openPangu-VL-7B: a Multi-Modal Large Language Model Designed and optimized for Ascend NPUs

openPangu Team, Huawei

openPangu@huawei.com

ABSTRACT

In this technical report, we investigate how to efficiently design and train multi-modal foundation models on Ascend NPUs. Specifically, we jointly optimize the model architecture, data pipeline, and training recipe to fully leverage Ascend's capabilities. We first introduce openPangu-ViT, an Ascend-co-designed vision encoder that achieves accuracy comparable to the encoder of Qwen2.5-VL while delivering approximately 15% higher inference throughput. On the system side, we develop an offline multi-modal packing strategy together with "Ascend-shaped" batching, and we further conduct a systematic study of key training factors. With these techniques, a more than 3.2T tokens training run on an Ascend cluster proceeds stably without loss spikes, resulting in a 7B multi-modal model that matches Qwen3-VL-8B across VQA, OCR, grounding, counting, and video understanding benchmarks.

The code and models are available at https://gitcode.com/ascend-tribe/.

1 Introduction

Multi-modal foundation models that process text, images, and video are rapidly becoming core infrastructure for intelligent assistants, educational tools, and in-vehicle copilots [8, 15, 16]. Most existing systems are trained on NVIDIA GPU clusters, which rely on high-bandwidth intra-node links and large-scale InfiniBand networks to enable efficient cross-node synchronization [19]. As Ascend NPUs continue to mature, they are emerging as a strong alternative for large-scale multi-modal training [2, 13, 42]. Unlike GPUs, Ascend adopts a distinct balance of compute, memory, and communication resources [22]. These differences motivate the development of training strategies and model designs that exploit Ascend's strengths, rather than assuming that configurations optimized for GPUs will remain optimal in all settings.

These considerations are further amplified in the multi-modal setting. Modern perception-and-understanding systems typically pair a language model with a vision encoder, introducing additional structural and data-flow complexity [2, 13, 38]. For example, multi-modal training involves text, images, documents, and videos with highly variable sequence lengths. Efficient hardware utilization in this context requires deliberate design of offline packing, length-aware batching, and load balancing, rather than directly reusing pipelines developed primarily for GPUs. In addition, several technical choices still lack consensus, such as the design of encoding

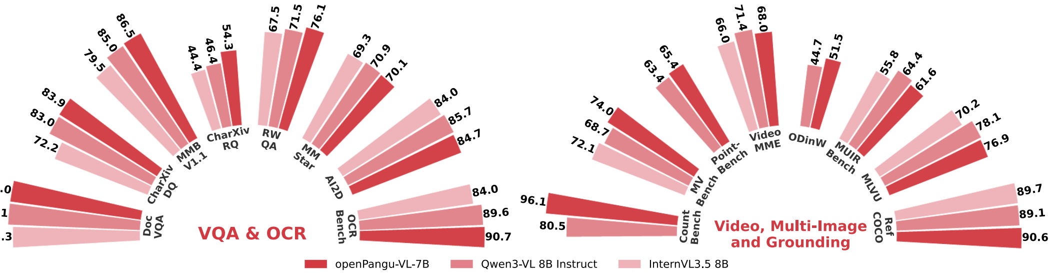

Figure 1: Comparison of benchmark results between openPangu-VL-7B and other competitive models.

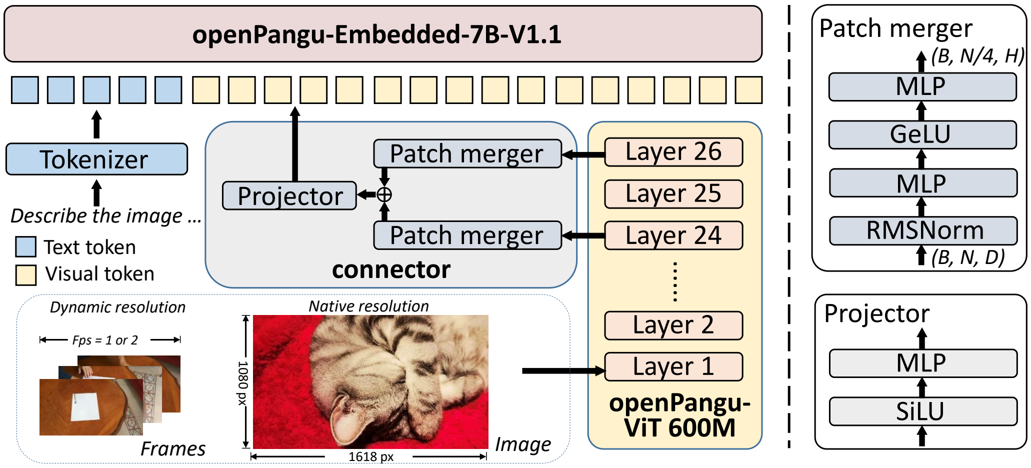

Figure 2: Overall Structure of openPangu-VL-7B model.

strategies for grounding tasks and the balancing of training losses across different modalities. These issues require further exploratory experiments before satisfactory solutions can be established.

In this technical report, we provide a detailed description of our data and training process, along with the practical know-how gained from our preliminary experiments. On the architecture side, we introduce an Ascend-co-designed openPangu-ViT vision encoder, which achieves accuracy comparable to the ViT used in Qwen2.5-VL [3] while delivering approximately 15% higher inference throughput on Ascend hardware. On the system side, we develop an offline packing and mixing scheme that groups heterogeneous modalities into "Ascend-shaped" batches, thereby improving compute density and reducing cross-node traffic. On the training side, we conduct extensive ablation studies to jointly examine key factors, including sequence length, learning-rate schedules, and data-mixing policies. These investigations yield training recipes that are tailored to Ascend's system characteristics.

We validate our design in a large-scale training run exceeding 3.2T tokens on an Ascend cluster, sustaining 884K NPU hours without disruption and achieving stable long-horizon training with no loss spikes or rollbacks. The resulting 7B-scale multi-modal model reaches Qwen3-VL-8B–level performance on general visual question answering, and surpasses Qwen3-VL-8B on OCR and chart/document understanding tasks such as DocVQA [27], CharXiv [40], and OCRBench [24] (see Table 4 for details). Our openPangu-VL-7B also demonstrates strong capability on grounding (RefCOCO [44] and ODinW-13 [21]) and counting (CountBench [28] and Point-Bench [7]), where it surpasses all other models of similar scale. Its performance on video understanding and multi-image reasoning is overall comparable to Qwen3-VL-8B [2].

2 Model Architecture

2.1 Overall Structure

We follow a classic VLM architecture consisting of a vision encoder, a large language model (LLM), and a lightweight multi-modal connector, shown in Figure 2.

Large Language Model. Our openPangu-VL-7B is built on top of the pretrained openPangu-Embedded-7B-V1.1 base model, which serves as the language backbone.

Connector. To bridge the vision encoder and the LLM, openPangu-VL-7B employs a lightweight hierarchical connector. Concretely, we choose multiple outputs of the vision encoder and reduce the effective number of tokens forwarded with the patch merger module. This hierarchical design allows the model to preserve fine-grained local information while controlling the computational cost of multi-modal inputs. And the outputs of hierarchical patch merger are mapped to the LLM embedding space with the projector.

Vision Encoder. We adopt a vision encoder that maps each input image or video frame into a sequence of visual tokens. In order to make the best use of the Ascend NPU, we redesign a vision encoder and train from scratch. The Ascend-affinity architecture design allows us to achieve faster inference speed with the same number of parameters and computational complexity. A detailed introduction to our model design and training

Table 1: The architecture of openPangu-ViT.

Depth	Full-attn layers	Pose embed	Activation	Head dim	Heads	FFN Hidden Size
26	6,13,20,26	2D RoPE	GELU	96	16	4608

pipeline is provided in the following sections. Native resolution is adapted to capture better visual features of input images with different shapes.

2.2 Vision Encoder designed for Ascend NPUs

Ascend-co-designed Architecture. We carefully design the vision encoder's architecture to align with the parallelism characteristics of Ascend Atlas 800T A2 chips. As shown in Table 1, we select a configuration that balances model size and inference speed. The resulting openPangu-ViT contains approximately 615M parameters, consists of 26 layers, and adopts a hybrid attention structure combining window attention and full attention. We use GELU as the activation function in openPangu-ViT, as it has fewer parameters and enables faster inference compared with SwiGLU. To further enhance Ascend affinity, we set the FFN hidden dimension to a multiple of 128, conforming to the hardware design of Ascend NPUs. The chosen dimension is 4,608, which is an integer multiple (i.e., three times) of the attention hidden size.

Vision Encoder Pre-training. To handle inputs of varying resolutions, particularly the high-resolution images common in OCR tasks, openPangu-ViT natively supports dynamic-resolution inputs. The training of openPangu-ViT emphasizes strong OCR capability and is conducted entirely from scratch. We organize the training into two major stages: (1) vision-only pretraining that builds robust object-, scene-, and text-level visual discrimination, and (2) multi-modal alignment training that connects the vision encoder to an LLM.

