

VLSI: Verbalized Layers-to-Interactions from Large to Small Vision Language Models

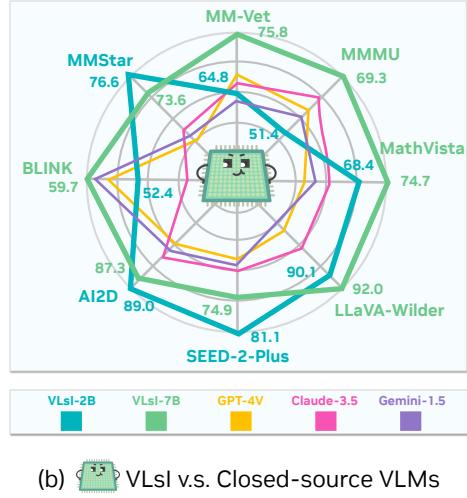
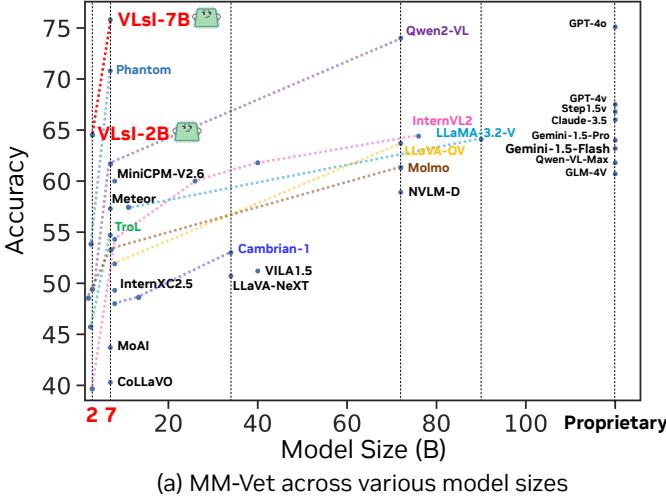
Byung-Kwan Lee^{1,2†}Ryo Hachiuma¹Yu-Chiang Frank Wang^{1,3}Yong Man Ro^{2‡} Yueh-Hua Wu^{1‡}¹NVIDIA, ²KAIST, ³National Taiwan University

Figure 1. Performance overview of VLSI on vision-language benchmarks. (a) Accuracy on MM-Vet [100] for various model sizes, showing that VLSI (2B and 7B) achieves competitive performance compared to proprietary closed-source VLMs. (b) Comparative evaluation on multiple challenging benchmarks, where VLSI (green and blue) outperforms leading closed-source VLMs, including GPT-4V [79], Claude-3.5-Sonnet [1], and Gemini-1.5-Pro [87], highlighting its efficiency and effectiveness across diverse tasks.

Abstract

The recent surge in high-quality visual instruction tuning samples from closed-source vision-language models (VLMs) such as GPT-4V has accelerated the release of open-source VLMs across various model sizes. However, scaling VLMs to improve performance using larger models brings significant computational challenges, especially for deployment on resource-constrained devices like mobile platforms and robots. To address this, we propose VLSI: Verbalized Layers-to-Interactions, a new VLM family in 2B and 7B model sizes, which prioritizes efficiency without compromising accuracy. VLSI leverages a unique, layer-wise distillation process, introducing intermediate “verbalizers” that map features from each layer to natural language space, allowing smaller VLMs to flexibly

align with the reasoning processes of larger VLMs. This approach mitigates the training instability often encountered in output imitation and goes beyond typical final-layer tuning by aligning the small VLMs’ layer-wise progression with that of the large ones. We validate VLSI across ten challenging vision-language benchmarks, achieving notable performance gains (11.0% for 2B and 17.4% for 7B) over GPT-4V without the need for model scaling, merging, or architectural changes. [Project Page](#).

1. Introduction

The integration of large language models (LLMs) with vision-language models (VLMs) has significantly enhanced the interpretive and processing capabilities of visual systems [5, 23, 78, 96]. By leveraging architectures such as CLIP-aligned vision encoders [11, 105], these VLMs have achieved unprecedented performance in understanding and responding to visual inputs. The core of these advance-

† Work Done during Internship.

‡ Corresponding Author.

ments is visual instruction tuning [18, 65], which pairs images with question-answer texts to provide VLMs with rich, instruction-based training. Closed-source VLMs like GPT-4V [78] and Gemini-Pro [87] have led the way, generating high-quality instruction tuning samples and setting new performance standards for visual language understanding.

In response, open-source VLMs of various sizes, including LLaVA-OneVision(OV) [56] and Qwen2-VL [92], have rapidly emerged. While these models demonstrate the advantage of scaling for performance gains in vision-language tasks, the computational cost of larger models presents a critical barrier to deployment in real-world, resource-limited settings, such as mobile devices and autonomous robots. Consequently, the challenge lies in designing high-performing, efficient VLMs capable of handling complex visual tasks without requiring extensive hardware resources.

Traditional approaches to address these constraints often involve adding specialized modules [53] or modifying model architectures [49]. However, these methods introduce significant engineering complexity and can lead to compatibility issues during deployment, particularly for on-device applications where low-latency and memory efficiency are paramount. Furthermore, recent evaluations using benchmarks like MM-Vet [100] and MMMU [102] reveal that these structural modifications still struggle with advanced visual reasoning tasks. This raises the question: *Can we achieve a similar or superior level of performance without scaling, merging, or architectural changes?*

To address this, we introduce  **VLSI**: Verbalized Layers-to-Interactions, a new VLM family that leverages an innovative, natural language-based distillation process to efficiently transfer knowledge from large to small VLMs. Unlike traditional distillation methods, which often directly imitate outputs from a larger model,  VLSI introduces a layer-wise approach where each intermediate layer generates verbal responses in natural language space, enhancing interpretability and alignment with larger models. This is achieved through a three-step process: (1) the *verbalization step*, which uses “verbalizers” to project intermediate features into the language space, making them interpretable as text-based responses; (2) the *interaction step*, which performs adaptive layer matching to align the reasoning progression between large and small VLMs; and (3) the *SFT step*, which finetunes the distilled VLMs for task-specific instruction-following responsiveness.

Our experiments validate  VLSI’s effectiveness across ten challenging benchmarks, demonstrating significant performance gains of 11.0% (2B model) and 17.4% (7B model) over GPT-4V. Notably, these improvements are achieved without increasing model size, merging modules, or modifying the architecture. Our contributions are as:

- We introduce  VLSI, a new VLM family that applies natural language-based, layer-wise distillation, offering

a scalable solution to high-performing yet efficient-scale VLMs without requiring scaling or structural changes.

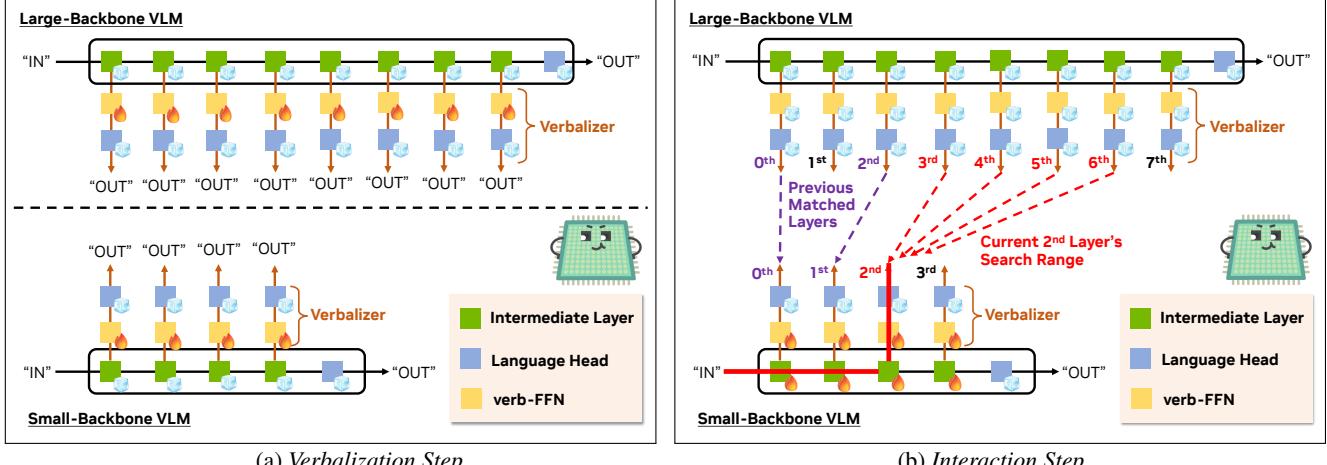
- We present sampling distribution-based dynamic layer matching before distillation, in order to address the layer number gap between large and small VLMs.
-  VLSI is easy to implement and adaptable across various model architectures, making it practical and deployable for real-world applications.

2. Related Works

Evolution of Vision-Language Models. The emergence of visual instruction tuning: LLaVA [65] and Instruct-BLIP [18] initially brings in not only introducing slight variations of VLMs [3, 8, 17, 48, 54, 107, 112], but also curating high-quality visual instruction samples [7, 9, 27, 32, 91, 99, 109]. Since that time, there has been a growing interest in enhancing visual understanding; thus the simple visual input technique of enlarging image resolution or dividing images into smaller sections with fixed or dynamic rules has got standardized [12, 33, 60, 66, 74, 94]. Furthermore, merging additional visual encoders [26, 46, 80, 105] or multiple computer vision models [13, 22, 76, 97] into LLMs have also become a major focus [34, 52, 53, 70, 82, 113]. Besides, Meteor [51] employs an additional rational projector that embeds multifaceted reasoning information to cover diverse capabilities, including chart, diagram, document, and math reasoning. More recently, Cambrian-1 [89], LLaVA-OneVision [56] InternVL2 [12], Molmo [20], and Qwen2-VL [92] have released large scale models in order to follow or surpass the performances of closed-source VLMs. While these advancements are both rapid and impactful, relatively few studies focus on achieving high-performing yet efficient-scale VLMs within limited architectures. Furthermore, many current approaches rely heavily on GPT-based instruction datasets, and only the final layer learns the target responses in visual instruction tuning. In contrast,

 VLSI presents how to effectively harness the capabilities of larger VLMs—already outperforming the closed-source ones—to transfer internally embedded knowledge from large to small VLMs.

Efficient Modeling Strategy. In the field of lightweight LLMs, MobiLlama [88], OpenELM [75], MobileLLM [68] have leveraged various engineering techniques such as shared feed-forward network (FFN) design, layer-wise scaling, and embedding-language head sharing to efficiently reduce model parameters. Their primary objective is not to close the gap with closed-source LLMs but rather to reduce parameters impacting less performance degradation. In the field of VLMs, there are variations about how to utilize pretrained lightweight LLMs [2, 35, 108] in order to make efficient-scale VLMs [14, 15, 62, 81, 110, 111],



(a) Verbalization Step

(b) Interaction Step

Figure 2. Illustration of the training process in VLSI, showing (a) the verbalization step and (b) the interaction step. (a) In the verbalization step, intermediate layers in both the large- and small-backbone VLMs are equipped with a “verbalizer”, allowing their outputs to be projected into natural language space. Autoregressive loss is applied to align these verbalized outputs with the target responses. (b) In the interaction step, each intermediate layer in the small-backbone VLM searches for a matching layer in the large backbone VLM within a specified range. For example, once the 2nd layer of the small VLM is matched with the 4th layer in the large VLM, the next matching search for the 3rd layer in the small VLM will proceed from the 5th to the 7th layers of the large VLM, ensuring progressive alignment. Note that, “fireball” icon represents parameters that are actively trained or updated, while the “ice cube” icon indicates untrained parameters.

but these works are also not a fundamental solution to embed more vision-language knowledge within limited structures. Notably, two VLM approaches, TroL [50] and Phantom [49], aim to expand learning own capabilities within limited structures by doubling forward propagation steps and enlarging the latent dimension without physically increasing model sizes, thereby showing large improvements. Unfortunately, these approaches face limitations such as key-value cache storage constraints and extensive architectural modifications, which may hinder direct application to real-world scenarios. Besides, a few works in distilling language models [61, 77, 85] have emerged, and TED [61] utilizes language head but it is not a version of LLM covering general tasks, and it does not handle layer number gap between large and small models. The other studies simply use the final layer distillation. For VLMs, LLaVA-MoD [83] and LLaVA-KD [4] have also been recently proposed, but they use the same way. Align-KD [28] distills both vision features and logits from the language head, while MoVE-KD [6] employs multiple vision encoders for large VLMs and a mixture of LoRA [40] for small VLMs. Regarding the distillation metric, DistILM [47] and MiniLLM [36] emphasize the effectiveness of reverse KL divergence, whereas LLAVADI [95] and Minitron [77] demonstrate that normal KL divergence is more effective. On the other hand,

VLSI makes layer-wise distillation process where we leverage natural language in order to make small VLMs mimic the reasoning progression of large VLMs across layers. We hope that incorporating natural language will facilitate smoother communication between large and small VLMs, alleviating the complexities of feature alignment.

3. VLSI: Verbalized Layers-to-Interactions

Overview of Model Architecture. As illustrated in Fig. 2, VLSI comprises two main components: the backbone VLM and verbalizers. For the backbone VLM, we use Qwen2-VL [92], selected for its high performance on the OpenVLM-Leaderboard [16]. The verbalizer consists of a simple FFN [90] and the language head from the backbone VLM. To distinguish between FFN components, we refer to the FFN within the verbalizer as “verb-FFN” to avoid confusion with the FFN in the LLM transformer decoder block. This design is inspired by the recent speculative decoding paradigm [59], which demonstrates the effectiveness of using a smaller LLM constructed with a frozen word embedding and the language head of larger LLMs to emulate the performance of those larger models. Building on this insight, we incorporate the language head of the backbone VLM into its intermediate layers. Specifically, the verb-FFN is placed between each intermediate layer and the language head, introducing trainable parameters that allow effective projection into the language space. In other words, the verb-FFN enables outputs from the intermediate layers to project to the language space via the language head. In the following sections, we detail the three critical training stages: *verbalization*, *interaction*, and *SFT*.

3.1. Verbalization Step

In this step, we introduce a verbalizer for each target intermediate layer (see Fig. 2(a)), allowing the outputs of these layers to be mapped to the natural language space. Each verbalizer comprises a verb-FFN (the yellow block in Fig. 2(a)) and a language head (the blue block in Fig. 2(a)), which is the same language head used in the corresponding backbone VLM. It is important to note that the outputs from in-

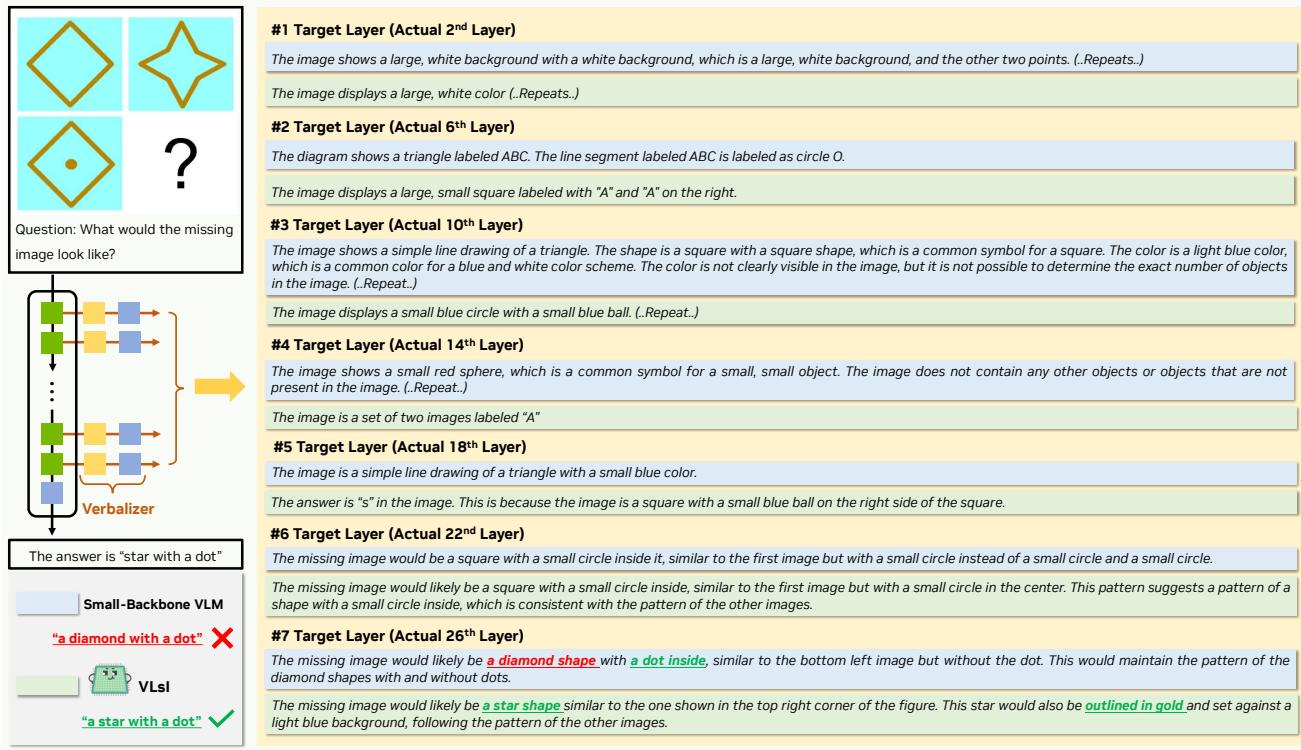


Figure 3. Example of verbalized outputs from each intermediate target layer in an alternative small-backbone VLM (without VLSI) and the VLSI. The visual question prompts VLM to predict the missing image in a sequence pattern. The outputs illustrate how each layer progressively interprets the visual cues, with VLSI accurately identifying the answer as ‘a star with a dot’ in the final layer, while the alternative small-backbone VLM incorrectly predicts ‘a diamond with a dot’. This demonstrates the improved interpretative capability of VLSI through layer-wise, language-based distillation.

intermediate layers are not directly translatable to the natural language space. To address this, we apply an additional verbalizer to process these intermediate layer outputs. To optimize the mapping from embeddings to natural language, we leverage instance pairs from visual instruction tuning datasets and apply an autoregressive loss to ensure the verbalized output aligns with the target response. Since the weights in both backbone VLMs are fixed, the gradient updates for each verbalizer at each layer remain independent. Our goal here is not for the intermediate layers to generate the correct response but rather to showcase their capacity to express verbal information given specific visual and text inputs. Fig. 3 presents the verbalization results, displaying the output of each layer along with its corresponding verbalizer. The results show a gradual improvement in reasoning and response accuracy as layers progress deeper. This verbalization enables our VLSI to track the evolution of verbal responses in the natural language space, facilitating a clearer understanding of the ‘key developments’ required to generate desired responses and thereby offering a more efficient distillation approach.

