Yiyang Zhou, Chenhang Cui, Jaehong Yoon, Linjun Zhang, Zhun Deng, Chelsea Finn, Mohit Bansal, and Huaxiu Yao. Analyzing and mitigating object hallucination in large vision-language models. *arXiv preprint arXiv:2310.00754*, 2023b.

A Limitations and Discussions

In this work, we observe intriguing findings regarding LLVMs under various experimental settings. To provide a clear and well-defined scope for our conclusions, we further discuss the limitations of the experimental setup for our findings (or claims), explore the most plausible application directions based on our findings, and offer meaningful insights for future research directions for each finding.

A.1 Limitations

Overall, our experiments have several limitations regarding model- and dataset-side generalizability, which are important for a more rigorous analysis. For instance, we primarily evaluate LLVMs on VQA-style tasks, including free-form and multiple-choice question types, and focus exclusively on the LLaVA family. To improve the generalizability of our findings, future work should explore experiments on other LLVMs, such as `Qwen2-VL` Wang et al. (2024), and extend evaluations to additional datasets (e.g., image captioning datasets). Furthermore, demonstrating the impact of model scaling would provide stronger support for our conclusions. Below, we present the specific limitations for each section.

Limitations: Section 3.3. In Figure 1, obtaining the results required running computations for the full number of visual patch tokens, which is highly resource-intensive. This is especially challenging given the large number of visual patch tokens required by recent LLVMs—for example, 576 for `LLaVA-1.5` and more than 1,000 for `Qwen2-VL` Wang et al. (2024).

Limitations: Section 3.4. Synthesized images were generated using `LLaVA-OneVision-7B` Li et al. (2024b) with the prompt template: "*Please generate a caption of this image.*" and `SDXL-Lightning` Lin et al. (2024). To improve robustness, future experiments should explore captions with varying levels of detail, from concise to highly detailed, by using alternative prompt templates, specialized captioning models (e.g., `ShareCaptioner` [4] Chen et al. (2023a)), or more advanced text-to-image generation models that outperform `SDXL-Lightning`. Incorporating these variations would enhance the reliability of our conclusions.

Limitations: Section 3.5. During patch-dropping, we employed the `dino-small` Caron et al. (2021) model for both `Salient PatchDrop` and `Non-Salient PatchDrop`. The impact of patch dropping is likely to vary depending on the size and type of self-supervised vision model used (e.g., large-scale DINO), potentially leading to differing patterns of performance degradation.

Limitations: Section 3.6. While we evaluated visual perception capabilities across various image datasets, many domain-specific image datasets exist in the real world. To draw more generalizable conclusions, it would be beneficial to evaluate additional datasets, such as the VTAB benchmark Zhai et al. (2019). Additionally, we investigated *catastrophic forgetting* by following existing experimental setups from the prior study Zhai et al. (2024). However, comparing LLVMs with contrastive approaches (e.g., `CLIP`) may be unfair due to multiple factors influencing LLVM performance, such as prompt variations and methods for calculating accuracy from the generated text. To enable a more rigorous analysis, future work should explore different prompt methods and fine-tune LLVMs on zero-shot image classification datasets (e.g., CIFAR-100) to assess whether perception capabilities improve. Regarding the LLM-dominance problem during visual instruction tuning, confirming this phenomenon is challenging. To test it effectively, LLVMs should be trained with identical datasets but varying LLM sizes and vision encoder scales. Alternatively, other types of LLVMs that incorporate external computer-vision models (e.g., segmentation models) such as `MoAI` Lee et al. (2024e) could be evaluated. Using visually enhanced LLVMs would strengthen this argument. In addition, for Figure 7, evaluating cross-modal alignment on a broader variety of datasets, such

[4] `https://huggingface.co/Lin-Chen/ShareCaptioner`

as CC12M Changpinyo et al. (2021), WIT Srinivasan et al. (2021), and RedCaps12M Desai et al. (2021), would provide a better understanding of the findings. Expanding this evaluation to various LLVMs, such as `LLaVA-OneVision` and `Qwen2-VL`, would also yield more comprehensive insights.

Limitations: Section 3.7. In Figures 8 and 9, obtaining the importance scores is computationally expensive. For a single run, we calculate the importance scores for each group-wise position (e.g., 36 positions for `LLaVA-1.5`), and we repeat the experiment K times (with $K = 10$). This results in a total of 360 experiments per benchmark. Similarly, the computation for layer importance is also resource-intensive.

A.2 Discussions

Here, we present several discussions based on our findings.

Findings: Permutation Invariance. We suggest that future work focuses on two key directions. First, it is essential to develop more challenging benchmarks that better explore LLVMs' capabilities. Such benchmarks should prioritize free-form question types and avoid including "blind" samples Fu et al. (2024); Li et al. (2024a) that models can solve using commonsense reasoning without actually perceiving the image. Building multi-turn interactive conversation benchmarks, like MMDU Liu et al. (2024d), could be particularly useful in this context. Second, since LLVMs generally exhibit permutation invariance, visual patch tokens can be treated as independent elements, allowing images to be represented as unordered sets of points. Applying paradigms like "Context Clusters," Ma et al. (2023) which rely on clustering algorithms rather than convolutions or attention mechanisms, could improve interpretability and training efficiency. Furthermore, this approach could facilitate generalization to other data domains, such as point clouds Ma et al. (2022), RGB-D data, or sensory images Yu et al. (2024), broadening the applicability of LLVMs.

Findings: Sensitivity to Spatial Structures. One future direction is to develop more robust LLVMs that can handle spatial disruptions. Real-world images often lack perfect clarity—details may be missing, images may be flipped, or other disruptions may occur. To address this, we propose incorporating randomly shuffled images into the training process. By framing this as a jigsaw puzzle Chen et al. (2023b) task, models can be trained to reconstruct the original positions of image patches. This approach could enhance their robustness to spatial variations, making them more applicable to real-world scenarios.

Findings: Catastrophic Forgetting. Balancing perception and cognitive reasoning capabilities is critical. The "catastrophic forgetting" problem Kirkpatrick et al. (2017) has been a long-standing issue in machine learning. A standard approach is to train models on mixed datasets Ke et al. (2020); Gururangan et al. (2020) with a carefully designed balance (a "golden ratio") between perception- and reasoning-related data. Continuously training LLVMs on perception-focused datasets following rehearsal methods Rebuffi et al. (2017) can minimize catastrophic forgetting by retaining knowledge of prior tasks while learning new ones. Knowledge distillation Jin et al. (2021) from large-scale LLVMs (e.g., 72B parameters) to smaller-scale models (e.g., 7B parameters) could help preserve perception capabilities while maintaining reasoning strength. Alternatively, fine-tuning adapters (e.g., p-tuning Liu et al. (2021), LoRA Hu et al. (2021), Q-LoRA Dettmers et al. (2024)) on task-specific datasets offers a lightweight solution to improve performance on new tasks without sacrificing existing capabilities.