1. **Vision-Only Pretraining with Contrastive Learning.** The goal of the first stage is to equip openPangu-ViT with strong feature discrimination, especially for fine-grained visual concepts and text. Inspired by Multi-Label Cluster Discrimination (MLCD) [1], we adopt a multi-label contrastive learning framework. While mainstream contrastively trained ViTs (e.g., SigLip) excel at visual concept discrimination, they typically lack text-recognition capability, which is essential for OCR and downstream multimodal reasoning. To address this, we explicitly introduce a text-focused discriminative component into the vision pretraining pipeline.

   We train openPangu-ViT from scratch using approximately 660 million open-source unlabeled images. A clustering algorithm assigns 10 pseudo-labels to each image, yielding a total of roughly 2 million categories. To reduce the computational cost associated with high-resolution inputs, we adopt a multi-resolution cascade schedule: training first at 224×224 for 30 epochs, then at 336×336 for 2 epochs, and finally at 560×560 for 1 epoch. This strategy preserves high-resolution discriminative ability while significantly reducing overall training time.

   To further enhance text-recognition capability, we incorporate an additional contrastive-learning stage focused on text images, drawing on the methodology of RICE [43]. This stage uses approximately 80 million text-centric samples and effectively strengthens openPangu-ViT's OCR-oriented feature representations.

2. **Multi-Modal Alignment with an LLM.** To build multi-modal understanding, we refine openPangu-ViT by coupling it with the pretrained openPangu-Embedded-7B-V1.1 language model using 85 million high-quality image-text pairs. Training follows a two-step next-token prediction procedure: in the first step, both the vision encoder and LLM are frozen to train the connector; in the second step, only the LLM remains frozen while the vision encoder is updated. Both steps use native-resolution image inputs. After the multi-modal alignment completes, we discard the temporary LLM and projector, and retain the resulting vision encoder as the final openPangu-ViT.

2.3 Positional Encoding

Our openPangu-VL-7B adopts the 3D Interleaved Rotary Position Embedding (3D-IM-RoPE), which is also mentioned in the Qwen series [2]. This technique is specifically designed to address inherent representational biases that can arise between modalities (such as image, video, and text) due to uneven distribution of frequencies components across different spatial and temporal dimensions. Our key modification lies in the allocation of six times more frequency bands to the spatial dimension, while assigning relatively fewer to the temporal dimension. Furthermore, to facilitate better differentiation between distinct spatial positions across multiple input sources (e.g., different images in a single prompt or successive frames in a video), we employ the accumulative strategy for applying the position encoding, which helps the model to effectively distinguish the unique relative position of the tokens.

2.4 Visual Prepocessing

Our openPangu-VL-7B adopts the native and dynamic resolution for image inputs and video inputs, separately. Similar to QwenVL, for static image inputs, openPangu-VL-7B employs a native-resolution strategy to handle arbitrarily high input resolution and aspect ratio, effectively mitigating the loss of high-frequency details. However, for video inputs, openPangu-VL-7B implements a frame-budget-aware dynamic resolution scaling mechanism, where the spatial resolution of each frame is inversely proportional to the total number of frames sampled ($\mathcal{N}_{\text{frames}}$) for the input video segment. This adaptive balancing ensures that the model can dynamically prioritize either spatial detail or temporal coverage based on the complexity and length of the input, leading to a more efficient and flexible allocation of resources. The openPangu-VL-7B uses the random frames-per-second (FPS) strategy for training on non-precise time-aware tasks such as video summarization, while employing a constant FPS of 1 for training on precise time-aware tasks like temporal localization, enabling the model to perceive temporal information to some extent.

3 Data Preparation

3.1 Data Categories and Task Types

We organize our training data corpus into several broad categories, including image and video captioning, OCR/Chart/Document understanding, STEM-oriented multi-modal reasoning, general vision–language question answering, visual grounding, and text. These data sources cover a wide spectrum of multi-modal tasks, ranging from open-ended description and instruction following to fine-grained text recognition, numerical and scientific reasoning, temporal event understanding, and precise spatial localization.

Captioning. Following recent work, we include large-scale image captioning data that pairs natural images, web photographs, and synthetic scenes with short or paragraph-level descriptions. Unlike other task types, captioning data are highly abundant on the internet. For example, COYO-700M [4] and LAION-COCO [31] offer more than 1000M image-text captioning. However, the associated captions are typically scraped web metadata (e.g., alt-text, titles, or surrounding HTML) and are often noisy, extremely short, or off-topic. To make this source more useful, we perform extensive cleaning and filtering, discarding samples with low-quality or non-linguistic text, boilerplate, or clear content mismatches. We further construct a label taxonomy to tag broad content categories and upsample captioning samples that encode factual and domain knowledge (e.g., scientific, diagrammatic, or instructional content), thereby increasing the share of knowledge-intensive data in the mixture. After these steps, the captioning portion of our pre-training corpus contains roughly 500M curated image–text pairs.

OCR, Chart and Document. We also allocate a substantial portion of data to text-rich images, including scanned documents, forms, invoices, receipts, slides, UI screenshots, tables, and charts. In this data category, each sample typically pairs a document image with questions, extraction instructions, or structured targets (e.g., key–value fields, table cells, chart values). Unlike captioning data, high-quality OCR and document-style supervision is relatively scarce on the open web, so we build data-construction pipelines, for example by applying OCR models to extract and normalize text from document images and then constructing aligned supervision from these outputs. In total, we use about 680M document-style image–text pairs of this form.

STEM-Oriented Data. To enhance mathematical and scientific reasoning capabilities, we introduce a multi-modal dataset centered on STEM (Science, Technology, Engineering, and Mathematics). These samples typically include diagrams, plots, geometric figures, experimental setups in physics, or chemical structures, paired with questions that require symbolic reasoning, unit conversion, or extraction of values from axes. Representative tasks encompass interpreting geometric diagrams, solving math word problems involving images or tables, understanding laboratory apparatus or circuit diagrams, and integrating textual problem descriptions with visual cues. The primary data formats include question-answer (QA) pairs and interleaved image-text data. In the QA dataset, we have curated approximately 15 million K12 educational problem instances, which are crucial for the learning and comprehension of core subjects, including mathematics, physics, chemistry, and biology. Additionally, we develop automated data synthesis pipelines for mathematical functions, graphical interpretation, and related reasoning tasks.

Interleaved image-text data are primarily sourced from crawled web content and parsed PDF documents, including academic books, research papers, and educational materials. For document-derived interleaved data, we applied the following processing strategies: (1) To enhance the density of visual elements in certain image-text samples, selected formulas, tables, and textual segments were preserved as image-based representations; (2) To improve model robustness in handling multilingual and structured content, certain formulas and tables were retained in LaTeX, HTML, and Markdown formats. Ultimately, we obtained 188 million interleaved image-text samples from books and 46 million from web sources. In total, we have leveraged approximately 260 million STEM-oriented multi-modal examples, significantly advancing the training and evaluation of multi-modal reasoning models in scientific and mathematical domains.

General Question Answering. We also include general-purpose vision–language QA data to cover everyday scenes. This category spans both single-turn and multi-turn dialogues, allowing the model to practice following instructions, asking clarification questions, and maintaining context while grounding its responses in visual evidence. Most of this data is sourced from open instruction-tuning corpora, complemented by a smaller portion of synthetic data, for a total of roughly 100 million examples, making it a relatively small fraction of our overall mixture.

Visual Grounding. To improve spatial precision, our training data also includes grounding-style supervision where the model must link text spans or referring expressions to explicit visual regions. This family of data covers captioning with grounding, visual grounding, grounded captioning, grounded QA, OCR with grounding, as well as pointing, counting and depth estimation with grounding. In practice, we run conventional detection and segmentation models on images to obtain region proposals and then convert them into grounding annotations. A key challenge is to express these heterogeneous geometric primitives in a unified, natural-language-like format that is both learnable and extensible. We describe the grounding label format in detail in Appendix A.1. This representation allows points and boxes to be combined within a single label while maintaining a consistent syntax. Previous work typically formulates visual grounding as localizing a single region in an image conditioned on a given textual description. In contrast, we extend visual grounding to encompass grounding (a) a single object, (b) multiple objects, or even (c) no object described by the textual input within an image. We utilize approximately 420 million visual grounding-related data for model training, including box-based and point-based data. To further improve generalization, we incorporate negative samples derived from queries targeting randomly selected objects that are absent from the image. Detailed construction of grounding data are provided in Appendix A.2.

Video Understanding. We explicitly scale to both short and long videos, using both high-quality open-source and in-house datasets that pair video clips or multi-minute recordings with dense captions and question–answer annotations. This data typically consists of video-level captions together with general-purpose video QA pairs. The tasks cover action recognition, temporal localization, temporal ordering, multi-step event reasoning, and fine-grained queries about specific frames or segments, etc. High-quality, annotated video data is relatively scarce on the open web, even though raw video content is abundant, so we adopt a frame-sampling and labeling strategy: we sample 2 frames per second (FPS) from each video and then use a in-house data generation pipeline to generate captions and QA-style supervision.

Text. In addition to multi-modal sources, we incorporate a large corpus of pure text data to maintain the language ability. This corpus is derived from various sources, including web content, books, multilingual, code, STEM (Science, Technology, Engineering, and Mathematics), industrial domains, reasoning, and synthetically generated data. We use about 250 million text data for training.

3.2 Data Processing

Pre-training data. For all data modalities, we adopt a multi-stage data pre-processing pipeline.

