeline to generate high-quality images and videos. Since these methods are trained from scratch, they consume substantial computing resources and

necessitate significant amounts of data. To adapt an off-the-shelf decoder-only language model to a multi-modal model, EVE (Diao et al., 2024) introduces a meticulously designed patch embedding layer and training strategies. However, they still exhibit a significant performance gap compared to encoder-decoder multi-modal language models, despite utilizing 35 M images.

3. Method

Our HaploVL is a single-transformer multi-modal model. Like popular LMMs (Liu et al., 2024c; Dai et al., 2023; Bavishi et al., 2023), it maps visual and textual input to the same latent space and takes them as conditions for text generation in an auto-regressive manner. Unlike other LMMs that always rely on highly compressed vision embeddings from a fixed vision encoder, our HaploVL fused the visual and textual input in the early stage and extracts the necessary vision information based on the text input. Compared to previous early-fusion and single-transformer LMMs (Bavishi et al., 2023; Diao et al., 2024; Team, 2024), our HaploVL is more efficient in training, as it absorbs the prior knowledge that the model learned. In the subsequent section, we begin by presenting a detailed description of HaploVL’s architecture, followed by a receipt of the efficient training procedure.

3.1. Architecture

From a holistic perspective, as illustrated on the right side of Figure 3, our HaploVL adopts a multi-modal end-to-end transformer architecture. Most of its parameters are attributed to the transformer decoder that processes sequences, regardless of the modality. HaploVL can generate the language response X_a conditioned on the visual input X_v and textual input X_t in an auto-regressive manner. This generation process is as clear as that of language models (Radford et al., 2019; Dubey et al., 2024). Given a sequence of length L , HaploVL calculates the probability of X_a by:

$$p(X_a|X_v, X_t) = \prod_{i=1}^L p_\theta(x_i|X_v, X_t, X_{a,< i}), \quad (1)$$

where $X_{a,< i}$ denotes the answer tokens before the current prediction token x_i . θ is the parameter of components that model conditional probability. Thus, θ of our HaploVL is from the whole model, while θ of compositional LMMs using separated vision encoders (Liu et al., 2024c; Karamchetti et al., 2024) is the parameter of the LLM.

From a detailed perspective, as depicted in Figure 3 (b), HaploVL can be decomposed into three primary components: (1) multi-modal embedding layers, (2) a pre-decoder, and (3) a post-decoder. These bottom-up modules work together to facilitate efficient training and enhance visual understanding and reasoning performance, especially in the

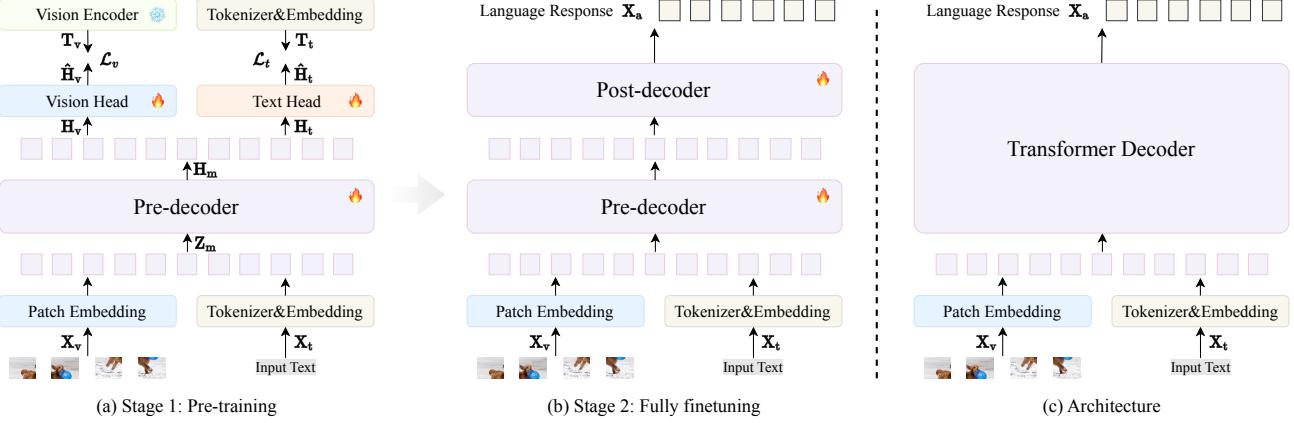


Figure 3: The diagram of HaploVL. It includes a transformer decoder made up of a pre-decoder and a post-decoder. During the pre-training stage (a), the pre-decoder is trained by distilling knowledge from the pre-trained vision encoder and the text embeddings of the LLM. Heads and teacher models are dropped after pre-training. In the full fine-tuning stage (b), the entire model is fine-tuned using visual instruction data.

fine-grained scene.

Multi-modal embedding layers. Regarding the input data, we use lightweight and modality-specific components with unshared parameters to map them into a shared latent space \mathbb{R}^d . Specifically, for the input RGB image X_v , we apply a simple patch embedding layer, *a single linear layer*, to compress local windows ($k \times k$) of pixels into a vision embedding Z_v within the shared latent space \mathbb{R}^d . This approach differs from existing compositional LMMs (Liu et al., 2024a; Karamcheti et al., 2024), which typically rely on one or more separate vision encoders to embed visual input. For the input text X_t , we leverage the pre-trained LLM’s embedding matrix \mathcal{W} to convert each text token into a vector within LLM’s space \mathbb{R}^l . These text vectors are then projected into text embeddings Z_t within the shared latent space \mathbb{R}^d by a text projector, also *a single linear layer*. The resulting vision and text embeddings, Z_v and Z_t , are combined to form a mixed multi-modality embedding sequence Z_m , which is fed into the subsequent transformer.

Pre-decoder. Upon the multi-modal sequence Z_m , the pre-decoder fuses it in the initial stage of HaploVL, extracting visual cues based on text embeddings. Then, it yields a multi-modal hidden state H_m . Each block of the pre-decoder consists of a multi-head self-attention layer and a 2-layer MLP with GELU (Hendrycks & Gimpel, 2016) nonlinearity in between. Its configuration, such as depth and width, mirrors that of the vision transformer, as it is required to inherit the prior vision knowledge of a pre-trained vision model. In practice, we default to leveraging CLIP-ViT-L (Radford et al., 2021) which has 24 blocks and an embedding dimension of 1024. Notably, although the pre-decoder inherits prior knowledge from a vision encoder, it differs from the vision encoder. For one thing, the pre-

decoder can process both visual and textual input, whereas the vision encoder only processes visual input. Furthermore, the text embeddings in the pre-decoder utilize a causal mask strategy, allowing the pre-decoder to predict the next token in an auto-regressive manner.

Post-decoder. Based on the multi-modal hidden state H_m , the post-decoder further processes it and outputs a language response. Each block of the post-decoder mirrors the Llama block (Dubey et al., 2024) as it needs to acquire prior textual knowledge the Llama model. Leveraging the inherited knowledge from extensive text data, the post-decoder can swiftly learn multi-modal knowledge and generate language responses based on multi-modal hidden states.

Masking strategy. HaploVL employs a mixed masking strategy within its self-attention layers. In a mixed multi-modal sequence, a causal mask is utilized for the textual part. This is consistent with GPT-like language models (Radford et al., 2019). For the visual part, a bidirectional mask is applied to embeddings from a single image, as correlations exist between image tokens regardless of their positional order. In addition, a causal mask is employed between multiple images, reflecting the temporal causal relationships in sequential data. This modeling approach aligns with prevalent vision models (Dosovitskiy et al., 2021).

