ERNIE 4.5 Technical Report

ERNIE Team, Baidu
ernie@baidu.com

Abstract

In this report, we introduce **ERNIE 4.5**, a new family of large-scale multimodal models comprising 10 distinct variants. The model family consist of Mixture-of-Experts (MoE) models with 47B and 3B active parameters, with the largest model having 424B total parameters, as well as a 0.3B dense model. For the MoE architecture, we propose a novel heterogeneous modality structure, which supports parameter sharing across modalities while also allowing dedicated parameters for each individual modality. This MoE architecture has the advantage to enhance multimodal understanding without compromising, and even improving, performance on text-related tasks. All of our models are trained with optimal efficiency using the PaddlePaddle deep learning framework, which also enables high-performance inference and streamlined deployment for them. We achieve 47% Model FLOPs Utilization (MFU) in our largest ERNIE 4.5 language model pre-training. Experimental results show that our models achieve state-of-the-art performance across multiple text and multimodal benchmarks, especially in instruction following, world knowledge memorization, visual understanding and multimodal reasoning. All models are publicly accessible under Apache 2.0 to support future research and development in the field. Additionally, we open source the development toolkits for ERNIE 4.5, featuring industrial-grade capabilities, resource-efficient training and inference workflows, and multi-hardware compatibility.

Github: https://github.com/PaddlePaddle/ERNIE

Huggingface: https://huggingface.co/baidu

June 29, 2025

Contents

1 Introduction

In recent years, the field of artificial intelligence has witnessed remarkable progress, largely driven by the development of large-scale foundation models. These models, powered by massive datasets and advanced training techniques, have demonstrated unprecedented capabilities across a wide range of domains. In the field of text understanding and reasoning, models such as GPT-4.1 (OpenAI, 2025a), GPT-4.5 (OpenAI, 2025b), o3 (OpenAI, 2025d), Qwen-3 (Yang et al., 2025a), DeepSeek-V3 (DeepSeek-AI et al., 2024b), DeepSeek-R1 (DeepSeek-AI, 2025), Claude 4 (Anthropic, 2025), Gemini 2.5 (DeepMind, 2025), and Llama-4 (Meta-AI, 2025) have set new state-of-the-art results, demonstrating impressive capabilities in comprehension, reasoning, and problem solving. For multimodal understanding, models like GPT-4 series (OpenAI, 2024; 2025a;b), Gemini 2.5 (DeepMind, 2025), Gemma 3 (Gemma-Team, 2025), and Qwen2.5-VL (Bai et al., 2025) have extended these abilities to visual data, enabling robust visual reasoning and interpretation. These state-of-the-art systems have not only set new benchmarks in natural language processing, image and video comprehension, but have also facilitated the emergence of powerful applications in reasoning, conversation, and creative generation.

The ERNIE 4.5 models listed in Table 1 include both Mixture-of-Experts (MoE) models and a dense model. Except for the 0.3B dense language model, all others are MoE-based. There are two kinds of MoE models: Large Language Models (LLM) and Vision-Language Models (VLM). Because we have separated parameters for each modality, LLMs have fewer total parameters than the VLMs. These models are built with the goal of achieving highly efficient and effective multimodal pre-training and inference, while also addressing the engineering and performance challenges posed by model scaling.

Model	Multimodal	MoE	Post-Trained	Thinking / Non-Thinking Mode
ERNIE-4.5-300B-A47B-Base	✗	✓	✗	-
ERNIE-4.5-300B-A47B	✗	✓	✓	non-thinking
ERNIE-4.5-21B-A3B-Base	✗	✓	✗	-
ERNIE-4.5-21B-A3B	✗	✓	✓	non-thinking
ERNIE-4.5-0.3B-Base	✗	✗	✗	-
ERNIE-4.5-0.3B	✗	✗	✓	non-thinking
ERNIE-4.5-VL-424B-A47B-Base	✓	✓	✗	-
ERNIE-4.5-VL-424B-A47B	✓	✓	✓	both
ERNIE-4.5-VL-28B-A3B-Base	✓	✓	✗	-
ERNIE-4.5-VL-28B-A3B	✓	✓	✓	both

Table 1: The overview of ERNIE 4.5 family.

Unlike traditional unimodal MoE models, ERNIE 4.5 uses a novel heterogeneous modality structure, which supports parameter sharing across modalities, including self-attention parameter sharing and expert parameter sharing, while also allowing dedicated parameters for each individual modality. With our proposed modality-isolated MoE routing technique and multimodal joint pre-training, ERNIE 4.5 not only enables efficient learning of visual information through dedicated vision experts, but also enhances the language model's original knowledge and reasoning capabilities during training. Furthermore, during post-training, our model is able to achieve an effective balance between thinking and non-thinking modes.

All of our models are trained using the PaddlePaddle framework. We efficiently pre-train ERNIE 4.5 by proposing a heterogeneous hybrid parallelism approach and a hierarchical load balancing solution tailored for multimodal large models. Through our extreme optimizations including efficient intra-node expert parallelism, FP8 mixed-precision training, and fine-grained recomputation methods, we achieve 47% Model FLOPs Utilization (MFU) in pre-training our largest ERNIE 4.5 language model on 2016 NVIDIA H800 GPUs. Based on our training approach, our largest ERNIE 4.5 language model is able to achieve the optimal training performance with limited compute resources, for example 96 GPUs.

ERNIE 4.5 series consists of MoE and dense models with varying parameter sizes, suitable for various deployment scenarios. All models are capable of performing inference directly using the BF16 and FP8 precisions. To improve inference efficiency, we propose a lossless and inference-friendly low-bit quantization solution with hardware-optimized operators to achieve enhanced memory reduction and computation acceleration. The overall size of parameters and the lossless compression techniques allow our largest ERNIE 4.5 model to be conveniently deployed with minimal computational resources (4x 80GB GPUs for 4-bit, 1x 141GB GPU for 2-bit). By further adopting Prefill-Decode (PD) disaggregation with expert parallelism, our largest ERNIE 4.5 language model achieves an inference throughput of 56k input TPS (Tokens Per Second) and 18k output TPS per H800 node.

Our model family is characterized by three key innovations:

1. **Multimodal Heterogeneous MoE Pre-Training:** Our models are jointly trained on both textual and visual modalities to better capture the nuances of multimodal information and improve performance on tasks involving text understanding and generation, image understanding, and cross-modal reasoning. To achieve this without one modality hindering the learning of another, we designed a *heterogeneous MoE structure*, incorporated *modality-isolated routing*, and employed *router orthogonal loss* and *multimodal token-balanced loss*. These architectural choices ensure that both modalities are effectively represented, allowing for mutual reinforcement during training.
2. **Scaling-Efficient Infrastructure:** We propose a novel heterogeneous hybrid parallelism and hierarchical load balancing strategy for efficient training of ERNIE 4.5 models. By using intra-node expert parallelism, memory-efficient pipeline scheduling, FP8 mixed-precision training, and fine-grained recomputation methods, we achieve remarkable pre-training throughput. For inference, we propose *multi-expert parallel collaboration* method and *convolutional code quantization* algorithm to achieve 4-bit/2-bit lossless quantization. Furthermore, we introduce PD disaggregation with dynamic role switching for effective resource utilization to enhance inference performance for ERNIE 4.5 MoE models. Built on PaddlePaddle, ERNIE 4.5 delivers high-performance inference across a wide range of hardware platforms.
3. **Modality-Specific Post-Training:** To meet the diverse requirements of real-world applications, we fine-tuned variants of the pre-trained model for specific modalities. Our *LLMs* are optimized for general-purpose language understanding and generation. The *VLMs* focuses on visual-language understanding and supports both thinking and non-thinking modes. Each model employed a combination of *Supervised Fine-tuning (SFT)*, *Direct Preference Optimization (DPO)* or a modified reinforcement learning method named *Unified Preference Optimization (UPO)* for post-training.

We conducted extensive evaluations of our models across a wide range of benchmarks, covering language understanding and generation, reasoning, and multimodal tasks. Our models consistently achieve strong performance, especially in instruction following, world knowledge memorization, visual understanding and multimodal reasoning.

All models in this release, including model weights and development toolkits, are fully open-sourced to encourage broad adoption and collaborative research. In the following sections of this report, we will provide detailed descriptions of our model architecture, training procedures, and comprehensive evaluation results. We hope that our efforts contribute meaningfully to the research community and help accelerate progress in the field of large-scale multimodal models.

2 Architecture

Figure 1 illustrates the Transformer architecture adopted by ERNIE 4.5. It supports image, video, and text modalities as input, and generates text as output. For visual modalities (images and videos), a variable-resolution ViT encoder is employed, followed by an adapter to project representations into a shared embedding space with text. ERNIE 4.5 then applies a fine-grained Mixture-of-Experts (MoE) architecture with multimodal positional embedding to model the unified hidden states across modalities. The key components of the architecture includes the following:

- **Heterogeneous MoE:** Text and vision features are routed to separate sets of experts, while they both also go through a group of shared experts as well as all self-attention parameters. Visual experts have one-third the parameters of textual experts.
- **Vision Encoder:** We employ a vision encoder equipped with an adaptive-resolution transformation and 2D Rotary Position Embedding (RoPE).
- **Adapter:** The adapter aligns the representations from the visual and textual modalities, and incorporates both spatial and temporal compression.
- **Multimodal Positional Embedding:** We utilize 3D RoPE within the vision-language model, encoding temporal, width, and height positions independently.

2.1 Heterogeneous MoE

ERNIE 4.5 is built upon a fine-grained MoE backbone. Text and vision inputs are routed to distinct sets of experts tailored to their respective characteristics, mitigating cross-modal interference. A subset of shared experts, together with all self-attention layers, is maintained for all tokens to facilitate cross-modal

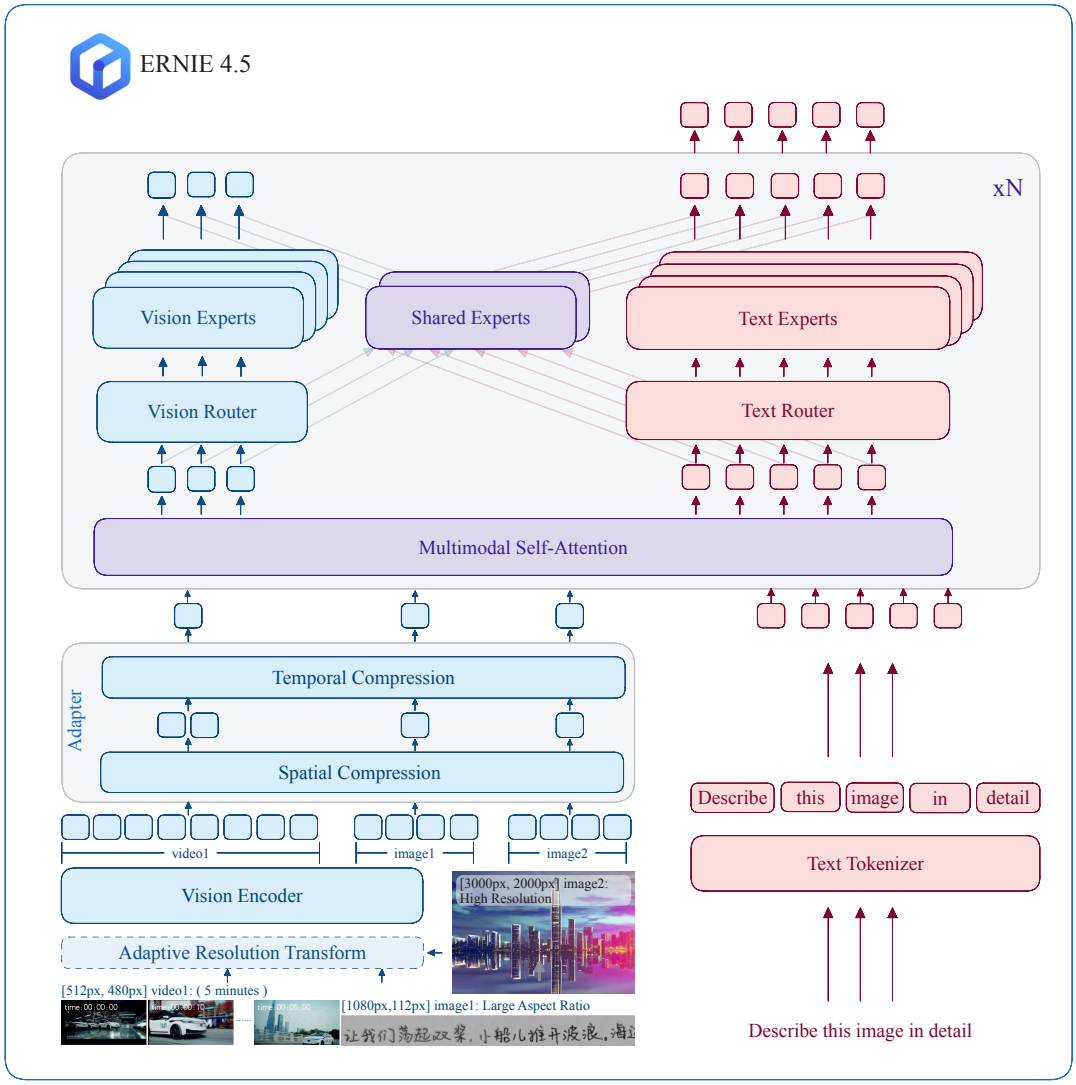


Figure 1: Architecture of ERNIE 4.5 supporting image, video, and text inputs with text outputs. The system comprises three core components: (1) **Heterogeneous MoE** routing text features (red) to text-specific experts and visual features (blue) to vision-specific experts, while shared experts and self-attention parameters (purple) process unified cross-modal hidden states; (2) **Adapter** projecting visual representations into shared embedding space via dual-compression layers; (3) **Vision encoder** implementing adaptive-resolution ViT for images/videos, featuring dynamic frame-resolution sampling and timestamps rendered at top-left corners.

knowledge integration. In addition, we introduce a modality-aware expert allocation strategy, where visual experts contain only one-third the parameters of textual experts, thereby improving the efficiency of visual information processing.

In multimodal modeling, the MoE router is prone to instability, especially when there is a sudden shift in the data distribution. For instance, extending a text-only MoE model to handle multimodal inputs may cause the router to collapse, leading to degradation of textual capabilities (Liang et al., 2024). To mitigate this issue, we propose a **modality-isolated routing** strategy. Figure 2 presents heatmaps of expert activations ratio across different layers on the hold-out dataset. It illustrates that textual experts exhibit concentrated activations while visual experts display more dispersed activation patterns. These findings substantiate the necessity of modality-separated MoE designs to ensure effective multimodal joint training.