3.2. Interaction Step

After the *verbalization step*, we proceed with a distillation process that leverages the natural language output from each

intermediate layer. The main objective of this *interaction step* is to establish an effective mapping between the layers of the large-backbone and small-backbone VLMs. This approach ensures that the small-backbone VLM mirrors the reasoning progression in the large one as layers deepen. Furthermore, because the computational requirements for each ‘key development’ in generating the desired response vary, we propose an adaptive layer-matching strategy that dynamically aligns corresponding layers.

Extracting Verbal Information for Distillation. To extract verbal information from intermediate layers, we use the vocabulary probabilities from the language head in the verbalizer based on input from the visual instruction training dataset. This method avoids the high computational cost of text generation (e.g., greedy or beam search [29]), which would be prohibitively resource-intensive for each layer across both the large and small VLMs.

Layer Matching for Distilling Reasoning Progression. To achieve effective layer matching, we develop a strategy to pair layers between the large- and small-backbone VLM, allowing the small model to learn the reasoning progression encoded in the large model. We define the number of target layers in the large and small VLMs as t_l and

VLMs	QBench	AI2D	ChartQA	POPE	HallB	MME	MathVista	MMB	MMB ^{CN}	MM-Vet	MMMU
LLaVA-NeXT-7B [66]	-	-	-	86.5	-	1851	34.6	69.6	63.3	43.9	35.1
LLaVA-NeXT-8B [66]	-	71.6	69.5	-	-	1972	37.5	72.1	-	-	41.7
LLaVA-NeXT-13B [66]	-	70.0	62.2	86.7	-	1892	35.1	70.0	68.5	47.3	35.9
MM1-7B [74]	-	-	-	86.6	-	1858	35.9	72.3	-	42.1	37.0
MM1-MoE-7B×32 [74]	-	-	-	87.8	-	1992	40.9	72.7	-	45.2	40.9
MiniGemini-HD-7B [60]	-	-	-	-	-	1865	32.2	65.8	-	41.3	36.8
MiniGemini-HD-13B [60]	-	-	-	-	-	1917	37.0	68.6	-	50.5	37.3
Cambrian-1-8B [89]	73.0	73.3	-	-	-	49.0	75.9	-	-	-	42.7
Cambrian-1-13B [89]	73.6	73.8	-	-	-	48.0	75.7	-	-	-	40.0
Eagle-8B [82]	76.1	80.1	-	-	-	52.7	75.9	-	-	-	43.8
Eagle-13B [82]	74.0	77.6	-	-	-	54.4	75.7	-	-	-	41.6
VILAI.5-8B [63]	-	-	-	85.6	-	-	75.3	69.9	43.2	-	38.6
VILAI.5-13B [63]	-	-	-	86.3	-	-	74.9	66.3	44.3	-	37.9
VILA ² -8B [27]	-	-	-	86.7	-	-	76.6	71.7	50.0	-	38.3
CogVLM2-8B [38]	-	73.4	81.0	-	-	1870	-	80.5	-	60.4	44.3
LLaVA-OneVision-7B [56]	-	81.4	80.0	-	-	1998	63.2	80.8	-	57.5	48.8
InternVL2-8B [11]	-	<u>83.8</u>	83.3	-	-	2210	58.3	81.7	<u>81.2</u>	54.2	49.3
MiniCPM-V2.5-8B [98]	-	-	-	-	-	2025	54.3	77.2	74.2	-	45.8
MiniCPM-V2.6-8B [98]	-	-	-	-	-	2348	60.6	-	-	60.0	49.8
TroL-7B [50]	73.6	78.5	71.2	87.8	65.3	2308	51.8	<u>83.5</u>	<u>81.2</u>	54.7	49.9
Phantom-7B [49]	<u>73.8</u>	79.5	87.8	87.7	65.4	2126	<u>70.9</u>	84.8	84.7	<u>70.8</u>	51.2
Qwen2-VL-7B [92]	77.5	77.5	83.0	88.9	<u>65.7</u>	2327	58.2	83.0	80.5	62.0	<u>54.1</u>
VLSI-7B	77.5	87.3	<u>86.1</u>	<u>88.6</u>	74.2	<u>2338</u>	74.7	86.3	85.5	75.2	69.3

Table 1. Evaluation of existing open-source VLMs and VLSI on various vision-language benchmarks: QBench [93], AI2D [44], ChartQA [72], POPE [58], HallB [64], MME [30], MathVista [71], MMB [67], MMB^{CN} [67], MM-Vet [100], and MMMU [102]. **Bold** and underline indicate the top and second-best results, respectively.

t_s , respectively, with ranges $i_l = 0^{\text{th}}, 1^{\text{st}}, \dots, (t_l - 1)^{\text{th}}$ and $i_s = 0^{\text{th}}, 1^{\text{st}}, \dots, (t_s - 1)^{\text{th}}$, where $t_s \leq t_l$. Our matching strategy iterates over the t_s layers in the small VLM, ensuring that each layer has a corresponding target layer in the large VLM. Note that, layers in the large-backbone VLM may not have corresponding layers in the small one. To maintain the reasoning progression as layers deepen, we employ a strategy that respects two key criteria: (i) Order Preservation—the matched layer j (large-backbone VLM) of layer i (small-backbone VLM) should be deeper than the matched layer k of layer $i - 1$, ensuring $j > k$; and (ii) Layer-wise Exploration (Multinomial Sampling)—to encourage novel and effective configurations for layer matching, we sample layers based on a distribution that is inversely proportional to the KL divergence between the verbal distributions of the matched layers in the large and small VLMs. Specifically, we compute this sampling distribution using a softmax of the scaled, negative KL divergence values, as summarized in Algorithm 1.

3.3. Supervised Finetuning (SFT) Step

In the final stage of our distillation framework, we finetune the entire small-backbone VLM, including word embeddings, multi-head attention, FFN, and the language head on the visual instruction dataset in a supervised-learning manner. This *SFT step* is inspired by pruning-based distillation methods [77, 85], which require additional training after pruning to counteract potential performance drops from structural changes. While our *interaction step* does not alter the model’s structure, fully absorbing the rich information from the large VLM remains challenging for the small VLM

Algorithm 1 Pseudo-Code for Interaction Loss

```

1: Input:  $t_s, t_l$ 
2: Initialize: loss: 0,  $i_l^*$ : 0,  $\epsilon$ : 1e-6, scale: 2
3: for  $i_s$  in  $0 \leq i_s < t_s$  do
4:    $kld-list = []$ 
5:   for  $i_l$  in  $i_l^* \leq i_l \leq t_l - t_s + i_s$  (Search Range) do
6:      $kld-list.append(compute-kld(i_s, i_l))$ 
7:   end for
8:    $T \leftarrow \frac{\text{scale}}{kld-list.max - kld-list.min + \epsilon}$  {Temperature}
9:    $p\text{-list} \leftarrow \text{Softmax}(-kld-list/T)$ 
10:   $i_l^* \leftarrow 1 + \text{Multinomial}(p\text{-list})$  {Sampling Index}
11:  loss  $\leftarrow$  loss +  $kld-list[i_l^*]$ 
12: end for
13: Return: loss

```

within a single *interaction step*. The SFT step allows the small VLM to better align and integrate the acquired knowledge across layers, enhancing response quality through autoregressive loss. We avoid optimizing autoregressive loss during the *interaction step* (where KL divergence loss is optimized instead) as this can degrade performance, as demonstrated in our ablation study (Sec. 4). Therefore, to improve task-specific instruction-following responsiveness and accuracy, we incorporate the SFT step with autoregressive loss to further finetune the small-backbone VLM.

4. Experiments

4.1. Implementation Details

We present four key technical components of VLSI to ensure reproducibility: (a) the configuration of the backbone vision-language model, (b) the architecture of the ver-

VLMs	QBench	AI2D	ChartQA	POPE	HallB	MME	MathVista	MMB	MMB ^{CN}	MM-Vet	MMMU
MiniCPM-2.4B [41]	-	56.3	-	-	-	1650	28.9	64.1	62.6	31.1	-
MiniCPM-V2-2.8B [41]	-	62.9	-	-	-	1809	38.7	69.1	66.5	41.0	-
MMI-3B [74]	-	-	-	87.4	-	1762	32.0	67.8	-	43.7	33.9
MMI-MoE-3B×64 [74]	-	-	-	87.6	-	1773	32.6	70.8	-	42.2	38.6
ALLaVA-3B [7]	-	-	-	-	-	1623	-	64.0	-	32.2	35.3
VILA1.5-3B [63]	-	-	-	85.3	-	-	-	62.8	52.2	38.6	33.3
InternVL2-4B [11]	-	78.9	81.5	-	-	2064	58.6	78.6	73.9	51.0	34.3
TroL-3.8B [50]	70.0	73.6	73.8	86.5	<u>62.2</u>	1980	55.1	79.2	<u>77.1</u>	51.1	37.5
Phantom-3.8B [49]	70.3	71.7	87.3	87.1	60.8	<u>2046</u>	60.6	<u>80.4</u>	<u>77.1</u>	<u>54.4</u>	39.2
DeepSeek-VL-1.3B [70]	-	-	-	87.6	-	-	31.1	64.6	62.9	34.8	32.2
MobileVLM-1.7B [14]	-	-	-	84.5	-	-	-	53.2	-	-	-
MobileVLM-V2-1.7B [15]	-	-	-	84.3	-	-	-	57.7	-	-	-
MoELLaVA-1.8B×4 [62]	-	-	-	87.0	-	-	-	59.7	-	25.3	-
Mini-Gemini-2B [60]	-	-	-	-	-	1653	29.4	59.8	-	-	31.7
InternVL2-2B [11]	-	<u>74.1</u>	76.2	-	-	1877	46.3	73.2	70.9	39.5	34.3
TroL-1.8B [50]	68.2	68.9	64.0	<u>88.6</u>	60.1	2038	45.4	76.1	74.1	45.1	35.2
Phantom-1.8B [49]	69.1	62.3	<u>87.0</u>	89.6	<u>62.2</u>	1885	<u>60.9</u>	76.6	75.1	54.1	40.6
Qwen2-VL-2B [92]	<u>70.8</u>	60.2	73.5	87.8	61.2	1872	43.0	74.9	73.5	49.5	<u>41.1</u>
 VLSI-2B	72.3	89.0	85.8	87.9	70.0	2022	68.4	81.7	78.8	64.8	51.4

Table 2. Comparison of smaller open-source VLMs and

 VLSI on the same evaluation benchmarks as in Table 1.

balizer, (c) the training and inference configuration, and (d) the structure of the visual-instruction dataset, which includes a variety of capabilities crucial for effectively building  VLSI.

(a) Configuration of the Backbone VLM. We select Qwen2-VL [92] as our backbone VLM due to its flexible model scaling options, which include Qwen2-1.5B, Qwen2-7B, and Qwen2-72B [96]. Importantly, the tokenizer’s vocabulary indices remain consistent across these model sizes, allowing for seamless integration without reordering the vocabulary. This structure enables us to focus on optimizing the LLM component, where Qwen2-1.5B and Qwen2-7B each contain 28 layers, and Qwen2-72B consists of 80 layers. For the vision encoder and projector, we adopt the same modules as those in Qwen2-VL: the vision encoder is a ViT model [21] adapted from DFN [25] and enhanced with visually-adapted rotary positional embeddings [86]. The vision projector comprises an MLP with two fully-connected layers interleaved with GELU activations [37].

(b) Architecture of the Verbalizer. We design the verbalizer as a sequential feed-forward network (FFN), similar to the FFN typically used in transformer blocks [90], and as a language head of the backbone VLM. Conventionally, an FFN consists of three MLPs responsible for dimensional expansion, gating, and reduction. This configuration first expands the hidden dimension, then applies importance weighting to emphasize relevant features, and finally reduces the features back to the original hidden dimension. To enhance computational efficiency in verbalization and interaction, we opt to maintain a consistent hidden dimension throughout the process, foregoing the typical expansion and reduction steps. This streamlined FFN (verb-FFN) design reduces computational complexity while preserving

overall performance. For intermediate target layers, we select i_s : 2nd, 6th, 10th, ..., and 26th layers, and i_t : 2nd, 6th, 10th, ..., and 78th layers.

(c) Training and Inference Configuration. Training and inference for  VLSI are conducted on 8 NVIDIA A100 80GB GPUs. To enable efficient training, we apply LoRA [40] to the LLM component, setting the rank and alpha parameters to 64. We use the AdamW optimizer [69] with a cosine annealing schedule, adjusting the learning rate from $1e^{-4}$ to $1e^{-6}$ over each training step. To handle large batch sizes effectively, we employ gradient accumulation with 16 steps and gradient checkpointing [84] to optimize memory usage. Specifically, batch configurations are four batches each for the 2B and 7B models and two batches for the 72B model, resulting in total gradient update counts of 512 (8×16×4) and 256 (8×16×2), respectively. For inference, we maintain the setup used in Qwen2-VL with a greedy search for text generation.

(d) Visual-instruction dataset. Following the methodology in [49], we compiled a diverse dataset spanning a broad range of vision-language capabilities, totaling 2.9 million visual instruction tuning samples from various sources. This dataset includes foundational image understanding samples sourced from datasets such as ShareGPT4o-Images (57K) [24], ShareGPT4V (755K) [9], ALLaVA-VFLAN/Text (548K) [7], and MiniGemini (27K) [60], which are focused on tasks like DocVQA [73], ChartQA [72], DVQA [42], and AI2D [44]. To support scientific and mathematical reasoning, we incorporated samples from LLaVA-HD (116K) [109], enhancing datasets like ArXivQA [57] and TextbookQA [45]. Additionally, we included document understanding samples from mPLUG-DocOwl1.5-Downstream/Reasoning (599K) [39]. For more

Benchmarks	OmniFusion-7B	DeepSeek-VL-7B	MoVA-7B	Eagle-8B	CoLLaVO-7B	MoAI-7B	Meteor-7B	VLSI-2B	VLSI-7B
MMB [67]	69.0	73.2	81.3	75.9	83.0	79.3	82.9	81.7	86.3
MathVista [71]	-	-	44.3	52.7	57.6	56.2	53.4	68.4	74.7
MM-Vet [100]	39.4	41.5	-	-	40.3	43.7	57.3	64.8	70.8
MMMU [102]	36.6	36.6	-	43.8	42.2	45.6	48.3	51.4	69.3

(a) Validation of open-source VLMs with additional modules and projectors compared to VLSI: OmniFusion [34], DeepSeek-VL [70], MoVA [43], Eagle [82], CoLLaVO [52], MoAI [53], and Meteor [51].