3.2. Training

We use a two-stage training recipe for HaploVL, as illustrated in Figure 3. In the first stage, the pre-decoder is trained through feature distillation. This enables it to effectively process both visual and textual inputs simultaneously, laying the foundation for subsequent stages. In the second stage, the model is trained to follow visual instruction,

which equals LLaVA’s visual instruction tuning.

Stage 1: Pre-training. As mentioned before, the pre-decoder inherits prior vision knowledge from the pre-trained ViT and can fuse visual and textual inputs. This stage mainly endows the pre-decoder to support vision and text modality. As illustrated in Figure 3 (a), the knowledge distillation approach (Hinton, 2015) is employed to train the pre-decoder, prompting the model to learn new text knowledge and avoiding the model forgetting the inherited vision knowledge. Given the visual input X_v and textual input X_t , the output of the pre-decoder is the hidden state H_m , which can be decomposed into the visual hidden state H_v and textual hidden state H_t based on their respective token positions.

To preserve the image processing capabilities of the pre-decoder, we adopt the pre-trained CLIP vision encoder (Radford et al., 2021) as a teacher model to guide the expansion process. This approach enables the pre-decoder to retain its inherited knowledge, ensuring its image abilities are not compromised. This vision loss can be formulated as:

$$\mathcal{L}_v = 1 - \frac{1}{hw} \sum_{i=1}^{hw} \cos(\hat{H}_{v,i}; T_{v,i}), \quad (2)$$

where \hat{H}_v is the projected H_v by a vision head; T_v is the feature extracted from CLIP vision encoder; and hw represents the number of vision embedding after patch partition.

For the textual input, the pre-decoder performs a simple identity mapping. This training target enables the pre-decoder to leverage the strengths of the post-decoder in handling complex generation tasks, thus alleviating the challenging multi-modal learning. More importantly, language is semantic (He et al., 2022). When the text and image are jointly input into the pre-decoder in a mixed way, semantic text embeddings can autonomously acquire the necessary vision cues from raw vision embeddings. For the split textual hidden state H_t , we utilize a learnable text head to align it with the teacher embedding, resulting in \hat{H}_t . We employ two types of loss functions to encourage this distillation.

(a) The first type is feature loss, which is formulated as:

$$\mathcal{L}_{feat} = 1 + \frac{1}{S} \sum_{i=1}^S [\|\hat{H}_{t,i} - T_{t,i}\|_2 - \cos(\hat{H}_{t,i}; T_{t,i})]. \quad (3)$$

Here, S denotes the length of text tokens in the input sequence; and T_t represents the text embedding directly obtained from the embedding matrix \mathcal{W} using indices of text tokens, which also serves as the input of the pre-decoder. Equation (3) involves the L_2 distance to align the magnitude and the cosine loss function to align the direction. This is because the magnitude and direction of text embeddings are crucial for alignment with the post-decoder, which has been retained in a certain input mode when inheriting knowledge from a pre-trained LLM.

(b) The second type of loss function is a current token prediction loss which can be formulated as:

$$\mathcal{L}_{ctp} = -\frac{1}{S} \sum_{i=1}^S \sum_{c=1}^C y_{i,c} \log\left(\frac{e^{\frac{x_{i,c}}{\tau}}}{\sum_{j=1}^C e^{\frac{x_{i,j}}{\tau}}}\right). \quad (4)$$

In this loss function, C refers to the vocabulary size of the tokenizer; y_i is the one-hot label of the i -th token; and $y_{i,c}$ is the label of c -th word in the vocabulary. $x_i = \hat{H}_{t,i} \cdot \mathcal{W}^T$ is the logit of $\hat{H}_{t,i}$. A learnable temperature τ is used to adjust the distribution of the logits as CLIP (Radford et al., 2021). This is very effective in reducing the magnitude of the output text embeddings, as the cross-entropy loss function minimizes the total loss by enlarging the logit magnitude. The difference between Equation (4) and the next token prediction loss (Radford et al., 2019) lies in that the target in Equation (4) is derived from the current token instead of the next token.

So far, we have introduced two types of loss functions, and the total text loss function used in the modal expansion stage is the sum of them: $\mathcal{L}_t = \mathcal{L}_{feat} + \mathcal{L}_{ctp}$. We combine interleaved image-text data and pure text data to train the pre-decoder. After modal expansion, we keep the pre-decoder, while discarding the heads.

Stage 2: Fully fine-tuning. This training stage is mainly for multi-modal learning. As illustrated in Figure 3 (b), we fine-tune all components of HaploVL in this stage. The next token prediction loss (Radford et al., 2019) is still adopted to maximize the log-likelihood of Equation (1). After tuning, our HaploVL performs capabilities in following human visual instructions.

4. Experiment

In this section, we first outline the experimental setup including training settings and dataset. Then, we compare our HaploVL with leading methods on various benchmarks. Finally, an analysis of training procedures and some qualitative results are given at the end of this section.

4.1. Experiment Setup

Implementation details. In this study, we instantiate HaploVL by allowing a pre-decoder to receive images and texts at the same time. The pre-decoder inherited the vision knowledge of the CLIP-ViT-L (Radford et al., 2021). The post-decoder inherits the text knowledge from Vicuna-7B (Chiang et al., 2023) and Llama-3-8B (Dubey et al., 2024), resulting in HaploVL-7B and HaploVL-8B, respectively. During the pre-training stage, we optimize the post-decoder for 40 K steps with $1 \times e^{-4}$ learning rate, a batch size of 256, and 2 K warm-up steps. In terms of the data, all models are trained on 665 K plus 558 K multi-modal samples from LLaVA-1.5 (Liu et al., 2024a) if there is no other

Method		Base	LLM	SEED	POPE	AI2D	RWQA	MMMU	MMB	MMS	VQAv2	GQA	SQA	MMVP
<i>Compositional LMM</i>														
InstructBLIP (Dai et al., 2023)	Vicuna-7B	58.8	-	33.8	37.4	30.6	36.0	-	-	-	49.2	60.5	16.7	
LLaVA-1.5 (Liu et al., 2024a)	Vicuna-7B	66.1	85.9	54.8	54.8	35.3	64.3	30.3	78.5*	62.0*	66.8	21.3		
LLaVA-1.6 (Liu et al., 2024b)	Vicuna-7B	70.2	86.5	66.6*	57.8	35.8	67.4	-	81.8*	64.2*	70.1	37.3		
ShareGPT4V (Chen et al., 2023)	Vicuna-7B	-	-	58.0	54.9	37.2	68.8	33.0	80.6*	63.3*	68.4	-		
VILA (Lin et al., 2024)	Llama-2-7B	61.1	85.5	-	-	-	68.9	-	80.8*	63.3*	73.7	-		
LLaVA-OV (Li et al., 2024a)	Qwen2-7B	75.4	-	81.4*	66.3	48.8	80.8	61.7	-	-	96.0*	-		
<i>Single-Transformer LMM</i>														
Fuyu-8B (Bavishi et al., 2023)	Persimmon-8B	-	74.1	64.5	-	27.9	10.7	-	74.2	-	-	-		
Chameleon-30B (Team, 2024)	-	-	-	-	-	-	37.6	-	69.6	-	-	-		
EVE-7B (Diao et al., 2024)	Vicuna-7B	54.3	83.6	-	-	-	49.5	28.2	75.4*	60.8*	63.0	19.3		
Emu3-8B (Wang et al., 2024b)	-	68.2	85.2	70.0*	57.4	31.6	58.5	-	75.1*	60.3*	89.2*	-		
HaploVL-8B (ours)	Llama-3-8B	75.1	88.6	79.2*	61.4	37.4	73.6	57.2	81.0*	65.5*	95.3*	45.3		
HaploVL-8B-MI (ours)	Llama-3-8B	75.5	88.2	79.6*	62.0	41.7	75.0	57.6	80.7*	65.0*	94.4*	46.0		
HaploVL-7B-Pro (ours)	Qwen2.5-7B	75.0	88.7	80.6*	64.3	48.7	80.5	61.4	81.1*	64.6*	96.9*	50.1		