1. **LLM-based text filtering.** We first perform LLM-based text filtering to remove low-quality content, and explicitly target "loop-like" repetitive patterns using document-level deduplication and LLM-based repetition detectors, following the observation from InternVL 2.5 that even a few thousand anomalous repetitive samples can noticeably degrade model behavior.

2. **Ruled-based image filtering.** We apply a rule-based pipeline that discards images that are blank or nearly blank, fall below a minimum resolution, exhibit extreme aspect ratios, or show heavy compression artifacts. Unsafe or NSFW content is removed via an automatic classifier combined with heuristic rules.
3. **Validation of multi-modal annotations.** All multi-modal annotations are checked against a unified schema. For example, for grounding and pointing data, we require: (i) special tags (e.g., object-reference markers and coordinate spans) appear in balanced pairs; (ii) coordinates fall within the corresponding image or frame; and (iii) any structured outputs (such as JSON bounding boxes or point lists) can be correctly parsed.

SFT Data. The SFT mixture includes dialog-style instruction data, reasoning-with-explanations, safety-aligned conversations, and task-specific formats (e.g., JSON, grounding tags). Building on the above pre-training preprocessing, we adopt a stricter workflow for SFT data, with a particular focus on instruction-following quality.

1. **Manual spot-checks and LLM-based refinement.** Emphasizing curated instruction data for alignment, we treat each SFT dataset as a dialog-style collection in which user instructions and assistant responses must be tightly coupled. For every dataset in the mixture, we conduct manual spot-checks to verify that responses satisfy explicit formatting and behavioral constraints specified in the prompt—for example, whether a "short answer only" instruction is respected, or whether a creative-writing prompt requesting an 800-word essay remains within the required length and style. LLM-based automatic refinement is used to further improve consistency and reduce annotation noise.
2. **Chain-of-thought enhancement.** We incorporate chain-of-thought (CoT) style multi-modal SFT data to encourage step-by-step reasoning while still adhering to user-imposed constraints. This promotes more structured reasoning traces and improves instruction-following robustness.
3. **Data adjustment based on training accuracy.** The SFT-trained model is further used to run inference on the SFT datasets, enabling a comparative analysis of accuracy between subsets used during training and the held-out control splits. This evaluation is essential for identifying datasets that exhibit large accuracy differentials, thereby guiding targeted data augmentation to ensure sufficient coverage and improved model performance.

RL Data. For the RL stage, we curate the training dataset from both open-source and proprietary sources. The open-source data is derived from ViRL39K [36], which has been labeled by pass rate generated by Qwen2.5-VL-7B [3]. The proprietary data consists of two parts. The first part involves filtering queries suitable for answer extraction and rule-based judgment from unused SFT data, which are then transformed into RL data format. These data cover STEM, General QA, Chart and Table, and other domains, and have undergone rigorous filtering and manual annotation to ensure all answers are correct and verifiable. The second part of the proprietary data is generated by sampling and filtering from previously used SFT data of knowledge domains, which are also converted into RL data format through the same process. These data are mainly intended to reinforce the knowledge learned in previous stages and further enhance related competencies. For all data, we use a preliminary checkpoint of our SFT model to sample 8 responses per query and only queries with an accuracy below or equal to 75% are retained to ensure effective supervision during training. Finally, we construct approximately $35,000$ samples for verifiable reinforcement learning training.

4 Training Recipe

4.1 Pre-Training Paradigm

We follow a three-stage curriculum for multi-modal pre-training, broadly aligned with mainstream VLM training pipelines [2, 13, 42].

Warm-up. In the first stage, we warm up the vision–language connector while freezing both the ViT and the LLM backbone. The model is exposed to roughly 35B tokens of image–text pairs, and only the lightweight connector is trainable. This stage focuses on stabilizing the mapping from visual features to the LLM token space, so that the subsequent joint training can start from a well-aligned initialization without corrupting the pre-trained language or vision representations.

Table 2: **Detailed configuration for different training stages of openPangu-VL-7B.** The table illustrates the vision configurations, dataset characteristics, and training hyperparameters.

	Settings	Warm-up	Pretraining	Annealing	SFT
Vision	**Resolution**	Min 56^2, Max 1344^2	Min 56^2, Max 1344^2	Min 56^2, Max 1344^2	Min 56^2, Max 1344^2
	# Tokens	Min 4, Max 2304	Min 4, Max 2304	Min 4, Max 2304	Min 4, Max 2304
Data	**Dataset**	∼35B	∼3.2T	∼430B	∼40B
	# Samples	∼65M	∼2270M	∼130M	∼27M
Training	**Trainable**	Connector	Full Model	Full Model	Full Model
	Batch Size	1024	1024	128	128
	Context Length	24576	24576	131072	131072
	Start LR	5×10^{-4}	3×10^{-5}	2.5×10^{-6}	2×10^{-5}
	End LR	3×10^{-5}	3×10^{-6}	2.5×10^{-6}	2×10^{-6}
	Encoder LR coeff	-	1.0	1.0	0.1
	Epoch	1	1	1	2

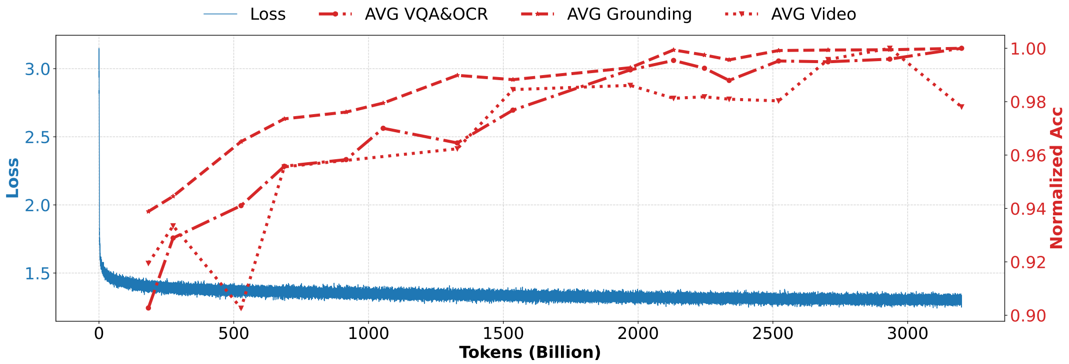

Figure 3: Loss curve and normalized average accuracy on VQA&OCR, grounding and video understanding tasks in the multi-modality pretraining stage.

Multi-modality Pretraining. In the second stage, we unfreeze all major components, including the ViT, the projector, and the LLM, and perform large-scale joint training on about 3.2T tokens of interleaved text, image, document, chart, and video data. This long-run phase is responsible for building the core multi-modal capabilities of the model: robust perception, cross-modal alignment, and general reasoning across diverse tasks such as captioning, OCR, document understanding, STEM diagrams, video understanding, and grounding. Compared with the Warm-up stage, optimization now adjusts both visual and language representations, allowing the system to co-adapt to multi-modal inputs while preserving the strong language prior from the base LLM. The loss curve and average accuracy trend in this training stage are shown in Figure 3. We can find that the normalized average accuracy on benchmarks is scaling up stably as the loss decreases steadily.

Annealing with Longer Context and Denser Knowledge. After the bulk of multi-modal capacity has been established, we introduce an annealing stage that serves as a bridge between generic pre-training and alignment. In this phase (about 430B tokens), we increase the maximum sequence length, enrich the data mixture with higher-knowledge-density samples (e.g., more document, STEM, and knowledge-intensive content), and raise the proportion of instruction-following style data. We also increase the share of pure text data so that the model further consolidates its language skills under long-context settings. This stage refines the model's ability to follow complex prompts over long sequences and to integrate dense factual or procedural knowledge, while still training all multi-modal modules jointly.

4.2 Post-Training

SFT for Instruction Following. We perform supervised fine-tuning on roughly 40B tokens of high-quality SFT data, focusing on instruction following, output format control, and response quality. At this stage, the LLM backbone and projector remain fully trainable, while the ViT is updated with a learning rate that is one order of magnitude smaller than that of the LLM. This asymmetric schedule preserves the visual encoder's mature representations, yet allows small adjustments to better support instruction-driven multi-modal tasks. The goal is no longer to expand raw capability, but to shape the model into a reliable assistant that follows instructions faithfully, produces safe and concise outputs, and leverages its multi-modal knowledge in a controlled manner.

DPO. Following SFT, we apply a standard DPO [29] procedure to reduce hallucinations and better align the model's preferences. We first use the SFT-trained model to generate rollouts for a selected subset of the SFT corpus. This subset primarily contains queries with multiple plausible correct answers; relying on a single reference response during SFT can limit the model's ability to explore alternative solutions. For each query, we generate four candidate responses and score them using proprietary evaluators on a 1–5 scale. The scoring criteria cover hallucination level, similarity to ground truth, linguistic quality, and consistency with the visual input. If a query contains at least one response scored above four and another scored below two, we construct a positive–negative pair for DPO training. In total, we use roughly 50,000 such preference pairs, covering both multi-modal and pure-text data. To maintain broader data coverage, we also proportionally mix in the remaining SFT data and train it with the standard next-token prediction (NTP) loss, while the preference data are optimized with the preference and generation losses defined in [37].