Specifically, the FFN experts in ERNIE 4.5 are categorized into three types: text experts, vision experts, and shared experts. Both text and vision tokens are processed by the shared experts in a non-routed manner, while each is independently routed to its corresponding modality-specific experts. Specifically, text experts exclusively process text tokens, vision experts exclusively process vision tokens, and shared

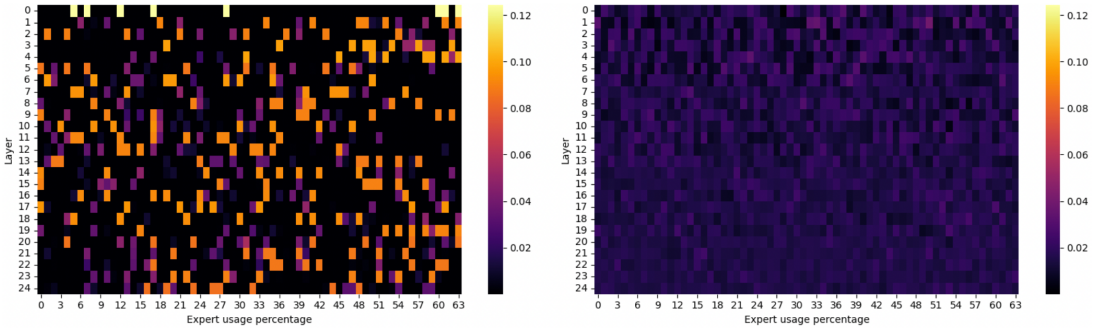


Figure 2: Expert usage across Transformer layers. Left: text modality expert group; right: vision modality expert group.

experts operate on both modalities. The final output of the MoE layer is obtained by summing the outputs from the shared and modality-specific experts.

Several engineering decisions further enhance the scalability and efficiency of our approach. Given the redundant nature of visual tokens, experts in the vision group use an intermediate dimension that is one-third the size of that used by text experts. Since the FLOPs of an FFN layer scale with the product of input and intermediate dimensions, this adjustment achieves a roughly 66% reduction in per-token FFN computation for vision tokens. We further exclude visual experts from the final Transformer layer, since their weights do not contribute to the cross-entropy loss. To resolve the device-level load imbalance problem caused by modality-isolated experts, we design a custom load-aware expert parallelism strategy, detailed in Section 5.2.1.

The heterogeneous MoE design brings several key advantages:

- **Unified Multimodal Modeling:** It enables the construction of a unified multimodal model in which all parameters—textual and visual—are optimized jointly. Compared to partial-tuning approaches, this design is both more data-efficient and more scalable, supporting model growth up to hundreds of billions parameters.
- **Routing Stability:** Vision experts can be introduced in the later stages of training, avoiding routing collapse. This staged training reduces overall computation while preserving performance, since visual understanding is largely grounded in prior textual knowledge.
- **Computational Efficiency:** Text and vision experts can be deployed separately. In text-only inference scenarios, vision experts can be skipped to reduce memory overhead. For multimodal inference, we support modality-aware partitioning of the inference pipeline. Specifically, allocating different inference budget for each modality, deploying `Prefill-Text`, `Prefill-Vision`, and `Decode-Text` modules independently can significantly reduce cross-device communication.

Compared to previous approaches, our modality-isolated fine-grained MoE strategy introduces several important innovations. Unlike Wang et al. (2023) and Wang et al. (2022), which use dense FFNs, we adopt a fine-grained MoE backbone that offers improved scalability. Unlike Liang et al. (2024), we retain dense attention layers to preserve cross-modal interactions, while restricting MoE routing to the FFN layers. In contrast to Lin et al. (2024d), which relies on expert choice gating mechanisms, we employ fine-grained top-k routing (DeepSeek-AI et al., 2024a;b). This maintains compatibility with autoregressive decoding and enables scaling to long context training.

2.2 Vision Encoder

Image Encoding. Vision Transformers (ViTs) are widely employed in vision-language models as vision encoders (Radford et al., 2021b; Zhai et al., 2023; Fang et al., 2024; Sun et al., 2023). However, existing ViTs are typically pre-trained on fixed-resolution inputs, necessitating that images be resized to a square shape prior to processing. In this work, we use an **adaptive-resolution** vision encoder. Rather than enforcing a square input, we independently resize the height and width of each input image to the nearest multiples of the ViT patch size. This approach approximately preserves the original aspect ratio, avoiding the distortions introduced by fixed-size resizing.

The adaptively resized image is subsequently divided into patches, resulting in a variable-length 1D sequence of tokens. To encode the 2D spatial origin of each patch, we employ **2D Rotary Position Embedding** (RoPE) (Su et al., 2024b), which separately encodes spatial information along the height and width dimensions. Additionally, we adopt the image packing technique proposed by Dehghani

et al. (2023b), which efficiently packs multiple images into a single batch while maintaining positional consistency among patches. This enables a more effective utilization of computational resources without requiring modifications to the model architecture.

Video Encoding. Videos are processed as sequences of sampled frames, but this quickly exhausts the model's limited sequence length budget and makes comprehensive temporal coverage challenging. We propose an **adaptive video sampling strategy** that dynamically adjusts both the number of frames and their spatial resolution based on each video's duration and the available sequence length. Specifically, frames are sampled at a predefined frame rate; if this exceeds the upper or lower frame count limits, uniform sampling is applied at the respective bound. If the total visual tokens exceed the limit, we reduce the resolution until it reaches a lower bound, and if needed, further decrease the number of frames. For multi-video inputs, the frame allocation is proportional to each video's length. This approach maximizes the use of sequence length, offering higher detail for short videos and adequate frame coverage for long videos.

To further enhance temporal modeling, we introduce a **timestamp rendering** technique that overlays absolute timestamps onto each frame. Unlike position embeddings (Bai et al., 2025) or textual token-based time encoding (Guo et al., 2025; Hong et al., 2024), our method is flexible to any frame rate, consumes no extra tokens, and provides the model with explicit temporal cues directly in the visual stream. This direct supervision reduces learning difficulty and enables more accurate temporal understanding.

2.3 Adapter

To align visual and textual representations in a unified embedding space, we design an Adapter that serves as a modality bridging module between the visual encoder and the language model. The Adapter incorporates spatial and temporal token compression to perform feature fusion and reduce sequence length. Specifically, the spatial compression operates on non-overlapping 2×2 patches, yielding a $4\times$ decrease in token count along spatial dimensions, while the temporal compression reduces the sequence length by a factor of 2. Both compression operations leverage pixel shuffle (Shi et al., 2016) which rearranges spatially or temporally adjacent token features into a more compact form. The rearranged features are then processed through MLP layers. To unify the processing of images and videos, each static image is treated as a synthetic two-frame video by duplicating the image feature, enabling consistent temporal modeling across modalities. Overall, the Adapter not only performs efficient token compaction via spatio-temporal fusion, but more importantly, aligns the multimodal feature space to the textual embedding space by training, facilitating deeper cross-modal interaction in the subsequent transformer layers.

2.4 Multimodal Position Embedding

To effectively handle multimodal sequences of text, images, and videos, we employ a unified 3D RoPE position embedding scheme for the input layer of the vision language transformer. 3D RoPE encodes temporal and spatial positions separately by assigning distinct frequency bands to each axis for visual inputs, and defaults to standard 1D RoPE for text tokens (Wang et al., 2024a). Specifically, lower frequencies are allocated to the temporal axis (which varies most slowly), while the remaining frequencies are interleaved between the spatial axes (height and width), enabling both symmetric spatial modeling and strong long-term temporal modeling (Wei et al., 2025). Unlike conventional 2D RoPE used for images and 1D for text, our method accommodates the extra temporal dimension in videos, enabling flexible and consistent position encoding across all modalities within a single embedding space. Empirical results demonstrate that 3D RoPE consistently enhances multimodal understanding, particularly in long video comprehension tasks that require sequence length extrapolation.

3 Pre-Training

In this section, we first describe the construction of our large-scale text and multimodal datasets (Section 3.1). Next, we introduce a data manager REEAO (Record Everything Everywhere All at Once) that supports bitwise-deterministic data processing, enabling reproducible and non-redundant data access across training runs. Subsequently, we present the training recipes for both textual and multimodal pre-training stages (Section 3.3). We also detail our innovations in the training objective (Section 3.4), and conclude by sharing our findings regarding the EMA process (Section 3.5).

3.1 Pre-Training Data

ERNIE 4.5 models are trained on data curated from web pages, academic papers, documents, images, videos, and synthetic modality conversion data. Given the diverse data sources and substantial noise in the raw datasets (Awadalla et al., 2024; Kim et al., 2022), we implement comprehensive data quality filtering pipelines for text, images, videos, and audio respectively, mainly including deduplication, removal of noise and irrelevant content. Subsequently, we perform data labeling and clustering to extract patterns and discover knowledge within the dataset. This process facilitates both knowledge discovery and provides information for data management, analysis, and optimal data mixing. Finally, we analyze model performance on our dataset to identify data weaknesses and guide optimization, establishing a human-model-in-the-loop iterative data refinement approach. The following presents the key steps on this process.

- **Data Denoising and Synthesis:** Our data filtering pipeline combines heuristic rules with model-based approaches. Heuristic filtering performs deduplication and low-quality data removal, while model-based filtering employs quality assessment models to automatically filter low-quality samples and ensure data quality. Noise filtering inevitably leads to data scale reduction, creating a quality-quantity dilemma. To this end, we introduce data synthesis solutions to supplement high-value data. For example, we adopt self distillation and multimodal conversion methods to enrich data sources, which significantly alleviates the data scarcity problem in high-value domains.
- **Data Analysis:** To better understand and manage data, we meticulously construct a pre-training data map to support data mining and data analysis. To build the data map, we categorize data from several aspects including language, knowledge, application, and quality. Through multi-perspective data analysis, we can optimize training data mixing configurations, staged selection, and model performance tracking analysis.
- **Human-Model-in-the-Loop Data Refinement:** To continuously improve data quality, we design a Human-Model-in-the-Loop pipeline. This pipeline includes stages of selecting core datasets, choosing candidate models for data evaluation, and manual result analysis. Through this data iteration loop, we can ensure the effectiveness of each filtering and data mining strategy, thereby improving the overall quality of text, image and video data.

In the following section, we describe the construction of various data sources, including knowledge-centric data, multimodal aligned data, and domain-specific data.

Knowledge-Based Data. Through large-scale data analysis, we observe that the amount of knowledge contained in natural language corpora is inherently uneven and can be systematically categorized into multiple levels. Inspired by the DIKW framework (Wikipedia, 2025), we define five distinct tiers of knowledge and develop a knowledge-level classification model to automatically annotate pre-training data according to these levels. This classification framework enables a deeper analysis of the value distribution within pre-training datasets.

Based on this framework, our analysis reveals that high-value data is scarce and constitutes only a small fraction of available corpora. To address this scarcity, we employ data synthesis to augment the limited high-value training data. Specifically, we propose a key-point-based data synthesis method capable of generating diverse and high-quality samples across domains such as mathematics (Wei et al., 2024b; Yang et al., 2025b; Li et al., 2024b; Yu et al., 2024), factual knowledge (Gunasekar et al., 2023; Wettig et al., 2024), and programming code (Chang et al., 2024; Wei et al., 2024b). To ensure broad knowledge coverage, we use textbooks and educational websites as seed sources to extract structured key points that guide the data generation process.

Furthermore, to enhance the model's performance on reasoning tasks, we conduct a series of targeted processing steps on reasoning-related corpora. These steps involve selecting representative samples, organizing them by reasoning type, grading difficulty levels, and filtering low-quality items to retain the valuable instances (Sun et al., 2021; Xie et al., 2023; Su et al., 2024a; Yang et al., 2025b). Empirical results suggest that this processing significantly contributes to improving the model's reasoning capability.

Interleaved Text-Image Data. Interleaved text-image data plays a crucial role in advancing multimodal learning capabilities (Alayrac et al., 2022; Lin et al., 2024c), yet existing datasets suffer from limited scale and weak alignment between visual and textual components. To address these challenges, we develop a comprehensive data curation strategy. We first systematically collect extensive web data, identifying high-quality web pages and documents with well-integrated visual elements. We then augment this dataset by extracting substantial interleaved content from video sources, leveraging the rich explanatory knowledge embedded in online videos through keyframe extraction and automatic

speech recognition (ASR) (Miech et al., 2019; Xu et al., 2023; Zellers et al., 2021). Finally, we implement rigorous quality enhancement procedures including intra-page image-text deduplication, filtering of low-resolution images and irrelevant content, removal of garbled text and advertisements, and detection of disordered content. Through this multi-faceted approach, we significantly enrich our interleaved dataset and observe that this enhanced interleaved data substantially improves the model's knowledge capacity and multimodal understanding capabilities.

Image-Text Pairs. Image-text pairs are essential for learning transferable representations in vision-language models (Radford et al., 2021a). Although such data is abundant on the internet (Schuhmann et al., 2022), it suffers from substantial noise, such as irrelevant descriptions, trivial or generic captions, redundant images, and uneven data quality. To address these challenges, we employ filtering techniques, including image-text similarity scoring (e.g. CLIP-score thresholding) to filter low-relevance pairs, followed by deduplication of both images and text (Zauner, 2010; Abbas et al., 2023). We then perform image classification and tagging, categorizing images into natural scenes, tables, screenshots, charts, documents, and other types. Finally, we sample a subset of the data for recaptioning (Betker et al.) to improve the quality of the image-text alignment. To further improve grid-style image understanding, we synthesize training samples by stitching multiple images into a grid layout and concatenating the corresponding captions into a matching structure. This strategy helps the model better localize and interpret information in composite visual inputs.

Domain-Specific Data. To strengthen the model's capability in domain-specific tasks, we construct large-scale datasets spanning vertical domains such as industry, finance, healthcare, consumer entertainment. Given the scarcity and specialized nature of high-quality domain data, we develop a diversified data sourcing strategy to address these limitations. Our approach encompasses two primary data sources:

- **Progressive Mining and Conditional Training:** Inspired by Shao et al. (2024), we employ progressive mining methods to systematically extract substantial amounts of domain-specific data. During pre-training, we find that conditional pre-training schemes (Korbak et al., 2023) significantly improve learning efficiency for this domain data, particularly for creative writing tasks.
- **Audio Transcription and Enhancement:** We utilize Auto Speech Recognition (ASR) models to transcribe valuable domain-specific content from audio sources, including video soundtracks and podcasts, developing a comprehensive rewriting and filtering pipeline to mitigate transcription noise while enriching our dataset with colloquial and conversational text data.

3.2 REEAO: Bitwise-Deterministic Pre-Training Data Manager

Training modern autoregressive language models at scale involves processing trillions of tokens across heterogeneous datasets and leveraging dynamically changing computational infrastructures. Large-scale training often involves frequent fluctuations, such as resuming from checkpoints, recovering from node failures, adjusting computational resources, handling variable sequence lengths, and updating datasets. These variations usually disrupt the underlying data pipeline, potentially resulting in significant issues such as inadvertent data duplication or omission.