Table 2: Comparison on multi-modal benchmarks, including SEED (Li et al., 2023a), POPE (Li et al., 2023c), AI2D (Kembhavi et al., 2016), RWQA (x.ai, 2024), MMMU (Yue et al., 2024), MMB (Liu et al., 2024d), MMStar (Chen et al., 2024a), VQAv2 (Goyal et al., 2017), GQA (Hudson & Manning, 2019), and SQA (Lu et al., 2022b). ‘*’ denotes images of related training datasets are observed during training. HaploVL-8B-MI is the model further fine-tuned on multi-image datasets.

statement. During the fully fine-tuning stage, the learning rate is set to $2 \times e^{-5}$ and batch size to 128. Regarding the data, our best model is optimized on the 4 M visual instruction data for 1 epoch (30K steps). For HaploVL-7B, we align it with LLaVA (Liu et al., 2024c). Thus, we first tune the connector between the pre-decoder and post-decoder using 558 K caption data and then fully tune the model using 665 K instruction data. For HaploVL-8B with the ability to input any resolution, we first tune the whole model using 1.2 M caption data (Chen et al., 2023) and then tune the model using 4 M instruction data (Li et al., 2024a). For the models that support the multi-image and video input, we continue training the single-image model using the mix of interleaved data and single-image data. For the ablation experiments, the models are optimized on the 0.6 M visual instruction data for 5 K steps. All models are optimized using the AdamW (Loshchilov & Hutter, 2019) optimizer and cosine scheduler on 32 GPUs with 64GB per-device memory. More details are recorded in the Appendix.

Dataset. The data is mainly from LLaVA (Liu et al., 2024a; Li et al., 2024a), MMC4 (Zhu et al., 2023b), dolphin (Computations, 2023), CC3M (Changpinyo et al., 2021), and COCO (Lin et al., 2014). Moreover, HaploVL is evaluated on widely adopted image-based benchmarks including GQA (Hudson & Manning, 2019), VQAv2 (Goyal et al., 2017), ScienceQA-IMG (SQA) (Lu et al., 2022b), AI2D (Kembhavi et al., 2016), MMBench-EN-dev (MMB) (Liu et al., 2024d), MMMU (Yue et al., 2024), RealWorldQA (x.ai, 2024), MMStar (MMS) (Chen et al., 2024a), POPE (Li et al., 2023c), SEED-Bench-IMG (SEED) (Li et al., 2023a), and MMVP (Tong et al., 2024).

Among these benchmarks, MMVP mainly focuses on fine-grained perception. More details are shown in the Appendix.

4.2. Main Results

We compare our model with existing multi-modal models, including both separate models and unified models with a single transformer, in Table 2. Notably, our model achieves superior performance compared to other unified models. Specifically, we outperform Emu3 (Wang et al., 2024b) by 15.1% on the MMBench (Liu et al., 2024d) and by 5.5% on the MMMU (Yue et al., 2024). Additionally, our model significantly surpasses EVE (Diao et al., 2024), a model that uses pre-trained weights, with a lead of 24.1% on MM-Bench (Liu et al., 2024d) and 20.8% on SEED-Bench (Li et al., 2023a). These results demonstrate the promising potential of our model in multi-modal capabilities. Furthermore, we also compare our model with separate models and find that our model has a significant advantage over previous separate models (Dai et al., 2023; Liu et al., 2024a; Chen et al., 2023; Lin et al., 2024). However, our performance still falls short of the state-of-the-art separate open-source models LLaVA-OneVision (Li et al., 2024a). We attribute this to the input resolution and context length. LLaVA-OneVision (Li et al., 2024a) uses 7290 tokens to represent an input image, while our model only uses up to 2304 tokens. Due to computational resource constraints, we can only set the context length to 6144, which affects the model’s effectiveness to some extent. Nevertheless, the performance of HaploVL-7B-Pro is nearly comparable to that of LLaVA-OneVision. Plus, we achieve a simple and efficient baseline for one multi-modal transformer, which outperforms other



Figure 4: Qualitative comparison of LLaVA-1.5-7B (Liu et al., 2024a) and our HaploVL-7B. The first line involves cases about fine-grained perception. The second line includes cases of logical reasoning that depend on fine-grained perception.

native LMMs using fewer resources. We expect to further improve the performance of such models based on this foundation.

4.3. Ablation study

Ablation for different LLMs, resolution, and visual instruction data. As shown in Table 4, we achieve enhanced performance by upgrading the language model, input resolution, and instruction data. Specifically, employing a more advanced language model (Llama-3 (Dubey et al., 2024)) yields an average performance gain of 2.5%. This highlights that multi-modal understanding capabilities are correlated with the capabilities of the language model. Increasing the resolution from 336×336 to 672×672 results in an average performance improvement of 3.3% using the same 665K dataset, especially showing a notable 3.7% gain on POPE (Li et al., 2023c). This underscores the importance of enabling the LMM to perceive finer-grained visual details. When expanding the visual instruction data at the 672×672 resolution, the average performance improves by 6.6% since LMM’s knowledge is enriched. These gains are particularly pronounced on benchmarks such as MMStar (Chen et al., 2024a) and MMVP (Tong et al., 2024), suggesting that fine-grained perception ability can be enhanced after expanding LMM’s vision knowledge. However, a slight performance decline is observed on GQA (Hudson & Manning, 2019).

Base LLM	Res.	VID	Avg	GQA	POPE	MMS	MMVP
Vicnua-7B	336	665K	51.8	62.5	85.4	34.5	24.7
Llama-3-8B	336	665K	54.3	63.1	84.8	39.4	30.0
Llama-3-8B	672	665K	57.6	65.7	88.2	42.0	34.3
Llama-3-8B	672	4M	64.2	65.5	88.6	57.2	45.3

Table 4: Ablation for different LLMs, resolution (Res.), and visual instruction data (VID).

Method	ST	MMVP	MMS						
			Avg	CP	FP	IR	LR	ST	
LLaVA-1.5-7B	✗	21.3	30.3	58.8	24.0	38.8	24.0	13.6	22.8
EVE-7B	✓	19.3	28.2	40.6	24.2	32.7	27.5	20.4	23.8
HaploVL-7B	✓	24.7	34.5	63.8	28.9	38.5	33.6	17.6	24.6

Table 5: In-depth comparison with LLaVA-1.5-7B (Liu et al., 2024a) and EVE (Diao et al., 2024) on MMVP (Tong et al., 2024) and MMStar (Chen et al., 2024a). ‘ST’ denotes whether the model belongs to the single-transformer LMM. CP: coarse perception, FP: fine-grained perception, IR: instance reasoning, LR: logical reasoning, ST: science and technology, and MA: mathematics.

This discrepancy may stem from differences in the distribution of the 4M instruction data compared to the GQA dataset.

Compared with the compositional LMM using the same LLM and training data.