Findings: Cross-modal Alignment in the Platonic Representation Hypothesis. Maintaining the original cross-modal alignment is critically important. Continual learning methods (presented above) could be applied to mitigate the loss of alignment during visual instruction tuning. Enhancing the visual perception capability of the projector during training could also help. For instance, employing models such as HoneyBee Cha et al. (2024), which incorporate convolution layer-based projectors, could improve localized understanding. Convolution layers are well-known for their strong inductive bias toward localized feature extraction, making them better suited for capturing fine-grained details in images. Even with the inclusion of complex instruction datasets (e.g., charts, math), a carefully designed projector that excels at extracting detailed and localized information from images could naturally improve both perception and reasoning capabilities. We hypothesize

that enhancing localized perception would inherently lead to improvements in reasoning, aligning the two capabilities more effectively.

Findings: Importance of Central Visual Tokens. Based on our observations, reducing redundant visual tokens in the projector could enhance training and inference efficiency, aligning with findings from prior studies Alayrac et al. (2022); Cha et al. (2024); Xue et al. (2024). Typically, the large number of visual tokens poses a computational burden. This is particularly relevant for real-world scenarios where interleaved format-style conversations Li et al. (2024c); Lee et al. (2024h) are predominant. High visual token counts can make it challenging to train more effective LLVMs for such interleaved conversational formats. Our findings provide a practical direction for reducing visual token counts while maintaining performance. By doing so, we can enable the training of interleaved-format LLVM models more efficiently, similar to approaches highlighted in previous research Xue et al. (2024).

Findings: Importance of Lower Layer. Based on our observations, we emphasize the importance of the traversing layers (TroL) approach Lee et al. (2024b), in improving generalization. In this approach, models are trained to revisit and leverage layer-specific information during the training process. The paper demonstrates that lower layers are more actively engaged, which aligns with our findings. These results suggest that the lower layers of LLVMs play a critical role in establishing a foundational understanding of the world. To enhance this capability, increasing the signal for world understanding in the lower layers during training could be a promising direction. One potential method is injecting noise information into the lower layers during training, as suggested in a prior study Jain et al. (2023). This technique could improve the robustness of LLVMs, further solidifying their foundational perception and reasoning capabilities.

Findings: Relative Importance of Modalities. While the textual modality appears more influential in higher layers, improving the visual perception capability in lower layers is crucial. This is because LLVMs rely heavily on understanding the given image during the initial processing stages. As suggested in prior works Cha et al. (2024); McKinzie et al. (2024), using a larger number of visual tokens, adopting high-resolution image processing Li et al. (2024c), or employing dynamic image processing methods Wang et al. (2024); Li et al. (2024b) is essential for enhancing performance. Furthermore, strengthening the projector's capability for localized visual understanding Cha et al. (2024) could be beneficial. For instance, after the initial image-caption alignment step (commonly the first step in LLVM training), an additional training phase called "empowering localized understanding" could be introduced before visual instruction tuning. This phase would involve adding an extra layer, referred to as the "AL" (Augmented Layer), on top of the simple linear layer. The AL would be trained using a masked autoencoder (MAE) approach He et al. (2022), where the model learns to predict masked image patches. This process would enhance localized visual understanding, ultimately improving the balance between visual and textual modalities and boosting overall model performance.

B Additional Explanation of Platonic Representation Hypothesis

In Section 3.6, we investigate how effectively a trained projector preserves cross-modal alignment, drawing on the *Platonic Representation Hypothesis* Huh et al. (2024). In this section, we provide a detailed explanation of (1) the definition of the Platonic Representation Hypothesis, (2) the alignment metric, and (3) the methodology used to measure alignment in our experiment.

B.1 Definition of the Platonic Representation Hypothesis

Traditionally, different types of AI models represent the world in fundamentally different ways. For instance, when presented with the same reality (e.g., an image, as illustrated in Figure 10), self-supervised vision models might focus on shapes, colors, and optical effects — features critical to visual understanding — while LLMs might emphasize semantic meanings and syntactic structures. Recently, researchers have developed LLVMs by jointly training vision models and LLMs, encouraging them to interpret and represent the world in a more unified manner. The Platonic Representation Hypothesis posits that neural networks, trained with distinct objectives on different data

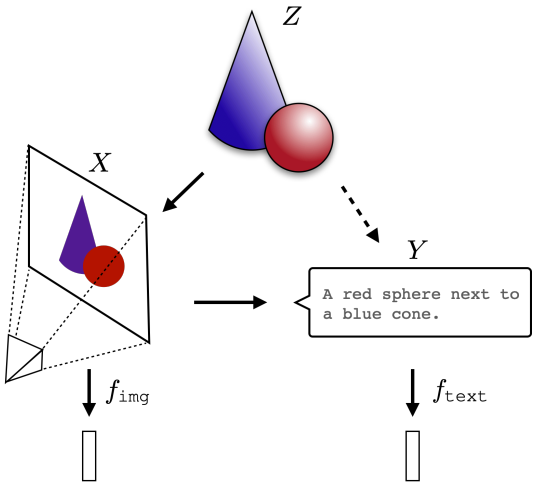

Figure 10: Images (X) and text (Y) are projections of a common underlying reality (Z). We conjecture that representation learning algorithms will converge on a shared representation of Z, and scaling model size, as well as data and task diversity, drives this convergence. For clarity, this figure and its caption have been taken exactly as they appear in the original paper Huh et al. (2024).

modalities, converge toward a shared statistical model of reality in their representation spaces. In the original paper introducing this hypothesis, the authors demonstrated a strong level of alignment between the representations of models trained on disparate modalities (e.g., Figure 3 in the original paper). Based on these findings, we argue that the alignment between models trained on different modalities should not only be preserved but potentially strengthened.

B.2 Alignment Measurement.

To evaluate the alignment between representations from two models, we employ the **Mutual k-Nearest Neighbor (MNN) Metric**. This metric focuses on local similarity by computing the intersection of the k-nearest neighbor sets for each sample from the two models' representation spaces. The alignment is then measured based on the size of these intersections, as detailed below.

Mutual k-Nearest Neighbor Metric. Let f and g denote the representation functions of two models, and let $\mathcal{X}$ represent the data distribution (e.g., an image-caption dataset).

1. The representations for a mini-batch of samples $\{x_i, y_i\}_{i=1}^{b}$ are defined as:
$$\phi_i = f(x_i), \quad \psi_i = g(y_i), \quad i = 1, \ldots, b,$$
where $\Phi = \{\phi_1, \ldots, \phi_b\}$ and $\Psi = \{\psi_1, \ldots, \psi_b\}$ represent the feature sets produced by models f and g, respectively.
2. For each feature ϕ_i and ψ_i, the k-nearest neighbor sets are computed as:
$$\mathcal{S}(\phi_i) = \{k \text{ nearest neighbors of } \phi_i\}, \quad \mathcal{S}(\psi_i) = \{k \text{ nearest neighbors of } \psi_i\}.$$
3. The alignment for a given pair of features (ϕ_i, ψ_i) is defined as the normalized size of the intersection of their k-nearest neighbor sets:
$$m_{\text{NN}}(\phi_i, \psi_i) = \frac{1}{k} |\mathcal{S}(\phi_i) \cap \mathcal{S}(\psi_i)|,$$
where $|\cdot|$ represents the size of the intersection.
4. The overall alignment for the mini-batch is computed as the average alignment across all samples:
$$M_{\text{NN}} = \frac{1}{b} \sum_{i=1}^{b} m_{\text{NN}}(\phi_i, \psi_i).$$

B.3 How to Measure in Our Experiment

To assess the alignment between a suite of large language models (LLMs) and vision models, we utilize the image-caption pair dataset DOCCI Onoe et al. (2024). Specifically, in DOCCI, the dataset consists of image-caption pairs

$$D = \{(x_i, y_i)\}_{i=1}^{|D|},$$

where x_i denotes the image and y_i denotes the corresponding caption text.