VLMs	MM-Vet	MM-Vet-v2	MMMU	MMStar	AI2D	SEED-2-Plus	MathVista	BLINK	CV-Bench	LLaVA-Wilder
LLaVA-NeXT-34B [66]	50.7	50.9	48.8	51.6	78.9	65.9	40.4	-	-	-
VILA1.5-40B [63]	51.2	-	55.1	55.2	77.8	-	49.5	-	-	-
Cambrian-34B [89]	53.2	-	50.4	54.2	79.5	65.1	50.3	-	76.9	-
Molmo-72B [20]	61.1	-	52.8	63.3	83.4	-	55.8	-	-	-
LLaVA-OV-72B [56]	60.6	-	56.6	65.8	86.2	-	68.4	-	-	72.0
LLaMA-3.2-Vision	64.1	-	60.3	55.3	69.5	68.2	58.3	48.0	-	-
Claude3.5-Sonnet [1]	66.0	71.8	65.9	62.2	80.2	71.7	61.6	28.2	-	83.1
NVLM-D-72B [19]	58.9	-	60.8	63.7	80.1	68.4	63.9	48.0	-	-
GPT-4V (0409) [9]	67.5	66.3	61.7	56.0	78.6	69.3	54.7	58.3	69.1	71.5
Gemini-1.5-Pro	64.0	66.9	60.6	59.1	79.1	70.8	57.7	59.1	-	-
InternVL2-76B [11]	64.4	68.4	58.3	67.1	87.6	70.0	65.6	57.5	-	-
GPT-4o (0806)	75.1	71.0	69.9	64.7	84.7	70.8	62.7	64.7	-	85.9
Qwen2-VL-72B [92]	73.9	68.7	64.3	68.6	88.3	72.3	69.7	60.5	74.3	84.1
TroL-1.8B [50]	45.1	-	35.2	45.5	68.9	-	45.4	-	-	-
TroL-7B [50]	54.7	-	49.9	51.3	78.5	-	51.8	-	-	-
Phantom-1.8B [49]	54.1	46.3	40.6	45.5	62.3	57.1	60.9	44.2	63.1	78.5
Phantom-7B [49]	70.8	60.6	51.2	57.3	79.5	65.5	70.9	58.9	74.9	82.9
VLSI-2B	64.8	60.8	51.4	76.6	89.0	81.1	68.4	52.4	90.1	90.1
VLSI-7B	75.8	70.0	69.3	73.6	87.3	74.9	74.7	59.7	89.1	92.0

(b) Comparison of VLSI with other open-source and closed-source VLMs on challenging benchmarks: MM-Vet [100], MM-Vet-v2 [101], MMMU [102], MMStar [10], AI2D [44], SEED-2-Plus [55], MathVista [71], BLINK [31], CV-Bench [89], and LLaVA-Wilder [56]. This comparison includes models embedding additional knowledge [49, 50] and larger open/closed-source VLMs.

Table 3. Detailed comparison of VLSI with various open and closed-source VLMs on challenging evaluation benchmarks. Appendix A provides detailed descriptions of the evaluation benchmarks listed in Tables 1 and 2.

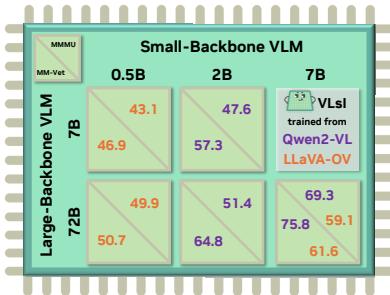


Figure 4. Comparison of performance on MM-Vet [100] and MMMU [102] across different model size combinations in large and small backbone VLMs. Each cell shows the evaluation results for various interaction configurations between 0.5B, 2B, and 7B small backbone VLMs trained with either Qwen2-VL [92] or LLaVA-OV [56] as the large-backbone VLM.

general mathematical tasks, our dataset features samples from GLLaVA (177K) [32], MathVision (3K) [91], Math-Instruct (262K) [103], and MathPlus (304K) [104].

4.2. Verbalization on VLSI

Fig. 3 illustrates the verbal responses generated at each intermediate layer in small-backbone VLM and VLSI. Using the verbalized outputs to trace each layer’s interpretive progression, this comparison highlights how both mod-

els gradually enhance understanding across layers. At the shallower layers, both models generate basic descriptions, focusing on large, simple shapes and colors. However, as

VLSI progresses to mid-level layers, it begins to recognize and articulate more complex visual structures, such as labeled shapes and their relative positions. In contrast, the small-backbone VLM’s verbal responses remain relatively vague or repetitive, often lacking in specific relational details. By the deeper layers, VLSI demonstrates a clear advantage: its verbalizations shift towards identifying the correct pattern, explicitly referring to shapes and colors in alignment with the target response: “star with a dot”. Meanwhile, the small-backbone VLM incorrectly predicts the missing image as a “diamond with a dot”, failing to capture the specific pattern. This example underscores the effectiveness of VLSI’s layer-wise verbalization, where each stage of verbal responses helps the small-backbone VLM align with the larger one. Note that, Appendix B shows the loss changes during verbalization. Additional examples of VLSI’s verbalization are available in Appendix C, highlighting its capacity to interpret layer-wise verbal responses.

4.3. Comparison on Evaluation Benchmarks

As shown in Tab. 1 and Tab. 2, VLSI achieves higher performance while maintaining an efficient model scale.

VLMs	MMB	MM-Vet	MMMU
LLaVA-OV-0.5B	52.1	29.1	31.4
VLSI-0.5B (LLaVA-OV-72B)	72.5	50.7	49.9
LLaVA-OV-7B	80.8	57.5	48.8
VLSI-7B (Qwen2-VL-72B)	86.3	75.8	69.3
VLSI-7B (LLaVA-OV-72B)	86.1	61.6	59.1
LLaVA-OV-72B	85.9	63.7	56.8

(a) Backbone VLMs

VLMs	SFT	MMB	BLINK	MM-Vet	MMMU
Qwen2-VL-2B	-	74.9	41.4	49.5	41.1
VLSI-2B	x	73.2	40.1	47.9	39.8
VLSI-2B (Beam)	x	77.6	48.3	58.1	45.9
VLSI-2B	✓	81.7	52.4	64.8	51.4
Qwen2-VL-7B	-	83.0	50.8	62.0	54.1
VLSI-7B	x	82.1	49.6	60.5	52.9
VLSI-7B (Beam)	x	85.2	56.3	70.9	63.5
VLSI-7B	✓	86.3	59.7	75.8	69.3
Qwen2-VL-72B	-	86.5	60.5	73.9	64.3

(b) Use of SFT Step

RS Training Percent.	MMB	BLINK	MM-Vet	MMMU
Qwen2-VL-7B	80.5	50.8	62.0	54.1
+50%	81.0	51.5	62.8	54.6
+100%	81.8	52.3	63.7	55.2
VLSI-7B	82.1	49.6	60.5	52.9
+50%	85.4	56.0	70.0	62.1
+100%	86.3	59.7	75.8	69.3

(c) Percentage of Training Iterations in RS

IL-Ops	LL-Ops	MMB	BLINK	MM-Vet	MMMU
CE	x	79.2	51.3	64.5	56.5
CE	CE	77.8	50.2	63.2	55.2
CE	KLD	81.0	53.5	67.2	59.0
L2	KLD	81.5	53.2	66.8	58.0
KLD	x	83.0	55.0	69.5	61.0
KLD	CE	81.5	54.3	68.5	59.8
KLD	KLD	86.3	59.7	75.8	69.3
L2	KLD	81.7	53.5	67.0	58.3

(d) Operations for Intermediate/Last Layers

VLMs	MMB	BLINK	MM-Vet	MMMU
Random Index	77.0	50.0	62.0	52.0
Uniform Index	79.5	51.9	66.5	55.3
Bottom-1 Index	81.2	53.8	67.8	57.7
Bottom-3 Index	81.5	54.0	68.0	58.0
Multinomial Sampling	82.0	54.5	68.5	58.5
+Search Range	83.5	55.5	69.8	60.0
+Order Preservation	86.0	59.2	75.2	68.5
+Adaptive Temperature	86.3	59.7	75.8	69.3

(e) Components in Matching Strategy

Structure	Params	MMB	BLINK	MM-Vet	MMMU
Decoder×2	3.3B	85.5	59.1	76.2	68.6
Decoder	1.6B	85.7	59.0	76.0	68.4
FFN×2	2.9B	86.3	59.2	75.9	69.3
FFN	1.4B	86.4	59.4	75.7	69.2
verb-FFN×2	539M	85.8	59.9	75.8	69.3
verb-FFN	269M	86.3	59.7	75.8	69.3
MLP×2	180M	84.2	57.5	74.1	67.0
MLP	90M	83.8	57.0	73.5	66.7

(f) Verbalizer Architecture

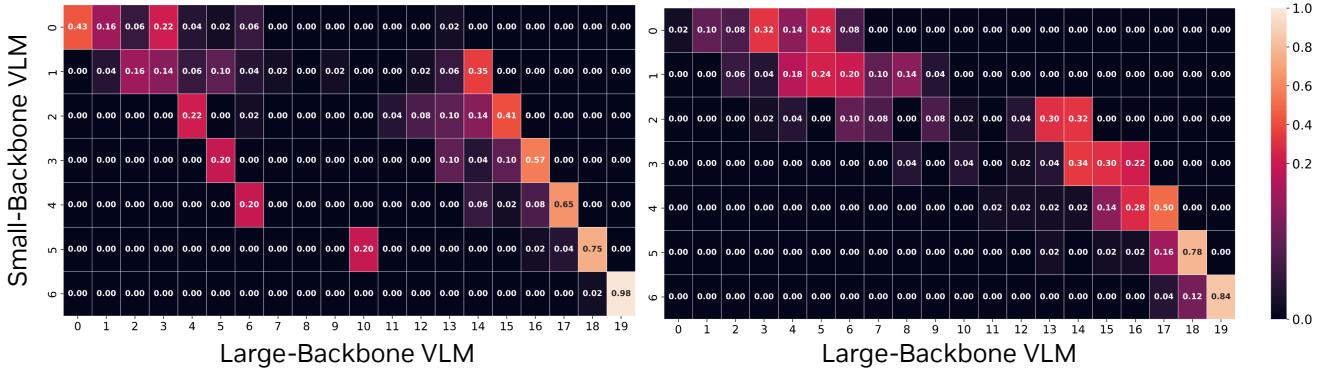
Table 4. Ablation studies examining the six main factors influencing the effectiveness of  VLSI.

Figure 5. Distribution changes of the matched indices between small-backbone and large-backbone VLMs at the interaction step. The left figure shows the distribution at the beginning of training, while the right figure shows it at the end.

Furthermore, Tab. 3 compares  VLSI’s performance with (a) open-source VLMs that incorporate multiple vision encoders, computer vision models, and additional rationale projector; (b) VLMs with modified architectures [49, 50] and various larger open- and proprietary closed-source VLMs. To evaluate where the effectiveness comes from, we analyze six key factors detailed in Fig. 4 and Tab. 4. Appendix D explains the detailed settings for ablation studies. These results indicate that (1) utilizing a more capable large-backbone VLM provides substantial performance benefits, suggesting that the choice of backbone significantly impacts the transfer of knowledge; (2) using larger-backbone VLM gets benefits; (3) KL divergence is more effective during interaction step than cross-entropy, and simultaneously using last-layer distillation boosts performances; and (4) additional SFT step is crucial for further performance gains, consistent with findings in [77, 85]. Appendix E represents  VLSI’s text generation quality.

4.4. Discussion and Limitations

For Fig. 3,  VLSI’s verbalizer is not re-trained but just utilized by the trained verbalizer of the small-backbone VLM. Interestingly, this verbalizer also works well at  VLSI, demonstrating flexibility and indicating that it may

serve as a practical medium of interpretability. Additionally, Fig. 5 illustrates that, as the interaction step progresses, the small-backbone VLM gradually tries to learn about deeper layers’ responses of the large-backbone VLM, which can be considered accelerating the process of reaching an answer. While  VLSI is highly effective, the large and small-backbone VLMs must share the same tokenizer and token index order when constructing VLSI. We will explore more general ways that accommodate different tokenizers and token index orders, potentially expanding  VLSI’s applicability and scalability.

5. Conclusion

We present a new VLM family with 2B and 7B model sizes,  VLSI: Verbalized Layers-to-Interactions, designed to build high-performing yet efficient-scale VLMs. This is accomplished by leveraging natural language-based distillation to transfer knowledge from large to small VLMs. We show  VLSI achieves strong vision-language performances, suggesting that natural language is an important key in transferring knowledge not only for humans but also for AI. We hope to keep progress in further utilization of natural language and larger VLMs.

References

- [1] Anthropic. The claude 3 model family: Opus, sonnet, haiku. <https://www.anthropic.com>, 2024. 1, 7
- [2] Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, Jianxin Ma, Rui Men, Xingzhang Ren, Xuancheng Ren, Chuanqi Tan, Sinan Tan, Jianhong Tu, Peng Wang, Shijie Wang, Wei Wang, Shengguang Wu, Benfeng Xu, Jin Xu, An Yang, Hao Yang, Jian Yang, Shusheng Yang, Yang Yao, Bowen Yu, Hongyi Yuan, Zheng Yuan, Jianwei Zhang, Xingxuan Zhang, Yichang Zhang, Zhenru Zhang, Chang Zhou, Jingen Zhou, Xiaohuan Zhou, and Tianhang Zhu. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023. 2
- [3] Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingen Zhou. Qwen-vl: A frontier large vision-language model with versatile abilities. *arXiv preprint arXiv:2308.12966*, 2023. 2
- [4] Yuxuan Cai, Jiangning Zhang, Haoyang He, Xinwei He, Ao Tong, Zhenye Gan, Chengjie Wang, and Xiang Bai. Llava-kd: A framework of distilling multimodal large language models. *arXiv preprint arXiv:2410.16236*, 2024. 3
- [5] Zheng Cai, Maosong Cao, Haojiong Chen, Kai Chen, Keyu Chen, Xin Chen, Xun Chen, Zehui Chen, Zhi Chen, Pei Chu, et al. Internlm2 technical report. *arXiv preprint arXiv:2403.17297*, 2024. 1
- [6] Jiajun Cao, Yuan Zhang, Tao Huang, Ming Lu, Qizhe Zhang, Ruichuan An, Ningning Ma, and Shanghang Zhang. Move-kd: Knowledge distillation for vlms with mixture of visual encoders. *arXiv preprint arXiv:2501.01709*, 2025. 3
- [7] Guiming Hardy Chen, Shunian Chen, Ruifei Zhang, Junying Chen, Xiangbo Wu, Zhiyi Zhang, Zhihong Chen, Jianquan Li, Xiang Wan, and Benyou Wang. Allava: Harnessing gpt4v-synthesized data for a lite vision-language model. *arXiv preprint arXiv:2402.11684*, 2024. 2, 6
- [8] Keqin Chen, Zhao Zhang, Weili Zeng, Richong Zhang, Feng Zhu, and Rui Zhao. Shikra: Unleashing multi-modal llm's referential dialogue magic. *arXiv preprint arXiv:2306.15195*, 2023. 2
- [9] Lin Chen, Jisong Li, Xiaoyi Dong, Pan Zhang, Conghui He, Jiaqi Wang, Feng Zhao, and Dahua Lin. Sharegpt4v: Improving large multi-modal models with better captions. *arXiv preprint arXiv:2311.12793*, 2023. 2, 6, 7
- [10] Lin Chen, Jisong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Jiaqi Wang, Yu Qiao, Dahua Lin, et al. Are we on the right way for evaluating large vision-language models? *arXiv preprint arXiv:2403.20330*, 2024. 7, 2
- [11] Zhe Chen, Jiannan Wu, Wenhui Wang, Weijie Su, Guo Chen, Sen Xing, Zhong Muyan, Qinglong Zhang, Xizhou Zhu, Lewei Lu, et al. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. *arXiv preprint arXiv:2312.14238*, 2023. 1, 5, 6, 7
- [12] Zhe Chen, Weiyun Wang, Hao Tian, Shenglong Ye, Zhangwei Gao, Erfei Cui, Wenwen Tong, Kongzhi Hu, Jiapeng Luo, Zheng Ma, et al. How far are we to gpt-4v? closing the gap to commercial multimodal models with open-source suites. *arXiv preprint arXiv:2404.16821*, 2024. 2
- [13] Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1290–1299, 2022. 2
- [14] Xiangxiang Chu, Limeng Qiao, Xinyang Lin, Shuang Xu, Yang Yang, Yiming Hu, Fei Wei, Xinyu Zhang, Bo Zhang, Xiaolin Wei, et al. Mobilevlm: A fast, reproducible and strong vision language assistant for mobile devices. *arXiv preprint arXiv:2312.16886*, 2023. 2, 6
- [15] Xiangxiang Chu, Limeng Qiao, Xinyu Zhang, Shuang Xu, Fei Wei, Yang Yang, Xiaofei Sun, Yiming Hu, Xinyang Lin, Bo Zhang, et al. Mobilevlm v2: Faster and stronger baseline for vision language model. *arXiv preprint arXiv:2402.03766*, 2024. 2, 6
- [16] OpenCompass Contributors. Opencompass: A universal evaluation platform for foundation models. <https://github.com/open-compass/opencompass>, 2023. 3
- [17] XTuner Contributors. Xtuner: A toolkit for efficiently fine-tuning llm. <https://github.com/InternLM/xtuner>, 2023. 2
- [18] Wenliang Dai, Junnan Li, Dongxu Li, Anthony Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven Hoi. InstructBLIP: Towards general-purpose vision-language models with instruction tuning. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. 2
- [19] Wenliang Dai, Nayeon Lee, Boxin Wang, Zhuolin Yang, Zihan Liu, Jon Barker, Tuomas Rintamaki, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. Nvlm: Open frontier-class multimodal llms. *arXiv preprint*, 2024. 7
- [20] Matt Deitke, Christopher Clark, Sangho Lee, Rohun Tripathi, Yue Yang, Jae Sung Park, Mohammadreza Salehi, Niklas Muennighoff, Kyle Lo, Luca Soldaini, et al. Molmo and pixmo: Open weights and open data for state-of-the-art multimodal models. *arXiv preprint arXiv:2409.17146*, 2024. 2, 7
- [21] Alexey Dosovitskiy. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 6
- [22] Yuning Du, Chenxia Li, Ruoyu Guo, Cheng Cui, Weiwei Liu, Jun Zhou, Bin Lu, Yehua Yang, Qiwen Liu, Xiaoguang Hu, et al. Pp-ocrv2: Bag of tricks for ultra lightweight ocr system. *arXiv preprint arXiv:2109.03144*, 2021. 2
- [23] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024. 1
- [24] et al. Erfei Cui. Sharegpt-4o: Comprehensive multimodal annotations with gpt-4o, 2024. <https://sharegpt4o.github.io/>. 6
- [25] Alex Fang, Albin Madappally Jose, Amit Jain, Ludwig Schmidt, Alexander Toshev, and Vaishaal Shankar. Data