To address these challenges, we introduce **REEAO** (Record Everything Everywhere All at Once) — a data flow manager built on five core principles: Reproducibility, Efficiency, Elasticity, Adaptivity, and Observability. REEAO chunks multimodal data sources into fixed-length records and fundamentally guarantees that the training process produces a bitwise-deterministic token sequence, which is fully determined immediately after the pre-training data is configured and before actual training begins. This guarantee holds even under complex scenarios, such as changes in the number of training nodes, distribution strategy, global batch size, or context length. Additionally, REEAO maintains a distributed-independent record of data source consumption to ensure that no data is duplicated — even when scaling resources dynamically or updating training data on the fly.

3.3 Pre-Training Recipe

We develop a series of Transformer-based models with diverse scales, attention configurations, and optional MoE modules, adopting architectural choices tailored to each model's parameter budget. Table 2 summarizes both the architectural hyperparameters and scale-aware training settings for the ERNIE 4.5 family under large-scale and lightweight configurations. Specifically, we report key architectural choices alongside training hyperparameters such as batch size, learning rate, and optimization strategies. For all models, we employ the Warmup-Stable-Decay learning rate schedule (Hu et al., 2024) during pre-training, with model-specific configurations.

Models	ERNIE-4.5-A47B-Base	ERNIE-4.5-A3B-Base	ERNIE-4.5-0.3B-Base
Text Params	300B	21B	0.36B
Total Params	424B	28B	–
Layers	54	28	18
Heads (Q/KV)	64/8	20/4	16/2
# Text Experts (Total/Activated)	64/8	64/6	–
# Vision Experts (Total/Activated)	64/8	64/6	–
# Shared Experts	–	2	–
Context Length	131,072	131,072	131,072
Learning Rate	2.2e-4	3.14e-4	4.4e-4
Batch Size	65M	50M	8M
Weight Decay	0.1	0.1	0.1
GradNorm Clip	1.0	1.0	1.0
ViT Layer-Wise LR Decay	0.9	0.9	-

Table 2: Model and training hyperparameters of three ERNIE-4.5-Base models.

Stages	Trainable Parameters	Sequence Length
Stage I: Text-Only Training		
Short-Context	LLM	4,096
Long-Context	LLM	32,768 → 131,072
Stage II: Vision-Only Training		
Vision Encoder	ViT	8,192
Vision Pre-Alignment	Adapter + Vision Experts	8,192
Vision Integration	ViT + Adapter + Vision Experts	8,192
Stage III: Joint Multimodal Training		
Short-Context Multimodal	Full Model	8,192
Long-Context Multimodal	Full Model	131,072

Table 3: The training stages of our ERNIE-4.5-VL-424B-A47B-Base and ERNIE-4.5-VL-28B-A3B-Base models.

To ensure the stability of multimodal joint training, we have designed a pre-training strategy in stages for ERNIE 4.5, as shown in Table 3.

3.3.1 Stage I: Text-Only Training

This stage is dedicated to establishing a robust language backbone, ensuring the model possesses strong linguistic understanding and efficient long-range dependency modeling. By progressively increasing context length and adapting positional encoding, the language model is equipped for both conventional and long-context tasks—providing a stable foundation for subsequent multimodal pre-training.

- **Short-Context:** We commence with large-scale pre-training on trillions of pure-text tokens sourced from diverse domains. This sub-stage primarily develops the core linguistic capabilities, the factual knowledge base, and the proficiency in text generation under a standard short-context(8k sequence length) configuration.
- **Long-Context:** To extend the model's context length to 128k tokens, in this substage, we first increase the maximum sequence length to 32k by raising the frequency base θ of the Rotary Position Embedding (RoPE) (Su et al., 2024b) from 10k to 160k, and continue training to adapt the model to longer sequences. Next, we further extend the sequence length to 128k tokens and increase the RoPE frequency base θ limit from 160k to 500k, and train the model using long-context data. During this process, we upsample documents with sequence lengths exceeding 16k tokens to ensure the model is adequately exposed to long-range dependencies. Benchmark evaluations demonstrate that this stage enables the model to support input sequences up to 128k tokens while maintaining its original capabilities on standard tasks.

3.3.2 Stage II: Vision-Only Training

This stage focuses on integrating visual understanding into pre-trained language models. By carefully designing alignment strategies, we ensure that visual knowledge is efficiently incorporated without

compromising the model's existing language capabilities, thereby laying a solid foundation for future multimodal learning.

- **Vision Encoder:** We first pre-train the vision encoder alongside a smaller language model, utilizing a large-scale dataset of image-text pairs. This process encourages the vision encoder to capture comprehensive visual knowledge.
- **Vision Pre-Alignment:** All LLM parameters are frozen while the vision adapter, vision experts, and vision router are trained. The adapter is initialized from scratch, and the vision experts are derived from text experts via structural pruning. This stage ensures visual modules integrate smoothly with the LLM backbone without degrading its performance.
- **Vision Integration:** The vision encoder is then unfrozen, allowing joint optimization of the full visual pathway. Training emphasizes high-quality image-text pairs, such as captions and alt-text, to align vision and language representations.

3.3.3 Stage III: Joint Multimodal Training

In the final stage, we unfreeze the entire model and jointly train on multimodal data with both standard and extended context lengths. This training stage enables the model to handle complex multimodal tasks in a long context.

- **Short-Context Multimodal:** We unfreeze the entire model and jointly train it on a mixture of text, image, and video data with standard context lengths. This stage serves as a fusion phase, consolidating the modality-specific alignments.
- **Long-Context Multimodal:** Finally, the joint training is extended to a 128k context length. This enables the model to generalize effectively in long-context multimodal tasks.

3.4 Model Optimization

Training multimodal MoE models faces challenges such as expert load imbalance and gradient instability due to input length variability. To address these issues, beyond standard loss functions such as the auxiliary loss and the z-loss (Lepikhin et al., 2021; Zoph et al., 2022), we introduce two novel loss functions: **Router Orthogonalization Loss** and **Token-Balanced Loss**. These proposed losses are specifically designed to promote balanced expert utilization and stabilize gradients, thereby enabling more robust optimization and more effective multimodal pre-training.

3.4.1 Router Orthogonalization Loss

Mixture-of-Experts (MoE) models often suffer from the **expert homogenization** problem, where different experts learn highly overlapping or redundant representations (DeepSeek-AI et al., 2024b). To address this issue, we propose the **router orthogonalization loss**, which encourages orthogonality among the router's expert weight, leading to more balanced routing and better expert specialization.

The orthogonalization loss is defined as:

$$L_{\text{orth}} = \sum_{i=1}^{k} \sum_{j=1}^{k} \left(\hat{\mathbf{w}}_i^\top \hat{\mathbf{w}}_j - \delta_{ij}\right)^2, \quad \text{where} \quad \hat{\mathbf{w}}_i = \frac{\mathbf{w}_i}{\|\mathbf{w}_i\|_2}. \tag{1}$$

Here, $\mathbf{w_i}$ corresponds to the weight vector for expert i. $\delta_{i,j}$ is the kronecker delta. By encouraging orthogonality among these column vectors, the router produces a more uniform expert selection distribution, which facilitates specialization among experts and improves generalization on out-of-distribution (OOD) tasks.

Like weight decay, this orthogonalization loss depends solely on the router weights. Directly incorporating this term into total loss disrupts Adam optimizer (Kingma, 2014) gradient estimates and leads to suboptimal training dynamics. To address this, we modify the Adam optimizer in a manner analogous to AdamW (Loshchilov & Hutter, 2019), enabling the orthogonalization loss to update the router weights directly without interfering with Adam's gradient estimates. The coefficient for the orthogonalization loss in ERNIE 4.5 is set to 1×10^{-3}, and unlike weight decay, it is not scaled by the learning rate.

In our ablation experiments, incorporating Router Orthogonalization Loss yields **+1.44** improvement on text benchmarks, detailed in Appendix A.1.

3.4.2 Token-Balanced Loss

Conventional cross-entropy loss averages the loss over all valid tokens in a sample. However, in multimodal training, only textual tokens contribute to the loss, and their proportion varies significantly between samples. This discrepancy can induce substantial gradient variance, undermining the stability and efficiency of optimization. To address this, we propose the Token-Balanced Loss, which normalizes the loss by the total sequence length, thereby reducing gradient variance and promoting more stable and consistent optimization in multimodal settings.

Formally, during multimodal training, image tokens and prompt positions are masked out and excluded from the cross-entropy loss computation. Let $\mathcal{M}_i$ and $\mathcal{U}_i$ denote the loss mask and its complement (unmasked region), respectively, for sample i. The conventional loss for sample i is:

$$L^{(i)} = -\frac{1}{|\mathcal{U}_i|} \sum_{j \in \mathcal{U}_i} \log P(y_j^{(i)} \mid y_{<j^{(i)}}; \theta). \tag{2}$$

This formulation inadvertently introduces a gradient imbalance: samples with fewer unmasked tokens contribute disproportionately larger gradients, thereby biasing the optimization process.

To address this gradient imbalance, we introduce the Token-Balanced Loss function:

$$L_{\text{balanced}}^{(i)} = \frac{1}{|\mathcal{U}_i| + |\mathcal{M}_i|} \sum_{j \in \mathcal{U}_i} \ell_j^{(i)}, \tag{3}$$

where $\ell_j^{(i)} = -\log P(y_j^{(i)} \mid y< j^{(i)}, \theta)$ represents the individual token loss contribution. The normalization factor $(|\mathcal{U}_i| + |\mathcal{M}_i|)^{-1}$ ensures that each sample's loss contribution is weighted by the inverse of its total sequence length, independent of the specific masking configuration.

3.5 Exponential Moving Average

In addition to loss function design, another crucial component influencing training stability and final performance is the use of parameter smoothing techniques. Among them, Exponential Moving Average (EMA) is widely adopted in large-scale pre-training to stabilize training dynamics and improve generalization. Despite its empirical success, the choice of the EMA decay coefficient α is often heuristic, lacking theoretical guidance—especially in the context of large-scale pre-training. To better understand its role, we conduct a theoretical analysis and demonstrate that EMA can be viewed as analogous to learning rate decay. This perspective offers a principled explanation for a commonly observed phenomenon: EMA models often match the final model checkpoints trained under explicit learning rate decay schedules (DeepSeek-AI et al., 2024b; Li et al., 2025). Building on this insight, we explore the relationship between the decay coefficient α and the effective decay window of EMA, which governs how much influence recent parameter updates have on the EMA-averaged model. Specifically, we introduce a framework for controlling the decay window's size to optimize model performance, which we will explain in the following section.

Analyzing EMA through Effective Learning Rate Decay. We show that EMA applies an exponential weighting to parameter updates, in a manner similar to learning rate decay, resulting in a monotonically decreasing "effective learning rate" over the course of training. Specifically, let $\delta_t = \theta_{t+1} - \theta_t$ be the update at step t; then, the EMA parameters after n steps can be written as:

$$\theta_n^{\text{EMA}} = \theta_0 + \sum_{i=0}^{n-1} \left(\eta_i^{(\alpha)}\right) \delta_i, \quad \eta_i^{(\alpha)} = 1 - \alpha^{n-i}, \tag{4}$$

where $\eta_i^{(\alpha)}$ represents the effective learning rate assigned to the i-th update. See section A.2 for derivation. This formalism reveals that, unlike vanilla parameter updates (which assign unit weight to all updates), EMA progressively downweights recent updates. Figure 3 further visualizes how the decay shape of $\eta_i^{(\alpha)}$ mirrors explicit learning rate schedules, such as cosine or warmup-stable decay, and demonstrates that larger α yields a smoother, longer decay window. Crucially, this perspective provides a principled way to select α—by directly linking it to a desired effective window size, rather than relying on rule-of-thumb choices.

It is important to note that while EMA exhibits a similar decay effect, it is not equivalent to an explicit learning rate decay schedule. In actual learning rate decay training, each update δ_i is computed based on the model after applying the decayed learning rate, whereas in EMA, the updates are aggregated after

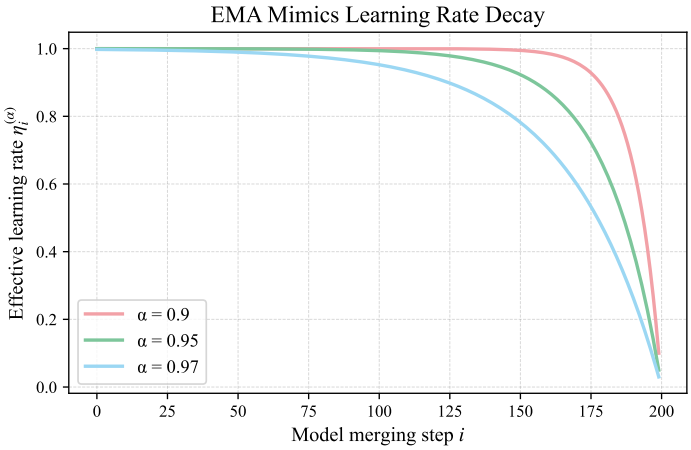


Figure 3: Learning rate decay shape of EMA at different coefficient α.

being computed using the original optimizer learning rates. However, through our empirical study, we find that applying continuous EMA during pre-training achieves comparable performance to explicit learning rate decay. Based on this observation, we propose a **"decay no more"** approach: instead of using repeated learning rate decay to capture the model's early-stage performance, we propose to simply use EMA.

Controlling the Effective Decay Window of EMA. In addition to shaping the decay behavior, the EMA decay coefficient α also leads to an *effective decay window*—the range of recent updates significantly affected by EMA smoothing. To precisely control the effective decay window size through α, we introduce a small threshold $\epsilon \in (0, 1)$ (e.g., $\epsilon = 0.001$). An update δ_i is considered to be *outside* the effective decay window if its effective learning rate satisfies $\eta_i^{(\alpha)} \geq 1 - \epsilon \approx 1$. In this case, the update is barely affected by the EMA smoothing, behaving as in a non-EMA model. Conversely, if $\eta_i^{(\alpha)} < 1 - \epsilon$, the update δ_i is regarded as *within* the effective decay window, indicating that it remains substantially influenced by the EMA process.

The relationship between the effective window size and α is given by Equation 5. This formulation enables precise control over the desired decay window size $\hat{W}$ by selecting an appropriate EMA decay coefficient $\hat{\alpha}$ for a specified threshold ϵ (see Appendix A.3 for the derivation):

$$\hat{\alpha} = \exp\left(\frac{1}{\hat{W}} \log \epsilon\right). \tag{5}$$

In practice, EMA is typically updated every s training steps, which we refer to as the *EMA interval*. Over the course of training, the EMA decay window spans $T = \hat{W} \cdot s$ training steps. Inspired by the decay behavior in explicit learning rate schedules (Hu et al., 2024), we set T to one-tenth of the total training steps. Moreover, our preliminary experiments suggest that higher merging frequencies lead to improved performance. Therefore, during the pre-training stage, we set $s = 4$ and determine α according to Equation 5. To enable high-frequency model merging without compromising training efficiency, we further introduce an asynchronous online EMA mechanism.