GRPO. To further enhance the reasoning capabilities, we adopt Reinforcement Learning with Verifiable Reward (RLVR), primarily targeting the fields of STEM (Science, Technology, Engineering and Mathematics). We employ GRPO [32] algorithm for RL training. We also incorporate several improvements from DAPO [45], including clip-higher and the removal of KL divergence. The reward system comprises the format reward and the accuracy reward. Following openPangu-Embedded-1B-V1.1, we use a training paradigm suitable for the fast-thinking model. The format reward only requires the model to place the answer within *boxed{}*, enabling it to better inherit the capabilities of the preceding model. The accuracy reward is calculated with a rule-based verifier based on the provided ground truth. The hyperparameters for RL training are configured as follows: learning rate of 10^{-6}, global batch size of 512, mini batch size of 2048, 8 rollout responses per query, temperature of 1.0, and sequence length of 24576.

Iterative Fine-Tuning. After the RL stage, we adopt iterative fine-tuning to further enhance the model capabilities. In this phase, we employ a reject sampling method to generate fine-tuning training data, aiming to gather challenging prompts that the model has not yet learned effectively during the previous post-training stage. Specifically, after the RL model is released, we process the SFT dataset by performing multiple rollouts for each query. The similar scorer used in the DPO phase is utilized to evaluate the rollout results. Rollouts scoring 5 are considered as confirming the correctness of the response. The score distribution of multiple rollouts for each query is then used to evaluate the query difficulty. Queries with at least two rollouts scoring below 2 are considered challenging. Finally, we select from the challenging queries those that contain at least one response scoring 5 to form the fine-tuning dataset, totaling approximately $20k$ examples. Fine-tuning is performed using the same training procedure as SFT, but with a smaller learning rate, starting at 10^{-6} and decaying to 10^{-7}.

4.3 Model Merging

During the post-training stage, we adjust the data recipe multiple times to find a balance among the various capabilities that we want our model to possess. As a result, we obtain several models. Among them, we select three models with diverse capability distributions and perform an arithmetic average of their model weights to attain one final model. We find that this simple merging technique can further boost the model's performance (see Figure 6). We attribute this performance improvement to a smoother feature space and enhance model robustness, and consider it an existing direction that is worth further exploration.

4.4 Balance of Per-sample and Per-token Loss

In the annealing and SFT stages, the length distribution of the training samples is highly imbalanced. We observed that the performance of short-response samples is easily affected by that of long-response samples (e.g., long textual response data). Therefore we combine the per-sample loss and per-token loss to balance the performance of long and short samples. Additionally, we further re-weight the loss of each token based on its position (order) and loss magnitude, enabling the model to focus more on earlier tokens and tokens with high

Table 3: Comparison of openPangu-VL-7B with other models on Text-Centric Benchmarks.

Benchmark	openPangu VL-7B	openPangu-Embedded 7B-V1.1 no-thinking	Qwen3-VL 8B Instruct
MMLU-Pro	**78.2**	69.6	71.6
MMLU-Redux	**87.3**	-	84.9
GPQA-Diamond	**65.2**	60.1	-
IF-Eval	83.0	-	**83.7**
C-Eval	**83.2**	78.5	-

loss values. The overall balance loss $\mathcal{L}$ is defined as:

$$\mathcal{L}_t = \frac{\Sigma_j \Sigma_i (w_{ji} \cdot \mathcal{L}_{ji})}{\Sigma_j \Sigma_i w_{ji}}, \mathcal{L}_s = \frac{1}{J} \Sigma_j \frac{\Sigma_i (w_{ji} \cdot \mathcal{L}_{ji})}{\Sigma_i w_{ji}}, \mathcal{L} = \lambda_1 \mathcal{L}_t + \lambda_2 \mathcal{L}_s, \tag{1}$$

where $\mathcal{L}_{ji}$ denotes the loss of the i-th token in the j-th sample within a piece of packed data (containing a total of J samples), w_{ji} is the weight of i-th token in the j-th sample, which is the sum of an order-based weight w_{ji}^{ord} and a magnitude-based weight w_{ji}^{mag}. Here, λ_1 and λ_2 are hyper-parameters that control the relative importance of per-token loss and per-sample loss, respectively. The order-based weight is calculated as: $w_{ji}^{ord} = ln(e + \tau_1 \cdot i)^{-1}$, where τ_1 is the temperature parameter (default valuse: 0.1). The magnitude-based weight is calculated by: $w_{ji}^{mag} = 2Sig(\tau_2 \cdot \mathcal{L}_{ji} - 0.5)$, where τ_2 is another temperature parameter (default value: 2) and Sig is the sigmoid function.

5 Benchmark Performance

5.1 Main Results

Evaluation of Text-Centric Abilities. We adopt MMLU-Pro [39], MMLU-Redux [12], GPQA-Diamond [30], IF-Eval [49], and C-Eval [14] to cover a broad spectrum of text-centric abilities, as shown in Table 3. On these benchmarks, openPangu-VL-7B largely preserves the language capabilities of its text-only backbone while closing much of the gap to stronger baselines. Compared with openPangu-Embedded-7B-V1.1 (no-thinking), openPangu-VL-7B shows consistent gains on knowledge and reasoning benchmarks such as MMLU-Pro and C-Eval, and slightly improves GPQA-Diamond, indicating that introducing the vision stack does not degrade – and can even enhance – general language understanding. Relative to Qwen3-VL-8B Instruct, openPangu-VL-7B achieves comparable or better performance on challenging knowledge benchmarks, outperforming it on MMLU-Pro and MMLU-Redux, while trailing slightly on instruction-following as measured by IF-Eval.

Evaluation of Visual Understanding Abilities. Across a broad spectrum of visual benchmarks, openPangu-VL-7B delivers competitive general-purpose visual understanding even when compared with some slow or hybrid thinking models. On general VQA benchmarks (MMBench [23], AI2D [18], RealWorldQA [41], MMStar [6]), openPangu-VL-7B performs largely on par with Qwen3-VL-8B Instruct [2]: it slightly trails on AI2D and MMStar, but matches or surpasses Qwen3-VL on MMBench and RealWorldQA, indicating comparable overall general-vision QA capability. In OCR, chart, and document understanding (OCRBench [24], TextVQA [34], ChartQA, DocVQA [27], CharXiv [40]), openPangu-VL-7B matches or slightly outperforms Qwen3-VL-8B and consistently outperforms most other competitive baselines, showing that introducing visual modules does not compromise, and often strengthens, text-heavy perception skills. By contrast, on STEM-oriented benchmarks (MMMU [46], MMMU-Pro [47], MathVista [25]), openPangu-VL-7B lags Qwen3-VL-8B and InternVL 3.5 by a few points, revealing remaining gaps in multi-modal scientific and mathematical reasoning. For multi-image reasoning (BLINK [11], MUIRBench [35]), openPangu-VL-7B is competitive with Qwen3-VL-8B while clearly outperforming other open competitive models, and, together with its strong performance on grounding and counting benchmarks such as RefCOCO [44], demonstrates a particular strength in aggregating information across views and localizing fine-grained regions. In video understanding, openPangu-VL-7B exhibits a clear advantage on short-form video reasoning, outperforming Qwen3-VL-8B on MVBench [20], but still falls slightly behind on longer-horizon datasets such as VideoMME [10] and MLVU [48], suggesting that modeling long-range temporal dependencies remains an important direction for future improvement.

Table 4: Comparison of openPangu-VL-7B with other models on Visual Benchmarks. The best result for each metric is highlighted in **bold**, and the second best one is highlighted with underline.

	Fast Thinking			**Slow & Hybrid Thinking**		
Benchmark	**openPangu VL-7B**	**Qwen3-VL 8B Instruct**	**MiMo-VL 7B-RL-2508 Non-Thinking**	**Keye-VL-1.5 8B-Auto-Think**	**InternVL 3.5 8B**	**MiniCPM-V 4.5 8B**
General VQA						
MMBench$_{\text{V1.1\_DEV}}$	**86.5**	85.0	-	-	79.5	84.2
AI2D$_{\text{test}}$	84.7	85.7	-	**88.6**	84.0	86.5
RealWorldQA	**76.1**	71.5	-	71.1	67.5	72.1
MMStar	70.1	70.9	-	**75.5**	69.3	72.1
OCR & Chart / Document Understanding						
OCRBench	**907**	896	881	833	840	890
TextVQA	**85.1**	-	-	-	78.2	82.2
ChartQA	88.3	89.6	**91.0**	-	86.7	87.4
DocVQA$_{\text{test}}$	96.0	**96.1**	-	-	92.3	94.7
CharXiv$_{\text{DQ}}$	**83.9**	83.0	-	-	72.2	-
CharXiv$_{\text{RQ}}$	**54.3**	46.4	-	-	44.4	-
STEM						
MMMU$_{\text{val}}$	65.2	69.6	63.1	63.4	**73.4**	67.7
MMMU-Pro$_{\text{overall}}$	52.6	**55.9**	-	-	-	-
MathVista$_{\text{mini}}$	75.0	77.2	-	70.9	78.4	**79.9**
Multi-Image						
BLINK$_{\text{val}}$	63.3	**69.1**	-	52.5	59.5	-
MUIRBench	61.6	**64.4**	-	-	55.8	-
Grounding & Counting						
RefCOCO-avg	**90.6**	89.1	90.3	-	89.7	-
ODinW-13	**51.5**	44.7	-	-	-	-
Point-Bench	**65.4**	63.4	-	-	-	-
CountBench	**96.1**	80.5	-	-	-	-
Video Understanding						
MVBench	**74.0**	68.7	-	-	72.1	-
VideoMME$_{\text{w/o sub}}$	68.0	**71.4**	70.1	59.7	66.0	67.9
MLVU	76.9	**78.1**	-	-	70.2	75.1

5.2 Ablation Study

In this section, we systematically examine how different training paradigms, data compositions, grounding strategies, and video modeling choices influence the performance of our multi-modal model. Unless otherwise specified, all experiments are conducted with the same model architecture and training budget. The results consistently reveal several design principles that guide the construction of large-scale multi-modal pre-training systems.