- filtering networks. *arXiv preprint arXiv:2309.17425*, 2023. 6
- [26] Yuxin Fang, Wen Wang, Binhui Xie, Quan Sun, Ledell Wu, Xinggang Wang, Tiejun Huang, Xinlong Wang, and Yue Cao. Eva: Exploring the limits of masked visual representation learning at scale. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19358–19369, 2023. 2
- [27] Yunhao Fang, Ligeng Zhu, Yao Lu, Yan Wang, Pavlo Molchanov, Jang Hyun Cho, Marco Pavone, Song Han, and Hongxu Yin. Vila²: Vila augmented vila. *arXiv preprint arXiv:2407.17453*, 2024. 2, 5
- [28] Qianhan Feng, Wenshuo Li, Tong Lin, and Xinghao Chen. Align-kd: Distilling cross-modal alignment knowledge for mobile vision-language model. *arXiv preprint arXiv:2412.01282*, 2024. 3
- [29] Markus Freitag and Yaser Al-Onaizan. Beam search strategies for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 56–60, Vancouver, 2017. Association for Computational Linguistics. 4
- [30] Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, et al. Mme: A comprehensive evaluation benchmark for multimodal large language models. *arXiv preprint arXiv:2306.13394*, 2023. 5, 1
- [31] Xingyu Fu, Yushi Hu, Bangzheng Li, Yu Feng, Haoyu Wang, Xudong Lin, Dan Roth, Noah A Smith, Wei-Chiu Ma, and Ranjay Krishna. Blink: Multimodal large language models can see but not perceive. *arXiv preprint arXiv:2404.12390*, 2024. 7, 2
- [32] Jiahui Gao, Renjie Pi, Jipeng Zhang, Jiacheng Ye, Wan-jun Zhong, Yufei Wang, Lanqing Hong, Jianhua Han, Hang Xu, Zhenguo Li, et al. G-llava: Solving geometric problem with multi-modal large language model. *arXiv preprint arXiv:2312.11370*, 2023. 2, 7
- [33] Chunjiang Ge, Sijie Cheng, Ziming Wang, Jiale Yuan, Yuan Gao, Jun Song, Shiji Song, Gao Huang, and Bo Zheng. Conyllava: Hierarchical backbones as visual encoder for large multimodal models. *arXiv preprint arXiv:2405.15738*, 2024. 2
- [34] Elizaveta Goncharova, Anton Razhigaev, Matvey Mikhalkchuk, Maxim Kurkin, Irina Abdullaeva, Matvey Skripkin, Ivan Oseledets, Denis Dimitrov, and Andrey Kuznetsov. Omnidfusion technical report. *arXiv preprint arXiv:2404.06212*, 2024. 2, 7
- [35] Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023. 2
- [36] Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. Minilm: Knowledge distillation of large language models. *arXiv preprint arXiv:2306.08543*, 2023. 3
- [37] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016. 6
- [38] Wenyi Hong, Weihan Wang, Ming Ding, Wenmeng Yu, Qingsong Lv, Yan Wang, Yean Cheng, Shiyu Huang, Jun-hui Ji, Zhao Xue, et al. Cogvlm2: Visual language mod- els for image and video understanding. *arXiv preprint arXiv:2408.16500*, 2024. 5
- [39] Anwen Hu, Haiyang Xu, Jiabo Ye, Ming Yan, Liang Zhang, Bo Zhang, Chen Li, Ji Zhang, Qin Jin, Fei Huang, et al. mplug-docowl 1.5: Unified structure learning for ocr-free document understanding. *arXiv preprint arXiv:2403.12895*, 2024. 6
- [40] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021. 3, 6, 7
- [41] Shengding Hu, Yuge Tu, Xu Han, Chaoqun He, Ganqu Cui, Xiang Long, Zhi Zheng, Yewei Fang, Yuxiang Huang, Weilin Zhao, et al. Minicpm: Unveiling the potential of small language models with scalable training strategies. *arXiv preprint arXiv:2404.06395*, 2024. 6
- [42] Kushal Kafle, Brian Price, Scott Cohen, and Christopher Kanan. Dvqa: Understanding data visualizations via question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5648–5656, 2018. 6
- [43] Oğuzhan Fatih Kar, Alessio Tonioni, Petra Poklukar, Achin Kulshrestha, Amir Zamir, and Federico Tombari. Brave: Broadening the visual encoding of vision-language models. *arXiv preprint arXiv:2404.07204*, 2024. 7
- [44] Aniruddha Kembhavi, Mike Salvato, Eric Kolve, Minjoon Seo, Hannaneh Hajishirzi, and Ali Farhadi. A diagram is worth a dozen images. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV I4*, pages 235–251. Springer, 2016. 5, 6, 7, 1
- [45] Aniruddha Kembhavi, Minjoon Seo, Dustin Schwenk, Jonghyun Choi, Ali Farhadi, and Hannaneh Hajishirzi. Are you smarter than a sixth grader? textbook question answering for multimodal machine comprehension. In *Proceedings of the IEEE Conference on Computer Vision and Pattern recognition*, pages 4999–5007, 2017. 6
- [46] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4015–4026, 2023. 2
- [47] Jongwoo Ko, Sungnyun Kim, Tianyi Chen, and Se-Young Yun. Distillm: Towards streamlined distillation for large language models. *arXiv preprint arXiv:2402.03898*, 2024. 3
- [48] Hugo Laurençon, Lucile Saulnier, Léo Tronchon, Stas Bekman, Amanpreet Singh, Anton Lozhkov, Thomas Wang, Siddharth Karamcheti, Alexander M Rush, Douwe Kiela, et al. Obelisc: An open web-scale filtered dataset of interleaved image-text documents. *arXiv preprint arXiv:2306.16527*, 2023. 2
- [49] Byung-Kwan Lee, Sangyun Chung, Chae Won Kim, Beomchan Park, and Yong Man Ro. Phantom of latent for large language and vision models. *arXiv preprint arXiv:2409.14713*, 2024. 2, 3, 5, 6, 7, 8

- [50] Byung-Kwan Lee, Sangyun Chung, Chae Won Kim, Beomchan Park, and Yong Man Ro. Trol: Traversal of layers for large language and vision models. *arXiv preprint arXiv:2406.12246*, 2024. 3, 5, 6, 7, 8
- [51] Byung-Kwan Lee, Chae Won Kim, Beomchan Park, and Yong Man Ro. Meteor: Mamba-based traversal of rationale for large language and vision models. *arXiv preprint arXiv:2405.15574*, 2024. 2, 7
- [52] Byung-Kwan Lee, Beomchan Park, Chae Won Kim, and Yong Man Ro. Collavo: Crayon large language and vision model. *arXiv preprint arXiv:2402.11248*, 2024. 2, 7
- [53] Byung-Kwan Lee, Beomchan Park, Chae Won Kim, and Yong Man Ro. Moai: Mixture of all intelligence for large language and vision models. *arXiv preprint arXiv:2403.07508*, 2024. 2, 7
- [54] Bo Li, Yuanhan Zhang, Liangyu Chen, Jinghao Wang, Jingkang Yang, and Ziwei Liu. Otter: A multi-modal model with in-context instruction tuning. *arXiv preprint arXiv:2305.03726*, 2023. 2
- [55] Bohao Li, Yuying Ge, Yi Chen, Yixiao Ge, Ruimao Zhang, and Ying Shan. Seed-bench-2-plus: Benchmarking multi-modal large language models with text-rich visual comprehension. *arXiv preprint arXiv:2404.16790*, 2024. 7, 1
- [56] Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Yanwei Li, Ziwei Liu, and Chunyuan Li. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*, 2024. 2, 5, 7
- [57] Lei Li, Yuqi Wang, Runxin Xu, Peiyi Wang, Xiachong Feng, Lingpeng Kong, and Qi Liu. Multimodal arxiv: A dataset for improving scientific comprehension of large vision-language models. *arXiv preprint arXiv:2403.00231*, 2024. 6
- [58] Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne Xin Zhao, and Ji-Rong Wen. Evaluating object hallucination in large vision-language models. *arXiv preprint arXiv:2305.10355*, 2023. 5, 1
- [59] Yuhui Li, Fangyun Wei, Chao Zhang, and Hongyang Zhang. Eagle: Speculative sampling requires rethinking feature uncertainty. *arXiv preprint arXiv:2401.15077*, 2024. 3
- [60] Yanwei Li, Yuechen Zhang, Chengyao Wang, Zhisheng Zhong, Yixin Chen, Ruihang Chu, Shaoteng Liu, and Jiaya Jia. Mini-gemini: Mining the potential of multi-modality vision language models. *arXiv preprint arXiv:2403.18814*, 2024. 2, 5, 6
- [61] Chen Liang, Simiao Zuo, Qingru Zhang, Pengcheng He, Weizhu Chen, and Tuo Zhao. Less is more: Task-aware layer-wise distillation for language model compression. In *International Conference on Machine Learning*, pages 20852–20867. PMLR, 2023. 3
- [62] Bin Lin, Zhenyu Tang, Yang Ye, Jiaxi Cui, Bin Zhu, Peng Jin, Junwu Zhang, Munan Ning, and Li Yuan. Moe-llava: Mixture of experts for large vision-language models. *arXiv preprint arXiv:2401.15947*, 2024. 2, 6
- [63] Ji Lin, Hongxu Yin, Wei Ping, Yao Lu, Pavlo Molchanov, Andrew Tao, Huizi Mao, Jan Kautz, Mohammad Shoeybi, and Song Han. Vila: On pre-training for visual language models. *arXiv preprint arXiv:2312.07533*, 2023. 5, 6, 7
- [64] Fuxiao Liu, Tianrui Guan, Zongxia Li, Lichang Chen, Yaser Yacoob, Dinesh Manocha, and Tianyi Zhou. Hallusionbench: You see what you think? or you think what you see? an image-context reasoning benchmark challenging for gpt-4v (ision), llava-1.5, and other multi-modality models. *arXiv preprint arXiv:2310.14566*, 2023. 5, 1
- [65] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. 2
- [66] Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. Llava-next: Improved reasoning, ocr, and world knowledge, 2024. 2, 5, 7
- [67] Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, et al. Mmbench: Is your multi-modal model an all-around player? *arXiv preprint arXiv:2307.06281*, 2023. 5, 7, 1
- [68] Zechun Liu, Changsheng Zhao, Forrest Iandola, Chen Lai, Yuandong Tian, Igor Fedorov, Yunyang Xiong, Ernie Chang, Yangyang Shi, Raghuraman Krishnamoorthi, et al. Mobilellm: Optimizing sub-billion parameter language models for on-device use cases. *arXiv preprint arXiv:2402.14905*, 2024. 2
- [69] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. 6, 7
- [70] Haoyu Lu, Wen Liu, Bo Zhang, Bingxuan Wang, Kai Dong, Bo Liu, Jingxiang Sun, Tongzheng Ren, Zhuoshu Li, Yaofeng Sun, et al. Deepseek-vl: towards real-world vision-language understanding. *arXiv preprint arXiv:2403.05525*, 2024. 2, 6, 7
- [71] Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts. *arXiv preprint arXiv:2310.02255*, 2023. 5, 7, 1
- [72] Ahmed Masry, Do Xuan Long, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. Chartqa: A benchmark for question answering about charts with visual and logical reasoning. *arXiv preprint arXiv:2203.10244*, 2022. 5, 6, 1
- [73] Minesh Mathew, Dimosthenis Karatzas, and CV Jawahar. Docvqa: A dataset for vqa on document images. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 2200–2209, 2021. 6
- [74] Brandon McKinzie, Zhe Gan, Jean-Philippe Fauconnier, Sam Dodge, Bowen Zhang, Philipp Dufter, Dhruti Shah, Xianzhi Du, Futang Peng, Floris Weers, et al. Mm1: Methods, analysis & insights from multimodal llm pre-training. *arXiv preprint arXiv:2403.09611*, 2024. 2, 5, 6
- [75] Sachin Mehta, Mohammad Hosseini Sekhavat, Qingqing Cao, Maxwell Horton, Yanzi Jin, Chenfan Sun, Iman Mirzadeh, Mahyar Najibi, Dmitry Belenko, Peter Zatoulakal, et al. Openelm: An efficient language model family with open-source training and inference framework. *arXiv preprint arXiv:2404.14619*, 2024. 2
- [76] Matthias Minderer, Alexey A. Gritsenko, and Neil Houlsby. Scaling open-vocabulary object detection. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. 2