We implement an asynchronous online EMA mechanism, which enables extremely high-frequency EMA by offloading GPU parameters directly into host memory without interrupting the training loop. An independent CPU-based worker asynchronously performs EMA accumulation and periodically writes the resulting checkpoint to disk. Related code has been open-sourced in PaddleNLP to facilitate further research.

4 Post-Training

Our model is designed to enable a clean separation between the text-only and vision-related components after multimodal pre-training. Specifically, by removing the multimodal experts, vision encoder, and adapter layers, the model reduces to a pure language model that can be used more efficiently in text-only

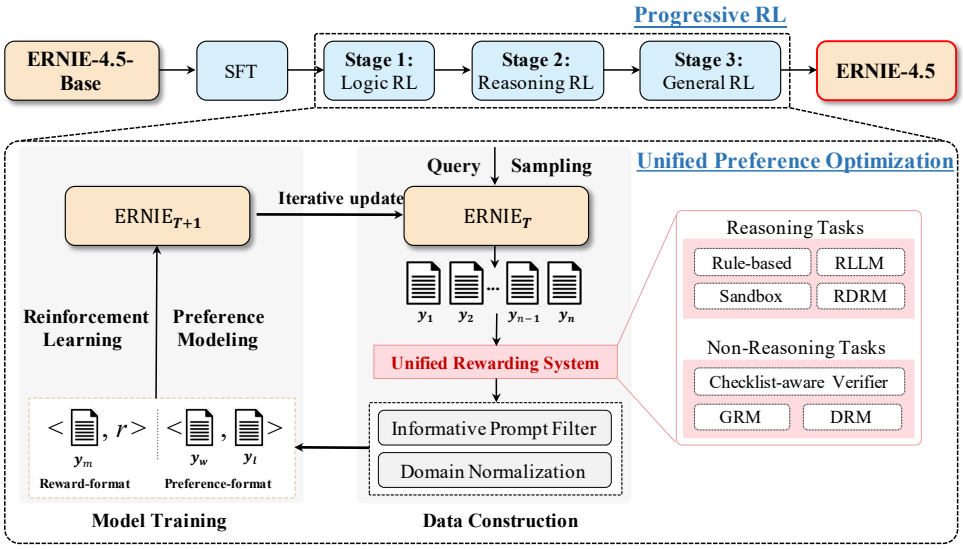


Figure 4: Illustration of LLM Post-Training Pipeline for ERNIE-4.5.

scenarios. Using this modular design, we post-train the text-specific parameters to obtain ERNIE-4.5 optimized for text-only tasks. The complete set of parameters, including both textual and visual components, is further tuned to obtain the multimodal model ERNIE-4.5-VL.

4.1 Post-Training of LLMs

The entire post-training pipeline for LLMs is illustrated in Figure 4. Supervised fine-tuning is an initial step as illustrated in Section 4.1.1. Then, reinforcement learning is carried out, leveraging the insights gained from the unified reward system to further enhance and fine-tune the model's performance. In Section 4.1.2 and Section 4.1.3, we respectively expound on our approaches regarding unified reward system and reinforcement learning.

4.1.1 Supervised Fine-Tuning

This section elaborates on the supervised fine-tuning (SFT) process implemented for ERNIE-4.5. To maximize the efficacy of the model, we implement a systematic taxonomy to categorize supervised fine-tuning (SFT) data into distinct topical domains. Specifically, we develop a comprehensive suite of ten distinct topical domains, encompassing areas including science & math, coding, logic, information processing, creative writing, multilingual, knowledge QA, multi-turn & role play and safety.

Beyond that, the SFT data are further systematically categorized into reasoning and non-reasoning tasks. The reasoning data comprises complex tasks that necessitate extended chains of thought (CoT) to ensure that the complexity and diversity of these tasks are fully captured. In contrast, the non-reasoning data comprises tasks requiring no in-depth reasoning, but ensuring the accuracy and conciseness of such data is vital for boosting the model's overall performance and versatility.

Furthermore, the emphasis on the quality and diversity of SFT data constitutes a foundational element for the subsequent reinforcement learning (RL) phase. To further enhance the diversity of the supervised fine-tuning (SFT) data, we introduce multiple responses with distinct reasoning contents under some queries within the reasoning tasks. This focus is instrumental in improving the foundational capabilities of the model and equipping it with the resilience necessary to engage in exploration during the RL training process. Based on the approaches defined in the preceding paragraph, we construct an SFT dataset containing 2.3 million samples. We then conduct an average of two training epochs on this dataset to optimize the model's performance.

4.1.2 Unified Rewarding System

This section demonstrates the unified rewarding system for subsequent reinforcement learning. This system is carefully crafted to accommodate both reasoning and non-reasoning tasks through employing distinct reward combinations. It offers precise and comprehensive feedback signals that facilitate preference learning, which provides a basis for subsequent preference optimization illustrated in Section 4.1.3.

For reasoning tasks that require precision and strict adherence to predefined criteria, we prioritize the deployment of rule-based verifiers. Nevertheless, rule-based verifiers inherently exhibit limited generalization capabilities. To mitigate this limitation and enhance the accuracy of feedback in reasoning tasks, we incorporate additional complementary mechanisms.

- **Reference-Guided LLM-as-a-Judge (RLLM):** The reference-guided LLM-as-a-judge component exploits the advanced capabilities of a Large Language Model (LLM) as an impartial evaluator, rigorously benchmarking model-generated outputs against a well-defined corpus of reference answers.
- **Sandbox:** The sandbox is a secure and isolated testing environment engineered to support the execution and systematic evaluation of computational tasks related to programming. By operating within a controlled and isolated context, model-generated responses undergo rigorous testing to directly assess their functionality, correctness, reliability, and adherence to specified requirements.
- **Reference-Guided Discriminative Reward Model (RDRM):** Motivated by the success of reference-guided LLM-as-a-judge, we introduce a novel reference-guided discriminative reward model (RDRM). Unlike traditional models that work in isolation, our RDRM is explicitly guided by reference answers during the scoring process. Instead of a closed-book exam that requires the model to rely solely on its internal knowledge, our RDRM has access to reference answers, effectively transforming the evaluation process into an open-book test. RDRM ensures that the model's outputs closely approximate the content and structural characteristics of the reference answers, thereby guaranteeing comprehensive coverage.

In the context of non-reasoning tasks, which are inherently open-ended and dependence on individual interpretative judgments, we implement a methodological paradigm tailored to effectively accommodate these specific epistemic characteristics:

- **Checklist-Aware Verifiers:** We introduce a novel method called checklist-aware verifiers, which draws inspiration from the RLVR (Lambert et al., 2024; DeepSeek-AI, 2025). Our method begins by meticulously defining a set of explicit criteria. These criteria are carefully crafted to be both clearly definable and objectively assessable, ensuring that there's no ambiguity in what the model's outputs should achieve. By instituting this rigorously defined yet adaptable evaluative framework, our checklist-aware verifiers ensure that the generated responses consistently meet established normative standards.
- **Generative Reward Models (GRM):** By further advances the evaluation process by incorporating multi-dimensional evaluation criteria and dynamic feedback mechanisms, GRM conducts a tailored evaluation for each query, thereby implementing a more systematic and nuanced assessment and enhancing both the accuracy and robustness of the evaluative outcomes.
- **Discriminative Reward Models (DRM):** DRM constitute a fundamental aspect of classical reinforcement learning frameworks, wherein reward functions are learned via discriminative tasks to effectively steer the model towards producing outputs that more accurately align with the intended objectives.

By systematically tailoring our reward system to the distinct requirements of both reasoning and non-reasoning tasks, we enable ERNIE-4.5 to demonstrate enhanced proficiency across a diverse spectrum of applications. The unified reward system not only substantively improves the model's overall performance but also facilitates a more nuanced elucidation of the latent preferences and evaluative criteria intrinsic to human judgment, thereby facilitating the progression toward more nuanced, sophisticated, and human-aligned interactions.

4.1.3 Reinforcement Learning

We conduct the RL training process of the ERNIE-4.5 within the Proximal Policy Optimization (PPO) (Schulman et al., 2017; Ouyang et al., 2022) framework. To enhance training stability and optimize the model's ultimate performance, we introduce the key techniques of our RL training recipe:

- **Progressive Reinforcement Learning (PRL):** PRL implements a three-stage Reinforcement Learning (RL) algorithm as shown in Figure 4, which employs a staged progression: (1) In the initial stage, the model is trained exclusively on logic corpora, which systematically build up the robust foundational ability for logical analysis and abstract reasoning, serving as the cornerstone for all subsequent stages of learning within the PRL framework. (2) Moving on to the second stage, the training corpus mainly includes mathematics and programming code. This incorporation facilitates the transfer of abstract reasoning skills to tasks that are characterized

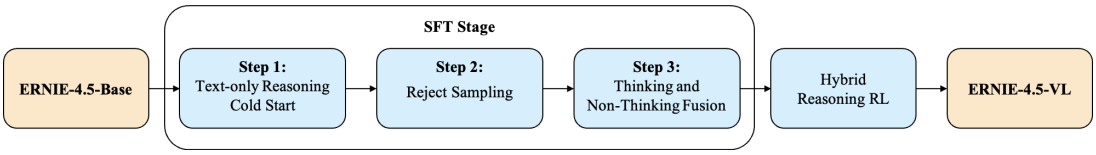


Figure 5: Illustration of VLM Post-Training for ERNIE-4.5-VL.

by stronger requirements for structural expressiveness and executable precision. (3) In the third phase, the model undergoes training on a general dataset that encompasses both non-reasoning and reasoning tasks, which enhances generalizability across a broad spectrum of tasks by systematically leveraging the knowledge acquired in earlier stages.

- **Unified Preference Optimization (UPO):** In conventional reinforcement learning algorithms such as PPO, training is designed to maximize the expected reward associated with a single response generated for each given query, and there is a lack of explicit pairwise comparative signals to guide the learning process. We introduce a novel UPO strategy. Specifically, UPO integrates the pairwise preference modeling loss, i.e., Direct Preference Optimization (DPO) loss, into the PPO framework. Based on the different ways of constructing pairwise preference data, the UPO algorithm can be categorized into online and offline versions. The online-UPO constructs preference pairs by employing a rejection sampling strategy on the multiple responses generated for each query at each reinforcement training iteration, while all preference data for each query is pre-generated before the RL training process in offline-UPO. By integrating learning from preference pairs that capture substantive behavioral distinctions, rather than relying exclusively on potentially unreliable reward signals, the UPO algorithm not only enhances the stability of reinforcement learning training, but also effectively mitigates the risk of reward hacking.

In contrast to traditional reinforcement learning, we systematically develop dedicated data for utilizing both the verifiers and the reward models listed in Section 4.1.2 across comprehensive topical domains introduced in Section 4.1.1, which leads to quite different reward scoring ranges and distributions. To further enhance training stability and optimize the model's ultimate performance, we implement a series of improvements. Specifically, we exclude prompts associated with 1 or 0 accuracy from the dataset tailored to support verifiers, whose reward signals exhibit explicit verifiability. For the remaining prompts, we filter prompts predicated on the intra-group variance of reward signals within each sample cohort. In other words, the prompts corresponding to groups characterized by insubstantial variance, indicating a lack of discriminative information, are excluded from the training process. Besides, within each training iteration, the rewards derived from verifiers and reward models are firstly disentangled, and further stratified according to specific topical domains, yielding multiple topical-domain-specific subsets. A reward normalization is independently applied to each subset. Empirical evaluations demonstrate that these improvements effectively reduce the heterogeneity of reward signals across different sources and domains, thereby enhancing stability and convergence during reinforcement learning.

4.2 Post-Training of VLMs

The entire VLM post-training procedure is shown in Figure 5, which consists of three SFT stages and one reasoning RL stage. Notably, the third SFT stage is designed to incorporate a mixture of thinking and non-thinking data, with the goal of promoting the capabilities of general vision understanding and complex vision reasoning.

4.2.1 Supervised Fine-tuning

We design a supervised fine-tuning framework to strengthen two key aspects of multimodal models: image understanding and reasoning capabilities. Accordingly, we focus on enhancing visual perception through targeted data construction, and unifying thinking and non-thinking behaviors via progressive training strategies.

Data. Through empirical observation during training, we identify that certain challenging tasks—such as puzzle tests, geometry problems, function analysis, and chart interpretation—require strong reasoning capabilities, yet vision-language models (VLMs) often struggle with foundational perceptual understanding (Rahmanzadehgervi et al., 2024; Chen et al., 2024). While enhancing this perceptual capacity is critical, a major obstacle lies in the scarcity of dense image-caption pairs in natural corpora.

To overcome this, we synthesize a large volume of perceptual data, including programmatically generated puzzles (Ghosal et al., 2024), geometric figures, and mathematical functions. These synthetic datasets

afford fine-grained control over spatial layout and structural properties, enabling the generation of high-quality visual-text pairs with minimal ambiguity. However, synthetic data, despite being clean and scalable, lack the visual variability, noise, and contextual richness of real-world images—limiting their generalizability and motivating a shift toward recaptioning natural STEM imagery.

Accordingly, we perform fine-grained caption synthesis on large-scale collections of real STEM images. Unlike synthetic data, natural images demand captions that are both informative and hallucination-resistant. To this end, we frame captioning as a constrained optimization problem: generating image descriptions that allow a text-only reasoning model to solve the associated question without visual input. Our pipeline begins with extracting problem-answer pairs from curated datasets. Captions are generated by a VLM and validated via repeated inference by a text-only model. Only samples yielding consistent correct answers are retained. Furthermore, we filter out samples solvable via visible text (e.g., OCR), ensuring that visual understanding is required.

By integrating this synthesized perceptual data during post-training, we achieve significant improvements in the model's ability to comprehend and reason about visually complex STEM tasks, including puzzle tests, mathematical problem solving, and related domains, which thereby lays a solid foundation for the model's image understanding capabilities.

Thinking and Non-Thinking Joint Training. Obtaining high-quality multimodal reasoning data for cold-start training presents significant challenges. While manually annotated samples can ensure a high degree of accuracy, they often fall short in diversity and coverage. To overcome these challenges and reduce the cost of extensive manual annotation, we propose a three-stage progressive training framework that leverages cross-modal transfer capabilities and expert merging techniques.