For our experiment, we prepare three models: an LLM (f_L), a vision encoder from a vision-language model without visual instruction tuning (f_V), and a vision encoder with a projector, representing a vision-language model with visual instruction tuning (f_{VP}). The vision encoder in f_{VP} is kept identical to f_V. For example, `CLIP-L/336px` is used as the vision encoder for both f_V and f_{VP} when paired with `LLaVA-1.5`.

In our experiment, we explore the degree of alignment lost after visual instruction tuning, guided by the Platonic representation hypothesis. We assume that in a successful LLVM, the projector should effectively represent the visual world and enable the LLM to understand and interpret the given image accurately. We calculate two alignment scores: one between f_L and f_V, and another between f_L and f_{VP}. The discrepancy between these scores reflects the extent to which alignment performance deteriorates.

To compute the alignment scores, we follow these steps:

1. Extract features from f_L by providing the input text y_i. We then apply average pooling to all the extracted hidden states.
2. Extract features from f_V by providing the image x_i, using only the feature corresponding to the `[CLS]` token.
3. Extract features from f_{VP} by providing the image x_i, applying average pooling to all visual patch tokens (e.g., 576 tokens in `LLaVA-1.5`) produced by the projector.

Finally, we calculate the alignment scores using these extracted features via the mutual nearest-neighbor alignment metric.

B.4 Motivation Behind Selecting the DOCCI Dataset

We posit that the ability to perceive and reason based on complex images (e.g., charts, mathematical representations, code snippets, and diagrams) is crucial for creating a helpful assistant. However, we believe that an LLVM must first excel at understanding more natural scenes to become a broadly applicable personal AI assistant, such as one integrated into smart glasses (e.g., Meta AI's glasses[5]) or real-time cameras (e.g., Project Astra [6]). To achieve effective alignment between the language and vision modalities, we require paired datasets where the captions provide detailed descriptions of the corresponding images. These descriptions must include essential visual features such as attributes, spatial relationships, object counts, objects, text rendering, viewpoints, optical effects, and world knowledge. Based on this criterion, we sought an image-caption pair dataset emphasizing (1) natural scenes and (2) highly descriptive captions. The DOCCI dataset meets these requirements effectively. Of course, other datasets could also be considered as candidates, such as Localized Narratives Pont-Tuset et al. (2020), CC12M Changpinyo et al. (2021), COCO-Caption Lin et al. (2014), WIT Srinivasan et al. (2021), or RedCaps12M Desai et al. (2021). In future work, we plan to conduct additional experiments to enhance the generalizability of our observations.

C Additional Explanation of Importance Score

In Section 3.7, we investigate the model's behavior to assess the importance of either a specific layer or a visual token when performing downstream tasks. We hypothesize that introducing arbitrary noise to a specific component — either a layer block or a visual token — will significantly drop the model's performance if that component is crucial to the reasoning process. To quantify this,

[5] `https://www.meta.com/smart-glasses/`

[6] `https://www.youtube.com/watch?v=nXVvvRhiGjI`

we define an *importance score* ($\mathcal{I}$), inspired by the concept of "sharpness of minima." This section provides a detailed explanation of how the importance score is computed.

How is Arbitrary Noise Introduced into Target Layers or Visual Tokens? Based on Equation (2), we prepare the constraint candidate set $\mathcal{C}_t$, defined as a squared boundary:

$$-\epsilon + |x_t| \leq z_t \leq \epsilon + |x_t|, \tag{4}$$

where $\epsilon \sim \text{Uniform}(-1, 1)$. At each iteration, we randomly sample a noise vector z_t and apply it to the target component. Below, we detail how this is done for visual tokens, layers, and modalities.

1. **Visual Token Importance:** When evaluating the importance of a visual patch token (Figure 8), the noise vector is injected into the group-wise visual patch token embeddings at the target position. For instance, Figure 8 illustrates 36 positions. To measure the importance of position 0, we add the noise vector to the corresponding visual patch token embeddings at position 0, while leaving all other patch token embeddings unchanged. These modified embeddings are then input into the LLM for further processing.
2. **Layer-Wise Importance:** To explore layer-wise importance, the noise vector is injected into the target layer before it is processed by the LLM. Specifically, the noise is applied directly to the layer's input embeddings before passing the target layer, ensuring that the perturbation affects only the selected layer.
3. **Modality Importance:** To calculate the importance of the textual modality ($\mathcal{I}_T$), the noise vector is injected only into the positions corresponding to text inputs within the target layer, while leaving the positions associated with visual patch tokens unchanged. Conversely, for visual modality importance ($\mathcal{I}_I$), the noise vector is injected into the positions corresponding to visual patch tokens within the target layer. The relative importance score for each modality is then computed as $\frac{\mathcal{I}_I}{\mathcal{I}_T}$.

To enable better interpretation across layers, all importance scores (both layer-wise and modality-specific) are normalized using `min-max normalization`.

D Additional Explanation of Experimental Setup

In this section, we provide a more detailed explanation of the experimental setup used to obtain our findings, including the required models, preparation of corrupted images, and other specifics. All experiments were conducted using eight A100 GPUs (40GB).

Experimental Setup: Section 3.3. We prepared ViT-variant vision encoder-equipped LVLMs that incorporate visual patch tokens. The experiments focus on visual patch tokens processed after the projector. Before conducting the "permutation invariance" experiments, we first demonstrated whether each visual patch token contains localized information. For the experiment on "sensitivity to spatial structure," shuffled images were used, as shown in Figure 2, following the methodology of a prior study Naseer et al. (2021).

Experimental Setup: Section 3.4. To generate synthesized images, we utilized an image captioner (`llava-hf/llava-onevision-qwen2-7b-ov-hf`) combined with a text-to-image generative model (`sdxl_lightning_8step_unet.safetensors`). Additionally, a prompt template was carefully designed for this purpose.

Experimental Setup: Section 3.5. We prepared occluded images using three masking methods as described in prior work Naseer et al. (2021): `Random PatchDrop`, `Salient PatchDrop`, and `Non-Salient PatchDrop`. To implement `Salient PatchDrop` and `Non-Salient PatchDrop`, we employed the `dino-small` model Caron et al. (2021). Furthermore, to evaluate the robustness of LVLMs to occlusion, we first verified whether ViT-variant encoders exhibit genuine robustness to occlusion by comparing them with CNN-based counterparts, such as `ResNet`.