- Conference on Neural Information Processing Systems*, 2023. 2
- [77] Saurav Muralidharan, Sharath Turuvekere Sreenivas, Raviraj Joshi, Marcin Chochowski, Mostafa Patwary, Mohammad Shoeybi, Bryan Catanzaro, Jan Kautz, and Pavlo Molchanov. Compact language models via pruning and knowledge distillation. *arXiv preprint arXiv:2407.14679*, 2024. 3, 5, 8
- [78] OpenAI. Gpt-4v(ision) system card, 2023. <https://openai.com/research/gpt-4v-system-card>, Last accessed on 2024-02-13. 1, 2
- [79] OpenAI. Gpt-4v(ision) technical work and authors, 2023. <https://openai.com/contributions/gpt-4v>, Last accessed on 2024-02-13. 1
- [80] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. 2
- [81] Yanyuan Qiao, Zheng Yu, Longteng Guo, Sihan Chen, Zijia Zhao, Mingzhen Sun, Qi Wu, and Jing Liu. Vi-mamba: Exploring state space models for multimodal learning. *arXiv preprint arXiv:2403.13600*, 2024. 2
- [82] Min Shi, Fuxiao Liu, Shihao Wang, Shijia Liao, Subhashree Radhakrishnan, De-An Huang, Hongxu Yin, Karan Sapra, Yaser Yacoob, Humphrey Shi, et al. Eagle: Exploring the design space for multimodal llms with mixture of encoders. *arXiv preprint arXiv:2408.15998*, 2024. 2, 5, 7
- [83] Fangxun Shu, Yue Liao, Le Zhuo, Chenning Xu, Guanghao Zhang, Haonan Shi, Long Chen, Tao Zhong, Wanggui He, Siming Fu, et al. Llava-mod: Making llava tiny via moe knowledge distillation. *arXiv preprint arXiv:2408.15881*, 2024. 3
- [84] Nimit S Sohoni, Christopher R Aberger, Megan Leszczynski, Jian Zhang, and Christopher Ré. Low-memory neural network training: A technical report. *arXiv preprint arXiv:1904.10631*, 2019. 6, 7
- [85] Sharath Turuvekere Sreenivas, Saurav Muralidharan, Raviraj Joshi, Marcin Chochowski, Mostafa Patwary, Mohammad Shoeybi, Bryan Catanzaro, Jan Kautz, and Pavlo Molchanov. Llm pruning and distillation in practice: The minitron approach. *arXiv preprint arXiv:2408.11796*, 2024. 3, 5, 8
- [86] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568: 127063, 2024. 6
- [87] Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023. 1, 2
- [88] Omkar Thawakar, Ashmal Vayani, Salman Khan, Hisham Cholakal, Rao M. Anwer, Michael Felsberg, Tim Baldwin, Eric P. Xing, and Fahad Shahbaz Khan. Mobillama: Towards accurate and lightweight fully transparent gpt, 2024. 2
- [89] Shengbang Tong, Ellis Brown, Penghao Wu, Sanghyun Woo, Manoj Middepogu, Sai Charitha Akula, Jihan Yang, Shusheng Yang, Adithya Iyer, Xichen Pan, et al. Cambrian-1: A fully open, vision-centric exploration of multimodal llms. *arXiv preprint arXiv:2406.16860*, 2024. 2, 5, 7
- [90] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 3, 6
- [91] Ke Wang, Junting Pan, Weikang Shi, Zimu Lu, Mingjie Zhan, and Hongsheng Li. Measuring multimodal mathematical reasoning with math-vision dataset. *arXiv preprint arXiv:2402.14804*, 2024. 2, 7
- [92] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Yang Fan, Kai Dang, Mengfei Du, Xuancheng Ren, Rui Men, Dayiheng Liu, Chang Zhou, Jingren Zhou, and Junyang Lin. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution, 2024. 2, 3, 5, 6, 7
- [93] Haoning Wu, Zicheng Zhang, Erli Zhang, Chaofeng Chen, Liang Liao, Annan Wang, Chunyi Li, Wenxiu Sun, Qiong Yan, Guangtao Zhai, et al. Q-bench: A benchmark for general-purpose foundation models on low-level vision. *arXiv preprint arXiv:2309.14181*, 2023. 5, 1
- [94] Ruyi Xu, Yuan Yao, Zonghao Guo, Junbo Cui, Zanlin Ni, Chunjiang Ge, Tat-Seng Chua, Zhiyuan Liu, Maosong Sun, and Gao Huang. Llava-uhd: an lmm perceiving any aspect ratio and high-resolution images. *arXiv preprint arXiv:2403.11703*, 2024. 2
- [95] Shilin Xu, Xiangtai Li, Haobo Yuan, Lu Qi, Yunhai Tong, and Ming-Hsuan Yang. Llavadi: What matters for multimodal large language models distillation. *arXiv preprint arXiv:2407.19409*, 2024. 3
- [96] An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, et al. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*, 2024. 1, 6, 7
- [97] Jingkang Yang, Yi Zhe Ang, Zujin Guo, Kaiyang Zhou, Wayne Zhang, and Ziwei Liu. Panoptic scene graph generation. In *European Conference on Computer Vision*, pages 178–196. Springer, 2022. 2
- [98] Yuan Yao, Tianyu Yu, Ao Zhang, Chongyi Wang, Junbo Cui, Hongji Zhu, Tianchi Cai, Haoyu Li, Weilin Zhao, Zhihui He, et al. Minicpm-v: A gpt-4v level mllm on your phone. *arXiv preprint arXiv:2408.01800*, 2024. 5
- [99] Qinghao Ye, Haiyang Xu, Guohai Xu, Jiabo Ye, Ming Yan, Yiyang Zhou, Junyang Wang, Anwen Hu, Pengcheng Shi, Yaya Shi, et al. Mplug-owl: Modularization empowers large language models with multimodality. *arXiv preprint arXiv:2304.14178*, 2023. 2
- [100] Weihao Yu, Zhengyuan Yang, Linjie Li, Jianfeng Wang, Kevin Lin, Zicheng Liu, Xinchao Wang, and Lijuan Wang. Mm-vet: Evaluating large multimodal models for integrated capabilities. *arXiv preprint arXiv:2308.02490*, 2023. 1, 2, 5, 7
- [101] Weihao Yu, Zhengyuan Yang, Linfeng Ren, Linjie Li, Jianfeng Wang, Kevin Lin, Chung-Ching Lin, Zicheng Liu, Li-

- juan Wang, and Xinchao Wang. Mm-vet v2: A challenging benchmark to evaluate large multimodal models for integrated capabilities. *arXiv preprint arXiv:2408.00765*, 2024. [7](#), [2](#)
- [102] Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, et al. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. *arXiv preprint arXiv:2311.16502*, 2023. [2](#), [5](#), [7](#)
- [103] Xiang Yue, Xingwei Qu, Ge Zhang, Yao Fu, Wenhao Huang, Huan Sun, Yu Su, and Wenhua Chen. Mammoth: Building math generalist models through hybrid instruction tuning. *arXiv preprint arXiv:2309.05653*, 2023. [7](#)
- [104] Xiang Yue, Tuney Zheng, Ge Zhang, and Wenhua Chen. Mammoth2: Scaling instructions from the web. 2024. [7](#)
- [105] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11975–11986, 2023. [1](#), [2](#)
- [106] Kaichen Zhang, Bo Li, Peiyuan Zhang, Fanyi Pu, Joshua Adrian Cahyono, Kairui Hu, Shuai Liu, Yuanhan Zhang, Jingkang Yang, Chunyuan Li, et al. Lmms-eval: Reality check on the evaluation of large multimodal models. *arXiv preprint arXiv:2407.12772*, 2024. [2](#)
- [107] Pan Zhang, Xiaoyi Dong Bin Wang, Yuhang Cao, Chao Xu, Linke Ouyang, Zhiyuan Zhao, Shuangrui Ding, Songyang Zhang, Haodong Duan, Hang Yan, et al. Internlm-xcomposer: A vision-language large model for advanced text-image comprehension and composition. *arXiv preprint arXiv:2309.15112*, 2023. [2](#)
- [108] Peiyuan Zhang, Guangtao Zeng, Tianduo Wang, and Wei Lu. Tinyllama: An open-source small language model. *arXiv preprint arXiv:2401.02385*, 2024. [2](#)
- [109] Yi-Fan Zhang, Qingsong Wen, Chaoyou Fu, Xue Wang, Zhang Zhang, Liang Wang, and Rong Jin. Beyond llava-hd: Diving into high-resolution large multimodal models. *arXiv preprint arXiv:2406.08487*, 2024. [2](#), [6](#)
- [110] Han Zhao, Min Zhang, Wei Zhao, Pengxiang Ding, Siteng Huang, and Donglin Wang. Cobra: Extending mamba to multi-modal large language model for efficient inference. *arXiv preprint arXiv:2403.14520*, 2024. [2](#)
- [111] Baichuan Zhou, Ying Hu, Xi Weng, Junlong Jia, Jie Luo, Xien Liu, Ji Wu, and Lei Huang. Tinyllava: A framework of small-scale large multimodal models. *arXiv preprint arXiv:2402.14289*, 2024. [2](#)
- [112] Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*, 2023. [2](#)
- [113] Zhuofan Zong, Bingqi Ma, Dazhong Shen, Guanglu Song, Hao Shao, Dongzhi Jiang, Hongsheng Li, and Yu Liu. Mova: Adapting mixture of vision experts to multimodal context. *arXiv preprint arXiv:2404.13046*, 2024. [2](#)

A. Description of Numeorus Evaluation Benchmarks

- **QBench** [93] is a comprehensive benchmark designed to evaluate the low-level visual abilities of multimodal large language models (MLLMs), focusing on perception, description, and quality assessment of visual attributes. It introduces datasets like LLVisionQA for diverse low-level attribute queries, LLDescribe for detailed expert-crafted image descriptions, and a unified softmax-based strategy for quantifiable image quality assessment. Q-Bench highlights that while MLLMs exhibit preliminary capabilities in handling low-level visual tasks, their outputs remain inconsistent and imprecise, emphasizing the need for further advancements to align with human perception and achieve general-purpose applications.
- **AI2D** [44] is a benchmark dataset developed to study diagram interpretation and reasoning, focusing on identifying diagram structures and semantic relationships. It introduces Diagram Parse Graphs (DPG), a graph-based representation that encodes the syntactic and semantic structure of diagrams. The dataset contains over 5,000 grade-school science diagrams with exhaustive annotations of constituents and relationships, as well as 15,000 multiple-choice questions for diagram-based reasoning tasks.
- **ChartQA** [72] is a large-scale benchmark designed to assess question-answering systems' ability to reason logically and visually about data visualizations like bar, line, and pie charts. It includes 9,608 human-authored questions and 23,111 machine-generated questions, focusing on complex reasoning tasks involving mathematical operations, visual attributes, and multi-step logical inferences. By utilizing both extracted data tables and visual features, the benchmark highlights challenges in handling real-world charts and emphasizes the gap in models' ability to process intricate visual and logical questions compared to human understanding.
- **SEED-Bench-2-Plus** [55] is a comprehensive benchmark designed to evaluate Multimodal Large Language Models (MLLMs) on their ability to comprehend and reason about text-rich visual content across three categories: Charts, Maps, and Webs, covering 63 diverse data types. It includes 2.3K meticulously crafted multiple-choice questions with human-verified answers, simulating real-world scenarios that combine complex text and visual data.
- **POPE** [58] is a polling-based evaluation framework designed to assess object hallucination in Large Vision-Language Models (LVLMs). It formulates hallucination detection as a binary classification task using simple yes/no questions (e.g., “Is there a chair in the image?”) to probe LVLMs. Unlike previous methods, POPE offers a stable and flexible approach by avoiding dependence on lengthy captions or complex parsing rules. It introduces three object sampling strategies—Random, Popular, and Adversarial—to explore hallucination patterns in models.
- **HallusionBench (HallIB)** [64] is an advanced diagnostic benchmark designed to evaluate and analyze the failure modes of Large Vision-Language Models (LVLMs) in handling both language hallucinations and visual illusions. Featuring 346 images and 1,129 human-crafted visual-question pairs, it tests models like GPT-4V and LLaVA-1.5 using unique control pairs and human-edited images to assess logical consistency, perception, and reasoning. Results highlight persistent challenges, with top models achieving only 31.42% accuracy, revealing their over-reliance on parametric memory, susceptibility to simple manipulations, and struggles with geometry, math, and temporal reasoning.
- **MME** [30] is a comprehensive benchmark designed to evaluate Multimodal Large Language Models (MLLMs) across perception and cognition abilities with 14 subtasks. The benchmark includes tasks like object recognition, OCR, common-sense reasoning, and numerical calculation, using manually curated instruction-answer pairs to ensure fairness and avoid data leakage. MME emphasizes concise instructions for consistency and quantitative assessment, highlighting that current MLLMs, despite their progress, face challenges such as instruction-following errors, limited perception and reasoning, and hallucinations.
- **MathVista** [71] is a benchmark designed to evaluate the mathematical reasoning abilities of foundation models in visual contexts. It comprises 6,141 examples sourced from 31 datasets, including three newly created datasets—IQTest, FunctionQA, and PaperQA—tailored to assess logical, algebraic, and scientific reasoning in visual settings. MathVista emphasizes diverse visual contexts, such as diagrams, charts, and academic figures, and covers seven types of reasoning across five core tasks.
- **MMB, MMB-Chinese (MMB^{CN})** [67] is a multilingual benchmark designed to evaluate the multimodal capabilities of vision-language models (VLMs) across 20 fine-grained abilities, including perception, reasoning, and relation understanding. It features over 3,000 high-quality multiple-choice questions in English and Chinese, enabling comparative analysis in a bilingual context. MMBench introduces novel evaluation strategies like CircularEval, which enhances robustness by testing models across shuffled choices, and employs GPT-4 for accurate choice extraction.
- **MM-Vet** [100] is a benchmark designed to evaluate the integrated vision-language capabilities of Large Multimodal Models (LMMs). It defines six core capabilities—recognition, OCR, knowledge, language generation, spatial awareness, and math—and examines their combinations across 16 emergent multimodal tasks, such as explaining memes, solving spatial math problems, and summarizing visual data. The benchmark introduces an LLM-based evaluator to assess open-ended

model outputs consistently, focusing on both accuracy and quality.

- **MM-Vet-v2** [101] builds upon the original MM-Vet benchmark by introducing a new core capability, image-text sequence understanding, to evaluate large multimodal models (LMMs) on processing arbitrarily interleaved sequences of images and text. With an expanded dataset of 517 high-quality evaluation samples and tasks requiring combinations of seven core capabilities, it assesses advanced real-world scenarios like temporal reasoning, spatial understanding, and multimodal comparisons.
- **MMMU** [102] is a benchmark designed to evaluate large multimodal models on 11550 college-level problems requiring expert knowledge and reasoning across six disciplines: Art, Business, Science, Medicine, Humanities, and Engineering. Spanning 30 subjects and incorporating 30 diverse image types like charts, medical scans, and diagrams, it challenges models to integrate complex text and image inputs while applying domain-specific knowledge. MMMU sets a high standard for advancing multimodal AI and plays a crucial role in developing Expert AGI.
- **MMStar** [10] is a vision-critical multimodal benchmark consisting of 1,500 meticulously curated samples designed to evaluate large vision-language models (LVLMs) across six core capabilities and 18 specific axes. By addressing two key issues in existing benchmarks—unnecessary reliance on textual knowledge and unintentional data leakage—MMStar ensures each sample requires genuine visual reasoning and minimal data recall. Incorporating metrics for multimodal gain and data leakage, it provides a robust platform for assessing the true multimodal capacities of LVLMs.
- **BLINK** [31] is a benchmark designed to evaluate the core visual perception abilities of multimodal large language models (MLLMs) across 14 tasks, such as depth estimation, visual correspondence, and spatial reasoning, inspired by classical computer vision problems. With 3,807 multiple-choice questions derived from 7,300 images, BLINK focuses on tasks that humans can solve “within a blink” but remain challenging for models, as even advanced models like GPT-4V achieve only 51.26% accuracy compared to 95.7% for humans. It highlights the gap in nuanced visual perception and suggests integrating specialized vision models as a pathway for improving MLLMs’ performance.
- **CV-Bench** [89] is a vision-centric benchmark introduced to evaluate the fundamental 2D and 3D visual understanding capabilities of Multimodal Large Language Models (MLLMs). With 2,638 manually inspected examples sourced from datasets like ADE20K, COCO, and Omni3D, it tests tasks such as spatial relationships, object counting, depth ordering, and relative distances. By transforming traditional vision benchmarks into VQA format, CV-Bench ensures robust assessment of models’ abilities in multimodal contexts. It addresses gaps in existing benchmarks by offering significantly more samples, better diversity, and a stronger focus on visual grounding, making it a critical tool for advancing multimodal AI systems.
- **LLaVA-Wilder** [106] is a dataset designed to evaluate large multimodal models (LMMs) in real-world scenarios. It comprises 128 image-text pairs, each featuring an image accompanied by a question and a detailed answer. The dataset includes a variety of images, such as indoor and outdoor scenes, memes, paintings, and sketches, to assess models’ generalization capabilities across diverse domains. By providing this resource, LLaVA-Bench-Wilder aims to facilitate the development and benchmarking of LMMs, ensuring their robustness and effectiveness in handling complex, real-world visual tasks.

B. Verbalization Loss Changes

Target Layer Index	10%	30%	50%	70%	90%	100%
#1	14.28	11.73	9.87	6.78	3.54	3.30
#2	12.60	10.48	8.04	5.98	3.10	2.90
#3	11.22	9.32	7.02	5.05	2.63	2.37
#4	10.48	8.11	6.41	4.57	2.35	2.18
#5	9.34	7.55	5.81	3.54	1.82	1.79
#6	8.16	6.18	4.27	2.09	1.56	1.47
#7	7.39	3.85	2.02	1.03	0.85	0.79

Showing loss changes during the verbalization step (% indicates training progress)

C. VLsI's Verbalization Examples



VLsI

Question: What is the atmosphere and setting depicted in the image, and how do the elements contribute to creating a relaxing environment?

#1 Target Layer (Actual 2nd Layer)
The image displays a moment of a large, white, and white color. (..Repeats..)

#2 Target Layer (Actual 6th Layer)
The image depicts a vibrant and vibrant scene of a young woman with a large, dark blue color scheme. The color palette is bright and vibrant, with a vibrant color scheme that suggests a more likely setting. The color palette is a common feature of the image, with the visible signs of the person's attire and the visible signs of the other person's attire. The color palette is a common feature of the image, with the visible elements of the image itself.