- **Step 1: Text-only Reasoning Cold Start.** We collect a diverse corpus of text-only reasoning data spanning mathematics, science, code generation, instruction following and dialogue. To ensure high-quality reasoning supervision, we apply a combination of agent-based and rule-based filters to remove samples exhibiting flawed logic, including circular reasoning, contradictions and conceptual errors. Remarkably, despite being trained exclusively on curated textual data and never exposed to visual inputs, the model exhibits emergent multimodal reasoning behaviors, for example, producing reflective cues such as "let me take another look at the image."
- **Step 2: Reject Sampling for Multimodal Enhancement.** Building upon the Step 1 model, we employ reject sampling to generate reasoning data for vision-related capabilities across STEM, comprehension tasks, chart and document analysis, and creative writing. This process systematically expands the coverage of reasoning capabilities while ensuring data quality through the verifiable reward mechanisms detailed in Section 4.2.2. Furthermore, during the RL training phase, we continuously track higher-quality response trajectories via the verification system. These superior trajectories are persistently recorded and progressively incorporated to update and enrich our supervised fine-tuning (SFT) multimodal reasoning dataset.
- **Step 3: Thinking and Non-Thinking Fusion.** After strengthening multimodal reasoning through targeted data generation in Step 2, we proceed to unify reasoning and non-reasoning capabilities into a single model through two approaches:
  1. **Mixed training with reasoning and non-reasoning data:** We conduct joint training using both reasoning (generated in step 2) and non-reasoning datasets. For all non-reasoning data, we prepend empty thinking tags `<think>\n\n</think>` to the response, with these tags masked and excluded from gradient updates. This approach enables the model to maintain non-reasoning capabilities while preserving reasoning competencies.
  2. **Experts Merging:** Following DeepSeek-R1T-Chimera (GmbH, 2025), we merge experts from the thinking and non-thinking models by transferring multimodal experts from the non-reasoning model to the reasoning model. This fusion strategy creates a unified model with both reasoning and non-reasoning capabilities, where non-reasoning performance surpasses the original baseline. This approach enables us to effectively combine models with distinct strengths in reasoning and visual perception.

This progressive training methodology successfully addresses the cold-start challenge for multimodal reasoning models while achieving superior performance across both reasoning and non-reasoning tasks .

4.2.2 Reinforcement Learning with Verifiable Rewards

Reinforcement Learning with Verifiable Rewards (RLVR) (DeepSeek-AI, 2025; Lambert et al., 2024) has emerged as a crucial paradigm for improving the alignment and performance of multimodal language models in domains where ground-truth verification is feasible. In this section, we introduce several types

of tasks employed in our multimodal RL training that leverage verifier-based reward mechanisms. These tasks include visual STEM, visual puzzles and UI2Code (Chen et al., 2018).

Visual STEM. Visual STEM (Science, Technology, Engineering, and Mathematics) problems consist of image-based questions that are accompanied by ground-truth answers, making them particularly suitable for use in RLVR. We curate a diverse collection of visual STEM questions from both open-sourced resources and proprietary K-12 educational resources. Similar to the Guo et al. (2025), we reformulate the multiple-choice questions into open-ended formats to discourage models from random guessing. Additionally, we filtered out examples that models consistently answered correctly or incorrectly, as such questions contribute little to learning progress and reduce training efficiency. This curation pipeline ensures a high-quality, challenging dataset conducive to effective policy learning.

Visual Puzzles. Visual puzzles are image-based reasoning tasks that require visual perception and cognitive reasoning to arrive at a correct answer, including pattern recognition and graph reasoning. We synthesize a dataset of over 10k visual puzzles along with their verified solutions for RLVR training. The preprocessing of visual puzzle data follows a similar approach to that used for visual STEM tasks. In contrast to conventional verification approaches, which prompt models to output their final answers in \boxed{} format and evaluate correctness via string matching (DeepSeek-AI, 2025), we employ two large language models (LLMs) to assess the correctness of the policy model's responses. One LLM is used to evaluate whether the response contains any internally inconsistent or conflicting answers, while the other verifies the correctness of the final answer. A response is considered correct only if both LLMs return positive evaluations. This evaluation strategy does not impose constraints on the response format of the policy model, thereby enabling more flexible and generalizable outputs.

UI2Code. To enhance the model's capability in practical multimodal applications, we collect the UI2Code (Chen et al., 2018) and Image2Struct (Roberts et al., 2024) datasets, which focus on generating HTML code from UI design images. We deploy a *UI2Code verifier* environment, which evaluates the visual fidelity between the user-provided reference image (typically a UI design mockup) and the UI rendered from the HTML code generated by the VLM. This ensures that the VLM learns to produce syntactically correct and visually faithful HTML representations.

Hybrid Reinforcement Learning. To enable the model to perform well in both reasoning and general capabilities, we design a unified reinforcement learning framework that integrates RLVR and RLHF (Ouyang et al., 2022). Accordingly, we develop a multimodal reward model trained using the Bradley-Terry reward modeling objective. The reward model is initialized from ERNIE-4.5-Base to effectively handle queries containing visual inputs. We adopt GRPO (Shao et al., 2024) as our reinforcement learning algorithm, incorporating improvements inspired by DAPO (Yu et al., 2025), including dynamic sampling, overlong filtering, etc. These strategies collectively ensure stable training dynamics and enhance exploratory capability.

5 Training Framework

The training of ERNIE 4.5 is supported by **PaddlePaddle** (Ma et al., 2019). The inherent heterogeneity of multimodal model coupled with the large-scale MoE architecture presents significant systemic challenges for distributed training at scale. We introduce an optimized training framework with the following key innovations:

1. **Heterogeneous Parallelism for Multimodal Model Training:** ERNIE 4.5 combines a ViT encoder with a multimodal MoE backbone. The fundamental divergences in parameter scale, computational complexity, and memory requirements between these components create challenges for homogeneous parallelism strategy. To mitigate this, we introduce a heterogeneous parallelism strategy for efficient joint training. Moreover, we propose a hierarchical load balance method to enhance scaling efficiency for variable-resolution training.
2. **Hybrid Parallelism for MoE Backbone:** Through meticulous co-design with ERNIE 4.5 architecture, we implement intra-node expert parallelism to eliminate the overhead associated with cross-node all-to-all communication. In addition, we propose a memory-efficient pipeline scheduling method to reduce activation memory during large-scale training.

In addition, we make other notable optimizations to further improve training performance and stability.

1. **FP8 Mixed Precision Training:** We introduce an FP8 mixed-precision training framework with fine-grained memory optimization, well-designed operator fusion and communication optimiza-

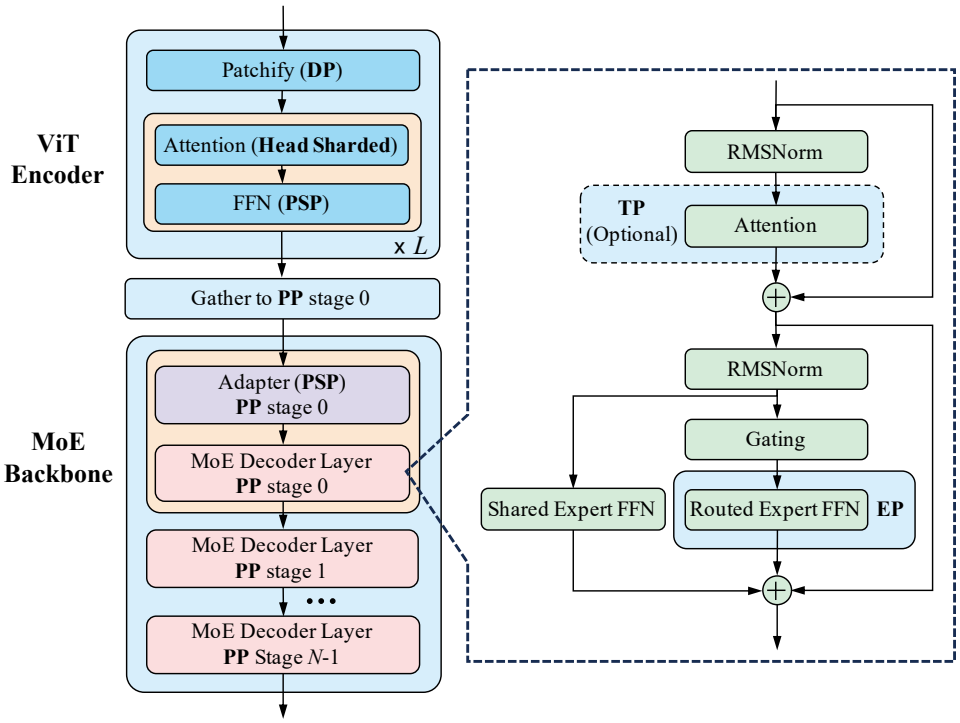


Figure 6: Overview of ERNIE 4.5's multimodal heterogeneous parallelism strategy.

tion. The precision for each operator and communication is well designed to simultaneously maximize training throughput while maintaining convergence.

2. **Computational Optimizations:** To minimize recomputation overhead, we propose an optimized recomputation strategy that achieves superior computation-memory tradeoffs. We also integrate FlashMask (Wang et al., 2025) to accelerate attention operators.
3. **Framework-Native Fault Tolerance System:** We introduce a fault tolerance system, which deeply integrates with our training framework, to overcome the challenges of frequent failures in large-scale training. Especially, we propose a Zero Cost Checkpoint technique, a superior checkpointing approach, to minimize the interruption cost.

Our largest ERNIE 4.5 language model employs an 8-way expert parallelism (EP) (Lepikhin et al., 2021), 12-way pipeline parallelism (PP) (Huang et al., 2019), and ZeRO-1 data parallelism (DP) (Rajbhandari et al., 2020) configuration. Through these comprehensive optimizations above, we achieve 47% Model FLOPs Utilization (MFU) in our largest ERNIE 4.5 language model, on 2016 NVIDIA H800 GPUs and RoCE interconnection with 4096 sequence length and 15120 global batch size.

5.1 Heterogeneous Parallelism for Multimodal Model Training

5.1.1 Heterogeneous Parallelism Architecture

As illustrated in Figure 1, ERNIE 4.5 supports unified training with mixed text, image, and video modalities. For ERNIE-4.5-VL-424B-A47B-Base, the vision inputs (images and videos) are processed by a unified ViT encoder comprising 630 million parameters. This encoder is jointly trained with a backbone network that employs a large-scale MoE architecture, totaling 424 billion parameters with 47 billion parameters activated during computation.

The large-scale MoE backbone necessitates hybrid parallelism training, combining expert parallelism (EP), pipeline parallelism (PP), and optional tensor parallelism (TP) (Shoeybi et al., 2019), integrated with ZeRO-1 data parallelism (DP). While a straightforward approach would place the ViT encoder only in the first pipeline stage of the MoE backbone, this induces severe workload imbalance across pipeline stages, significantly degrading training efficiency. Given its substantially smaller parameter count, the ViT encoder is optimally suited for data parallelism. To enable efficient joint training of the ViT encoder and MoE backbone, we propose the heterogeneous parallelism strategy illustrated in Figure 6. The ViT encoder parameters are replicated across all devices, and the data parallelism dimension of the ViT encoder is nested within the hybrid parallelism topology of the MoE backbone. The internal

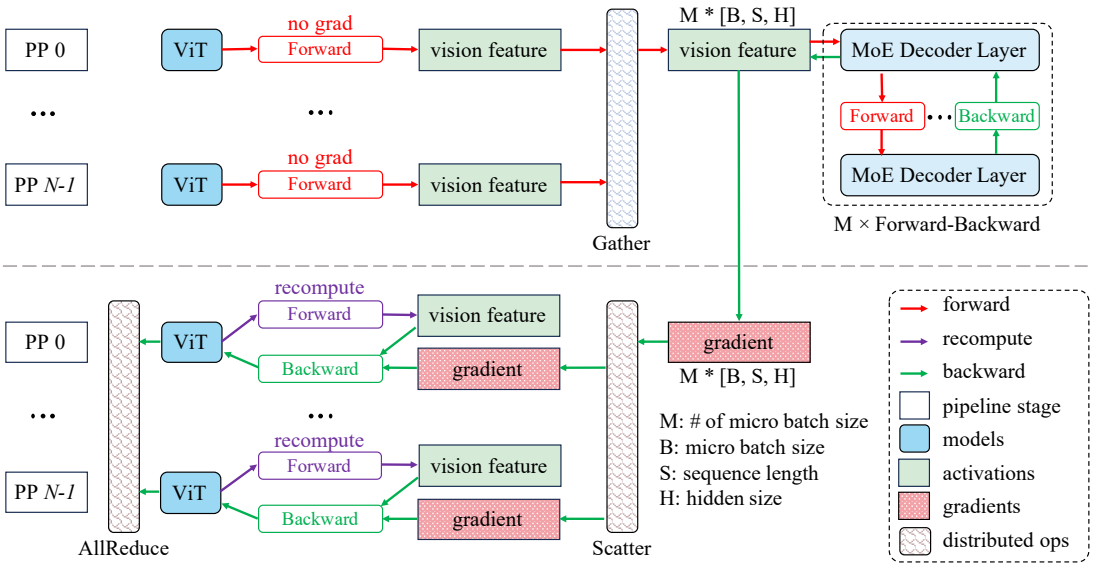


Figure 7: Joint training of the ViT encoder and the MoE backbone.

parallelism architecture of the ViT encoder and the adapter in Figure 6 would be comprehensively detailed in Section 5.1.2.

In the forward pass, the ViT encoder on each device computes vision features independently. These features are then *gathered* to the first pipeline stage of the MoE backbone. Subsequent forward propagation occurs through the MoE backbone's pipeline parallelism stages. However, during the backward pass, the gradients of the ViT encoder parameters cannot be computed directly. This is because the automatic differentiation backpropagation of the pipeline parallelism naturally ends in the first module of the MoE backbone but not the ViT encoder.