LLVMs	MMVP	Q-Bench	MME	MMStar	MM-Vet	LLaVA^W	MathVista	SQA^I	ChartQA	AI2D	Avg. Δ
LLaVA-1.5	34.67	59.73	1850.07	34.20	31.50	67.50	24.70	65.59	16.92	53.34	
+ Perm.	36.00 (▲ 1.33)	59.60 (▼ 0.13)	1874.60 (▲ 24.53)	33.33 (▼ 0.87)	30.40 (▼ 1.10)	66.20 (▼ 1.30)	21.20 (▼ 3.50)	65.44 (▼ 0.15)	14.08 (▼ 2.84)	52.69 (▼ 0.65)	▼ 0.59
LLaVA-NeXT	36.67	63.55	1874.42	37.80	43.50	75.50	32.00	62.12	66.06	64.02	
+ Perm.	37.33 (▲ 0.67)	62.54 (▼ 1.00)	1890.19 (▲ 15.78)	36.87 (▼ 0.93)	43.40 (▼ 0.10)	75.80 (▲ 0.30)	21.70 (▼ 10.30)	62.12 (▼ 0.00)	34.55 (▼ 31.51)	64.02 (▼ 0.00)	▼ 2.71
LLaVA-OneVision	60.67	77.26	1982.5	59.87	57.80	87.40	61.80	94.00	93.52	81.25	
+ Perm.	59.33 (▼ 1.33)	76.99 (▼ 0.27)	1964.3 (▼ 18.2)	54.93 (▼ 4.93)	47.60 (▼ 10.20)	82.30 (▼ 5.10)	53.50 (▼ 8.30)	89.24 (▼ 4.76)	58.26 (▼ 35.26)	75.58 (▼ 5.67)	▼ 9.40
QwenVL-2	50.67	77.06	2356.70	55.27	62.60	94.10	59.80	0.00	94.83	80.21	
+ Perm.	48.67 (▼ 2.00)	77.19 (▲ 0.13)	2266.96 (▼ 89.74)	53.47 (▼ 1.80)	62.20 (▼ 0.40)	93.20 (▼ 0.90)	53.10 (▼ 6.70)	0.00 (▼ 0.00)	83.59 (▼ 11.25)	77.43 (▼ 2.78)	▼ 12.82
Fuyu-8B	30.00	40.33	0.00	19.67	16.30	0.00	0.00	0.00	15.81	0.00	
+ Perm.	28.67 (▼ 1.33)	38.80 (▼ 1.54)	0.00 (▼ 0.00)	18.93 (▼ 0.73)	10.90 (▼ 5.40)	0.00 (▼ 44.00)	0.00 (▼ 0.00)	0.00 (▼ 0.00)	7.50 (▼ 8.31)	0.00 (▼ 0.00)	▼ 6.92

Table 4: Results of drop ratio (Δ) when random permutation is applied. We run five experiments.

Experimental Setup: Section 3.6. We curated image classification datasets containing realistic and natural images across various domains. To explore the platonic representation hypothesis Huh et al. (2024), we first thoroughly examined its definition, as detailed in Appendix B. This process involved preparing a diverse set of LLMs, vision encoders, and vision encoders equipped with projectors in LVLMs. We also selected datasets for verifying cross-modal alignment, ensuring that they included natural and realistic images.

Experimental Setup: Section 3.7. We first clarified the definition of "importance score" and determined how to introduce noise into the visual patch tokens. This procedure is described in Appendix C.

E Additional Experimental Results

E.1 Permutation Invariance.

As shown in Table 4, we investigate the extent to which other LVLMs exhibit permutation invariance under the same experimental settings described in Table 1. Overall, the Qwen2-VL-7B Wang et al. (2024) and Fuyu-8B models Bavishi et al. (2023) demonstrate permutation invariance on average, displaying patterns similar to those observed in the LLaVA-family models. A more detailed analysis across benchmarks reveals interesting patterns. In perception-focused benchmarks, such as MMVP, Q-Bench, MME, and MMStar (the latter two being integrated capability benchmarks that include perception-related tasks), the performance drop due to permutation is negligible. However, in text-rich benchmarks like MathVista and ChartQA, the performance drops significantly. These benchmarks require an understanding of detailed numerical information and highly structured geometric graphs, where maintaining the spatial structure of visual patch tokens is critical.

Difficulty of Benchmark. Interestingly, in the SQA^I benchmark, which includes science-related datasets, and the AI2D benchmark, which consists of diagram images, the relatively small performance gap is noteworthy, even though these images are rich in detail. We speculate that this phenomenon might be influenced by the difficulty of the benchmark, particularly the "question type." Benchmarks typically include two question formats: (1) free-form and (2) multiple-choice questions (MCQ). We hypothesize that:

1. LLMs can often solve questions using their extensive commonsense reasoning, even without image perception. Li et al. (2024a); Fu et al. (2024)
2. MCQ formats may be easier for models compared to free-form questions due to the presence of preferred answer patterns or inherent biases in selection.

To investigate further, we conduct additional experiments comparing the difficulty of MathVista, ChartQA, SQA^I, and AI2D. We randomly select 500 samples from each dataset and, for MCQ

Datasets	Question Type	Accuracy (%)	Don't Know (%)
MathVista	Free-Form	0.3	82.1
	MCQ	36.8	0
	Overall	13.6	52.2
ChartQA	Free-Form	0	90
SQA^I	MCQ	64.2	0
AI2D	MCQ	53.2	1.6

Table 5: Accuracy results of ChatGPT on four benchmarks for two different question types.

samples, include only those with four options. We then prompt ChatGPT (i.e., `gpt-3.5-turbo`) to answer these questions using the following templates:

Prompt Template for MCQ

Question: `{question}`
Choices:
`{choices}`
E: I don't know.

Please MUST generate only one option (A, B, C, D, E). Do not generate any explanation.
Answer:

Prompt Template for Free-Form

Question: `{question}`

Please provide your answer. If it is difficult to provide an answer, respond with "I don't know."

We added the "*I don't know*" option to prevent the model from guessing randomly. Table 5 show that ChatGPT performs better on MCQ-type benchmarks compared to free-form types. Moreover, ChatGPT achieves higher accuracy on AI2D and SQA^I compared to MathVista and ChartQA. This supports the observation that LLVMs exhibit less permutation invariance in these text-rich benchmarks, possibly due to the nature of the datasets and their question formats. For free-form questions, the "don't know" response rate is significantly higher, indicating that these benchmarks are more challenging. This highlights the need to minimize "blind" samples — questions solvable by LLMs without image perception — in benchmark design. Benchmarks should prioritize free-form questions to reduce potential selection bias Zheng et al. (2023a), as argued by recent studies Li et al. (2024a).

E.2 Sensitivity to Spatial Structures

As shown in Figure 11, we randomly shuffle image patches to evaluate their impact on model performance and observe that `Qwen2-VL` exhibits a similar tendency to LLaVA-family models. Specifically, we found that `Qwen2-VL` and `LLaVA-OneVision` are highly sensitive to spatial structures in text-rich benchmarks (e.g., MathVista, AI2D), which contain detailed numerical information. Notably, the performance of the `Qwen2-VL` model dropped significantly when the grid size was 2. To understand why `Qwen2-VL` is particularly sensitive, we hypothesize that this behavior is linked to its use of enhanced multi-modal rotary position embeddings (M-ROPE) Wang et al. (2024). This embedding mechanism likely contributes to the performance degradation observed when image patches are shuffled. Conversely, the model is relatively insensitive to spatial structures in perception-centric benchmarks (e.g., MMVP).