We compare the performance of our method with that of LLaVA-1.5-7B (Liu et al., 2024a), a typical compositional LMM, using the same LLM (Vicuna-7B) and instruction data (665K) from (Liu et al., 2024a). Since the LLM and instruction data primarily affect performance on different benchmarks, we restrict the data to ensure a fair comparison between our method and LLaVA-1.5-7B. This allows us to verify whether the LMM using one single transformer has advantages over separate models. As shown in Table 5, on the MMVP (Tong et al., 2024) benchmark, our model obtains 3.4% and 5.4% gains than LLaVA-1.5-7B and EVE-7B (Diao et al., 2024), respectively; and, on the MMStar (Chen et al., 2024a) benchmark, our model outperforms LLaVA-1.5-7B and EVE-7B by 4.2% and 6.3%, respectively. We further analyze the detailed scores on MMStar (Chen et al., 2024a), including coarse perception (CP), fine-grained perception (FP), instance reasoning (IR), logical reasoning (LR), science and technology (ST), and mathematics (MA). Notably, our HaploVL-7B model exhibits a 4.9% improvement in fine-grained perception and a 9.6% improvement in logical reasoning over LLaVA-1.5-7B. This suggests that fusing raw image and text embeddings in a single transformer is beneficial for fine-grained perception and subsequently enhances image-based logical reasoning. In

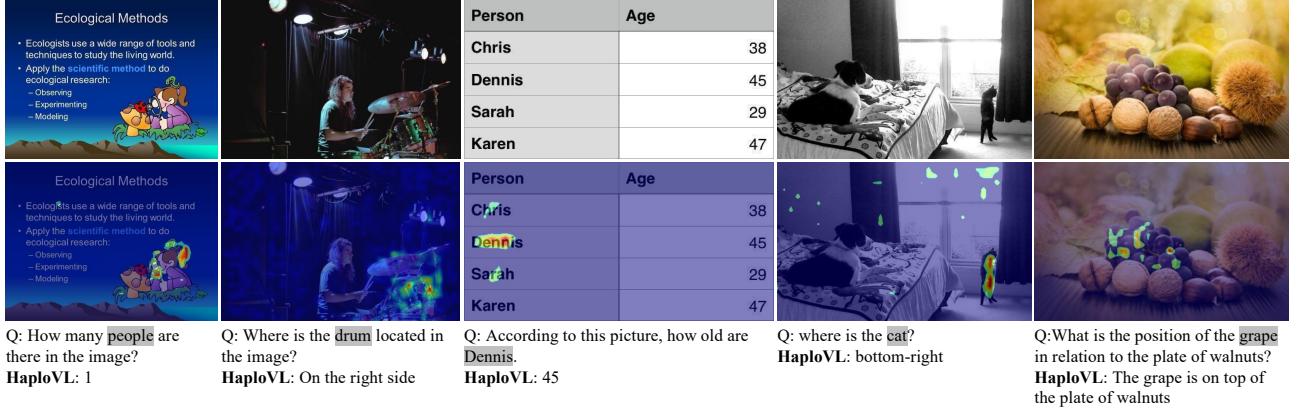


Figure 5: Visualization for the early fusion mechanism of our single transformer. The second row illustrates the attention map of the gray words concerning the vision embeddings after the pre-decoder.

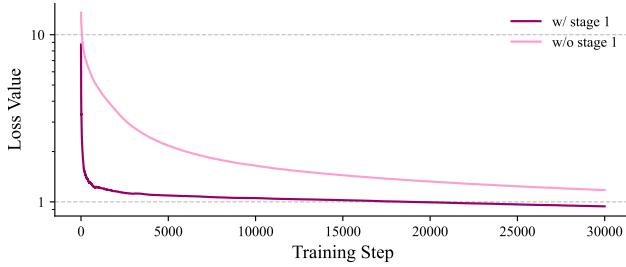


Figure 6: Loss curve of the full fine-tuning stage. We use ema to smooth the actual loss value for better visualization.

Base LLM	Res.	S-1	Avg	GQA	POPE	MMB	MMS	MMVP
Llama-3-8B	336	X	55.5	56.6	84.0	64.3	46.0	26.7
Llama-3-8B	336	✓	60.5	60.7	84.5	72.9	51.3	33.3

Table 3: Ablation for the stage one (S-1). Both models are trained using 4M instruction tuning data. The model with the pre-training stage shows faster convergence and superior performance.

contrast, separate models using high-level semantic embeddings from CLIP-ViT encoder (Radford et al., 2021) directly may obscure fine-grained image information, thereby impairing the model to perform tasks that rely on image details. This is consistent with previous study (Tong et al., 2024).

To further illustrate the differences in fine-grained perception and logical reasoning, we provide qualitative results in Figure 4. The first row shows cases of fine-grained perception, where LLaVA-1.5-7B fails to recognize the color of small objects and the number of objects outside the image center. For instance, LLaVA-1.5-7B incorrectly identified the color of the NBA player’s socks. The second row shows examples of logical reasoning, where the lack of fine-grained perception ability leads LLaVA-1.5-7B to failure in tasks that rely on it, such as edge object perception and reasoning, and highlighting regions in images. In contrast, our

HaploVL, fusing raw image embeddings after the patch embedding layer, enhances its ability to perceive fine-grained image information. Therefore, it shows better performance on tasks relying on the capability of fine-grained perception.

Is it possible to use the next token prediction loss directly? To validate the effectiveness of the modal expansion, we directly optimized the model using next-token prediction loss without the first stage. As shown in Figure 6, the model converged slowly when optimized directly, as it had to perform both modality fusion and text generation simultaneously. In contrast, the model with the modal expansion stage converged significantly faster. Furthermore, as shown in Table 3, we found that the model without the modal expansion stage exhibits a 4.3% performance drop.

4.4. Visualization study

In order to investigate whether text embeddings can dynamically capture visual clues, we visualize the attention map between text embeddings and visual embeddings after the pre-decoder, as illustrated in Figure Figure 5. It is observable that the text exhibits an automatic response to regions of higher relevance. For example, it demonstrates responsiveness to objects located at the image edges as well as to textual elements within the image. These findings suggest that the early fusion mechanism of our single-transformer model is effective for fine-grained perception tasks, thereby corroborating the results presented in Table 5.

5. Conclusion

This work presents a simple baseline for the multi-modal model with a single transformer architecture and a corresponding efficient training approach. By fusing raw vision and text embeddings in the early stages, our model enhances its fine-grained perception capabilities, enabling it to capture subtle relationships in the image better. Furthermore,

our model builds upon prior knowledge from pre-trained single-modal models. This allows it to achieve superior performance with relatively few training tokens and bridge the performance gap between single-transformer multi-modal models and compositional models. Consequently, it demonstrates the potential of single-transformer architectures for multi-modal tasks. We expect our work can provide a foundation for future research on single-transformer multi-modal models, offering new insights and opportunities for advancement in this field.