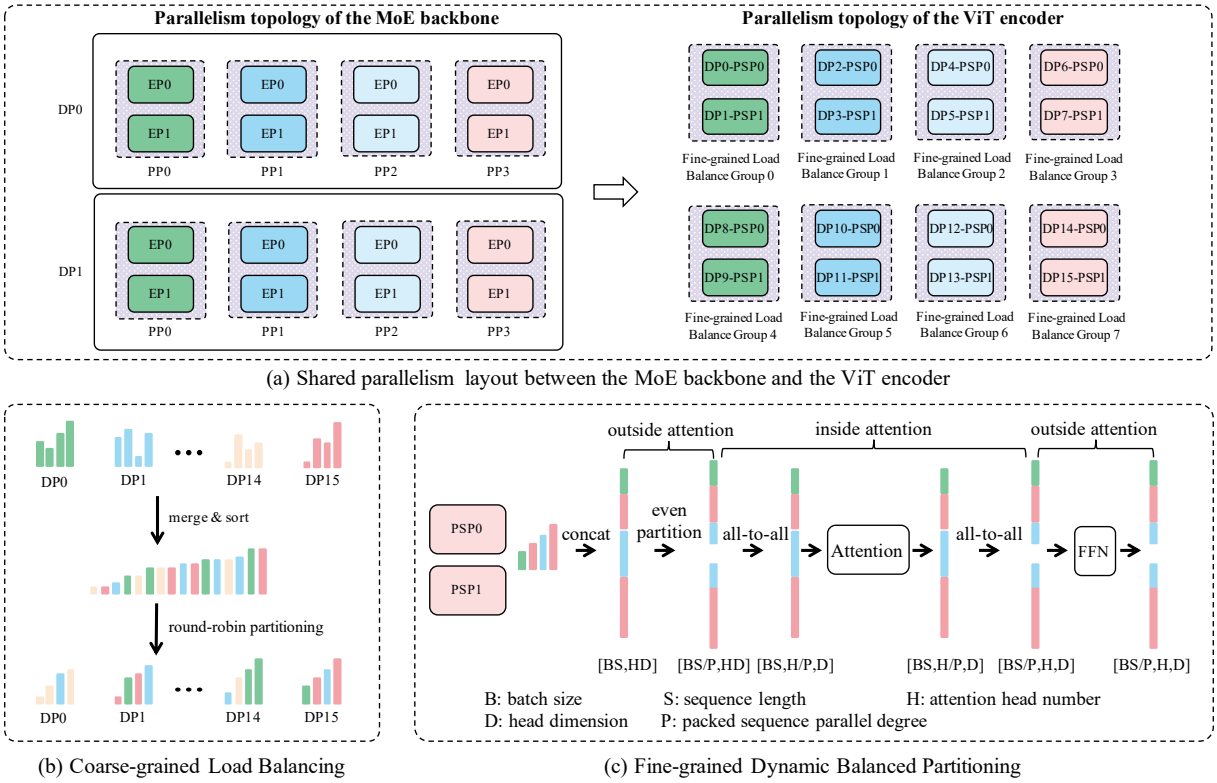
To address this challenge, we implement a customized backpropagation mechanism in Figure 7 to train the ViT encoder. Upon completing the MoE backbone's backward pass, all vision feature gradients become available in the first pipeline stage. These gradients are then *scattered* to each pipeline stage of the MoE backbone, allowing each ViT encoder with different data parallel ranks to receive gradients specific to its local vision features. In this way, we can perform standard automatic differentiation backpropagation through the ViT encoder. Finally, since ViT operates in data parallel mode, its parameter gradients should be synchronized across all devices via an *all-reduce* communication to ensure consistent parameter updates. Notably, the recomputation of the ViT encoder in Figure 7 is optional and serves to reduce the activation memory of the ViT encoder.

5.1.2 Hierarchical Load Balance Strategy

ERNIE 4.5 demonstrates enhanced adaptability by supporting input images and videos with arbitrary and continuously variable resolutions. To enable ViT encoder training with variable resolutions, we organize tokens from patchified images or video frames into packed sequences (Dehghani et al., 2023a). However, multimodal data exhibits significant imbalance challenges: it is not only that the image and video data differ in spatial resolutions, but also video data presents more severe imbalance due to temporal length variations. Therefore, both the number of packed sequences per training sample and the token count per packed sequence exhibit significant variation. This variability leads to extreme computational and memory imbalances between different data parallel ranks of the ViT encoder.

To address these challenges, we propose a hierarchical multimodal load balance strategy, as illustrated in Figure 8. Our load balance method is summarized as follows:

Level 1: Coarse-grained Load Balance. First, we collect and sort all packed sequences in ascending order by their token counts across the data parallel group of the ViT encoder. Then, using a round-robin partitioning algorithm, we distribute the packed sequences to each device to ensure approximately balanced total token counts. This redistribution of packed sequences achieves coarse-grained computational and memory load balance, as shown in Figure 8(b).



(a) Shared parallelism layout between the MoE backbone and the ViT encoder

(b) Coarse-grained Load Balancing

(c) Fine-grained Dynamic Balanced Partitioning

Figure 8: Illustration of hierarchical load balance strategy for variable resolution multimodal training.

Level 2: Fine-grained Dynamic Balanced Partitioning. Due to the natural token count divergences in each packed sequence, the total token counts on each device may differ a lot after coarse-grained load balance. Specifically, we perform dynamic partitioning methods inside and outside attention operators to achieve load balance further, as illustrated in Figure 8(c):

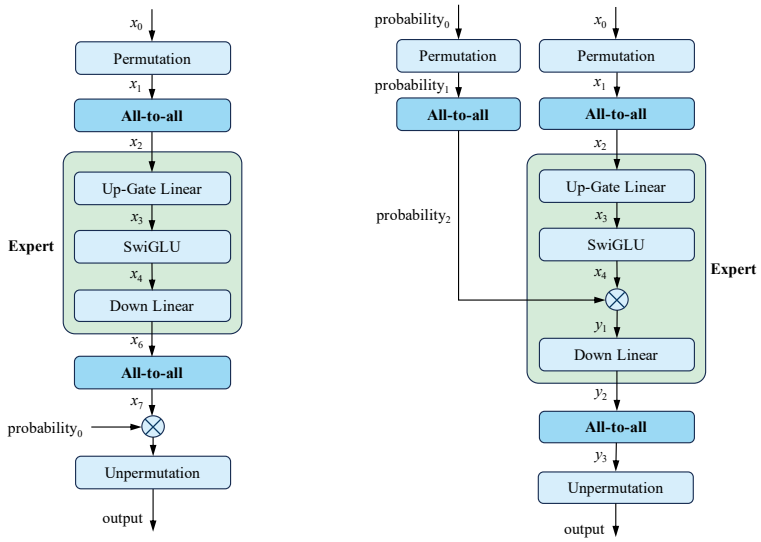
- **Outside Attention Operations:** We propose the packed sequence parallelism (PSP) strategy for the operators outside the attention. Unlike the sequence parallelism proposed in Megatron-LM (Korthikanti et al., 2023) which builds upon tensor parallelism and partitions the model parameters, our proposed packed sequence parallelism method concatenates the packed sequences and evenly partitions along the sequence length dimension. The packed sequence parallelism is also applied in the adapter module in Figure 6.
- **Inside Attention Operations:** Since attention operators require the full sequence length dimension for each packed sequence, we perform an all-to-all communication before attention computation to exchange the sequence length and attention head dimensions. After attention computation, these dimensions are exchanged back to resume packed sequence parallelism.

Through our hierarchical load balance strategy, the resource utilization efficiency of computation, memory and communication is significantly improved. Experimental results demonstrate that ERNIE-4.5-VL-424B-A47B-Base achieves up to 32% overall performance improvement in end-to-end multimodal training compared to baseline approaches without load balance.

5.2 Hybrid Parallelism for MoE Backbone

To effectively scale the training of our largest ERNIE 4.5 language model, we employ three-dimensional parallelism on MoE backbone, specifically, expert parallelism, pipeline parallelism and ZeRO-1 data parallelism, for text pre-training. As for its multimodal model pre-training, we incorporate tensor parallelism to accommodate the increased sequence length and extra vision experts parameters.

Furthermore, we introduce several techniques to optimize the memory footprint and reduce communication overhead, ultimately enhancing the training efficiency. These innovations ensure that our approach provides superior scalability and performance in the training of large-scale MoE models.



(a) Conventional MoE implementation (b) ERNIE 4.5's MoE implementation

Figure 9: Comparsion of different MoE implementions.

5.2.1 Intra-Node Expert Parallelism

We design the model architecture and configuration to avoid costly inter-node expert parallelism communication. By confining expert parallelism communication to intra-node, we implement MoE all-to-all communication based on NCCL-compatible collective primitives. This approach achieves end-to-end throughput comparable to DeepEP-based MoE implementations (Zhao et al., 2025) on ERNIE 4.5, and can be easily deployed on AI clusters without NVIDIA GPUs and InfiniBand (IB) networks.

As shown in Figure 9(a), conventional MoE implementations apply gating probability multiplication operator after the second all-to-all communication. This method necessitates retaining the output tensor of the second all-to-all communication for backpropagation, creating significant memory pressure. Our solution in Figure 9(b) repositions the gate probability multiplication operator within the expert computation block. This architectural modification enables immediate release of the second all-to-all output tensor after consumption. While introducing minor overhead through probability permutation and an additional lightweight all-to-all operation, this optimization significantly reduces the peak memory usage, and eliminates numerous recomputations during backpropagation.

5.2.2 Memory-Efficient Pipeline Scheduling

When scaling to larger cluster training, maintaining a fixed global batch size necessitates reducing the gradient accumulation steps, which in turn increases pipeline bubble time fraction, significantly degrading training throughput. Virtual Pipeline Parallelism (VPP) (Narayanan et al., 2021) is usually adopted to reduce the pipeline bubble time fraction.

Generally, the first pipeline stage of VPP is expected to consume the most activation memory. However, the last pipeline stage involves loss function computations, which can become a memory bottleneck instead. To address this, we propose a memory-efficient virtual pipeline scheduling strategy. Once the last pipeline stage completes the forward computation of the loss function, it immediately begins its backward computation and releases the activation memory of the loss function. In this way, the last pipeline stage should atmost retain the activation memory of a single VPP chunk. Figure 10 and 11 illustrate our memory-efficient pipeline scheduling under forward-then-backward (F-then-B) and one-forward-one-backward (1F1B) .

When the gradient accumulation steps are fewer than twice of the PP degree, we can only use the F-then-B scheduling method. In these scenarios, we propose a parameter gradient release technique to reduce the memory usage. At the end of each training step, we release the memory allocated for parameter gradients. The parameter gradient release method significantly reduces peak memory usage in F-then-B scheduling, especially when maintaining FP32 gradients during BF16 or FP8 mixed-precision training.

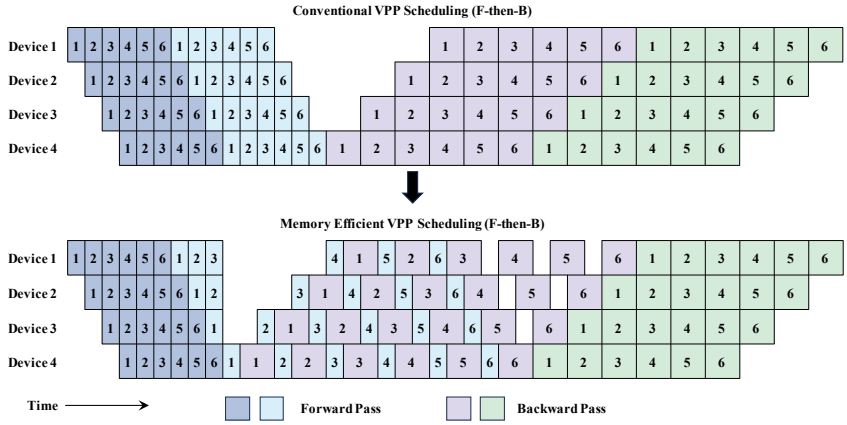


Figure 10: Memory-efficient F-then-B virtual pipeline scheduling.

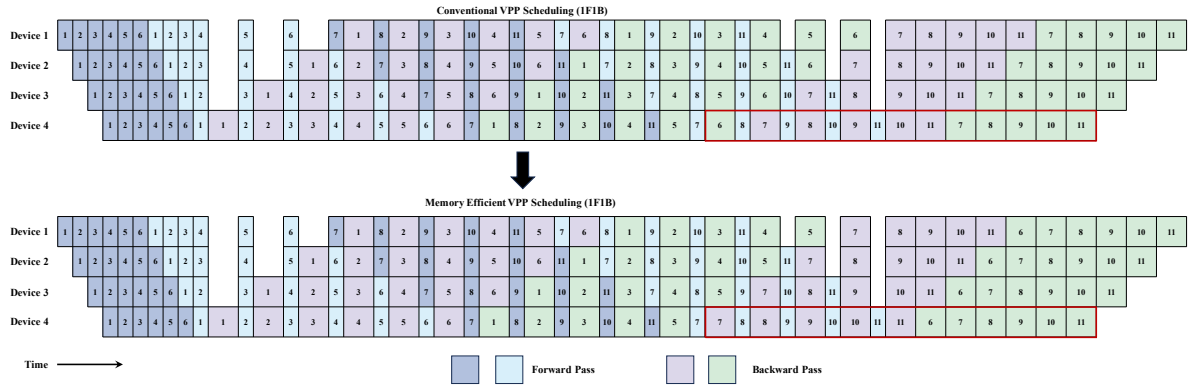


Figure 11: Memory-efficient 1F1B virtual pipeline scheduling.

5.3 FP8 Mixed Precision Training

FP8 format (Micikevicius et al., 2022) reduces the bit-width by half compared to BF16, offering significant advantages in large model training including improved computational throughput, reduced memory consumption, and decreased communication overhead (NVIDIA, 2024; Peng et al., 2023; torchao, 2024). ERNIE 4.5 adopts a similar quantization strategy to DeepSeek-V3 (DeepSeek-AI et al., 2024b) in the MoE FFN modules, utilizing E4M3 FP8 numerical format with an online quantization strategy that employs block-wise quantization for weights and tile-wise quantization for activations. The FP8 mixed precision training strategy for ERNIE 4.5 is illustrated in Figure 12. We highlight our engineering insights from efficient end-to-end FP8 training to benefit the community.

Fine-Grained Memory Optimization on FP8 Training. The primary benefit of FP8 mixed precision training stems from memory savings, enabling us to reduce the most expensive recomputations to improve the throughput. In the MoE FFN module, the major activation memory comes from the input activations of the up-gate linear, down linear, SwiGLU and gate probability multiplication.

1. For the up-gate linear, we retain its FP8 input activations X_{FP8} for the backward pass rather than the BF16 tensors X_{BF16}. In the backward pass, the FP8 quantization version of transposed X_{BF16} is required to compute the weight gradient. Therefore, we need to apply a *dequantize-transpose-quantize* operation to X_{FP8} during weight gradient computation. In this way, we can reduce the memory usage of the up-gate linear, and the first all-to-all communication can be performed in FP8 precision to save the communication cost. It is a tradeoff between the memory, communication and precision, and we found that this method can keep the same convergence rates with the baseline implementation.
2. For the down linear, there are two options to save the memory: (1) retain the BF16 output tensor of the up-gate linear; (2) recompute up-gate linear using the X_{FP8} tensor above to generate the BF16 output tensor of the up-gate linear. Both of these two methods requires a lightweight recomputation of the SwiGLU and gate probability multiplication operators, so that we can save the memory of the input tensors of these two operators.