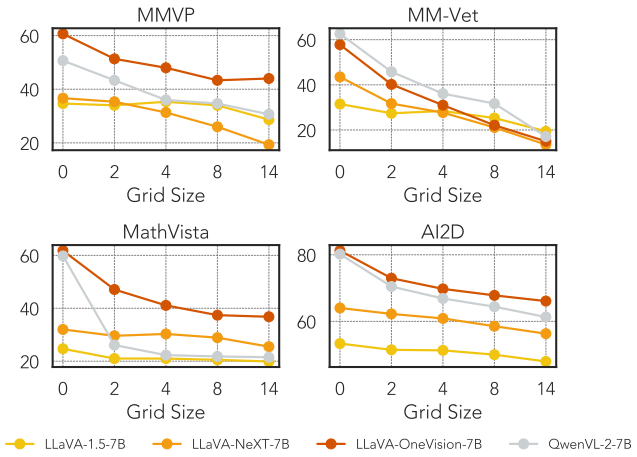

Figure 11: We present the performance across different grid sizes (2, 4, 8, 14) on the MMVP, MM-Vet, MathVista, and AI2D datasets, using four models: `LLaVA-1.5`, `LLaVA-NeXT`, `LLaVA-OneVision`, and `Qwen2-VL`.

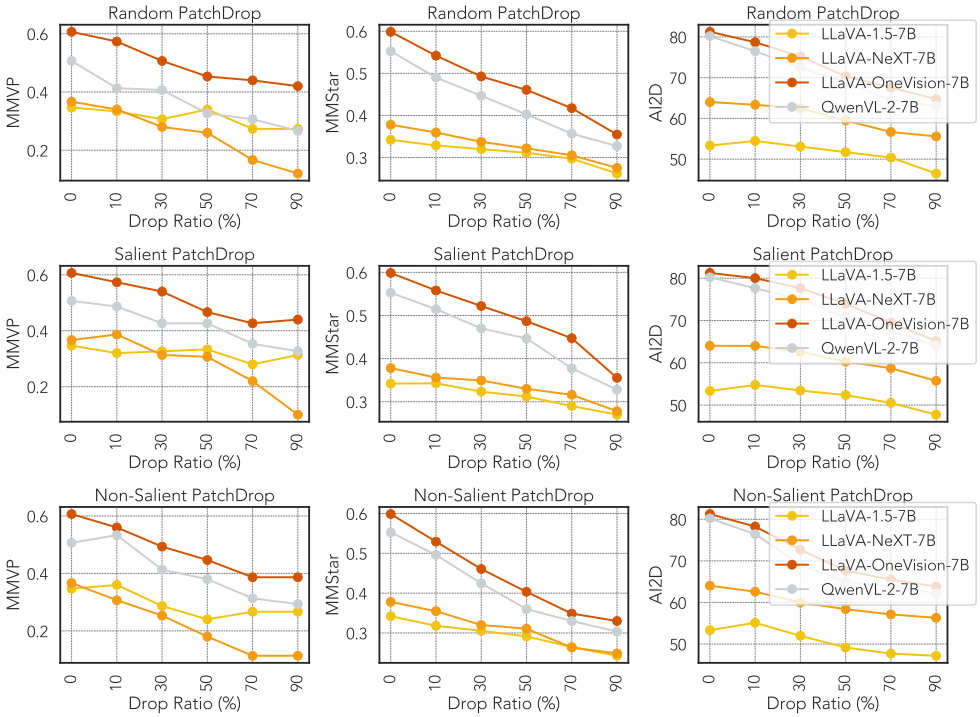

Figure 12: We present robustness performance under occlusion conditions.

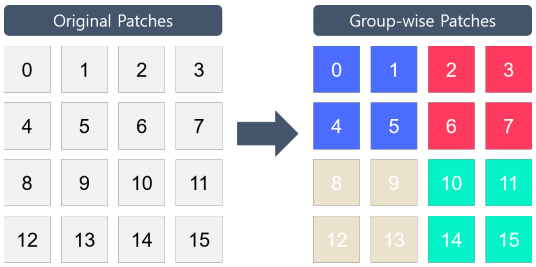

Figure 13: An illustration of group-wise patching.

E.3 Occlusions

In Figure 12, we observe that the `Qwen2-VL` model exhibits a similar tendency to the LLaVA family models. Notably, the performance trend slope of the `Qwen2-VL` model closely resembles that of `LLaVA-OneVision`, suggesting that both models — currently high-performing LVLMs — share similar patterns. This alignment supports the generalizability of our observations. Specifically, LVLMs demonstrate relatively strong performance under occlusion. For instance, in the AI2D dataset, even when 50–70% of image patches are missing, the models can still provide correct answers to some extent. Moreover, in these scenarios, the `Qwen2-VL` and `LLaVA-OneVision` models outperform `LLaVA-1.5` and `LLaVA-NeXT`, even when no patches are missing. These results indicate that state-of-the-art LVLMs possess strong visual understanding capabilities. This suggests that improving visual understanding during training contributes significantly to high performance and robustness against occlusion.

E.4 Varying Grid Size for Grouping Strategy

In Figure 1 and Figure 8, we group the nearest patches. For clarification, we visualize how the patches are grouped, as shown in Figure 13. Similar to the operation of a convolution layer, we group neighboring patches into a single group (indicated by the same color) and feed these groups into the model. Here, we vary the grid size, which corresponds to changing the number of elements in each group, and investigate whether the pattern observed in Figure 1 changes. We conduct additional experiments using a 3×3 grid of patches in Figure 14. We observe that increasing the number of grid patches leads to more precise observations. Compared to a 6×6 grid of patches, a 3×3 grid yields less precise observations. While conducting experiments on all visual patch tokens (576 for `LLaVA-1.5`) would provide the most precise interpretations, this approach is computationally intensive, as mentioned in Section 3.3. Therefore, we believe our chosen grid size strikes a reasonable balance for obtaining meaningful interpretations.

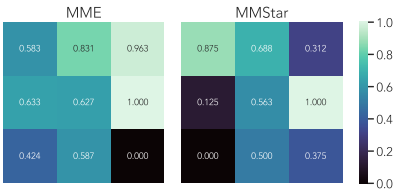

Figure 14: We demonstrate the extent to which group-wise visual tokens capture region-specific information (`PIL`) for `LLaVA-1.5-7B` on the MMStar (Chen et al., 2024a) and MME (Fu et al., 2023) when a 3×3 gird of patches. Darker regions indicate areas where the model retains more localized information for those specific groups.