Impact Statements

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

- Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., et al. Gpt-4 technical report. *arXiv:2303.08774*, 2023.
- Bavishi, R., Elsen, E., Hawthorne, C., Nye, M., Odena, A., Soman, A., and Taşırılar, S. Introducing our multimodal models, 2023. URL <https://www.adept.ai/blog/fuyu-8b>.
- Changpinyo, S., Sharma, P., Ding, N., and Soricut, R. Conceptual 12m: Pushing web-scale image-text pre-training to recognize long-tail visual concepts. In *CVPR*, 2021.
- Chen, L., Li, J., Dong, X., Zhang, P., He, C., Wang, J., Zhao, F., and Lin, D. Sharegpt4v: Improving large multi-modal models with better captions. *arXiv:2311.12793*, 2023.
- Chen, L., Li, J., Dong, X., Zhang, P., Zang, Y., Chen, Z., Duan, H., Wang, J., Qiao, Y., Lin, D., et al. Are we on the right way for evaluating large vision-language models? *arXiv:2403.20330*, 2024a.
- Chen, Z., Wu, J., Wang, W., Su, W., Chen, G., Xing, S., Zhong, M., Zhang, Q., Zhu, X., Lu, L., et al. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *CVPR*, 2024b.
- Chiang, W.-L., Li, Z., Lin, Z., Sheng, Y., Wu, Z., Zhang, H., Zheng, L., Zhuang, S., Zhuang, Y., Gonzalez, J. E., Stoica, I., and Xing, E. P. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality, March 2023. URL <https://lmsys.org/blog/2023-03-30-vicuna/>.
- Chu, Y., Xu, J., Zhou, X., Yang, Q., Zhang, S., Yan, Z., Zhou, C., and Zhou, J. Qwen-audio: Advancing universal audio understanding via unified large-scale audio-language models. *arXiv:2311.07919*, 2023.
- Computations, C. <https://huggingface.co/datasets/cognitivecomputations/dolphin>, 2023.
- Dai, W., Li, J., Li, D., Tiong, A. M. H., Zhao, J., Wang, W., Li, B., Fung, P., and Hoi, S. C. H. Instructblip: Towards general-purpose vision-language models with instruction tuning. In *NeurIPS*, 2023.
- Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *CVPR*, pp. 248–255. Ieee, 2009.
- Diao, H., Cui, Y., Li, X., Wang, Y., Lu, H., and Wang, X. Unveiling encoder-free vision-language models. *arXiv:2406.11832*, 2024.
- Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021.
- Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Yang, A., Fan, A., et al. The llama 3 herd of models. *arXiv:2407.21783*, 2024.
- Goyal, Y., Khot, T., Summers-Stay, D., Batra, D., and Parikh, D. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *CVPR*, 2017.
- He, K., Chen, X., Xie, S., Li, Y., Dollár, P., and Girshick, R. B. Masked autoencoders are scalable vision learners. In *CVPR*, 2022.
- Hendrycks, D. and Gimpel, K. Gaussian error linear units (gelus). *arXiv:1606.08415*, 2016.
- Hinton, G. Distilling the knowledge in a neural network. *arXiv:1503.02531*, 2015.
- Hudson, D. A. and Manning, C. D. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *CVPR*, 2019.
- Karamcheti, S., Nair, S., Balakrishna, A., Liang, P., Kollar, T., and Sadigh, D. Prismatic vlms: Investigating the design space of visually-conditioned language models. *arXiv:2402.07865*, 2024.
- Kembhavi, A., Salvato, M., Kolve, E., Seo, M., Hajishirzi, H., and Farhadi, A. A diagram is worth a dozen images. In *ECCV*, 2016.

- Li, B., Wang, R., Wang, G., Ge, Y., Ge, Y., and Shan, Y. Seed-bench: Benchmarking multimodal llms with generative comprehension. *arXiv:2307.16125*, 2023a.
- Li, B., Zhang, Y., Guo, D., Zhang, R., Li, F., Zhang, H., Zhang, K., Li, Y., Liu, Z., and Li, C. Llava-onevision: Easy visual task transfer. *arXiv:2408.03326*, 2024a.
- Li, F., Zhang, R., Zhang, H., Zhang, Y., Li, B., Li, W., Ma, Z., and Li, C. Llava-next-interleave: Tackling multi-image, video, and 3d in large multimodal models. *arXiv preprint arXiv:2407.07895*, 2024b.
- Li, J., Li, D., Savarese, S., and Hoi, S. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *ICML*, 2023b.
- Li, Y., Du, Y., Zhou, K., Wang, J., Zhao, W. X., and Wen, J.-R. Evaluating object hallucination in large vision-language models. *arXiv:2305.10355*, 2023c.
- Li, Y., Zhang, Y., Wang, C., Zhong, Z., Chen, Y., Chu, R., Liu, S., and Jia, J. Mini-gemini: Mining the potential of multi-modality vision language models. *arXiv:2403.18814*, 2024c.
- Liao, H., Han, H., Yang, K., Du, T., Yang, R., Xu, Q., Xu, Z., Liu, J., Lu, J., and Li, X. Baton: Aligning text-to-audio model using human preference feedback. In *IJCAI*, 2024.
- Lin, J., Yin, H., Ping, W., Molchanov, P., Shoeybi, M., and Han, S. Vila: On pre-training for visual language models. In *CVPR*, 2024.
- Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. Microsoft coco: Common objects in context. In *ECCV*, 2014.
- Liu, H., Li, C., Li, Y., and Lee, Y. J. Improved baselines with visual instruction tuning. In *CVPR*, 2024a.
- Liu, H., Li, C., Li, Y., Li, B., Zhang, Y., Shen, S., and Lee, Y. J. Llava-next: Improved reasoning, ocr, and world knowledge, 2024b. URL <https://llava-vl.github.io/blog/2024-01-30-llava-next/>.
- Liu, H., Li, C., Wu, Q., and Lee, Y. J. Visual instruction tuning. In *NeurIPS*, 2024c.
- Liu, Y., Duan, H., Zhang, Y., Li, B., Zhang, S., Zhao, W., Yuan, Y., Wang, J., He, C., Liu, Z., et al. Mmbench: Is your multi-modal model an all-around player? In *ECCV*, 2024d.
- Loshchilov, I. and Hutter, F. SGDR: stochastic gradient descent with warm restarts. In *ICLR*. OpenReview.net, 2017.
- Loshchilov, I. and Hutter, F. Decoupled weight decay regularization. In *ICLR*, 2019.
- Lu, J., Clark, C., Zellers, R., Mottaghi, R., and Kembhavi, A. Unified-io: A unified model for vision, language, and multi-modal tasks. In *ICLR*, 2022a.
- Lu, P., Mishra, S., Xia, T., Qiu, L., Chang, K.-W., Zhu, S.-C., Tafjord, O., Clark, P., and Kalyan, A. Learn to explain: Multimodal reasoning via thought chains for science question answering. In *NeurIPS*, 2022b.
- OpenAI. Chatgpt. <https://openai.com/blog/chatgpt/>, 2023.
- Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C. L., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., Schulman, J., Hilton, J., Kelton, F., Miller, L., Simens, M., Askell, A., Welinder, P., Christiano, P. F., Leike, J., and Lowe, R. Training language models to follow instructions with human feedback. In *NeurIPS*, 2022.
- Podell, D., English, Z., Lacey, K., Blattmann, A., Dockhorn, T., Müller, J., Penna, J., and Rombach, R. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv:2307.01952*, 2023.
- Qi, Z., Fang, Y., Sun, Z., Wu, X., Wu, T., Wang, J., Lin, D., and Zhao, H. Gpt4point: A unified framework for point-language understanding and generation. In *CVPR*, 2024.
- Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. Language models are unsupervised multitask learners. *OpenAI blog*, 2019.
- Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021.
- Rafailov, R., Sharma, A., Mitchell, E., Manning, C. D., Ermon, S., and Finn, C. Direct preference optimization: Your language model is secretly a reward model. *NeurIPS*, 2024.
- ShareGPT. <https://sharegpt.com/>, 2023.
- Su, J., Ahmed, M. H. M., Lu, Y., Pan, S., Bo, W., and Liu, Y. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 2024.
- Taori, R., Gulrajani, I., Zhang, T., Dubois, Y., Li, X., Guestrin, C., Liang, P., and Hashimoto, T. B. Stanford alpaca: An instruction-following llama model. https://github.com/tatsu-lab/stanford_alpaca, 2023.