#3 Target Layer (Actual 10th Layer)
The image depicts a young man with a red and white design on his shirt. The setting is a room with a large window, a small, small blue and white design on the top of the pizza. The setting suggests a dining area, possibly a dining area or a dining area with a small dining table. The setting suggests a dining area with a large window, a small blue and right side of the image, and a small greenery, which suggests a natural setting.

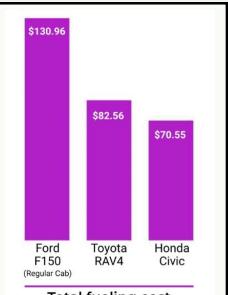
#4 Target Layer (Actual 14th Layer)
The image depicts a young woman in a room with a large, large window and a large window with a window on the right. The setting is a large, large room with a large window and a small window on the right. The setting is a large, greenery-lit interior space, which is a common theme in the image. The color palette is soft, with a soft light and a light color palette. The color scheme is a mix of green and greenish hues, which are often associated with the natural and natural elements of the setting. The setting is a large, greenery-chashy, and even the elements are present in the image.

#5 Target Layer (Actual 18th Layer)
The image depicts a scene of a young man in a white shirt and black shorts, wearing a black and white patterned shirt and a black top, wearing a red and red top. The person is wearing a red and black top and a black and white patterned shirt. The setting is a large, open space with a large, open space. The color palette is rich and vibrant, with a mix of hues that are rich and vibrant. The color scheme is vibrant and vibrant, with a mix of colors that are more than a number. The color scheme is more than a color, but it is not a color. (..Repeats..)

#6 Target Layer (Actual 22nd Layer)
The image portrays a serene and cozy setting, likely a home or a cozy room, with a person lying on a couch, engaging in a relaxed activity. The setting is characterized by a warm color palette, with warm, warm tones that contribute to a cozy and inviting atmosphere. The presence of a large window with a window adorned with a plant adds a touch of nature, enhancing the tranquility of the setting. The presence of a bookshelf with books and a collection of art pieces on the walls adds a personal and personalized touch, suggesting a space that is personal and inviting. The use of soft, soft furnishings, such as a plush couch and scattered pillows, contributes to the cozy and cozy atmosphere. The use of soft, soft furnishings, such as a plush couch and scattered pillows, contributes to the cozy and cozy atmosphere.

#7 Target Layer (Actual 26th Layer)
The image portrays a serene and cozy setting, likely designed to evoke a sense of relaxation and tranquility. The main subject is a young woman seated cross-legged on the floor, absorbed in music through headphones. This position suggests a relaxed and leisurely atmosphere, as she is not

The image portrays a serene and cozy setting, likely designed to evoke a sense of relaxation and tranquility. The warm color palette, dominated by soft tones, creates a gentle and inviting atmosphere. The presence of cushioned cushions and a plush couch adds to the comfort and leisurely mood. The coffee table is adorned with teacups and books, suggesting a leisurely pastime, possibly reading or unwinding with a cup of coffee. The window allows natural light to permeate the space, enhancing the welcoming ambiance. The art on the walls and the potted plant introduce a personal touch, making the environment feel more inviting and unique. The overall arrangement of the room, with its warm lighting and comfortable seating, contributes to a peaceful and relaxing atmosphere.



VLsI

Question: What is the average total fueling cost excluding the Ford F150?

#1 Target Layer (Actual 2nd Layer)
The answer is 10

#2 Target Layer (Actual 6th Layer)
The first number is the number of people who are in the first 10

#3 Target Layer (Actual 10th Layer)
100

#4 Target Layer (Actual 14th Layer)
The answer is 30.99

#5 Target Layer (Actual 18th Layer)
The total salary for the first two states is 200.25. The answer is 200.25.

#6 Target Layer (Actual 22nd Layer)
To find the average total fueling cost of the Toyota V24 and the Honda C202, we need to add their respective costs and then divide by 2.
The cost of the Toyota V24 is \$82.56, and the cost of the Honda C202 is \$70.55.
The combined cost is \$82.56 + \$70.55 = \$153.11.
To find the average, we divide the combined cost by 2:
\$153.11 / 2 = \$76.55.
Therefore, the average total fueling cost is \$76.55.

#7 Target Layer (Actual 26th Layer)
Excluding the Ford F150, the average total fueling cost is calculated by adding the fueling costs of the Toyota RAV4 and Honda Civic, which are \$82.56 and \$70.55 respectively, and then dividing by two.
The calculation is as follows:
$$\text{Average cost} = \frac{82.56 + 70.55}{2} = \frac{153.11}{2} = 76.55$$

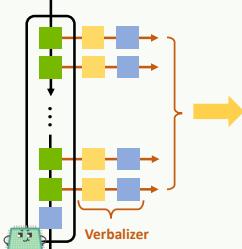
Therefore, the average total fueling cost is \$76.55.

To find the average total fueling cost excluding the Ford F150, we consider the costs of the Toyota RAV4 and Honda Civic. The Toyota RAV4 has a fueling cost of \$82.56, and the Honda Civic has a fueling cost of \$70.55.
To calculate the average, we add these two amounts together and divide by two:
$$\text{Average cost} = \frac{82.56 + 70.55}{2} = \frac{153.11}{2} = 76.55$$

Rounding to two decimal places, the average total fueling cost is approximately \$76.55.



Question: What is located to the right of the shampoo?



VLSI

#1 Target Layer (Actual 2nd Layer)

The image displays a moment of a large, white, and white color. The image depicts a large, white background, with a small, white background. (..Repeats..)

#2 Target Layer (Actual 6th Layer)

The image depicts a large, small, and black color scheme with a red and white text. The text is a black and white line, which is a common feature in the image.

#3 Target Layer (Actual 10th Layer)

The text "100% of (..Repeats..)

#4 Target Layer (Actual 14th Layer)

The text on the sign is the text on the sign.

#5 Target Layer (Actual 18th Layer)

The first answer is the first answer to the question.

#6 Target Layer (Actual 22nd Layer)

To the right of the shampoo, there is a "Densifying conditioner for fine & thin hair" and a "Densifying manger for fast quick results." These are part of the "Therrier & getDefault" series by Charles W. Hagan.

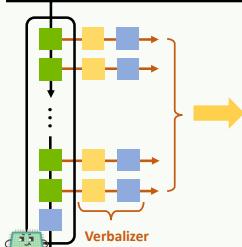
#7 Target Layer (Actual 26th Layer)

To the right of the shampoo, you will find the densifying conditioner and the densifying manger from the "Thicker & Fuller" line by Charlesworth London.

To the right of the shampoo, you will find the densifying conditioner. This product is part of the "Thicker & Fuller" line and is designed to enhance hair volume and fullness. It typically follows the shampoo in a hair care routine, providing additional benefits to the hair's texture and appearance.



Question: Which chest x-ray disease can be found? Options include:
Atelectasis; Cardiomegaly; Effusion;
Infiltration; Mass; Nodule; Pneumonia;
Pneumothorax; Consolidation; Edema;
Emphysema; Fibrosis; Pleural
Thickening; Hernia.)



VLSI

#1 Target Layer (Actual 2nd Layer)

The image is a large, white, and white, with a small, white, and white. (..Repeats..)

#2 Target Layer (Actual 6th Layer)

The image is a black and white line with a visible pattern of color.

#3 Target Layer (Actual 10th Layer)

The image is a black and white image of a large, black and white image of a small, small, black and white image of a small, small, small, and small. The text is in English and is in the text of the text.

#4 Target Layer (Actual 14th Layer)

The image is a black and white image of a large, dark coloration, which is a common coloration for a high-quality, high-quality image. The image is a black and white image, which is a common color scheme for a large and more likely than a small, small size.

#5 Target Layer (Actual 18th Layer)

The most likely type of corruption of the image is the answer to the question.

#6 Target Layer (Actual 22nd Layer)

Based on the given options and the characteristics of the X-ray, the most likely visible condition is "Béral". This is a specific type of "Furball," which is a term often used to describe a large, bright, or bright and bright image in a MRI image. (..Repeats..)

#7 Target Layer (Actual 26th Layer)

The chest X-ray indicates the presence of a mass; however, the specific condition is not clearly visible from the image provided. To accurately determine the nature of the mass, a detailed analysis by a medical professional is necessary. Common chest X-ray findings include:
- Atelectasis: collapse of lung tissue.
- Cardiomegaly: enlarged heart.
- Effusion: fluid accumulation.
- Infiltration: tissue or fluid accumulation.
- Mass: a non-lobulated mass.
- Nodule: a smoothly marginated mass.
- Pneumonia: inflammation.
- Pneumothorax: air in the pleural space.
- Consolidation: consolidation of lung tissue.
- Edema: tissue swelling.
- Emphysema: lung tissue damage.
- Fibrosis: tissue scar.
- Pleural thickening: thickening of the pleural membrane.

The chest X-ray indicates the presence of infiltration. Infiltration refers to the accumulation of substances, such as fluid or cells, within the lung tissue, which can appear as areas of increased opacity on the X-ray. This can be indicative of various underlying conditions, including infection, inflammation, or the presence of substances like blood or fluid. Identifying infiltration is crucial for further diagnostic evaluation and treatment planning.



Question: In which country was this photo taken?

VLsI

#1 Target Layer (Actual 2nd Layer)

The image displays a moment of a large, white, and white color. (...Repeats..)

#2 Target Layer (Actual 6th Layer)

The first of the two people are in the image, which is a large, black and white line with a red and white color scheme. The text is a common feature of the image, with the text "12" on the top and the "20" on the right.

#3 Target Layer (Actual 10th Layer)

The woman in the image is likely a young man, likely a young man with a high level of attention to detail, likely a young man with a high level of energy and energy.

#4 Target Layer (Actual 14th Layer)

The question is about the question about the image of the pizza and the question about the text.

#5 Target Layer (Actual 18th Layer)

The setting of the image is likely the United States, as the setting and setting of the setting is a typical setting for a dining table. The setting is likely a dining table, which is a common way to display a sign or a sign that is often used to display a sign. (...Repeats..)

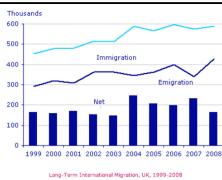
#6 Target Layer (Actual 22nd Layer)

The question asks about the location of the setting in the image. The answer suggests that the setting is in a location where the "klebe" is used. This is likely a reference to the "klebe" (...Repeats..)

#7 Target Layer (Actual 26th Layer)

The photograph was taken in Australia. This conclusion is based on the context provided, which indicates that the image captures a moment of celebration likely related to a significant event in the country, such as a major sports tournament. The trophy being held is reminiscent of those given out at major Australian events, reinforcing the likelihood that this scene is set in Australia.

The photograph was taken in Australia. This conclusion is based on the context provided, which indicates that the image captures a moment of celebration likely related to a significant event in the country, such as a major sports tournament. The trophy being held is reminiscent of those given out at major Australian events, reinforcing the likelihood that this scene is set in Australia.



Question: The graph below shows the long-term international migration, UK, 1999-2008. Summarize the information by selecting and reporting the main features, and make comparisons where relevant. You should write at least 150 words.

VLsI

#1 Target Layer (Actual 2nd Layer)

The image displays a moment of a large, white, and 1000 (...Repeats..)

#2 Target Layer (Actual 6th Layer)

The image is a black and white line with a red line labeled "A" and "A" on the top and "1000" on the right. The text is labeled "A" and "1000" on the right.

#3 Target Layer (Actual 10th Layer)

The image displays a collection of 2000-100000 people in the United States, with the number of people in the United States is 100000. The data is labeled as "The 100000"

#4 Target Layer (Actual 14th Layer)

The first question is to answer the question about the data on "The answer is (...Repeats..)

#5 Target Layer (Actual 18th Layer)

The chart illustrates the "The total number of people who are in the United States with a potential for a potential for the same amount of data data data. The data is 2002, (...Repeats..)

#6 Target Layer (Actual 22nd Layer)

The graph illustrates the trend of net migration in the UK from 1999 to 2008, alongside the individual actions of migration. The data is divided into three distinct categories: income, expenditure, and net (...Repeats..)

#7 Target Layer (Actual 26th Layer)

The graph illustrates the trends in UK's long-term international migration from 1999 to 2008, broken down into three categories: immigration, emigration, and net migration. Overall, the data reveals a consistent upward trend in both immigration and emigration. However, the rate of increase varied throughout the period. Between 1999 and 2008, immigration consistently outpaced emigration, resulting in a net migration flow that was positive throughout the years. In the initial years, both immigration and emigration numbers were relatively low, hovering around the 200 thousand mark. By 2008, both had notably increased, with immigration reaching nearly 600 thousand and emigration slightly below 500 thousand. The net migration flow, which is the difference between immigration and emigration, was positive but varied throughout the years. It was at its lowest point in 2001 and highest in 2004. Despite these fluctuations, the net migration flow remained positive throughout the period, reflecting the consistently higher numbers of immigrants compared to emigrants. In summary, the graph clearly shows a marked increase in both immigration and emigration, with the former consistently outpacing the latter. The resulting net migration flow was positive but exhibited some fluctuations, highlighting the dynamic nature of migration patterns in the UK during this time.

The graph illustrates the trends in immigration, emigration, and net migration in the UK from 1999 to 2008. Throughout this period, immigration consistently outpaced emigration, resulting in a net migration gain. Notably, immigration numbers reached a peak of nearly 600,000 in 2007, marking the highest level recorded during this time frame. Conversely, emigration levels varied but generally stayed below 200,000. The net migration figure, which is the difference between immigration and emigration, was positive every year, with the highest net migration recorded in 2004 at approximately 300,000. Despite some fluctuations, the net migration remained positive, indicating a net migration gain throughout the period.

Dataset 1

Department	Name	Salary
Administration	Emily	\$2,000
Finance	Nick	\$1,800
Marketing	John	\$2,400
Marketing	Gloria	\$1,800
Administration	Jane	\$2,500
Finance	Max	\$2,000
Administration	Tyler	\$3,000
Finance	Wilkins	\$2,500
Marketing	Miles	\$2,800
Administration	Sabine	\$1,900

Question: Which department is the person who has the highest salary from?

VLSI

#1 Target Layer (Actual 2nd Layer)
The answer is 10

#2 Target Layer (Actual 6th Layer)
The answer is 1.

#3 Target Layer (Actual 10th Layer)
The first two states of the data is the first answer

#4 Target Layer (Actual 14th Layer)
The answer is

#5 Target Layer (Actual 18th Layer)
The first answer is the answer to the question.

#6 Target Layer (Actual 22nd Layer)
The person with the highest salary, with a salary of \$3,000, belongs to the "Accounting" department. This is determined by comparing the salaries of the individuals in each department. The data points are as follows: "Amazon" with a salary of \$2,000, "Finance" with a salary of \$2,000, "Marketing" with a salary of \$2,000, "A advertising" with a salary of \$3,000, and "A advertising" with a salary of \$1,900. Among these, the highest salary is found in the "A agriculture" category.

#7 Target Layer (Actual 26th Layer)
The individual with the highest salary is from the Finance department. In the provided dataset, the salaries are as follows:
Administration: \$2,000 (Emily)
Finance: \$2,500 (Max)
Marketing: \$2,600 (John)
Marketing: \$1,800 (Gloria)
Administration: \$3,000 (Tyler)
Administration: \$1,900 (Sabine)
Marketing: \$2,800 (Miles)
Among these, Tyler earns the most with \$3,000, which is the highest salary in the dataset. Therefore, the person with the highest salary is from the Finance department.

The individual with the highest salary is from the Administration department. In the provided dataset, the salaries are as follows:
Administration: \$2,000 (Emily)
Finance: \$2,000 (Nick)
Marketing: \$2,000 (Jane)
Finance: \$2,500 (Wilkins)
Marketing: \$2,600 (John)
Marketing: \$1,800 (Gloria)
Administration: \$3,000 (Tyler)
Administration: \$1,900 (Sabine)
Marketing: \$2,800 (Miles)
Among these, Tyler earns the most with \$3,000, which is the highest salary in the dataset. Therefore, the person with the highest salary is from the Administration department.

VLSI

#1 Target Layer (Actual 2nd Layer)
The image depicts a moment of a large, white, and white color. The color is a large, white, and white, with a small, white, and white.

#2 Target Layer (Actual 6th Layer)
The image displays a large, small, and white, white, white, and white. The color palette is a common feature of the image, with the visible elements visible in the image. The color palette is a common feature of the image, with the visible elements visible in the image.

#3 Target Layer (Actual 10th Layer)
The image of a large, large blue and white, white and white, and white in the image is a common common way to form a color scheme with a more traditional design. The color palette is a common common way to create a more traditional and traditional color scheme.

#4 Target Layer (Actual 14th Layer)
The image of the large, vibrant red and reds of the buildings in the image are a significant portion of the color scheme of the image. The color scheme is a significant part of the design, and the color scheme is a significant part of the design. The color scheme is a significant part of the design, and the color scheme is a significant part of the text.