Table 5: Ablation study of grounding encoding.

Norm	**Padding**	**3-token**	**rA**	**rB**	**rVal**	**AVG**
×	×	×	88.8	81.7	86.6	85.7
×	×	✓	87.3	80.7	85.7	84.6
✓	×	×	90.9	83.3	88.1	87.4
✓	×	✓	91.2	85.0	89.1	88.4
✓	✓	✓	91.2	85.8	89.0	88.7

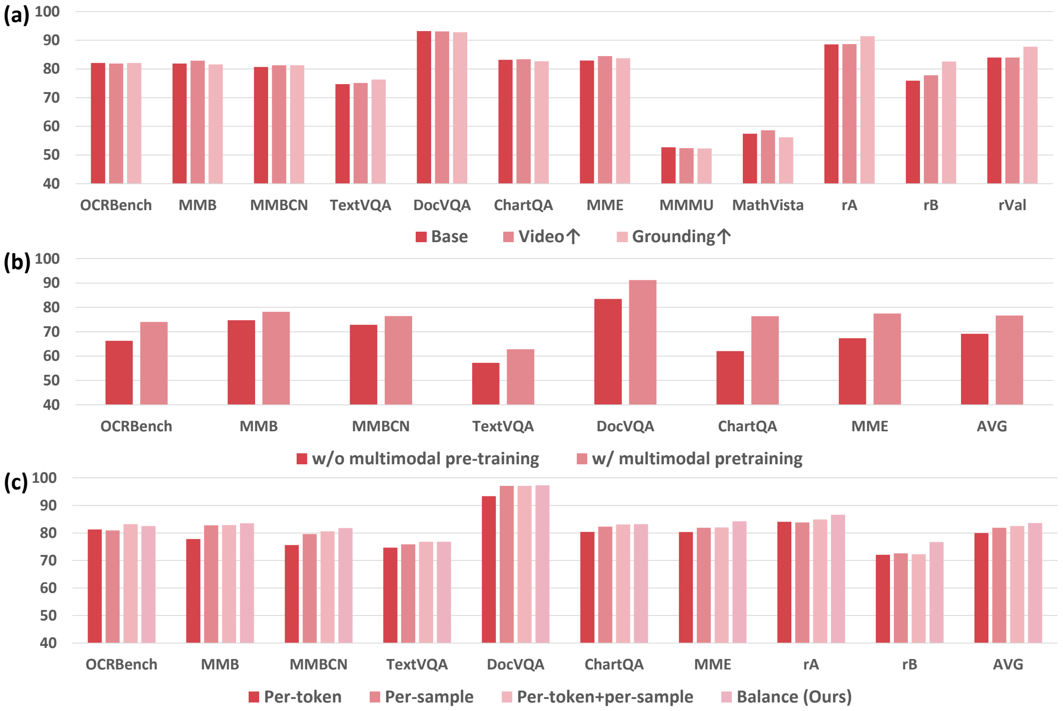

Figure 4: (a) Effects of pre-training data composition. (b) Effect of pre-training openPangu-ViT with LLM on image-caption tasks. (c) Performance of the model with various losses.

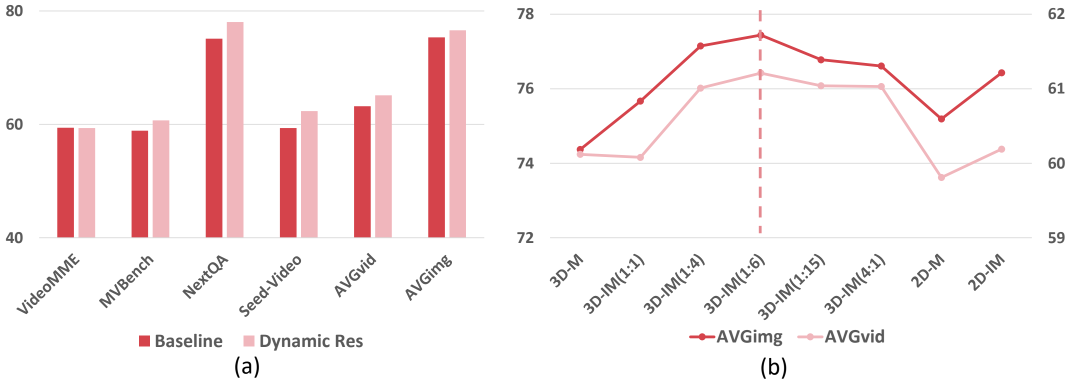

Figure 5: (a) Effect of dynamic resolution on video and image tasks. (b) Effect of IM-RoPE on image and video tasks. Ablation explores the proportion of temporal and spatial (i.e., t:h+w) dimensions.

5.2.1 Data Composition

We conduct some experiments to show the effectiveness of different data composition, shown in Figure 4(a).

Observation. Increasing domain-specific data (e.g., video, localization) does not harm general image–text performance. Increasing grounding data from 7% to 14% results in more than 3% improvement across RefCOCO splits, with no degradation on general vision–language tasks. This confirms that grounding tasks are highly data-dependent and respond strongly to increased spatial supervision. Besides, introducing more video data yields about 1% gain on average image–text benchmarks.

5.2.2 Training Paradigm

Observation 1. Pre-training with LLM increases the multi-modal understanding capability of vision encoder. Since openPangu-ViT is trained from scratch in a multi-label contrastive learning style, it is of vital importance to build a bridge between images and text through multimodal data training. Experiments on a carefully sampled pilot dataset presented in Figure 4(b) shows that, with pre-training the vision encoder together with openPangu-Embedded-7B-V1.1 base model, the performance is improved by 7.5% on average.

Observation 2. Unification of per-sample loss and per-token loss improves short-response tasks. In Figure 4(c), we compare the performance of different loss types across multiple datasets. Results show that a combination of per-token loss and per-sample loss improves the overall performance of these tasks, while the proposed strategy, incorporating order-based and loss-based reweighting, can further boost performance.

Observation 3. Performance monitoring at different training stages. In Table 6, we record the performance changes of one of the candidate models that we ultimately merge during its training process. As we can see, the pre-training stage continuously boosts the model's general visual capabilities, while the post-training stage can further enhance the model, especially its advanced capabilities.

Observation 4. Model averaging improves the model's overall capabilities. As shown in Figure 6, the model merging operation introduced in Section 4.3 can integrate the strengths of diverse models. During the post-training stage, model-1/2/3 are trained with different focus, thus excelling in grounding capability, general capability, and STEM respectively. After merging, the obtained final model is able to inherit the strengths of different configurations.

5.2.3 Grounding Representation Strategy

Observation 1. Multi-token coordinate encoding yields consistently better grounding performance. As shown in Table 5, encoding three-digit coordinate numbers into independent tokens consistently outperforms treating them as a single token. The advantage is most pronounced on RefCOCO-testB split, which features scenes with more complex spatial relations. Since each digit (hundreds, tens, units) conveys unique semantic information, tokenizing coordinates digit-by-digit (e.g., splitting '123' into '1', '2', '3') preserves fine-grained structure, facilitating more effective learning under the next-token prediction paradigm.

Observation 2. Zero-padded three-digit formatting stabilizes coordinate learning. Using normalized coordinates formatted as "008" rather than "8" yields slightly higher scores. The padding improves distribution regularity and simplifies token-to-value mapping.

Observation 3. Relative coordinates outperform absolute coordinates. Relative (normalized) coordinate encoding yields significantly stronger grounding performance than absolute pixel-based coordinates. Normalization mitigates scale variation and leads to better generalization across heterogeneous image resolutions.

5.2.4 Video Resolution and Encoding Strategy

Observation 1. Video dynamic resolution improves temporal understanding with no cost to image quality. As shown in Figure 5(a), the dynamic-resolution mechanism substantially enhances MVBench and NextQA (e.g., +2.6 and +2.9), while maintaining or slightly improving image-based metrics. Dynamic scaling regularizes the visual backbone and better adapts to naturally varying video frame resolutions.

Observation 2. Hybrid 3D high/low-frequency modeling balances image and video performance. We compare three positional modeling strategies: 2D HW-interleaved, 3D THW non-interleaved, and 3D THW-interleaved high/low-frequency decomposition. The 2D design is superior for image tasks, and the pure 3D design excels at temporal reasoning. As shown in Figure 5(b), the hybrid THW-interleaved approach provides the best overall trade-off and is therefore adopted as our final configuration.