- Team, C. Chameleon: Mixed-modal early-fusion foundation models. *arXiv:2405.09818*, 2024.
- Team, G., Georgiev, P., Lei, V. I., Burnell, R., Bai, L., Gulati, A., Tanzer, G., Vincent, D., Pan, Z., Wang, S., et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv:2403.05530*, 2024.
- Tong, S., Liu, Z., Zhai, Y., Ma, Y., LeCun, Y., and Xie, S. Eyes wide shut? exploring the visual shortcomings of multimodal llms. In *CVPR*, 2024.
- Van Den Oord, A., Vinyals, O., et al. Neural discrete representation learning. In *NeurIPS*, 2017.
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. Attention is all you need. In *NIPS*, 2017.
- Wang, P., Bai, S., Tan, S., Wang, S., Fan, Z., Bai, J., Chen, K., Liu, X., Wang, J., Ge, W., et al. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024a.
- Wang, X., Zhang, X., Luo, Z., Sun, Q., Cui, Y., Wang, J., Zhang, F., Wang, Y., Li, Z., Yu, Q., et al. Emu3: Next-token prediction is all you need. *arXiv:2409.18869*, 2024b.
- x.ai. Grok-1.5 vision preview, 2024. URL <https://x.ai/blog/grok-1.5v>.
- Yang, A., Yang, B., Hui, B., Zheng, B., Yu, B., Zhou, C., Li, C., Li, C., Liu, D., Huang, F., et al. Qwen2 technical report. *arXiv:2407.10671*, 2024a.
- Yang, R., Ma, H., Wu, J., Tang, Y., Xiao, X., Zheng, M., and Li, X. Scalablevit: Rethinking the context-oriented generalization of vision transformer. In *European Conference on Computer Vision*, 2022.
- Yang, R., Song, L., Li, Y., Zhao, S., Ge, Y., Li, X., and Shan, Y. Gpt4tools: Teaching large language model to use tools via self-instruction. *NeurIPS*, 2024b.
- Yue, X., Ni, Y., Zhang, K., Zheng, T., Liu, R., Zhang, G., Stevens, S., Jiang, D., Ren, W., Sun, Y., et al. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. In *CVPR*, 2024.
- Zhai, X., Mustafa, B., Kolesnikov, A., and Beyer, L. Sigmoid loss for language image pre-training. In *ICCV*, 2023.
- Zhan, J., Dai, J., Ye, J., Zhou, Y., Zhang, D., Liu, Z., Zhang, X., Yuan, R., Zhang, G., Li, L., et al. Anygpt: Unified multimodal llm with discrete sequence modeling. *arXiv:2402.12226*, 2024.
- Zhou, H., Tang, L., Yang, R., Qin, G., Zhang, Y., Hu, R., and Li, X. Uniqqa: Unified vision-language pre-training for image quality and aesthetic assessment. *arXiv preprint arXiv:2406.01069*, 2024.
- Zhu, D., Chen, J., Shen, X., Li, X., and Elhoseiny, M. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv:2304.10592*, 2023a.
- Zhu, W., Hessel, J., Awadalla, A., Gadre, S. Y., Dodge, J., Fang, A., Yu, Y., Schmidt, L., Wang, W. Y., and Choi, Y. Multimodal C4: An open, billion-scale corpus of images interleaved with text. *arXiv:2304.06939*, 2023b.

In the appendix, we first provide more details in Appendix A and more results in Appendix B. Secondly, implementation details are presented in Appendix C, which involves the proposed current token prediction loss, positional embedding, the dataset used in each stage, and training settings. Then, more qualitative results are showcased in Appendix D. Finally, we discuss the future work in Appendix E.

A. Model Details

Current token prediction loss. The PyTorch-like pseudo code for the proposed current token prediction loss is presented in Algorithm 1. The target for a text embedding is derived from the current token index instead of the next token index, which is a key distinction from the next token prediction loss (Radford et al., 2019).

Algorithm 1 Current Token Prediction Loss

```

def loss(hidden_state, target_ids, embed_tokens):
    """
    hidden_state: [N, C]
    target_ids: [N]
    embed_tokens: [K, C]
    # N: the sequence length
    # K: the vocabulary size
    # C: the dimension of hidden_state
    """
    # get logits
    logits = hidden_state @ embed_tokens.transpose(-2, -1) # [N, K]
    # scale logits
    logits = logit_scale * logits
    # calculate loss
    loss = F.cross_entropy(logits, target_ids)
    return loss

```

embed_tokens is frozen.

The positional embedding. Positional embedding plays a crucial role in models (Dosovitskiy et al., 2021; Yang et al., 2022; Zhou et al., 2024) based on the attention mechanism (Vaswani et al., 2017), as it enables the self-attention module to capture the spatial relationship between embeddings. Our pre-decoder adopts a similar architecture to CLIP-ViT-L (Radford et al., 2021) but with the extra capability of accepting both image and text inputs. Therefore, pre-decoder must consider text position information in addition to image position information. To address this, we retain the learnable position embedding for image embeddings in the patch embedding layer and incorporate Rotary Position Embedding (RoPE) (Su et al., 2024) in each self-attention layer to inject positional information into the multi-modal embeddings.

B. More Results

Data in the pre-training stage. The pre-training stage is a critical step in our training receipt since it enables the pre-decoder to acquire new text knowledge while retaining its inherited vision knowledge. The data composition in this stage can impact the model in processing multi-modal inputs, as it influences the model to learn effective representations of text and images. Therefore, we conducted an exploratory analysis of the training data used in this stage. Specifically, we omitted the alignment stage and proceeded directly to instruction tuning after completing the pre-training stage. Initially, we employed Vicuna-7B (Chiang et al., 2023) as the language model and CLIP-ViT-L-14 (Radford et al., 2021) with an input resolution of 224 as the vision teacher. We directly mix 665 K instruction data and 558 K pre-train data of LLaVA-1.5, termed ‘mix-v1’. When training with this data, the average result was 59.0%. In this setup, the image position was fixed in the sequence, which may lead the model to learn shortcuts related to the position. This can hinder multi-modal fusion since the model may not learn to effectively integrate image and text embeddings. Therefore, we created an interleaved sequence by randomly combining 665 K instruction data and 558 K pre-train data, denoted as ‘mix-v2’, where each sequence may have multiple images. This data resulted in an average performance of 59.2% across multiple benchmarks. To further investigate the effectiveness of interleaved data in improving the performance on subsequent understanding tasks, we collected 730 K samples from MMC4 (Zhu et al., 2023b) for training. Our findings indicate that using interleaved data for modal expansion improves the performance on downstream tasks, particularly on ScienceQA (Lu et al., 2022b).

Next, we expanded the input resolution and used CLIP-ViT-L-14 (Radford et al., 2021) with an input resolution of 336 as the vision teacher for pre-training. When training with the ‘mix-v2’ data, the increased resolution resulted in a 1.5% improvement in average performance, with notable gains on GQA (Hudson & Manning, 2019), MMBench (Liu et al., 2024d),

LLM	Res.	Data-S1	Avg	GQA	SQA	POPE	MMB	MMS
Vicuna-7B	224	mix-v1	59.0	59.0	61.7	84.6	56.9	32.8
Vicuna-7B	224	mix-v2	59.2	60.7	63.7	83.6	55.2	32.8
Vicuna-7B	224	mmc4	59.4	59.1	65.5	82.9	57.3	32.5
Vicuna-7B	336	mix-v2	60.7	61.8	63.8	84.6	59.5	33.6
Vicuna-7B	336	mmc4	61.1	61.9	67.0	84.4	58.3	33.7
Vicuna-7B	336	mix-v3	61.2	61.2	67.5	84.8	59.0	33.6
Vicuna-7B	336	mix-v4	61.3	62.0	67.3	84.7	59.5	33.3
Llama-3-8B	336	mix-v4	64.2	63.1	70.5	84.8	63.0	39.4

Table 6: Ablation for data used in the first stage (Data-S1). The quantity of instruction data used in the second stage is 665 K. ‘Res.’ denotes the resolution of the vision teacher.

and MMStar (Chen et al., 2024a). At this resolution, we found that replacing MMC4 resulted in a similar improvement to that observed at the 224 resolution. Furthermore, we combined 730 K data from MMC4, 665 K instruction data, and 558 K pre-train data to create ‘mix-v3’. This mixture of data only brings a marginal improvement. Because our goal is to teach the model new text knowledge, we deemed it necessary to incorporate pure text data into the training process. Therefore, we mixed 665 K instruction data, 558 K pre-train data, and 600 K pure text data (Computations, 2023; Taori et al., 2023; ShareGPT, 2023) to create ‘mix-v4’. Training with this data resulted in a slight performance improvement. Nevertheless, we confirm the importance of using pure text data to teach the model new knowledge. Building on this data, we replaced Vicuna-7B with Llama-3-8B (Dubey et al., 2024) as the LLM, which brings a significant improvement in performance from 61.3% to 64.2%. This suggests that the ability conditioned on multi-modal sequences for effective reasoning is crucial in multi-modal understanding and reasoning.