#5 Target Layer (Actual 18th Layer)
The second image, which features a collection of colorful and vibrant colors, is likely to be more likely to be considered as a more likely answer for the question as it is likely to be a more likely answer for the question as it is likely to be a more likely answer for the image provided.

#6 Target Layer (Actual 22nd Layer)
Based on the given context, the second image with the colorful buildings with vibrant colors and intricate design elements is likely to be more likely to be perceived as having a higher aesthetic appeal. This is often due to the visual appeal of vibrant colors and the intricate design elements that are often associated with the art and design of the past, which often have a more complex and complex design elements than the more straightforward and less colorful architecture of the first image.

#7 Target Layer (Actual 26th Layer)
Based on societal standards and norms, the second image is likely to be perceived as having a higher aesthetic appeal. This assessment is based on the vibrant colors and charming design elements of the buildings, which often align with current aesthetic preferences for vibrant and playful color schemes. The use of bright hues and intricate details can evoke a sense of vibrancy and creativity, which may be more appealing to a wider audience compared to the more muted and practical colors of the first image.

Based on societal standards and norms, the second image is likely to be perceived as having a higher aesthetic appeal. This is due to its vibrant colors and charming design, which often attract positive attention and admiration. The use of bright hues and the artistic arrangement of elements can evoke a sense of beauty and creativity, making it more visually appealing to a wider audience.

D. Comprehensive Experimental Setup for Ablation Studies

Training and Inference Configuration of LLaVA-OV [56]. Different from Qwen2-VL [92]-based VLSI, we select intermediate target layers of LLaVA-OV-based VLSI to i_s : 4th, 12th, 20th layers, and i_t : 10th, 30th, 50th, and 70th layers. Although LLaVA-OV’s language model is Qwen2 [96] that is equal with that of Qwen2-VL, the required number of image tokens in LLaVA-OV are 4 to 10 times more number than that of Qwen2-VL, depending on the height-to-width ratio of the images, given the same pixel count. To accommodate this increased computational demand on 8 NVIDIA A100 80GB GPUs, we reduce the number of intermediate target layers. For efficient training, we equally employ LoRA [40] with rank and alpha parameters to 64, use the AdamW optimizer [69] with a cosine annealing schedule, adjusting the learning rate from $1e^{-4}$ to $1e^{-6}$ over each training step, and use gradient accumulation with 16 steps and gradient checkpointing [84]. The only different training configuration is batch sizes where two batches each are used for the 0.5B and 7B model sizes, and one batch is used for the 72B model sizes. It finally results in 256 ($8 \times 16 \times 2$) and 128 ($8 \times 16 \times 1$) batches, respectively. We conduct inference experiments under the equal setup used in Qwen2-VL, where we use a greedy search for text generation.

VLMs	MM-Vet	MMMU
LLaVA-OV-7B	57.5	48.8
Qwen2-VL-7B	62.0	54.1
LLaVA-OV-72B	63.7	56.8
Qwen2-VL-72B	74.0	64.5

Fig. 4 provides the challenging evaluation benchmarks’ performances: MM-Vet [100] and MMMU [102]. Each cell represents the performances on their evaluation benchmarks, where the orange colored-values represent LLaVA-OV-based VLSI’s result and the purple ones represent Qwen2-VL-based result. This figure reveals consistent trends that using large- and small-backbone VLMs with more bigger model sizes enhances VLSI’s performances across all configurations. Besides, we can easily infer that the baseline performances of Qwen2-VL before buliding VLSI are also higher than those of LLaVA-OV as shown in the above table. Furthermore, Tab. 4(a) shows VLSI’s generalization ability to 0.5B and 7B model sizes in LLaVA-OV, but similarly insists that using better large-backbone VLMs provides benefits from performances.

Tab. 4(b) collectively illustrate the impact of incorporating the *SFT step*. Table (b) highlights the significant performance gains achieved by applying additional SFT. Interestingly, the performances without additional SFT step are not satisfactory when doing greedy decoding, but once we utilize beam decoding ($N=5$), its performances are dramatically enhanced. This suggests that the interaction step implicitly alters and expands the probability space over the language head, while SFT aligns this expanded space to better fit instruction-following. Thus, significant improvements arise from the combination of interaction and SFT step.

Tab. 4(c) It examines whether these improvements result from the fine-tuning effects of 2.9 million visual instruction tuning samples. The results demonstrate that VLSI’s performance becomes markedly superior to Qwen2-VL as SFT training progresses from 50% to 100%, confirming that additional SFT plays a much more critical role in driving performance enhancements than the contribution of the visual instruction tuning samples alone.

IL-Ops	LL-Ops	vL-Head	MM-Vet	MMMU
L2	X	X	63.3	53.5
L2	KLD	X	64.6	55.9
L2	L2	X	63.9	54.4
X	L2	-	65.2	56.8
X	KLD	-	66.5	57.5
KLD	KLD	✓	75.8	69.3

Tab. 4(d) evaluates the impact of different operations applied to intermediate (IL-Ops) and last layers (LL-Ops) on VLSI’s performances. The results clearly demonstrate that KL divergence (KLD) operations applied to both intermediate and

last layers yield the best performances, confirming the effectiveness of KLD over cross-entropy (CE) or L2 for intermediate layer alignment and final layer interaction. To further assess the effectiveness of verbalization in transferring the knowledge via distillation, we conduct an ablation study by removing the language head in verbalizer (vL-head). In this setup, the *verbalization step* is skipped, and the *interaction step* is first performed by aligning the hidden dimensions of the verb-FFNs in large- and small-backbone VLM. Here, the verb-FFN’s hidden dimension in large-backbone VLM is kept but the hidden dimension in small-backbone VLM’s verb-FFN is enlarged to match the large one. In addition, the verb-FFNs of large- and small-backbone VLM are interacted with only L2, which means that small-backbone VLM’s verb-FFNs try to naively follow those of large-backbone VLM. After *interaction step*, *SFT step* is equally conducted. However, as shown in the above table, this naive approach results in significantly lower performance compared to the version equipped with the language head. Interestingly, using only last layer distillation by L2 and KLD provides more benefits than using intermediate distillation without vL-Head. These findings suggest that directly imitating outputs from the large-backbone VLM, without verbalization from vL-Head, introduces instability and can lead to suboptimal results, highlighting the critical role of the language head in achieving effective distillation.

Pseudo-Code for Random Index (Search Range \mathcal{X})

```

1: Input:  $t_s, t_l$ 
2: Initialize: loss: 0,  $i_l^*$ : 0,  $\epsilon$ : 1e-6, scale: 2
3: for  $i_s$  in  $0 \leq i_s < t_s$  do
4:    $kld-list = []$ 
5:   for  $i_l$  in  $0 \leq i_l < t_l$  do
6:      $kld-list.append(compute-kld(i_s, i_l))$ 
7:   end for
8:    $r \leftarrow \text{Random-Select}(kld-list)$ 
9:   loss  $\leftarrow$  loss  $+r$ 
10: end for
11: Return: loss

```

Pseudo-Code for Uniform Index (Search Range \mathcal{X})

```

1: Input:  $t_s, t_l$ 
2: Initialize: loss: 0,  $i_l^*$ : 0,  $\epsilon$ : 1e-6, scale: 2
3: layer-gap-ratio  $\leftarrow \text{floor}(\frac{t_l}{t_s})$ 
4: for  $i_s$  in  $0 \leq i_s < t_s$  do
5:    $i_l \leftarrow i_s \times \text{layer-gap-ratio}$ 
6:    $u \leftarrow \text{compute-kld}(i_s, i_l)$ 
7:   loss  $\leftarrow$  loss  $+u$ 
8: end for
9: Return: loss

```

Tab. 4(e) highlights the effectiveness of various components in the matching strategy. Random Index, Uniform Index, and Bottom-1 or 3 Index, yield lower scores, underscoring the limitations of simpler selection mechanisms. Note that, the above and below algorithms represent the their detailed experimental setup. Multinomial sampling provides improvements and incorporating the Search Range further enhances performances (e.g., 83.5 on MMB and 69.8 on MM-Vet). Adding order preservation results in a significant leap, particularly on BLINK (59.2) and MM-Vet (75.2), demonstrating the importance of maintaining matched indices’ sequence alignment during *interaction step*. Finally, using all together with adaptive temperature achieves the best results across all benchmarks (e.g., 86.3 on MMB and 69.3 on MMMU), showcasing its ability to dynamically control the distribution. These exploration underscore the necessity of advanced sampling strategies and adaptive mechanisms for maximizing the efficiency of transferring the knowledge.

Pseudo-Code for Bottom- k Index (Search Range X)

```
1: Input:  $t_s, t_l$ 
2: Initialize: loss: 0,  $i_l^*$ : 0,  $\epsilon$ : 1e-6, scale: 2
3: for  $i_s$  in  $0 \leq i_s < t_s$  do
4:    $kld-list = []$ 
5:   for  $i_l$  in  $0 \leq i_l < t_l$  do
6:      $kld-list.append(compute-kld(i_s, i_l))$ 
7:   end for
8:    $bottom-k-kld-list \leftarrow Bottom-k(kld-list)$  {length( $bottom-k-kld-list$ ):  $k$ }
9:    $a \leftarrow Average(bottom-k-kld-list)$ 
10:  loss  $\leftarrow$  loss  $+a$ 
11: end for
12: Return: loss
```

Tab. 4(f) highlights verb-FFN’s efficiency among different verbalizer architectures in terms of performance and parameter count. While larger architectures such as Decoder $\times 2$ (3.3B) and FFN $\times 2$ (2.9B) achieve strong performance across MM-Vet and MMMU, the much smaller verb-FFN architecture (269M) delivers comparable results. In contrast, simpler structures like MLP $\times 2$ and MLP, while efficient in terms of parameter count, fall short in performance. These results emphasize the effectiveness of the verb-FFN architecture in a lightweight yet high-performing solution. Its efficiency becomes even more pronounced as the number of intermediate target layers increases and larger-backbone VLMs are utilized.

E. VLsI's Text Generation Quality



Question: How many gallons of supreme gasoline can I get with \$50?



VLsI-2B

To determine how many gallons of supreme gasoline you can buy with \$50, we need to calculate the price per gallon and then apply that to the total amount you have.

1. **Calculate the price per gallon:**

- The price for regular gasoline is \$3.49.
- The price for plus gasoline is \$3.55.
- The price for diesel No. 2 is \$3.60.

The price per gallon for each type of gasoline is:

- Regular: $\$3.49 / 10 = \0.349 per gallon
- Plus: $\$3.55 / 10 = \0.355 per gallon
- diesel No. 2: $\$3.60 / 10 = \0.36 per gallon

2. **Convert the total amount of money to cents:**

- \$50 is equal to 5000 cents.

3. **Calculate the number of gallons of supreme gasoline you can buy:**

- Divide the total amount of money by the price per gallon for supreme gasoline:
 $5000 \text{ cents} \div \$0.36 \text{ per gallon} = \text{approximately } 1388.89 \text{ gallons}$

Therefore, with \$50, you can buy approximately 1388.89 gallons of supreme gasoline.



VLsI-7B

To determine how many gallons of supreme gasoline you can purchase with \$50, you need to consider the price per gallon and subtract it from your budget until you reach zero.

1. **Calculate the cost per gallon:**

- The price for supreme gasoline is \$3.659 per gallon.

2. **Divide your budget by the price per gallon:**

- Divide \$50 by \$3.659 to find out how many gallons you can buy.
 $\frac{50}{3.659} \approx 13.63$ gallons.

3. **Determine the exact amount:**

- You can buy 13 full gallons of supreme gasoline with \$50.

So, with \$50, you can purchase 13 full gallons of supreme gasoline.



Question: What is the step after using probe to spread tire puncture?



VLsI-2B

After using the probe to widen the tire puncture, the next step is to take the repair strip out of the plastic packaging. This prepares the repair strip for insertion into the puncture, ensuring it's ready to seal the hole effectively.

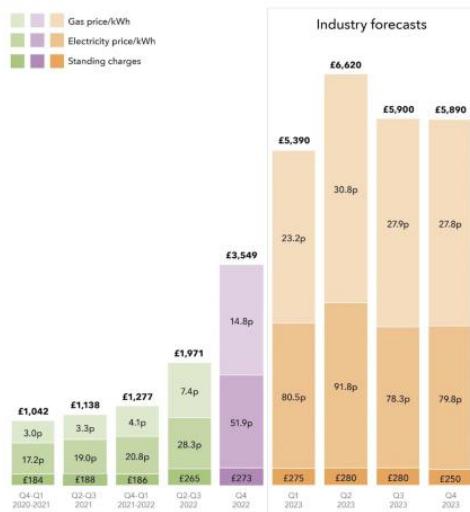


VLsI-7B

After using the probe to widen the tire puncture, the next step is to take the repair out of the plastic. This involves carefully removing the repair material from its packaging, ensuring it's ready for insertion into the tire.

Energy price cap update

ICAEW chart of the week



Question: What is the estimated average standing charge for the year 2023?



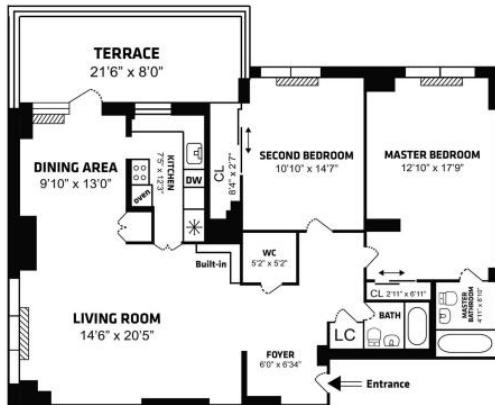
VLsi-2B

The projected average standing charge for the year 2023 is estimated to be £280. This figure represents the expected cost that consumers will pay in the form of standing charges, which are fixed amounts that cover the costs of supplying energy to consumers. The standing charge is distinct from the other components of energy pricing, such as gas and electricity prices, and is calculated based on the actual usage of energy services.



VLsi-7B

The projected average standing charge for the year 2023 is estimated to be approximately £275. This figure represents the projected cost associated with maintaining energy services, independent of actual energy usage. The standing charge is a fixed amount that consumers pay to cover the expenses of providing energy services, such as infrastructure maintenance and service delivery.



For illustration purposes only.



Question: In which direction does the living room window face?



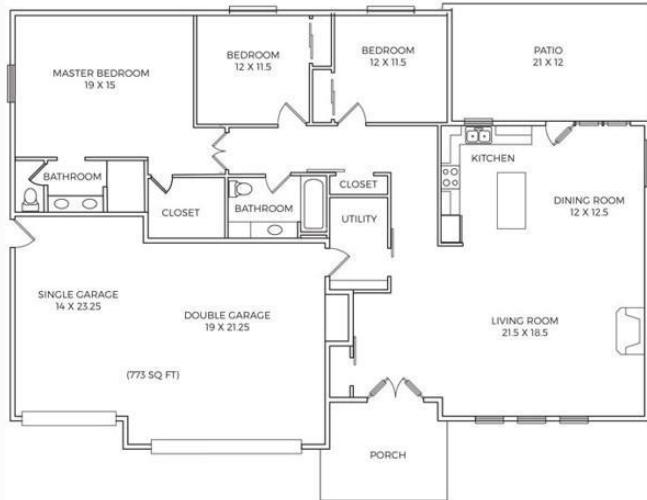
VLsl-2B

The living room window is oriented towards the north. This is indicated by the compass rose in the floor plan, which points to the north. The compass rose is a visual aid that helps in identifying the direction of various elements in a plan, such as rooms and their respective compass positions. In this case, the living room is positioned on the north side of the building, as indicated by the compass rose.



VLsl-7B

The living room window is oriented towards the north. This is indicated by the compass rose in the floor plan, which shows the compass directions. The north arrow points upward, and the living room is positioned to the south of this arrow, suggesting that the window is directed northward.



Question: Which room is bigger, the double garage or the living room?



VLsi-2B

The double garage is larger than the living room. The dimensions of the double garage are 19 feet by 21.25 feet, resulting in an area of 387.5 square feet. In comparison, the living room measures 21.5 feet by 18.5 feet, giving it an area of 397.75 square feet. Therefore, the double garage has a greater area than the living room.



VLsi-7B

The double garage is larger than the living room. The double garage measures 19 feet by 21.25 feet, giving it a total area of 404.75 square feet. In comparison, the living room measures 21.5 feet by 18.5 feet, resulting in an area of 397.75 square feet. Therefore, the double garage has a slightly larger area than the living room.



Question: Can you give a short introduction to this painting?