Stage	**ocr**	**mmb**	**mmb$_{cn}$**	**textvqa**	**docvqa$_{val}$**	**chartqa**	**mme**	AVG_7
Pretraining	832.0	76.6	75.1	80.3	93.0	83.1	2198.8	82.3
Annealing	863.0	85.0	82.2	82.0	95.3	85.5	2266.9	86.2
SFT	908.0	85.7	85.7	82.6	97.0	85.6	2248.8	87.7
DPO	910.0	86.3	85.7	83.2	96.9	86.0	2287.3	88.2
GRPO	917.0	86.3	85.6	82.8	96.7	85.8	2286.1	88.1
Iterative	915.0	86.3	85.7	82.8	96.6	85.8	2261.0	88.0
Stage	**rA**	**rB**	**rVal**	AVG_3	**mmmu**	**mmmupro**	**mathvista**	AVG_3
Pretraining	93.5	87.3	91.6	90.8	-	-	-	-
Annealing	94.5	88.5	92.6	91.9	-	-	-	-
SFT	94.3	88.9	91.8	91.7	58.2	44.7	61.0	54.6
DPO	94.4	89.3	92.0	91.9	59.2	44.9	65.4	56.5
GRPO	94.5	89.2	92.1	91.9	63.9	47.9	71.6	61.1
Iterative	94.5	89.4	92.1	92.0	63.0	49.7	75.2	62.6

Table 6: Performance trends across different training stages.

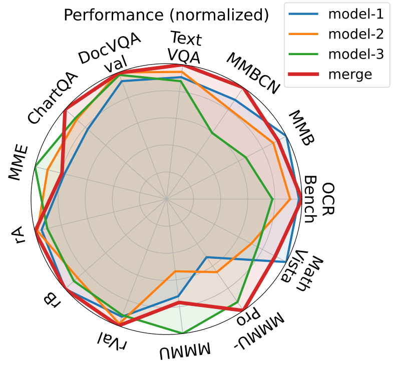

Figure 6: Performance comparison before and after model merge.

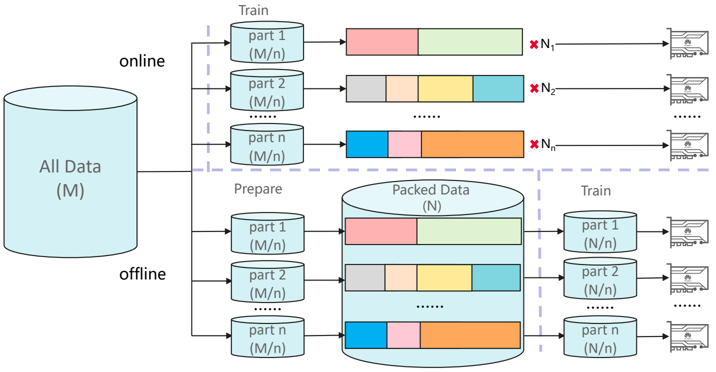

Figure 7: Compare of offline and online mixing and packing scheme

6 Training and Inference Infrastructure

6.1 Training Optimization

The core objective of multi-modal large model training lies in simultaneously enhancing model performance and training efficiency, which are highly dependent on data, parallelization strategies, and underlying computational kernels. Based previous training framework, we analyze data distribution, the time consumption of parallel strategies and the model computation path. Subsequently, we implement systematic optimizations targeting these aspects which synergistically improve the training of multi-modal large models and train openPangu-VL-7B with an MFU of 42.5% on Ascend Atlas 800T A2 clusters.

Offline Mixing and Packing Scheme. We observe that online data mixing and packing during training leads to a significant imbalance in the number of packed samples across Data Parallelism. This imbalance could introduce training bias and result in underutilization of data. To address this issue, we redesigned the data processing pipeline, transitioning from an online paradigm to an offline paradigm. As shown in Figure 7, Offline Data Mixing and Packing Scheme performs data mixing and sequence packing operations before the training phase begins. The packed samples are stored locally and loaded/reassigned during training, thereby resolving the data quantity imbalance. To accelerate this preparatory stage, we implemented the following key optimizations within the offline scheme:

- Token Calculation Based on Metadata: During offline packing, the number of image tokens is derived directly from metadata (image width, height, frame rate), eliminating the need to decode and transform actual images. This removes expensive I/O and image scaling operations, accelerating the packing process by approximately 4x compared to the baseline online method.

- Scalability: The offline packing process inherently supports distributed execution and multi-processing, enabling efficient handling of large-scale datasets.

Balanced Encoder Data Parallelism for ViT. The heterogeneity of multi-modal models and the complexity of the ViT Any-Resolution Schema (e.g., large resolution variations) collectively lead to an imbalanced workload in the ViT encoder during training. To alleviate this imbalance, we introduce a Balanced Encoder Data Parallelism strategy within the TP and CP communication groups of the LLM. Specifically, we comprehensively consider workload based on image resolution to allocate sequences across devices, ensuring a more balanced workload partitioning for ViT and thereby improving overall training efficiency.

Asynchronous Gradient Accumulation. Traditional distributed training with gradient accumulation typically requires inter-device communication (e.g., gradient synchronization) after processing every micro-batch within a gradient accumulation step. This introduces significant communication overhead and performance degradation within the global batch. To mitigate this issue, we implemented Asynchronous Gradient Accumulation. Global gradient synchronization is performed only once, after the final micro-batch of the global batch is processed. This approach significantly reduces the number of communication rounds and eliminates the performance degradation associated with per-micro-batch synchronization, while maintaining mathematical equivalence.

Kernel Optimizations. To maximize training throughput, we utilize Ascend fused operator kernels on several critical computations, including Fused COC, RMSNorm and Fused RoPE. Simultaneously, we performed a series of compute-efficient operator replacements (e.g., replacing 3D convolution with batch matrix multiplication).

6.2 Inference Optimization Based on vLLM

In our performance evaluation of multi-modal processing within the vLLM framework, we observed a phenomenon where increasing image resolution leads to significantly higher Time-To-First-Token (TTFT) latency, far surpassing the inference latency of text-only inputs with equivalent token lengths. This difference highlights the performance overhead caused by high-resolution visual data processing in current multi-modal inference systems.

This phenomenon stems from the inter-process communication (IPC) mechanism in the vLLM architecture, where executor processes send requests to worker processes via message queues. Despite employing shared memory optimization techniques, as image sizes continue to grow, the communication overhead also increases significantly, becoming a critical path bottleneck when generating the first token. To address this challenge, we propose a two-stage optimization method:

- Computation-Communication Co-Design: By migrating image preprocessing operations (rescaling and normalization) into the model's forward pipeline, we effectively reduce tensor payloads from FP32 to UINT8 precision. This transformation, combined with NPU-accelerated computation, achieves 4× communication volume reduction.
- Structural Optimization: We integrate Vision Transformer (ViT) temporal patching into the model's forward pipeline, eliminating redundant preprocessing steps. This restructuring further reduced the need for inter-process communication, achieves 2× communication volume reduction.

In addition, we demonstrate performance improvements optimized for the Ascend NPU, specifically reflected in the following aspects:

- Communication-Computation Co-Scheduling: Enables asynchronous overlapping of cross-device communication and matrix computation through NPU fusion kernel, effectively hiding data transfer latency across multiple accelerators.
- Operator Fusion: add with RMSNorm, 3D-IM-RoPE, and flash attention operations.

On Ascend Atlas 800T A2, we conduct performance tests on the openPangu-VL-7B model. When processing a single 1080p resolution image, the model's forward inference latency during the prefill stage was approximately 350ms; when processing a single 720p resolution image, the forward inference latency was approximately 161ms. During the decoding stage, the forward inference latency was approximately 16ms. This time does not include the performance overhead of the inference framework and only refers to the operator execution time of the model on the NPU.

7 Conclusion

We develop a fast-thinking multi-modal model with a scale of 7B parameters based on Ascend NPU and the openPangu series of language models. This model demonstrates capabilities comparable to those of the Qwen3-VL model of the same scale and generation in general visual question answering (VQA), grounding tasks, OCR and chart/document understanding, multi-image and video understanding. In this technical report, we introduce our approach to designing the model architecture, data construction methods, overall training process, evaluation results, and the design of the training and inference systems. We also share some preliminary experimental results to help others better build general-purpose multi-modal models on Ascend NPUs. In the next step, we will continue to enhance the model's STEM capabilities, develop slow-thinking models, and integrate fast and slow thinking models to achieve breakthroughs on more challenging evaluation datasets.

References

[1] Xiang An, Kaicheng Yang, Xiangzi Dai, Ziyong Feng, and Jiankang Deng. Multi-label cluster discrimination for visual representation learning. In *ECCV*, 2024.

[2] Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-VL technical report. *arXiv preprint arXiv:2511.21631*, 2025.

[3] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2.5-VL technical report. *arXiv preprint arXiv:2502.13923*, 2025.

[4] Minwoo Byeon, Beomhee Park, Haecheon Kim, Sungjun Lee, Woonhyuk Baek, and Saehoon Kim. COYO-700M: Image-text pair dataset. `https://github.com/kakaobrain/coyo-dataset`, 2022.

[5] Zhipeng Cai, Ching-Feng Yeh, Hu Xu, Zhuang Liu, Gregory Meyer, Xinjie Lei, Changsheng Zhao, Shang-Wen Li, Vikas Chandra, and Yangyang Shi. DepthLM: Metric depth from vision language models. *arXiv preprint arXiv:2509.25413*, 2025.