Zero-shot accuracy on ImageNet. In the pre-training stage, we leverage the vision encoder of CLIP-ViT-L (Radford et al., 2021) as a teacher model to teach the visual knowledge for pre-decoder. To evaluate the effectiveness of this stage, we assess the zero-shot accuracy on ImageNet (Deng et al., 2009) in Table 7. Specifically, when optimized directly using the next token prediction loss, pre-decoder fails to retain image classification capabilities as shown in the first line. In contrast, with the first stage, pre-decoder exhibits a minimal performance drop compared to its teacher model. This outcome confirms the efficacy of the first stage in our training receipt.

LLM	Res.	w/ S1	Data-S1	Avg	IN1K	IN1K-Teacher
Llama-3-8B	336	✗	-	-	0.1	76.6
Vicuna-7B	224	✓	mix-v2	59.2	71.7	75.5
Vicuna-7B	224	✓	mmc4	59.4	71.6	75.5
Vicuna-7B	336	✓	mix-v2	60.7	72.3	76.6
Vicuna-7B	336	✓	mmc4	61.1	73.6	76.6
Vicuna-7B	336	✓	mix-v3	61.2	73.6	76.6
Vicuna-7B	336	✓	mix-v4	61.3	71.9	76.6
Llama-3-8B	336	✓	mix-v4	64.2	72.4	76.6

Table 7: Zero-shot accuracy on ImageNet-1K (IN1K) (Deng et al., 2009) after the pre-training stage. ‘w/ S1’ denotes the model tuned by the pre-training stage. ‘Avg’ refers to the average results on GQA (Hudson & Manning, 2019), ScienceQA-IMG (SQA) (Lu et al., 2022b), POPE (Li et al., 2023c), MMBench (MMB) (Liu et al., 2024d), and MMStar (Chen et al., 2024a) benchmark. Vision teacher is the CLIP-ViT-L (Radford et al., 2021), and ‘IN1K-Teacher’ denotes its zero-shot accuracy on ImageNet-1K. The first line is the results of the model optimized directly by the next-token prediction loss.

C. Implementation Details

C.1. Dataset

Pre-training data. We illustrate the construction of mix-v4 in Table 3 of the main text. The main samples include 665 K visual instruction data (Liu et al., 2024c) and 665 K textual instruction data from dophin (471 K) (Computations, 2023),

Alpaca (51 K) (Taori et al., 2023), and ShareGPT (143 K) (ShareGPT, 2023). The auxiliary samples involve 558 K image caption data (Liu et al., 2024c) and 1 M pure text data (Computations, 2023). The auxiliary samples are randomly combined with the main samples during training. Assume a main sample is S_m and a auxiliary sample is S_a , the combined samples can be one of $\langle S_a, S_m \rangle$ and $\langle S_m, S_a \rangle$. In this way, we obtain interleaved samples.

Single-Image instruction data. 665 K instruction data is from LLaVA-1.5 (Liu et al., 2024a). 4 M instruction data is partly from LLaVA-OneVision (Li et al., 2024a), where we filter some error samples.

Multi-Image instruction data. We use the interleaved data collected by (Li et al., 2024b) to endow the model (HaploVL-8B-MI) with the ability to process multiple images.

C.2. Experiment Settings

We summarize the training settings of each stage in Table 8 and Table 9. To maintain the training stability of the pre-training stage, we reduce the value of β_2 as Siglip (Zhai et al., 2023) rather than change the model structure as Chameleon (Team, 2024). For HaploVL-7B, we train it as LLaVA (Liu et al., 2024c). After pre-training, we first tune the connector between the pre-decoder and post-decoder using 558 K caption data (Liu et al., 2024a) and then fully tune the model using 665 K instruction data (Liu et al., 2024a). For HaploVL-8B with the ability to input any resolution, we first tune the whole model using 1.2 M caption data (Chen et al., 2023) because the prior knowledge of the pre-decoder doesn't have any-resolution vision knowledge. Then, we tune the model using 4 M instruction data. For HaploVL-8B-MI that supports the multi-image and video input, we continue training the single-image model (HaploVL-8B) using the mix of interleaved data and single-image data.

config	value
optimizer	AdamW (Loshchilov & Hutter, 2019)
base learning rate	1.0×10^{-4}
weight decay	1.0×10^{-4}
optimizer momentum	$\beta_1 = 0.9, \beta_2 = 0.95$
batch size	256
learning rate schedule	cosine decay (Loshchilov & Hutter, 2017)

Table 8: The first stage setting.

config	value
optimizer	AdamW (Loshchilov & Hutter, 2019)
base learning rate	2.0×10^{-5}
weight decay	0.0
optimizer momentum	$\beta_1 = 0.9, \beta_2 = 0.98$
batch size	128
learning rate schedule	cosine decay (Loshchilov & Hutter, 2017)

Table 9: The second stage setting.

D. Qualitative Results

We provide additional examples that demonstrate the capabilities of HaploVL in various tasks, including visual fine-grained perception, logistic reasoning, and image captioning. As shown in Table 10, HaploVL excels in fine-grained visual perception, including recognizing subtle color differences, object locations, and object quantities. Furthermore, HaploVL demonstrates correct logistic reasoning based on visual information, as illustrated in Table 11. This is attributed to its exceptional fine-grained perception ability, which is facilitated by early fusion. In terms of image captioning, HaploVL generally produces accurate descriptions, as shown in Table 13. Notably, HaploVL-8B pays closer attention to image details compared to HaploVL-7B, due to the higher quality and quantity of its instruction data. Interestingly, models trained on 665 K instructions from LLaVA (Liu et al., 2024a) exhibit a similar response pattern, suggesting that the inherent patterns in the data can influence the behaviors of the model.

E. Future Work

In this paper, we present a simple yet effective baseline for multi-modal models using a single transformer. By processing raw image embeddings instead of high-level embeddings, our model avoids the biases inherent in separate encoders. We anticipate that future work can further improve the performance of our model by leveraging more data, setting longer context lengths, and adopting dynamic resolution as representative studies (Wang et al., 2024a). In addition, future work can employ such unified multi-modal models to act as agents (Yang et al., 2024b) for tool use. Furthermore, our qualitative results (Table 10 and Table 11) reveal that models trained on the same 665K instruction data exhibit similar response templates. This suggests that the patterns present in the data can impact the model's behavior. Therefore, future research can focus on developing more diverse formats to enhance the flexibility of LMMs. Plus, the alignment technique (Ouyang et al., 2022; Liao et al., 2024; Rafailov et al., 2024) can be used to align the LMMs with human preference. Finally, we envision that

our model can be extended to generation tasks using a similar approach, thereby achieving a unified understanding and generation framework without the need for separate encoders (Radford et al., 2021; Zhai et al., 2023) and decoders (Podell et al., 2023) in a training efficient way.

Visual fine-grained perception examples:



User What color is the ribbon that the man on the right is holding?