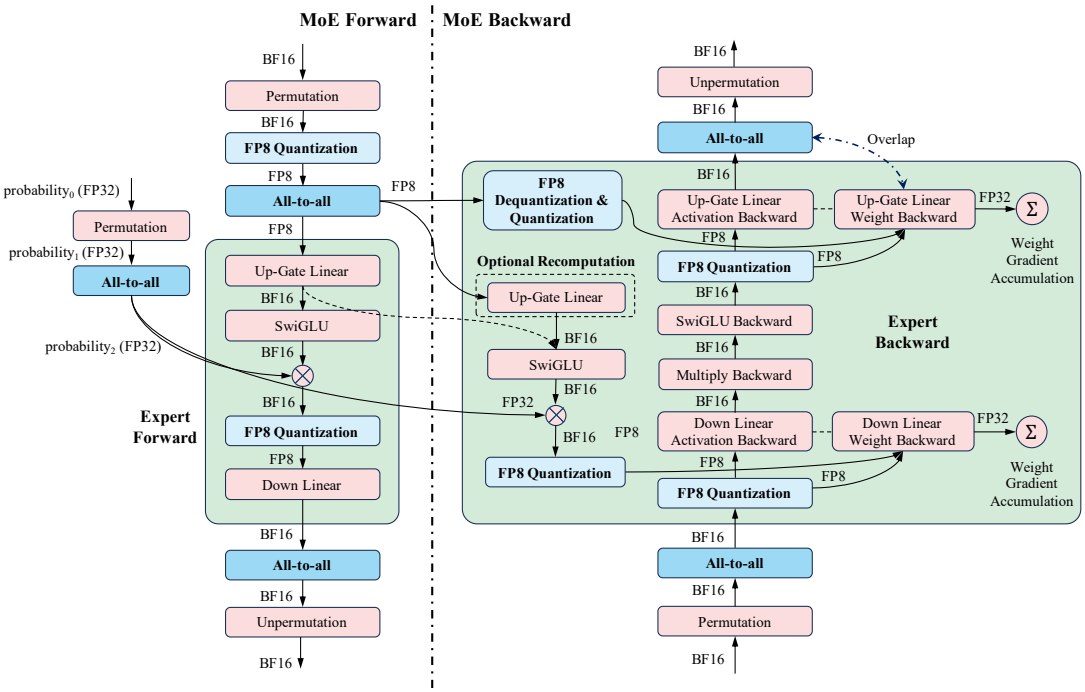


Figure 12: FP8 mixed precision training strategy.

FP8 Quantization Operator Fusion Optimization. We reduce data movement overhead and improve computational intensity through operator fusion, specifically: (1) fusion of permutation and FP8 quantization in the forward pass, and (2) fusion of SwiGLU, gate probability multiplication, and FP8 quantization in both forward and backward passes.

FP8 Communication Optimization and Overlap. In the forward pass, the first all-to-all communication is performed in FP8 precision to reduce the communication costs compared to BF16. In the backward pass, the second all-to-all communication is overlapped with the computation of the up-gate linear weight gradient.

5.4 Computational Optimizations

5.4.1 Recomputation with Best Computation-Memory Tradeoffs

The conventional recomputation strategy operates at the module level to minimize computational overhead by focusing on cost-effective modules. In contrast, ERNIE 4.5 adopts an operator-level recomputation strategy, offering a finer-grained balance between memory and computation, thereby further optimizing training performance.

Consider three operators in Algorithm 1, *op1*, *op2*, and *op3*, each needing to retain input tensors for the backward pass. If we opt not to recompute *op2*, the conventional recomputation method would retain x, $y1$, and $y2$ for backward computation. However, since $y1$ will be recomputed during the backward pass of *recompute(op1, x)*, retaining $y1$ during the forward pass is inefficient and unnecessary. This illustrates that conventional recomputation method is suboptimal at the operator level.

We propose an operator-level recomputation method in Algorithm 2 for training ERNIE 4.5, where we retain the output tensor of *op2* rather than its input tensor. This optimization allows us to retain only x and $y2$ for backward computation while excluding $y1$, thereby reducing memory overhead.

To develop an optimal recomputation strategy, we conducted a detailed analysis of each operator in the model, assessing its memory usage against computational time. By selectively applying operator-level recomputation to the most cost-effective operators—those offering significant memory savings with minimal runtime penalties—we devised an optimal checkpointing scheme that maximizes training efficiency.

Algorithm 1 Pseudocode of the conventional recomputation method

```
def my_module(x):
    y1 = op1(x)
    y2 = op2(y1)
    y3 = op3(y2)
    return y3

def my_module_with_conventional_recomputation(x):
    y1 = recompute(op1, x)
    y2 = op2(y1)
    y3 = recompute(op3, y2)
    return y3
```

Algorithm 2 Pseudocode of our operator-level recomputation method

```
class OperatorLevelRecomputation:
    output_tensor: paddle.Tensor

    def forward(self, op, x):
        if not is_grad_enabled(): # the first forward
            self.output_tensor = op(x)
            return self.output_tensor
        else: # the second forward
            return self.output_tensor

    def backward(self, op_grad, y_grad):
        x_grad = op_grad(y_grad)
        return x_grad

def my_module_with_operator_level_recomputation(x):
    y1 = op1(x)
    y2 = OperatorLevelRecomputation()(op2, y1)
    y3 = op3(y2)
    return y3

y3 = recompute(my_module_with_operator_level_recomputation, x)
```

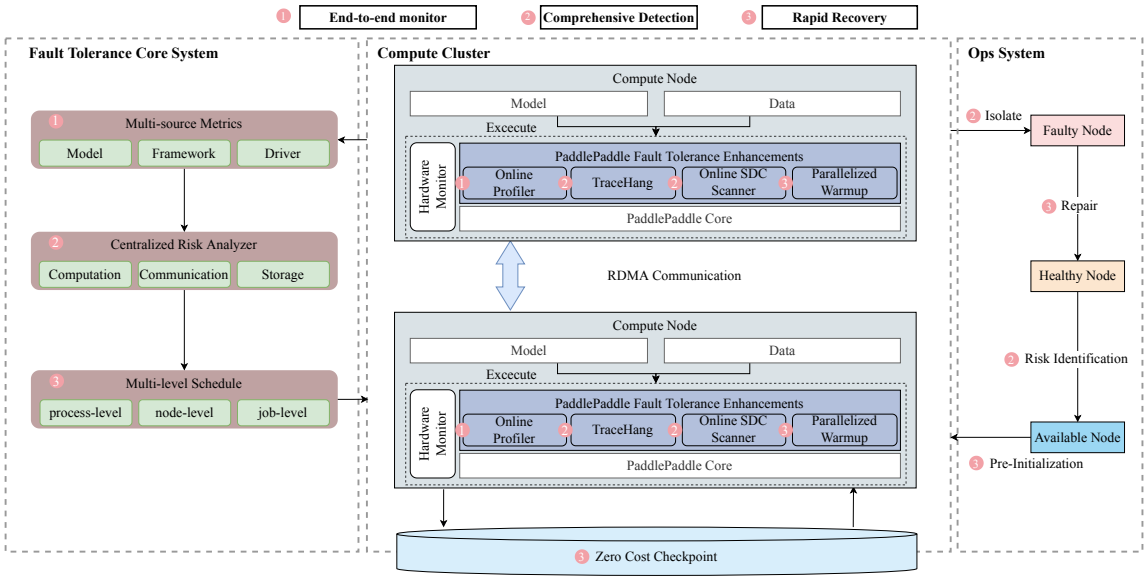


Figure 13: Framework-native fault tolerance system.

5.4.2 FlashMask for Flexible Attention Mask and Long Context Training

We propose FlashMask (Wang et al., 2025) to accommodate the diverse attention masks required in ERNIE 4.5 multimodal pre-training. FlashMask introduces a memory-efficient representation for attention masks across various tasks in both textual and multimodal training, reducing memory complexity from $\mathcal{O}(N^2)$ to $\mathcal{O}(N)$. We also apply FlashMask in Supervised Fine-Tuning (SFT), Direct Policy Optimization (DPO), and Reinforcement Learning training, particularly for long-context training, to reduce memory usage and improve the throughput.

5.5 Framework-Native Fault Tolerance System

The use of large-scale GPU clusters introduces elevated failure interruptions, posing significant challenges to training efficiency. To mitigate the interruption frequency and minimize recovery costs during ERNIE 4.5 training, we propose a novel framework-native fault tolerance system that spans the entire model, framework, and underlying hardware, as illustrated in Figure 13. Unlike traditional fault tolerance systems, our system leverages detailed information on the computation and communication workloads of the training process, enabling robust fault detection and rapid recovery from interruptions. To illustrate the benefits of deep integration of the training framework, we present *TraceHang*, *Online SDC Scanner* for comprehensive detection and *Parallelized Warmup*, *Zero Cost Checkpoint (ZCC)* for rapid recovery.

TraceHang. Hangs without explicit failures can be particularly troubling when training on large-scale clusters. To address this issue, we propose the TraceHang technique that leverages parallelism information and communication records to automatically diagnose the initiator of such hangs. By systematically analyzing these data, TraceHang can identify the root cause of the hang, facilitating quicker resolution and minimizing downtime.

Online SDC Scanner. Silent Data Corruption (SDC) poses a significant threat to model convergence due to its elusive nature (Dixit et al., 2021), which traditionally requires costly offline diagnostic procedures for detection. We observe that pipeline parallelism naturally introduces idle periods on devices, known as pipeline bubble times. During these bubble times at each stage of pipeline parallelism, we perform computations and communications with fixed inputs, subsequently verifying the results against a ground truth. This online method has successfully identified several SDC nodes without adversely affecting training throughput.

Parallelized Warmup. It is well known that the initial step of the training process often exhibits temporary performance degradation due to the initialization of resources such as the cuBLAS handle and NCCL communication handle. However, when training models with a large pipeline degree, this performance degradation is exacerbated by a factor of *pipeline degree p* due to data dependencies present in the pipeline warmup stages. To address this, we conducted a mock pipeline chunk operation across all pipeline stages simultaneously, effectively warming up all devices at the onset of training. This approach

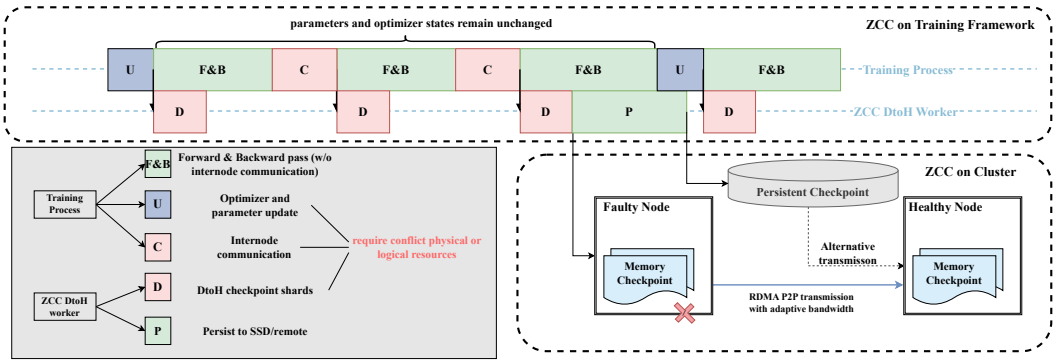


Figure 14: Zero Cost Checkpoint.

reduces the latency of the first step to $1/p$ of the baseline.

Zero Cost Checkpoint (ZCC). When a node failure occurs, it typically necessitates the removal of the faulty node, replacement with a new one, and subsequent resumption of training from the most recent checkpoint. The frequency of checkpointing is a pivotal factor influencing recovery time after interruptions; however, excessive checkpointing can severely impact training efficiency. To overcome this challenge, we introduce Zero Cost Checkpoint (ZCC) technique. This innovative approach enables checkpoint saving at every training step without any overhead for training throughput, thereby ensuring no training progress is lost during interruptions. As shown in Figure 14, ZCC operates on two sides: the training framework side and the cluster side.

On the training framework side, we introduce a zero-cost saving technique based on a key observation: parameters and optimizer states remain unchanged except during optimizer execution. This insight provides ample opportunity to overlap checkpoint routines (e.g., DtoH copies and flushing to persistent storage) with ongoing training routines (typical computation and communication tasks). However, DtoH copies during checkpointing and inter-node communication during training contend for the same physical resources on the PCIe bus. To ensure that asynchronous checkpointing does not degrade training throughput, we divide training routines into two types: conflict routines (involving inter-node communication, such as pipeline parallelism send/receive and ViT vision feature gather/scatter) and non-conflict routines (such as attention operators, dense feed forward networks and intra-node communication). DtoH copies are finely decomposed and allocated to these non-conflict routines to avoid impacting the training process.

On the cluster side, we propose a zero-cost transmission technique. Upon detecting a faulty node, we leverage all available healthy Network Interface Cards (NICs) to adaptively maximize the bandwidth, subsequently performing fast RDMA Peer-to-Peer transmission of the latest in-memory checkpoint to a new healthy node. Moreover, this transmission process is fully overlapped with the subsequent recovery operations, such as the initialization of the new node's execution environment. In cases where the fault node's memory is not accessible, recovery from a checkpoint on persistent storage serves as an alternative method.

Overall, we present a framework-native fault tolerance system, which is seamlessly integrated with PaddlePaddle. This integration enables it to fully comprehend the intricate parallelism inherent in the training processes, thereby offering a superior solution for ensuring stability in large-scale training clusters. Specifically, we have reduced the end-to-end automatic recovery time, which is measured from the timestamp of the latest step before interruption to the timestamp of resuming exactly the same step, to less than 8 minutes for ERNIE 4.5. Furthermore, given the assumption that interruption frequency scales linearly with cluster size, training models on a 10,000-GPU cluster leveraging our fault-tolerant framework can sustain over 98% effective training time.