[6] Lin Chen, Jinsong Li, Xiaoyi Dong, et al. Are we on the right way for evaluating large vision-language models? *arXiv preprint arXiv:2403.20330*, 2024.

[7] Long Cheng, Jiafei Duan, Yi Ru Wang, Haoquan Fang, Boyang Li, Yushan Huang, Elvis Wang, Ainaz Eftekhar, Jason Lee, Wentao Yuan, et al. PointArena: Probing multimodal grounding through language-guided pointing. *arXiv preprint arXiv:2505.09990*, 2025.

[8] Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025.

[9] Haodong Duan, Junming Yang, Yuxuan Qiao, Xinyu Fang, Lin Chen, Yuan Liu, Xiaoyi Dong, Yuhang Zang, Pan Zhang, Jiaqi Wang, et al. Vlmevalkit: An open-source toolkit for evaluating large multi-modality models. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 11198–11201, 2024.

[10] Chen Fu et al. Video-MME: The first-ever comprehensive evaluation benchmark of multi-modal llms in video analysis. In *CVPR*, 2025.

[11] Xingyu Fu, Yushi Hu, Bangzheng Li, Yu Feng, Haoyu Wang, Xudong Lin, Dan Roth, Noah A Smith, Wei-Chiu Ma, and Ranjay Krishna. BLINK: Multimodal large language models can see but not perceive. In *ECCV*, 2024.

[12] Aryo Pradipta Gema, Joshua Ong Jun Leang, Giwon Hong, Alessio Devoto, Alberto Carlo Maria Mancino, Rohit Saxena, Xuanli He, Yu Zhao, Xiaotang Du, Mohammad Reza Ghasemi Madani, Claire Barale, Robert McHardy, Joshua Harris, Jean Kaddour, Emile van Krieken, and Pasquale Minervini. Are we done with MMLU? *arXiv preprint arXiv:2406.04127*, 2024.

[13] Dong Guo, Faming Wu, Feida Zhu, Fuxing Leng, Guang Shi, Haobin Chen, Haoqi Fan, Jian Wang, Jianyu Jiang, Jiawei Wang, et al. Seed1.5-VL technical report. *arXiv preprint arXiv:2505.07062*, 2025.

[14] Yuzhen Huang, Yuzhuo Bai, Zhihao Zhu, Junlei Zhang, Jinghan Zhang, Tangjun Su, Junteng Liu, Chuancheng Lv, Yikai Zhang, Jiayi Lei, Yao Fu, Maosong Sun, and Junxian He. C-Eval: A multi-level multi-discipline chinese evaluation suite for foundation models. In *NeurIPS*, 2023.

[15] Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. GPT-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.

[16] Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, et al. OpenAI o1 system card. *arXiv preprint arXiv:2412.16720*, 2024.

[17] Sahar Kazemzadeh, Vicente Ordonez, Mark Matten, and Tamara Berg. Referitgame: Referring to objects in photographs of natural scenes. In *EMNLP*, 2014.

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

[19] David B Kirk and W Hwu Wen-Mei. *Programming massively parallel processors: a hands-on approach*. Morgan Kaufmann, 2016.

[20] Kunchang Li, Yali Wang, Yinan He, Yizhuo Li, Yi Wang, Yi Liu, Zun Wang, Jilan Xu, Guo Chen, Ping Luo, et al. MVBench: A comprehensive multi-modal video understanding benchmark. In *CVPR*, 2024.

[21] Liunian Harold Li, Pengchuan Zhang, Haotian Zhang, Jianwei Yang, Chunyuan Li, Yiwu Zhong, Lijuan Wang, Lu Yuan, Lei Zhang, Jenq-Neng Hwang, Kai-Wei Chang, and Jianfeng Gao. Grounded language-image pre-training. In *CVPR*, 2022.

[22] Xiaoyao Liang. *Ascend AI Processor Architecture and Programming: Principles and Applications of CANN*. Elsevier, 2020.

[23] Yuan Liu, Haodong Duan, Yuanhan Zhang, et al. MMBench: Is your multi-modal model an all-around player? In *ECCV*, 2024.

[24] Yuliang Liu, Zhang Li, Mingxin Huang, Biao Yang, Wenwen Yu, Chunyuan Li, Xu-Cheng Yin, Cheng-Lin Liu, Lianwen Jin, and Xiang Bai. OCRBench: on the hidden mystery of OCR in large multimodal models. *Science China Information Sciences*, 2024.

[25] 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.

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

[27] Minesh Mathew, Dimosthenis Karatzas, and CV Jawahar. DocVQA: A dataset for vqa on document images. In *WACV*, 2021.

[28] Roni Paiss, Ariel Ephrat, Omer Tov, Shiran Zada, Inbar Mosseri, Michal Irani, and Tali Dekel. Teaching CLIP to count to ten. In *ICCV*, 2023.

[29] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D. Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *NeurIPS*, 2023.

[30] David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. GPQA: A graduate-level google-proof Q&A benchmark. *arXiv preprint arXiv:2311.12022*, 2023.

[31] Christoph Schuhmann, Richard Vencu, Romain Beaumont, Robert Kaczmarczyk, Clayton Mullis, Aarush Katta, Theo Coombes, Jenia Jitsev, and Aran Komatsuzaki. Laion-400M: Open dataset of clip-filtered 400 million image-text pairs. *arXiv preprint arXiv:2111.02114*, 2021.

[32] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

[33] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012.

[34] Amanpreet Singh, Vivek Natarajan, Yu Jiang, Xinlei Chen, Alan Yuille, Devi Parikh, and Harsh Agrawal. Towards vqa models that can read. In *CVPR*, 2019.

[35] Fei Wang, Xingyu Fu, James Y Huang, Zekun Li, Qin Liu, Xiaogeng Liu, Mingyu Derek Ma, Nan Xu, Wenxuan Zhou, Kai Zhang, et al. Muirbench: A comprehensive benchmark for robust multi-image understanding. *arXiv preprint arXiv:2406.09411*, 2024.

[36] Haozhe Wang, Chao Qu, Zuming Huang, Wei Chu, Fangzhen Lin, and Wenhu Chen. Vl-rethinker: Incentivizing self-reflection of vision-language models with reinforcement learning. *arXiv preprint arXiv:2504.08837*, 2025.

[37] Weiyun Wang, Zhe Chen, Wenhai Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Jinguo Zhu, Xizhou Zhu, Lewei Lu, Yu Qiao, and Jifeng Dai. Enhancing the reasoning ability of multimodal large language models via mixed preference optimization. *arXiv preprint arXiv:2411.10442*, 2024.

[38] Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, et al. InternVL3.5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *arXiv preprint arXiv:2508.18265*, 2025.

[39] Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhu Chen. MMLU-Pro: A more robust and challenging multi-task language understanding benchmark. In *NeurIPS*, 2024.

[40] Zirui Wang, Mengzhou Xia, Luxi He, Howard Chen, Yitao Liu, Richard Zhu, Kaiqu Liang, Xindi Wu, Haotian Liu, Sadhika Malladi, et al. CharXiv: Charting gaps in realistic chart understanding in multimodal llms. *NeurIPS*, 2024.

[41] xAI. Realworldqa. https://x.ai/news/grok-1.5v, 2024. Benchmark for real-world spatial understanding.

[42] Bingquan Xia, Bowen Shen, Dawei Zhu, Di Zhang, Gang Wang, Hailin Zhang, Huaqiu Liu, Jiebao Xiao, Jinhao Dong, et al. MiMo: Unlocking the reasoning potential of language model–from pretraining to posttraining. *arXiv preprint arXiv:2505.07608*, 2025.

[43] Yin Xie, Kaicheng Yang, Xiang An, Kun Wu, Yongle Zhao, Weimo Deng, Zimin Ran, Yumeng Wang, Ziyong Feng, Roy Miles, et al. Region-based cluster discrimination for visual representation learning. In *ICCV*, 2025.

[44] Licheng Yu, Patrick Poirson, Shan Yang, Alexander C. Berg, and Tamara L. Berg. Modeling context in referring expressions. In *ECCV*, 2016.

[45] Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. Dapo: An open-source LLM reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*, 2025.

[46] Xiang Yue et al. Mmmu: A massive multi-discipline multimodal understanding benchmark for expert-level foundation models. *arXiv preprint arXiv:2311.16502*, 2024.

[47] Xiang Yue et al. Mmmu-pro: A more challenging multi-discipline multimodal benchmark for expert-level foundation models. *arXiv preprint arXiv:2406.16796*, 2025.

[48] Bo Zhou, Zheng Shu, et al. MLVU: A comprehensive benchmark for multi-task long video understanding. *arXiv preprint arXiv:2406.04264*, 2024.

[49] Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. Instruction-following evaluation for large language models. *arXiv preprint arXiv:2311.07911*, 2023.

A Details of Visual Grounding

A.1 Design of Grounding Data

Grounding data are essential for training a multi-modal model with strong spatial precision. However, unlike pure captioning or VQA supervision, grounding requires a unified textual encoding of *heterogeneous geometric primitives*—points, axis-aligned boxes, and quadrilaterals. These structures originate from diverse annotation tools or detector outputs, and must be transformed into a learnable, language-like representation that the model can parse, reason over, and generate consistently.