E.5 Detailed Analysis of Numerical Information

As shown in the above Table 6, in overall, the `Org.` ratio of LLVM generating "1" in free-form question types is reduced compared to the `Syn.` cases. This results suggest that LLVMs can effectively interpret and understand the detailed numerical information in the given image, thereby, the phenomenon that LLVM tend to use their commonsense reasoning is reduced. However, considering the ratio of `LLaVA-1.5` (44%), this ratio is not negligible. Therefore, in the future, we need to build more challenging benchmark that do not rely on the commonsense reasoning.

Additionally, we observe that most LVLMs prefer to answer "no" for `yes/no` question types in multiple-choice question (MCQ) formats. This suggests that, when presented with synthesized images, LVLMs struggle to solve the given questions effectively. Instead of attempting to provide an

	Syn.				Orig.		
Model	Freq. of 1	No (%)	Precision	Recall	Freq. of 1	Precision	Recall
LLaVA-1.5	81.0	64.0	49.2	36.8	44.4	59.4	43.7
LLaVA-NeXT	50.0	54.4	50.0	47.1	13.8	54.0	39.1
Meteor	9.5	82.8	55.2	18.4	7.5	78.0	36.8
LLaVA-OneVision	12.0	70.5	54.9	32.2	8.3	72.4	72.4

Table 6: Detailed analysis of the Syn. and Orig. versions of MathVista Lu et al. (2023). Precision and recall are reported for the yes/no question type.

Datasets	Prompt Template for CLIP	Prompt Template for LLVM
Caltech101	a photo of a {c}.	What is the object in the image? Please answer only a single object in {class_labels}.
CIFAR-100	a photo of a {c}.	What is the object in the image? Please answer only a single object in {class_labels}.
Food101	a photo of {c}, a type of food	What is the type of food in the image? Please answer only a single type of food in {class_labels}.
Pets	a photo of a {c}, a type of pet.	What is the type of pet in the image? Please answer only a single type of pet in {class_labels}.
Country211	a photo showing the country of {c}.	What is the country in the image? Please answer only a single country in {class_labels}.
EuroSAT	a centered satellite photo of {c}.	What is the type of centered satellite in the image? Please answer only a single type of centered satellite in {class_labels}.
AirCraft	a photo of a {c}, a type of aircraft.	What is the type of aircraft in the image? Please answer only a single type of aircraft in {class_labels}.

Table 7: Prompt templates used for evaluating CLIP and LLMs on zero-shot image classification tasks. The c represents a single class label, while class_labels refers to all class labels provided by each dataset.

answer based on the limited or unclear information available in the synthesized images, LVLMs tend to decline by answering "no," leading to an increased frequency of "no" responses compared to "yes." Furthermore, across all models, the Org. dataset consistently yields better performance in both precision and recall. This indicates that LVLMs face significant challenges in solving questions based on synthesized information. In the Syn. case, precision is consistently higher than recall, reflecting the tendency of LVLMs to output "no" answers more frequently than "yes" answers. This behavior underscores the challenges LVLMs face in effectively using synthesized visual information to provide accurate answers to yes/no questions.

E.6 Additional Results of Cross-Modal Alignment

How to evaluate the zero-shot image classification task? To evaluate CLIP models on the zero-shot classification task, we use the prompt templates provided by CLIP-Benchmark [7]. All the prompt templates we used are presented in Table 7. For evaluating LLVMs on the zero-shot image classification task, we design prompt templates inspired by those used for the CLIP model. Using these templates, the LLVM predicts a single class label. Based on the LLVM's generated answer, we then use ChatGPT to verify the prediction. Specifically, we utilize the following prompt: *Please only answer the question in yes or no. Is the "Prediction" correctly predicting the right 'Label"? Label: label; Prediction: outputs.* This evaluation method strictly follows the approach used in an existing study Zhai et al. (2024).

E.7 Additional Results on Image Captioning Task

The evaluation benchmarks used in our experiments primarily consist of VQA tasks, which focus on binary, multiple-choice, and free-form question types. To address whether our claim regarding "permutation invariance" generalizes to other datasets, we conduct additional experiments using image captioning tasks. These tasks inherently require "visual processing capabilities," such as understanding attributes, viewpoints, scenes, and objects. For this investigation, we evaluate three standard datasets: COCO-Captions Lin et al. (2014) (Karpathy test set), NoCaps Agrawal et al. (2019) (validation set), and TextCaps Sidorov et al. (2020) (validation set). To generate captions, we followe the default prompting setup from LMMs-Eval [8], which uses the prompt: "*Please carefully observe the image and come up with a caption for the image.*" We employ standard evaluation metrics — ROUGE-L Lin (2004) and CIDEr Vedantam et al. (2015) — to assess performance.

[7] https://huggingface.co/clip-benchmark

[8] https://huggingface.co/lmms-lab

	COCO-Captions		NoCaps		TextCaps	
LLVMs	ROUGE-L	CIDEr	ROUGE-L	CIDEr	ROUGE-L	CIDEr
LLaVA-1.5	22.01	0.97	25.34	1.52	22.46	6.09
+ `Perm.`	22.62	1.26	26.05	2.89	22.94	7.28
Avg. Δ	▲ 0.62	▲ 0.29	▲ 0.71	▲ 1.37	▲ 0.48	▲ 1.19
LLaVA-NeXT	21.63	8.12	22.78	6.26	21.49	15.94
+ `Perm.`	21.86	7.64	22.68	5.81	20.19	12.30
Avg. Δ	▲ 0.24	▼ 0.48	▼ 0.10	▼ 0.44	▼ 1.29	▼ 3.65
LLaVA-OneVision	57.23	116.25	56.09	86.60	44.58	72.69
+ `Perm.`	56.70	116.17	56.36	85.94	44.19	68.18
Avg. Δ	▼ 0.53	▼ 0.08	▲ 0.26	▼ 0.66	▼ 0.39	▼ 4.52
Qwen2-VL	39.98	44.61	44.01	39.37	35.80	46.86
+ `Perm.`	37.19	39.29	42.70	38.35	35.31	44.64
Avg. Δ	▼ 2.79	▼ 5.33	▼ 1.31	▼ 1.02	▼ 0.49	▼ 2.22

Table 8: Results of drop ratio (Δ) when random permutation is applied. We run five experiments.

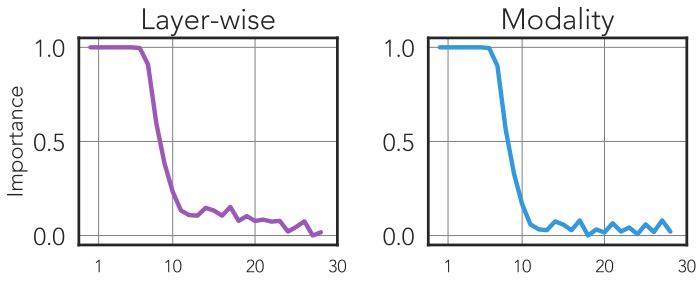

Figure 15: We present the results of (left) layer-wise importance and (right) modality importance within the layers on MME Fu et al. (2023) dataset.