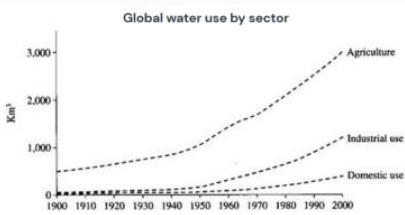
VLSI-2B

This painting is known as "The Return of the Tibetan Envoy," created by Zhang Zeduan in 1369. It is a significant work in Chinese history, depicting the return of a Tibetan Envoy to the Tang dynasty. The painting is celebrated for its artistic excellence and historical significance. It is often referred to as "China's Mona Lisa" and is one of the most famous pieces of Chinese art. The painting is housed in the Palace Museum in Beijing.



VLSI-7B

"Emperor Taizong Receiving the Tibetan Envoy" is a silk painting by Yan Liben, created during the Tang dynasty. Measuring 199 centimeters in length and 49.5 centimeters in width, this artwork is housed in the National Palace Museum in Taipei. The painting vividly portrays the historic meeting between the Tang dynasty's Emperor Taizong and a Tibetan Envoy, highlighting the amicable relations between the two cultures during that era.



Water consumption in Brazil and Congo in 2000

Country	Population	Irrigated land	Water consumption per person
Brazil	176 million	26,500 km ²	359 m ³
Democratic Republic of Congo	5.2 million	100 km ²	8 m ³

Question: The graph and table below give information about water use worldwide and water consumption in two different countries. Summarise the information by selecting and reporting the main features, and make comparisons where relevant. You should write at least 150 words.



VLsi-2B

The graph illustrates the global water usage across agriculture, industry, and domestic sectors from 1900 to 2000. In 1900, agricultural water usage was the lowest, followed by industrial and then domestic. By 2000, agricultural water usage had increased significantly, surpassing industrial and domestic. The table provides data on Brazil and the Democratic Republic of Congo in 2000, highlighting their population, irrigated land, and water consumption per person.

In Brazil, the population was 176 million, with 26,500 km² of irrigated land and a water consumption per person of 359 m³. In contrast, the Democratic Republic of Congo had a population of 5.2 million, with 100 km² of irrigated land and a water consumption per person of 8 m³. Despite having a smaller population, the Congo utilized more water per capita, highlighting the disparity in water consumption between the two countries.

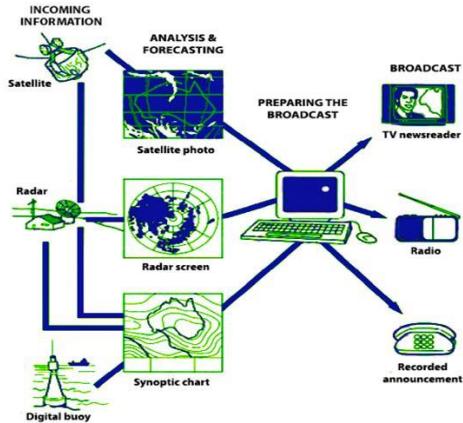


VLsi-7B

The graphs illustrate global water usage across agriculture, industry, and domestic sectors from 1900 to 2000, alongside data on water consumption in Brazil and the Democratic Republic of Congo in 2000.

Overall, global water consumption increased dramatically from 1900 to 2000, with agriculture being the predominant sector. In 1900, agricultural water usage was minimal, but it surged by 2000, surpassing industrial and domestic usage. Industrial water consumption rose steadily but was consistently lower than agriculture's. Conversely, domestic water usage saw a more moderate increase.

In 2000, Brazil exhibited significantly higher water consumption per capita compared to the Democratic Republic of Congo. Brazil's population was 176 million, with 26,500 km² of irrigated land, resulting in a per capita water consumption of 359 m³. In contrast, the Democratic Republic of Congo, with a population of 5.2 million and 100 km² of irrigated land, had a much lower per capita water consumption of 8 m³. This stark difference highlights the substantial disparity in water usage between the two countries.



Question: The diagram below shows how the Australian Bureau of Meteorology collects up-to-the-minute information on the weather in order to produce reliable forecasts. Write a report for a university lecturer describing the information shown below. You should write at least 150 words.



VLSI-2B

The diagram illustrates the methodology employed by the Australian Bureau of Meteorology to generate accurate weather forecasts. It begins with the collection of data from various sources, including satellites, radar, and digital buoys. These data points are then processed through analysis and forecasting models, which utilize satellite photos and radar screens to generate synoptic charts. These charts provide detailed information on weather patterns and conditions across different regions.

The collected data undergoes further analysis to predict weather changes, ensuring that the forecast is timely and relevant. Once the analysis is complete, the prepared forecast is broadcasted through various channels, including television newsreaders, radio, and recorded announcements. This comprehensive approach ensures that the public receives timely and accurate weather updates, contributing to better preparedness and informed decision-making.



VLSI-7B

The diagram illustrates the methodology employed by the Australian Bureau of Meteorology to generate accurate weather forecasts. The process begins with the collection of current weather data from various sources, including satellites, radar systems, and drifting buoys. This data is then processed and analyzed to create a comprehensive understanding of the weather patterns.

Initially, satellite imagery provides a wide view of cloud formations and temperature differences across the globe. This information is crucial for predicting large-scale weather systems. Simultaneously, radar data is gathered from multiple locations to track local weather conditions, such as rainstorms or snowstorms, offering detailed insights into specific regions.

To enhance the accuracy of these data sources, synoptic charts are utilized. These charts depict pressure systems and wind patterns, allowing meteorologists to identify high and low pressure areas, which are key factors in determining weather movements.

The processed data is then used to create detailed weather forecasts. This involves analyzing the collected information to predict changes in temperature, pressure, and wind speeds. The resulting forecast is prepared for dissemination through various channels, including television, radio, and recorded telephone announcements.

Finally, the prepared forecast is broadcasted to the public, ensuring timely and reliable weather updates. This multi-faceted approach allows the Bureau to provide accurate and timely weather information, enabling individuals and businesses to make informed decisions based on the forecasted conditions.



Question: What is this diagram telling?



VLsi-2B

This diagram illustrates a cloud architecture designed for a healthcare facility using AWS services. It details the setup of a web application hosted on AWS, along with its dependencies and security measures.

Components and Their Roles:

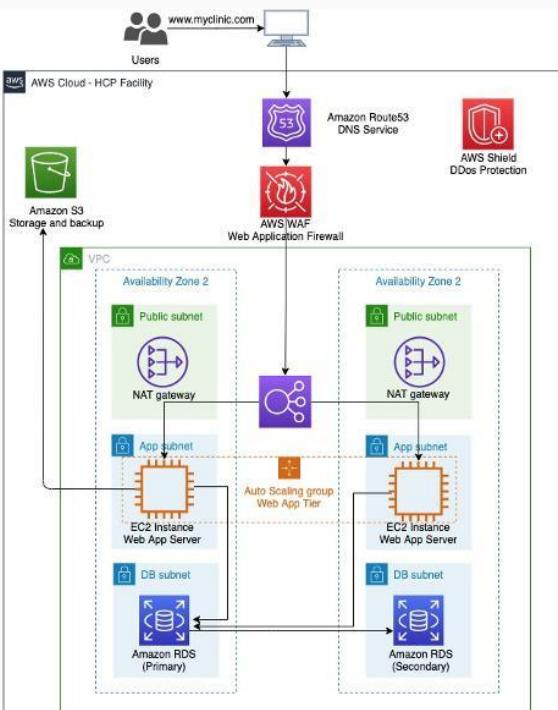
1. **Amazon S3**: Used for storage and backup of medical records.
2. **Amazon Route53**: Manages DNS services for domain names associated with the cloud infrastructure.
3. **AWS WAF**: 提供防护，防止DDoS攻击。
4. **VPC (Virtual Private Cloud)**:
 - **Availability Zone 2**: Two Availability Zones are configured for high availability.
 - **NAT gateway**: Converts traffic between public and private networks.
 - **App subnet**: Contains instances of the web application.
 - **DB subnet**: Manages databases for patient records.
5. **Autoscaling Group**: Manages load balancing and resource optimization.
6. **Amazon RDS (Rapid Deployment Service)**:
 - **Primary**: The primary database for patient records.
 - **Secondary**: A secondary instance for redundancy.
7. **CloudFront**: Manages content delivery to ensure fast load times for users.
8. **CloudTrail**: track and manage AWS events for security and monitoring.

Security and Protection:

- **AWS Shield**: Ensures secure internet access by providing DDoS protection.
- **CloudTrail**: Logs AWS activities, enhancing security and monitoring.

Conclusion:

This architecture provides a robust framework for a healthcare facility to manage its cloud services efficiently. It includes secure components like AWS WAF, RDS, and CloudTrail, as well as auto-scaling and load balancing to ensure reliable and scalable operations.



VLsi-7B

The diagram illustrates a cloud architecture using AWS services, designed for secure and reliable hosting of a healthcare application. Here's a detailed explanation:

1. **User Interaction**: Users access the application through a URL, which is directed to an Amazon Route53 DNS service.
2. **Application Security**: The DNS service is protected by an AWS WAF (Web Application Firewall), which filters and protects against web exploits and malicious traffic.
3. **Application Deployment**: The application is hosted on an auto-scaling group within an EC2 (Elastic Compute Cloud) instance, ensuring high availability and load management.
4. **Database Management**: The database is stored on Amazon RDS (Database instance), providing reliable and secure data management.
5. **Network Security**: The network is segmented into multiple subnets within two Availability Zones, offering fault tolerance and network resilience.
6. **Application Access**: The application is accessible through NAT (Network Address Translation) gateways, which facilitate secure internet connections.
7. **Cloud Storage**: Amazon S3 is used for secure storage and backup of application data.

This architecture combines multiple services to ensure the application is secure, reliable, and scalable.



VLsi-2B

To interpret the blood sugar test results accurately, follow these steps:

1. **Look at the Date and Time**:** The test was conducted on a specific date and time. This helps in comparing the results with the expected values for that day.
 2. **Examine the Blood Sugar Level**:** The meter displays the blood sugar level. A normal range for blood sugar is typically between 70 and 110 mg/dL. If the level is outside this range, it may indicate a potential health concern.
 3. **Consider the Time of Day**:** The blood sugar level can change throughout the day. For example, it may be higher in the morning and lower in the evening. Monitoring at different times can provide more comprehensive data.
 4. **Look at the Time of Measurement**:** The meter may have a feature to indicate the time of measurement. This can help in understanding if the blood sugar level was taken at a peak or a trough.
 5. **Compare with a Blood Sugar Tracker**:** If you have a blood sugar tracker, compare the readings with it to see if there are consistent patterns or fluctuations.
 6. **Consult with a Doctor**:** If the blood sugar level is high, it's important to consult with a healthcare professional for personalized guidance.
- By following these steps, you can better understand and interpret the blood sugar test results.

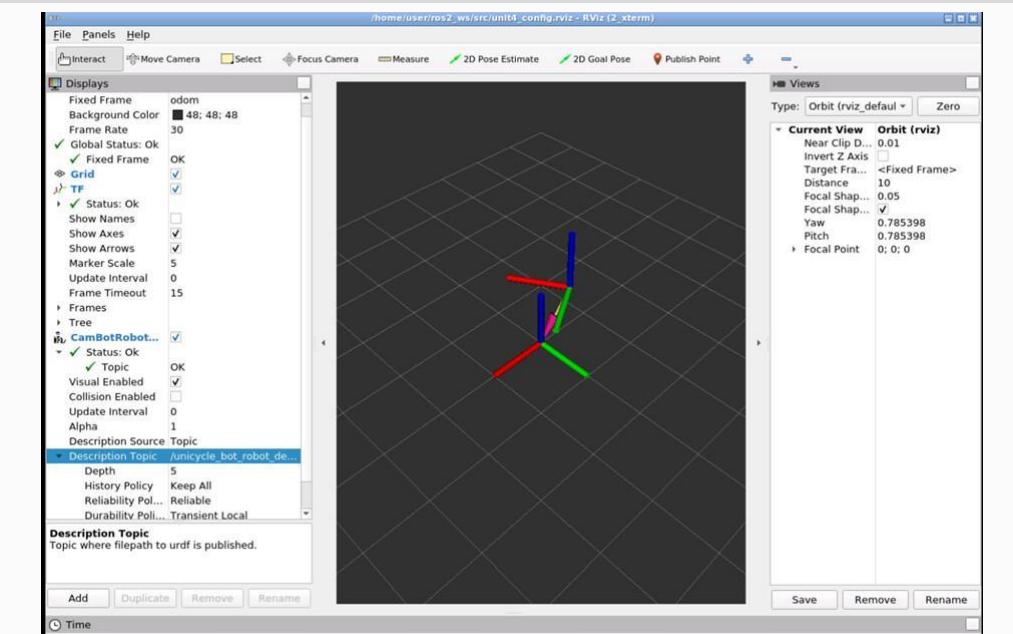


Question: How to interpret the test results in this image?



VLsi-7B

The blood glucose meter displays a reading of 97 mg/dL, which is considered a normal blood sugar level. This indicates that the individual's glucose level is within the typical range for someone without diabetes. The meter is easy to use, providing quick results in just 6 seconds, and requires a minimal amount of blood. It also features four reminder alarms to help manage blood sugar levels effectively. The large display ensures easy reading of the test results.



Question: What is in this image?



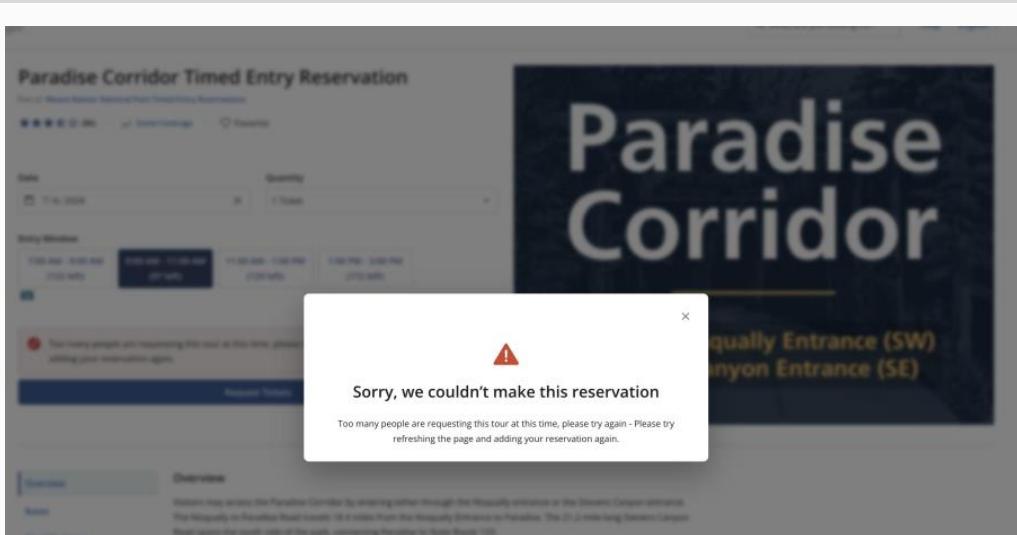
VLsi-2B

The image depicts a robotics simulation environment, likely from a robotics simulation software like RViz. This software is used for visualizing and debugging robotic systems. In the image, you can see a 3D grid representing a simulated environment, with a camera positioned to capture the scene. The grid is colored in red, green, and blue, likely indicating different sensor readings or data points. The top right corner of the image displays parameters for the simulation, such as the robot's orientation and the distance from the camera. This setup is crucial for developers and engineers to test and refine their robotic systems before physical deployment.



VLsi-7B

The image depicts a screenshot of a robotics simulation software interface, likely from the Robot Operating System (ROS) environment. The interface is designed for visualizing and debugging robotic systems. The left side of the screen shows a list of components or sensors, such as "camBotRobot_ee" and "odometry," indicating the various parts of a robotic setup. The central area displays a 3D grid with axes, which is a common way to represent robotic positioning and movement. On the right, there are options for adjusting the view and parameters, such as "Orbit (rviz_default)" and "Zero," which are likely tools for navigating and observing the simulated robotic environment. This setup is crucial for developers and engineers to test and refine robotic algorithms and movements.



Question: What is happening in this image?



VLsi-2B

The image depicts a computer screen with a booking system for a "Paradise Corridor Timed Entry Reservation." The user is attempting to book a ticket for the "Paradise Corridor" but is unable to proceed due to a "Too many people are requesting this tour at this time" error. The message suggests refreshing the page and trying again, indicating that the system is experiencing high demand for the available time slot.



VLsi-7B

The image shows a computer screen with an error message encountered while attempting to make a reservation for the Paradise Corridor at Mount Rainier National Park. The message indicates that the reservation could not be completed due to high demand, with too many people trying to book the tour simultaneously. It suggests refreshing the page and trying to submit the reservation again. This situation suggests that the website is experiencing heavy traffic, likely due to popular times or events, which is causing booking issues.