- | | |
|-------------------|---|
| LLaVA | The man on the right is holding a gold ribbon. (✗) |
| Chameleon | The man on the right is holding a white and red ribbon. (✗) |
| EVE | The ribbon that the man on the right is holding is green . (✓) |
| HaploVL-7B | The man on the right is holding a green ribbon. (✓) |



User Where is the cat? Options: top-right, top-left, bottom-left, bottom-right

- | | |
|-------------------|--------------------------------------|
| LLaVA | Top-left. (✓) |
| Chameleon | The cat is on top of the blanket.(✗) |
| EVE | Top-right. (✗) |
| HaploVL-7B | Top-left. (✓) |

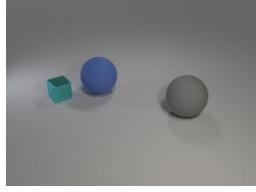


User How many balls are on the table? Options: A: 6, B: 4, C: 5, D: 3

- | | |
|-------------------|---|
| LLaVA | A. (✗) |
| Chameleon | The answer is B: 4. There are four balls on the table.(✓) |
| EVE | A. (✗) |
| HaploVL-7B | B. (✓) |

Table 10: Visual fine-grained perception examples. HaploVL excels in recognizing subtle colors, object locations, and object quantities.

Visual logistic reasoning examples:



User	Subtract all large gray things. Subtract all small brown metallic balls. How many objects are left?
LLaVA	After subtracting all large gray objects and small brown metallic balls, there are three objects left on the table: two blue balls and one gray ball. (X)
Chameleon	There are 12 objects left. (X)
EVE	There are three objects left: a blue sphere, a green cube, and a gray sphere. (X)
HaploVL-7B	After subtracting all large gray things and small brown metallic balls, there are two objects left: a blue sphere and a green cube. (✓)



User	What is the basketball player on the left doing in the image?
LLaVA	The basketball player on the left is trying to steal the ball from the player on the right. (✓)
Chameleon	The basketball player on the left is dribbling the ball with his left hand and is about to shoot the ball with his right hand. (X)
EVE	The basketball player on the left is holding the ball and dribbling it while standing on the court. (X)
HaploVL-7B	The basketball player on the left is attempting to steal the basketball from the player on the right. (✓)



User	What will happen next?
LLaVA	In the image, a young boy with a green shirt is crying and has his eyes closed . It is not possible to predict the exact next event, but it is likely that the boy's emotional state will continue, and he may eventually open his eyes and explain the reason for his crying . Alternatively, he might be comforted by someone or something, which could help him stop crying. The situation highlights the importance of empathy and support for children during their emotional moments. (✓, Hallucinations)
Chameleon	The image shows a young boy with his mouth wide open . He has light brown hair and is wearing a green striped t-shirt. There is a black background. He is likely to blow out some candles on a cake . (X, Hallucinations)
EVE	In the image, a young boy with blond hair is making a funny face , possibly sticking out his tongue. It is not possible to predict exactly what will happen next, but it is likely that the boy will continue to make a funny face or gesture, possibly to entertain himself or others around him. The boy's facial expression suggests that he is in a playful mood, and he might engage in some lighthearted antics or jokes. (X)
HaploVL-7B	In the image, a young boy with blonde hair is shown crying with his eyes wide open . It is difficult to predict exactly what will happen next, but considering the context of the image, it is likely that the boy's emotions will continue to be expressed through his tears. He might be expressing his feelings about something that has upset him or causing him distress. Alternatively, the boy could be reacting to a recent event or situation that has affected him deeply. The image captures a moment of vulnerability and emotionality , and it is important to be empathetic and supportive when encountering such situations. (✓)

Table 11: Visual logistic reasoning examples. HaploVL shows correct reasoning for the open-ended questions.

Visual caption example:



User	Describe this image in detail.
LLaVA	The image features three professional basketball players posing for a photo on the cover of an NBA Sports magazine. They are all wearing yellow sports jerseys, showcasing their team spirit. The players are standing close to each other, with one player on the left, another in the middle, and the third on the right side of the image. In the background, there is a sports ball, likely a basketball, placed near the center of the scene. The players are all smiling and appear to be enjoying the moment, capturing their camaraderie and dedication to the sport. (Distracted by text in images)
Chameleon	This image features two posters advertising the "All-Star Weekend" of the National Basketball Association (NBA). The image is a portrait orientation and features two advertisements in the center of the image. The advertisement on the left is titled "KINGS" in white text with a bold, black outline. The text is placed above a black-and-gold basketball that is shown from the side, with the black side facing the viewer. The advertisement below the title features two athletes, one with their back to the viewer and one facing forward. The athlete on the left is wearing a yellow Nike jersey with black sleeves and a black number "20" on the back. They are holding a black basketball in their right hand and have their left arm raised in the air. The athlete on the right is wearing a gold Nike jersey with black trim and a black number "35" on the back. They are also holding a black basketball in their right hand and have their left arm raised in the air. The athletes are both wearing black shorts and white socks. The background of the advertisement is a black wall with a large, yellow text that reads "ALL-STAR WEEKEND" in bold, white text. Both advertisements feature the Nike logo in the top left corner of the advertisement, and the text "Nike" is printed in white in the top right corner of each advertisement. (Hallucinations)
EVE	The image features three men dressed in yellow basketball uniforms, standing close together and posing for a picture. They are all wearing shorts and sports shoes, and one of them is holding a basketball. The men appear to be athletes, likely part of a basketball team. The image also includes a few other elements, such as a sports ball located near the center of the scene, and a couple of books placed on the ground. The books are positioned on the left side of the image, and they seem to be unrelated to the main focus of the scene, which is the three men in their basketball uniforms. (Hallucinations)
HaploVL-7B	The image is a magazine cover featuring three basketball players from the Cavaliers team. The cover is titled "Sports Illustrated" and has a subtitle "All the King's Men" with a tagline "Lebron James takes his team-building talents to Cleveland." The three players are standing next to each other, with the player in the middle holding a basketball. The cover also includes a note that the Cavaliers have won the NBA championship and a mention of the team-by-team entertainment index. The magazine cover is designed to attract readers interested in basketball and the Cavaliers team. (More consistent with the image)

Table 12: Visual caption example. HaploVL provides a description that is more consistent with the image.

Visual question answering example:



User	Do you know who drew these paintings? Please explain each image.
HaploVL-8B-MI	<p>The first painting is a famous work by Vincent van Gogh, titled "The Starry Night". It depicts a swirling night sky with bright stars and a bright crescent moon, set against a backdrop of dark, rolling hills and a small village with lit windows. The painting is characterized by its bold and expressive brushstrokes, and vibrant use of color. It is widely regarded as one of the most iconic and influential works of post-impressionist art.</p> <p>The second painting is also by Vincent van Gogh, titled "Self-Portrait with Pipe". It shows a man with a beard and pipe, wearing a blue hat, a white collar, and a green coat. The background is a solid orange color, which creates a strong contrast with the man's features. The painting is characterized by its bold and expressive use of color, as well as its swirling brushstrokes, which give it a dynamic and vibrant quality.</p> <p>The third painting is also by Vincent van Gogh, titled "Self-Portrait". It shows a man with a beard and pipe, wearing a blue suit and a white shirt. The background is a swirling pattern of blues and greens, which gives the impression of a dynamic and vibrant energy. The painting is characterized by its bold and expressive use of color, as well as its swirling brushstrokes, which give it a dynamic and vibrant quality.</p> <p>In summary, the first painting is "The Starry Night" by Vincent van Gogh, which depicts a swirling night sky with bright stars and a bright crescent moon. The second painting is "Self-Portrait with Pipe" by Vincent van Gogh, which shows a man with a beard and pipe, wearing a blue hat, a white collar, and a green coat. The third painting is "Self-Portrait" by Vincent van Gogh, which shows a man with a beard and pipe, wearing a blue suit and a white shirt, set against a swirling pattern of blues and greens.</p>

Table 13: Visual question answering example. HaploVL able to process multiple images.