As shown in Table 8, we observe similar trends across image captioning datasets: **most LLMs exhibit permutation invariance.** Interestingly, on the TextCaps dataset, the performance drop is more pronounced compared to other datasets, suggesting relatively greater permutation variance. TextCaps contains more complex images (e.g., those with detailed numerical information) compared to the other datasets, which may explain this phenomenon. When comparing these findings to those in Table 1, we note that in perception-related tasks (e.g., involving natural scenes), LLVMs generally exhibit permutation invariance. However, in reasoning-related tasks (e.g., MathVista) involving images with complex structures (e.g., charts or diagrams), LLMs demonstrate greater permutation variance. This suggests that maintaining the geometric or positional structure of plots and charts is crucial.

E.8 Additional Results on Layer & Modality Importance

Figure 15 (left) shows that the lower layers (< 10) play a crucial role in handling integrated capabilities. Meanwhile, Figure 15 (right) demonstrates that in the lower layers (< 12), the image modality is more important than the text modality. Overall, the tendencies observed on the MME dataset are similar to those on the MMStar dataset, as shown in Figure 9. However, a key difference lies in the layer index at which the modality importance shifts; for the MME dataset, this transition occurs at a higher layer index. Based on these results, we hypothesize that LLVMs allocate more effort to understanding the given images on the MME dataset compared to the MMStar dataset. One of the possible reason is that the images in the MME dataset are more challenging for the model

to comprehend, but we can not guarantee this reason is correct, therefore, Further investigation is required to validate this assumption in future studies.

F Additional Related Works

Model-Stitching. The model-stitching (Lenc & Vedaldi, 2015; Bansal et al., 2021) is a technique first introduced to study the internal representations of neural networks by measuring the representational similarity between two given models. Consider two models defined as $f = f^m \circ \cdots \circ f^1$ and $g = g^n \circ \cdots \circ g^1$. Specifically, the *stitched* model is formalized as $\mathcal{F} = g^n \circ \cdots \circ g^{k+1} \circ s \circ f^k \circ \cdots \circ f^1$, where s is a simple stitching layer (e.g., a linear layer or a 1×1 convolution). Therefore, even if the two models f and g differ in training methodology (e.g., supervised vs. self-supervised) or modalities (e.g., text vs. image), if $\mathcal{F}$ exhibits good performance, then f and g have strongly correlated and compatible internal representations at layer k, apart from the stitching layer s. Merullo et al. (2022) have the similar concept of *model-stitching* to verify their strong hypothesis that the conceptual representations from a frozen LLM and a visual encoder are sufficiently similar such that a simple linear mapping layer can align them.

G Additional Examples of Synthesized Images

We provide additional examples of synthesized images in Figure 16.

H Additional Examples of Shuffled Images

We provide additional examples of shuffled images in Figure 17.

I Additional Examples of Occluded Images

We provide additional examples of occluded images in Figure 18.

J Description of Evaluation Benchmarks

- **MM-Vet** (Yu et al., 2023) dataset is a benchmark designed to evaluate large vision-language models (LVLMs) across six core vision-language (VL) capabilities: recognition, knowledge, optical character recognition (OCR), spatial awareness, language generation, and mathematical reasoning. The dataset includes open-ended, real-world questions based on image-text pairs, requiring models to integrate multiple capabilities to solve complex tasks. MM-Vet benchmark consists of 200 images paired with 218 open-ended questions.
- **Q-Bench** (Wu et al., 2023) evaluates the capabilities of large vision-language models in three main areas related to low-level vision tasks. These tasks focus on evaluating how well LVLMs can perform basic low-level perception tasks that are traditionally associated with human visual perception. In the Q-Bench dataset, the questions are of three types: Yes-or-No, What, and How.
  - **Low-Level Visual Perception**: Assesses how accurately LVLMs can answer questions about low-level image attributes (e.g., clarity, color, distortion). LLVisionQA dataset includes 2,990 images, each with a corresponding question about low-level features.
  - **Low-Level Visual Description**: Evaluates the ability of LVLMs to describe images. LLDescribe dataset has 499 images with expert-labeled descriptions averaging 58 words each. LVLMs are compared against these to assess completeness, preciseness, and relevance.
  - **Visual Quality Assessment**: Evaluates LVLMs' ability to predict quantifiable quality scores for images by assessing how well they align with human-rated mean opinion scores (MOS) on low-level visual appearances, using 81,284 samples.

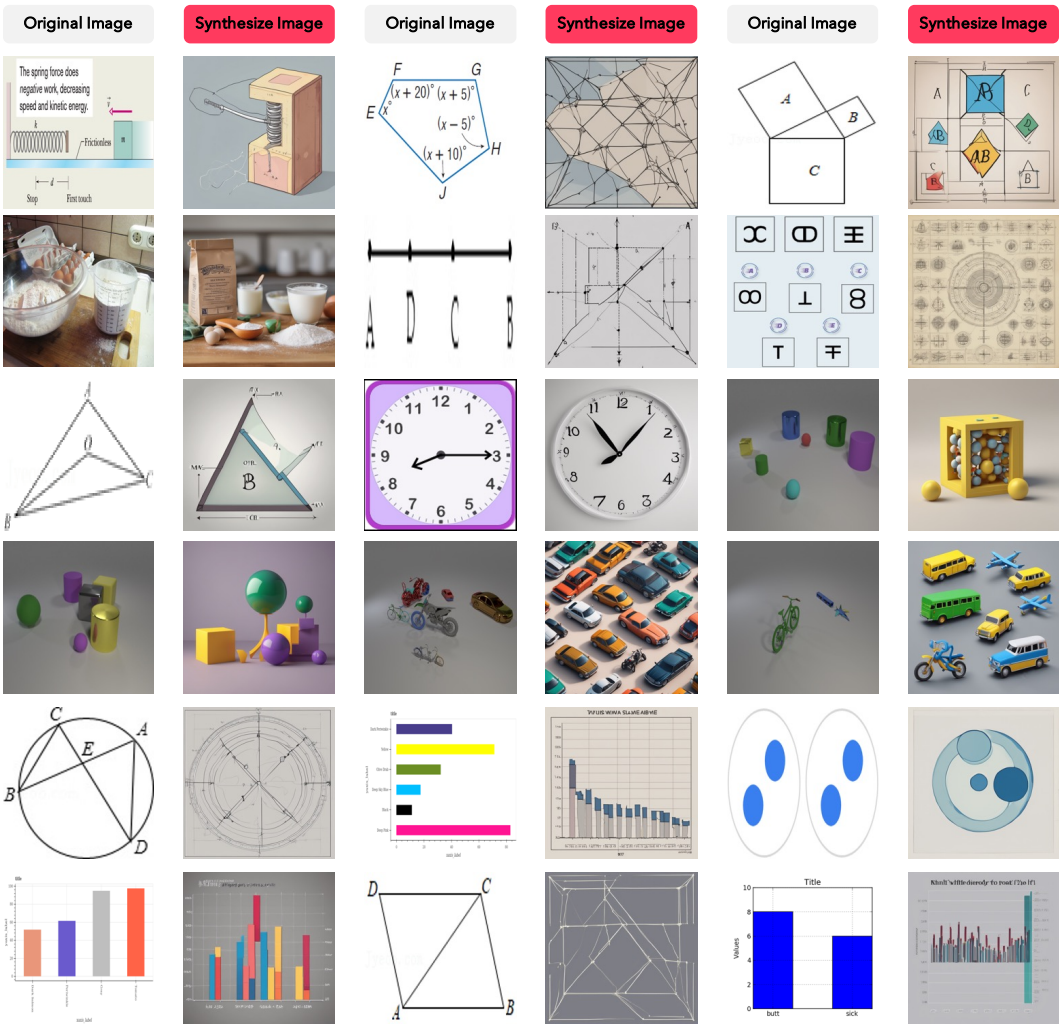

Figure 16: Examples of synthesized images from MathVista Lu et al. (2023).