Design Goals. Our grounding formatting strategy is guided by three principles:

1. Unification. All shapes—from a single point to a free-form polygon—must share a coherent syntax so that the model learns a single "grammar" for spatial expressions. This allows arbitrary combinations such as a box + polygon or point + polyline to appear in one annotation.
2. Extensibility. New shapes, attributes, or object-level metadata should be addable without breaking old formats. Inline tags with clearly delimited boundaries support this extensibility.
3. Learnability. The representation should behave like natural language: sequential, tokenizable, and stable. We therefore avoid whitespace inside tags, normalize all coordinates to integers in $[0, 999]$, and ensure each shape is enclosed in a fixed tag pair.

Following the above design principles, grounding labels within text begin with `<|object_ref_start|>{object_name}<|object_ref_end|>`, followed by a sequence of position labels whose shapes may include points, axis-aligned bounding boxes, or quadrilaterals. The `object_ref` tag is optional when no specific object or region is being referred to. The sequence may contain a mixture of different shapes. Each shape label is enclosed within a dedicated start–end tag pair, and no whitespace is allowed between consecutive tags within a single grounding label.

A.2 Detailed Formatting of Grounding Data

This section introduces the unified representation used for grounding and pointing annotations in our dataset. A grounding label begins with an optional object reference, followed by one or more geometric shape labels. Supported shapes include points, axis-aligned bounding boxes, and quadrilaterals. All coordinates are defined in image space with the origin at the upper-left corner, the x-axis aligned with image width, and the y-axis aligned with image height. Coordinates are normalized to integers in $[0, 999]$ based on the image resolution. Preliminary studies show that relative coordinates are generally easier for the model to learn than absolute coordinates. The unified representation can be easily extended to shapes including polygons, polylines, and segmentation masks.

Each shape label is enclosed in a pair of special tags, and no spaces are allowed within a single bounding label. The overall representation supports rich combinations, including multi-shape grounding for the same object.

A.2.1 Object Reference

A grounding label may contain an object reference. This component is optional when no explicit object or region is mentioned. The canonical format is:

```
<|object_ref_start|>{object_name}<|object_ref_end|>
```

A.2.2 Supported Geometric Shapes

Below, we list all supported spatial shapes. All coordinates must be integers within $[0, 999]$. No spaces are allowed inside the shape tags.

2D Point A point is represented as:

```
<|point_start|>(x,y)<|point_end|>
```

Axis-Aligned Bounding Box A bounding box is defined by the upper-left and lower-right corners:

```
<|box_start|>(x1,y1),(x2,y2)<|box_end|>
```

Quadrilaterals A quadrilateral is specified by four corners, preferably in clockwise order starting from the upper-left. The format is:

```
<|quad_start|>(x1,y1),(x2,y2),(x3,y3),(x4,y4)<|quad_end|>
```

When both a bounding box and quadrilateral are appropriate, a bounding box is preferred. Functions such as `cv2.BoxPoints` may be used to convert polygons into minimum-area quadrilaterals.

A.3 Detailed Evaluation Results of Grounding

We perform a comprehensive evaluation of the openPangu-VL-7B on benchmarks related to visual grounding compared with other VLMs with leading grounding capabilities. For box-based grounding, we evaluate openPangu-VL-7B on the referring expression comprehension benchmarks RefCOCO/+/g [17, 26] and the multi-target open-vocabulary detection benchmark ODinW-13 [21]. With comprehensive capability in phrase comprehension, openPangu-VL-7B achieves SOTA performance, reaching 90.6% accuracy (Acc@IoU=0.5) on RefCOCO/+/g. For ODinW-13, we use mean Average Precision (mAP) as the evaluation metric with confidence set to a constant value. During evaluation, we create a query task where openPangu-VL-7B is asked to detect all categories present in the image, and it achieves a state-of-the-art result of 51.5 mAP on ODinW-13. For point-based grounding, we evaluate pointing capabilities using Point-Bench [7], a benchmark comprising approximately 982 pointing tasks across five reasoning categories: spatial, affordance, counting, steerable, and reasoning. On this benchmark, openPangu-VL-7B achieves an accuracy of 65.38. For counting, leveraging our curated pointing and counting data, openPangu-VL-7B achieves state-of-the-art counting performance on CountBench [28], reaching an accuracy of 96.1 with a "point-based counting" prompt (92.8 with "direct counting"). Notably, by performing pointing first, point-based counting incurs extra inference overhead but leads to improved counting performance. For depth estimation, we evaluate on the NYU-DepthV2 [33] validation set by querying the depth at point coordinates located on visible objects in the image. With a lower AbsRel indicating better performance, openPangu-VL-7B achieves 10.25, outperforming Seed1.5-VL[13]'s 11.6 and DepthLM[5]'s 12.14.

B Evaluation Prompts

We list the prompts and resolution we used to evaluate our model in this section to improve reproducibility. As for the benchmarks that need LLM-based answer extraction, we adapt the default settings in the VLMEvalKit [9] framework by using GPT-5.2 as a judger.

B.1 STEM

MMMUPro

standard:
<image, resolution: $[672^2, 2016^2]$>
{question}
{options}
Answer the preceding multiple choice question. Please analyze and solve this problem step by step, and put your final answer within \\boxed{}.

---

vision:
<image, resolution: $[672^2, 2016^2]$>
Write out the multiple-choice question in the image and then solve it. Please analyze and solve this problem step by step, and put your final answer within \\boxed{}.

MMMU

<image, resolution:$[672^2, 2016^2]$>
{question}
{options}
Please analyze and solve this problem step by step, and enclose the correct option letter in \\boxed{}.

Mathvista

multi_choice:
<image, resolution:$[224^2, 2016^2]$>
Please answer the question, please analyze and solve this problem step by step, and provide the correct option letter, e.g., A, B, C, D, within \\boxed{} at the end.
Question: {question}
Choices: {choices}
Solution:

integer:
<image, resolution:$[224^2, 2016^2]$>
Please answer the question requiring an integer answer, please analyze and solve this problem step by step, and provide the final value, e.g., 1, 2, 3, within \\boxed{} at the end.
Question: {question} (Unit: {unit})
Solution:

float1:
<image, resolution:$[224^2, 2016^2]$>
Please answer the question requiring a floating-point number with one decimal place, please analyze and solve this problem step by step, and provide the final value, e.g., 1.2, 1.3, 1.4, within \\boxed{} at the end.
Question: {question} (Unit: {unit})
Solution:

float2:
<image, resolution:$[224^2, 2016^2]$>
Please answer the question requiring a floating-point number with two decimal places, please analyze and solve this problem step by step, and provide the final value, e.g., 1.23, 1.34, 1.45, within \\boxed{} at the end.
Question: {question} (Unit: {unit})
Solution:

list:
<image, resolution:$[224^2, 2016^2]$>
Please answer the question requiring a Python list as an answer, please analyze and solve this problem step by step, and provide the final list, e.g., [1, 2, 3], [1.2, 1.3, 1.4], within \\boxed{} at the end.
Question: {question} (Unit: {unit})
Solution:

B.2 Visual Grounding

RefCOCO-avg

<image, resolution:$[1344^2, 2016^2]$>
Please provide the bounding box coordinate of the region this sentence describes: {Referring expression.}

ODinW-13

<image, resolution:$[1344^2, 2016^2]$>
Detect all {category_name} in the image.

Point-Bench

<image, resolution:$[1344^2, 2016^2]$>
{Prompts provided by Dataset.}

CountBench

<image, resolution:$[1344^2, 2016^2]$>
How many {category_name} are there in the image? Point to them and output the total count.

NYUDepthV2

<image, resolution:640×480>
Here are the detailed camera parameters for the image. Camera intrinsic parameters: Focal length f_x=518.86, f_y=519.47. Principal point coordinate locates at the center of the image, c_x=325.6 and c_y=253.7, when image width 640 and height 480. We do not consider distortion parameters here. Therefore, the intrinsic matrix K = [[518.86, 0, 325.6], [0, 519.47, 253.7], [0, 0, 1]].
Here, we take the camera coordinate system as the world coordinate system.
Estimate the absolute depth between the photographer and the points at {points}.
Respond with their point coordinates and corresponding absolute depth in meters.

B.3 GeneralVQA

MMBench | AI2D_test | RealWorldQA | MMStar

<image, resolution:$[1344^2, 2016^2]$>
Question: {question}
Options:
{options}
Please select the correct answer from the options above.

B.4 OCR & Chart / Document Understanding

OCRBench

<image, resolution:$[224^2, 2016^2]$>
{question}

CharXiv

<image, resolution:$[1344^2, 2016^2]$>
{question}

TextVQA | DocVQA_test

<image, resolution:[$1344^2, 2016^2$]>
{question}
Answer the question using a single word or phrase.

ChartQA

<image, resolution:[$1344^2, 2016^2$]>
{question} Please answer the question directly.

B.5 Multi-Image

BLINK

<image, resolution:[$1344^2, 2016^2$]>
Question: {question}
Options:
{options}
Please select the correct answer from the options above.

MUIRBench

<image_1><text_1><image_2><text_2>...<image_n, resolution:[$672^2, 2016^2$]><text_n>
Answer with the option's letter from the given choices directly.