Figure 17: Examples of synthesized images from MM-Vet Yu et al. (2023).

Figure 18: Examples of occluded images from MME Fu et al. (2023).

- **SQA-IMG** (Lu et al., 2022a) is a portion of the Science Question Answering (SQA) dataset that contains questions from a wide range of scientific domains, each paired with corresponding image contexts. The dataset includes 10,332 examples of multimodal multiple-choice questions, along with lectures and explanations that detail the reasoning behind the correct answers.
- **ChartQA** (Masry et al., 2022) dataset is a benchmark designed to test AI models on their ability to perform question-answering tasks over various types of charts. It focuses specifically on questions requiring complex reasoning, such as visual and logical interpretation, going beyond simpler template-based datasets. ChartQA includes 9,608 human-authored open-ended questions as well as 23,111 questions that are automatically generated from chart summaries.
- **SEED-IMG** (Li et al., 2023), a subset of SEED-Bench, focuses on evaluating spatial comprehension of images by testing models on various dimensions like scene understanding, object identification, and spatial relationships. In terms of scale, the dataset includes 19,000 multiple-choice questions that evaluate both image and video comprehension, covering 12 evaluation dimensions such as scene understanding, instance identity, spatial relations, and action recognition.
- **MME** (Fu et al., 2023) evaluates both perception and cognition abilities of LVLMs. It features 14 subtasks, including recognition tasks (such as object existence, count, position, color) and reasoning tasks (such as commonsense reasoning, numerical calculation, text translation, and code reasoning). MME uses manually created instruction-answer pairs, ensuring no overlap with public datasets. MME uses "yes/no" responses for quantitative evaluations.
- **MathVista** (Lu et al., 2023) is a benchmark designed to evaluate the mathematical reasoning capabilities of foundation models in visual contexts. It integrates challenges from diverse mathematical and visual tasks, with a focus on fine-grained, deep visual understanding and compositional reasoning. MathVista consists of 6,141 examples including 3,392 multiple-choice questions and 2,749 free-form questions derived from 28 existing multimodal datasets and 3 newly created datasets: IQTest, FunctionQA, and PaperQA.
- **LLaVA-W** (Liu et al., 2024c) is a challenging evaluation benchmark created to assess the generalization and instruction-following capabilities LVLMs in complex, real-world situations. It consists of 24 images and 60 questions, including diverse scenes like indoor environments, outdoor settings, memes, paintings, and sketches. Each image is associated with a highly detailed and manually curated description, and the questions focus on extracting intricate details and reasoning about the visual content. LLaVA-W involves a variety of tasks, including detailed descriptions, conversational answers, and complex reasoning.
- **MMStar** (Chen et al., 2024a) is a vision-dependent multimodal benchmark designed to evaluate the multimodal capabilities of LVLMs. It addresses two main issues identified in previous benchmarks: the reliance on textual information without visual input and data leakage during training. MMStar is composed of 1,500 samples carefully selected to ensure that visual content is necessary for solving each problem. MMStar evaluates six core capabilities across 18 detailed axes, which include tasks like image perception and logical reasoning. MMStar uses multiple-choice as the primary answer type.
- **MMVP** (Tong et al., 2024) evaluates the visual grounding capabilities of large vision-language models by identifying scenarios where they fail to distinguish simple visual patterns in images. These patterns include aspects like orientation, counting, viewpoint, and relational context. The benchmark is constructed using 150 pairs of images, resulting in 300 multiple-choice questions.

K Description of Evaluation LVLMs

- **LLaVA-1.5** (Liu et al., 2024a) incorporates academic task-oriented datasets to enhance performance in VQA tasks and features an MLP vision-language connector, which improves upon the original linear layer utilized in LLaVA (Liu et al., 2024c). It uses CLIP ViT-L/14 (Radford et al., 2021) with a 336px resolution as its vision encoder, resulting in a total of $(336/14)^2$ = 576 visual tokens. LLaVA-1.5 is built on Vicuna with either 7B or

13B parameters. The training dataset includes 558K samples for pre-training and 665K for fine-tuning, totaling 1.2M image-text pairs from publicly available datasets

- **LLaVA-NeXT** (Liu et al., 2024b) (also known as LLaVA-1.6) enhances visual reasoning, OCR, and world knowledge, offering four times higher image resolution (up to 1344x336) and improved performance in visual conversations. Its architecture includes a CLIP ViT-L/14 as a vision encoder, paired with Vicuna models ranging from 7B to 34B as a backbone language model. It utilizes 1.3M visual instruction tuning data samples for training, maintaining efficiency with approximately one day of training on 32 A100 GPUs. The architecture's high resolution and dynamic grid scheme improve detailed image processing capabilities.
- **LLaVA-OneVision** (Li et al., 2024c) is a LVLM designed for task transfer across single-image, multi-image, and video scenarios, with strong capabilities in video understanding through image-to-video task transfer. Its architecture consists of a Qwen2 language model (Yang et al., 2024) with 8B to 72B parameters, and the SigLIP vision encoder (Zhai et al., 2023), which processes images at a base resolution of 384x384, producing 729 visual tokens. The model employs a 2-layer MLP projector. The training utilized 3.2M single-image data samples and 1.6M multi-modal data samples, focusing on high-quality visual instruction tuning data to enhance its multimodal capabilities.
- **Meteor** (Lee et al., 2024c) is a large vision-language model that uniquely embeds multi-faceted rationales using a Mamba-based architecture (Gu & Dao, 2023), enabling efficient processing of lengthy rationales to enhance its vision-language understanding. This approach allows Meteor to achieve superior performance without scaling up model size or employing additional vision encoders. Its architecture includes a CLIP-L/14 vision encoder with an image resolution of 490x490, comprising 428M parameters, and InternLM2-7B (Cai et al., 2024) as a foundational LLM. Meteor was trained on 2.1M question-answer pairs, with 1.1M curated triples.
- **TroL** (Lee et al., 2024b) uses a unique characteristic called layer traversing, which reuses layers in a token-wise manner, allowing it to simulate retracing the answering process without physically adding more layers, making it efficient despite smaller model sizes. TroL uses CLIP-L and InternViT as vision encoders, containing 428M and 300M parameters, respectively, and supports 24 layers. The image resolution is adjusted using MLPs in the vision projector. For its foundational LLM, TroL utilizes Phi-3-mini with 3.8B parameters and InternLM2 with 1.8B and 7B parameters. The training dataset comprises 2.3M visual instruction tuning samples.