6 Inference and Deployment

ERNIE 4.5 series consists of MoE and dense models with varying parameter sizes, suitable for various deployment scenarios. Owing to the compact parameter size of ERNIE 4.5, even its largest model can be efficiently deployed. To accommodate diverse application scenarios and hardware platforms, we offer multiple quantization schemes including FP8, INT8, INT4, and even 2-bit weight-only quantization. Specifically, ERNIE-4.5-A47B model can run on 8 GPUs with 8-bit parameters or 4 GPUs with 4-bit parameters. Additionally, we provide Prefill-Decode (PD) disaggregation deployment with large-scale expert

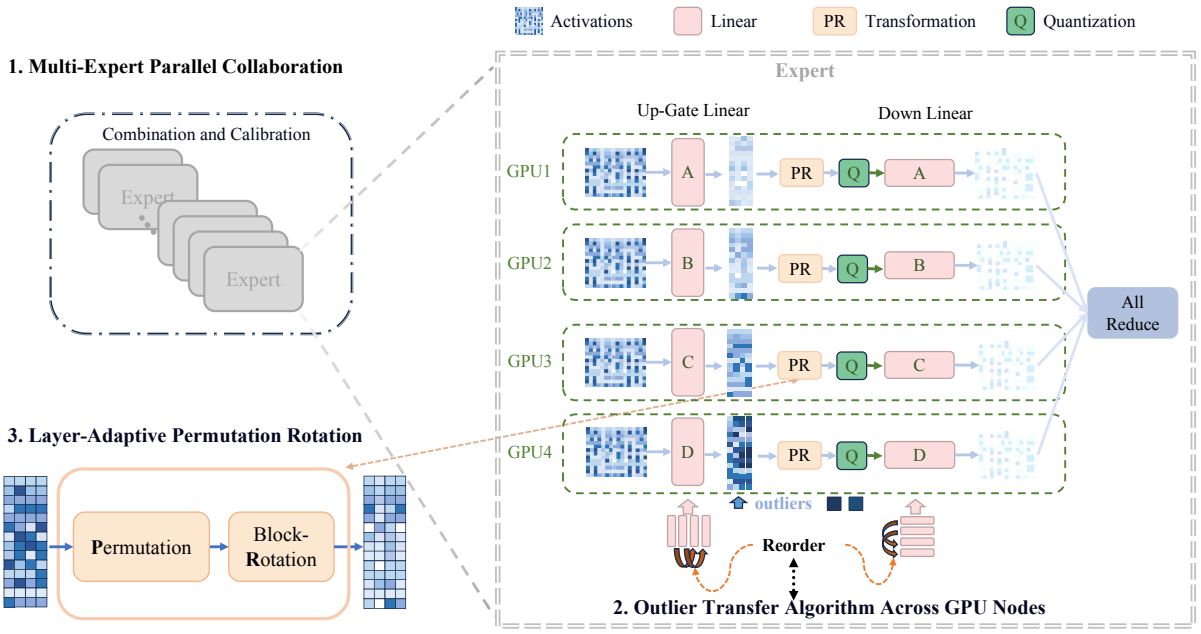


Figure 15: Quantization scheme in the MoE module. Three key methods: 1) Multi-Expert Parallel Collaboration, combines all experts into one tensor to calibrate efficiently and collaboratively; 2) Outlier Transfer Algorithm Across GPU Nodes, aggregates all outliers onto GPU4 by reordering weights of linear; 3) Layer-Adaptive Permutation Rotation, mitigates the impact of intra-node outliers by permutation and rotation.

parallelism. First, we will delve into our mixed-precision quantization solution, which is designed to reduce GPU memory usage and accelerate matrix computation. Next, we will introduce high-performance optimizations in our inference engine, such as hardware-level acceleration and speculative decoding. Finally, we present system-level optimization strategies for production deployment.

6.1 Quantization

To further enhance inference efficiency and support a wider range of hardware, we provide not only BF16 and FP8 inference capabilities but also a variety of lower-precision inference options. We develop multiple quantization strategies to ensure that low-precision models achieve performance comparable to BF16 models. For low-cost scenarios, we adopt W4A8 precision to improve inference throughput. For resource-constrained scenarios, we employ 2-bit weight-only quantization to reduce the deployment barrier. For long-context scenarios, we optimize memory and computation by applying KV cache quantization and attention quantization, respectively.

6.1.1 W4A8 Quantization

In the ERNIE-4.5-300B-A47B model, expert weights constitute over 90% of the total parameters, consuming about 40% of the inference time during the prefilling stage and about 60% during the decode phase of inference. To enable fast and cost-effective inference, we employ W4A8 (INT4 weights and INT8 activations) quantization on the General Matrix Multiplications (GEMMs) associated with the expert components. Through the innovative design of the inference-friendly W4A8 quantization algorithms, we achieve speedup with no accuracy drop and minimal inference-time overhead.

To strike an optimal balance between accuracy and inference cost, we employ channel-wise static INT4 quantization for expert weights and tensor-wise static INT8 quantization for activations.

During the MoE compression process of ERNIE 4.5, several challenges arise:

- **Slow GPTQ on MoE:** GPTQ is a widely-used approach for weight quantization (Frantar et al., 2022). However, for the GPTQ of MoE modules, a large dataset is required not only for activating all the experts but also for optimizing thousands of linear layers. This process would cause significant calibration costs, especially when there are 64 experts in ERNIE-4.5-300B-A47B.
- **Inter-Node Outliers:** When deployed using Tensor Parallel (TP), weights, activations, and their outliers are split into all the GPU nodes, causing globally distributed quantization errors.

- **Intra-Node Outliers:** For each GEMM operation within the MoE module, outliers present in both the weights and activations. The effectiveness of methods such as SmoothQuant (Xiao et al., 2023) and Adaptive Weight Quantization (AWQ) (Lin et al., 2024b) is limited, as they tend to shift outliers between activations and weights, rather than fundamentally mitigating the underlying issue.

We design a quantization framework to enhance and ensure lossless quantization accuracy with minimal inference-time overhead, as illustrated in Figure 15. In detail, we introduce the *Multi-Expert Parallel Collaboration (MEPC)* algorithm to streamline the calibration process. Additionally, we devise a framework incorporating *Outlier Transfer* and *Layer-Adaptive Permutation Rotation* techniques to mitigate outliers, which significantly reduces quantization error. We evaluate our quantized model against the W8A16 baseline across six core task categories. Results demonstrate that our method effectively preserves model performance: performance on general tasks (-0.49%), instruction following (-0.06%), reasoning tasks (+0.11%), math tasks (-0.02%), and code tasks (-1.10%) remains comparable to baseline levels, confirming the quantization's effectiveness.

MEPC: Multi-Expert Parallel Collaboration Quantization. To address the aforementioned bottlenecks of MoE-LLMs activations calibration and GPTQ, we adopt a prefill-only approach during the calibration process to ensure that a sufficient number of experts are activated. For experts remaining unactivated, we employ a shared quantization parameter strategy. Specifically, the quantization parameters for unactivated experts are set as the average of those from all activated experts within the same layer. This method is motivated by our empirical observation that experts in the same layer tend to have highly similar quantization parameters during the quantization process.

Inspired by QMoE (Frantar & Alistarh, 2024), we process all experts in parallel when applying GPTQ by concatenating their weights, thereby fully utilizing the parallel computation capabilities of GPUs and enabling highly efficient GPTQ quantization. Furthermore, we enhance this process with a hotspot-expert parallel GPTQ update strategy. Here, hotspot experts—those deemed more important based on their token activation frequency—are prioritized for optimization during quantization. Specifically, since parallel GPTQ quantization can impose a substantial computational burden due to the large Hessian matrix in the Up-Gate Linear, we employ a targeted GPTQ update strategy for hotspot experts within this layer. In contrast, the Down Linear contains a much smaller Hessian matrix, which allows us to efficiently perform full GPTQ updates for all experts in this layer without excessive computational cost.

Outlier Transfer Algorithm Across GPU Nodes. As previously discussed, activation outliers distributed across GPU nodes can result in quantization loss for all nodes. In the context of TP, Down Linear is partitioned across N GPU nodes, with each node maintaining its own quantization scale. Inspired by RPTQ (Yuan et al., 2023), we implement an outlier transfer algorithm that aggregates all outliers onto a single node, while distributing the regular (non-outlier) activations to the remaining nodes. Specifically, we first collect the channel-wise absolute maximum activation values from all N GPUs. Using these statistics, we globally rank the columns of Up-Gate Linear and the rows of Down Linear, co-locating weight channels with similar activation ranges onto the same GPU. This reordering adjusts the local weight and activation layouts without affecting the final model output, as subsequent AllReduce operations restore consistency through element-wise summation. Notably, our approach performs this reordering offline, resulting in a quantization-friendly model without compromising inference efficiency.

Layer-Adaptive Permutation Rotation Quantization. INT8 static quantization is widely adopted for its superior inference efficiency and broad hardware compatibility. However, it suffers from significant performance degradation in the presence of intra-node outliers. Alternative approaches such as dynamic quantization or FP8, while effective at handling outliers, are often limited by computational overhead or hardware support. To address this challenge, we propose a layer-adaptive permutation rotation quantization framework that efficiently mitigates the impact of outliers while maintaining hardware friendliness.

Inspired by Quarot (Ashkboos et al., 2024), we first employ a block-rotation strategy to smooth outliers in either weights or activations. To further enhance the robustness of this approach, we integrate a permutation mechanism (Lin et al., 2024a), enabling a more flexible redistribution of outliers within each block. Our method begins by pre-analyzing the outlier distribution in each layer's activations to determine whether outlier smoothing should prioritize weights or activations. For layers exhibiting clustered outliers, we apply the permutation-rotation procedure to redistribute these outliers within the rotated blocks, thereby reducing quantization noise and preserving model accuracy. This design achieves a balance between quantization granularity, computational efficiency, and hardware compatibility.

6.1.2 2-Bit Quantization

To further lower the entry barrier for ERNIE 4.5, we implement a nearly lossless 2-bit quantization algorithm, thereby reducing its model size by 80% from the BF16 baseline. The 2-bit ERNIE-4.5-300B-A47B can be deployed on one 141GB H20 GPU.

For scalar-based methods, compressing a model to 2 bits would result in a significant collapse of the model's effectiveness. Currently, the state-of-the-art weight-only quantization algorithms primarily focus on employing vector quantization (Gray, 1984; Liu et al., 2024a) and incoherent processing (Tseng et al., 2024a) to optimize model performance under extremely low-bit quantization. However, both of these approaches introduce computational overhead during inference.

- **Exponential Cost in Dequantization:** Vector quantization necessitates an additional codebook to store the centroid vectors, and the dequantization process involves indexing operations within a vast codebook space to reconstruct the weights, which is a time-consuming task.
- **Computational Overhead from Incoherent Processing:** Meanwhile, although incoherent processing in weight-only quantization can mitigate quantization errors, it requires extra real-time processing of weights during the inference phase.

To address the aforementioned issues, we propose *Convolutional Code Quantization (CCQ)*, a scalar quantization algorithm based on the convolutional code. The approach not only retains the high-precision data quantization capability of vector quantization but also preserves the low computational complexity of scalar quantization. Combined with scale quantization and optimization, we achieve the highest possible compression ratio while simultaneously minimizing inference overhead.

Convolutional Codebook. Inspired by QTIP (Tseng et al., 2024b), we innovatively integrate convolutional code with scalar quantization through a series of meticulously designed coding structures. Based on convolutional codes, we construct a lookup-free codebook that achieves a linear mapping between the codebook and weight vectors, thereby optimizing inference performance. Meanwhile, by drawing on the concept of data mapping from vector quantization, we minimize the performance degradation of the model under extremely low-bit conditions.

Hybrid Encoding. We employ convolutional codes with varying coding configurations to accommodate the storage of encoded values in INT8 and INT16 formats. As a result, we successfully compress 4-bit scalar quantization to an equivalent of 2.75 bits and 3-bit scalar quantization to 2.5 bits.

Code Clustering. Furthermore, by analyzing the distribution of encoded values across each channel, we observe that they conform to a normal distribution, enabling deeper compression along the coding dimension. Through clustering of the convolutional codes, we can compress any coding configuration to an equivalent of 2 bits, thereby further enhancing the model compression rate.

6.1.3 Attention and KV Cache Quantization

Supporting larger batch sizes and longer context lengths is essential for reducing the cost per request and accommodating diverse application scenarios. The GPU memory overhead of the Key-Value (KV) cache and the computational cost of attention mechanisms escalate linearly or even quadratically, posing new bottlenecks for inference computation.

Mainstream approaches like KIVI (Zirui Liu et al., 2023) and KVQuant (Hooper et al., 2024) maintain accuracy during KV cache quantization by employing dynamic token-wise quantization and outlier isolation at inference time. However, this dynamic processing **incurs inference latency**. To alleviate this issue, we propose a **static quantization solution** for both the KV cache and attention mechanisms (illustrated in Figure 16). In pursuit of the optimal trade-off between model accuracy and inference performance, we introduce several key strategies.

- **Flexible Quantization Configurations:** Our proposed method supports two distinct quantization granularities, head-wise and channel-wise. We provide a variety of precision levels, such as FP8, INT8, and INT4 to cater to diverse inference requirements and hardware constraints.
- **Static Quantization:** We utilize static quantization, employing a dedicated sampled dataset derived from the SFT phase, which circumvents the need for scale collections during the inference process.
- **Light-Weight RHT:** We leverage blocked Random Hadamard Transform (RHT) as proposed in Liu et al. (2024d) and Lin et al. (2024a). This technique is employed to mitigate the impact

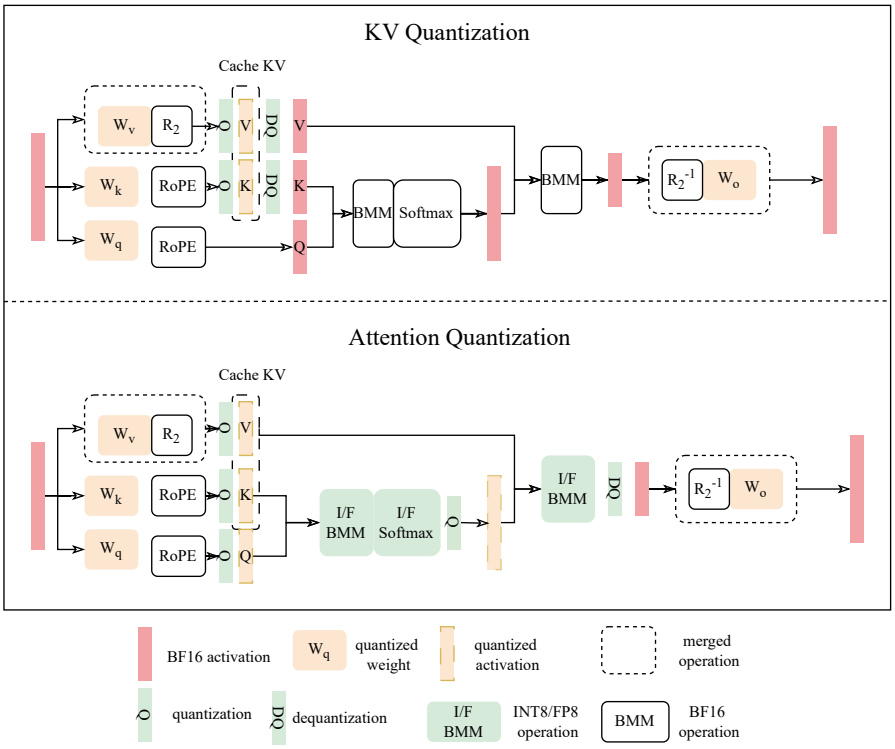


Figure 16: Quantization scheme in the attention module: KV Cache is quantized as 4 bits or 8 bits for memory reduction. Batched Matrix Multiplication (BMM) and Softmax are computed in 8-bit precision for latency reduction. Both quantization recipes involve RHT to eliminate outliers in the activations.

of outliers in the activations. Notably, the RHT operation can be seamlessly integrated into the matrices W_v and W_o, thereby incurring no additional computational overhead during inference.

To further enhance the precision of FP8 quantization, we adopt a strategy of reducing the representable range of FP8 values, which yields substantial improvements in quantization performance. Experimental results demonstrate that this approach achieves approximately 2% accuracy improvement on reasoning tasks. We attribute this improvement to the characteristics of the E4M3 FP8 format, which exhibits significant sparsity and large quantization errors near its extreme values (± 448). To address this issue, we proactively clip or exclude certain outliers during the calibration process. Specifically, we set a customized value for fp8_max (distinct from the default 448), as shown in Equation 6:

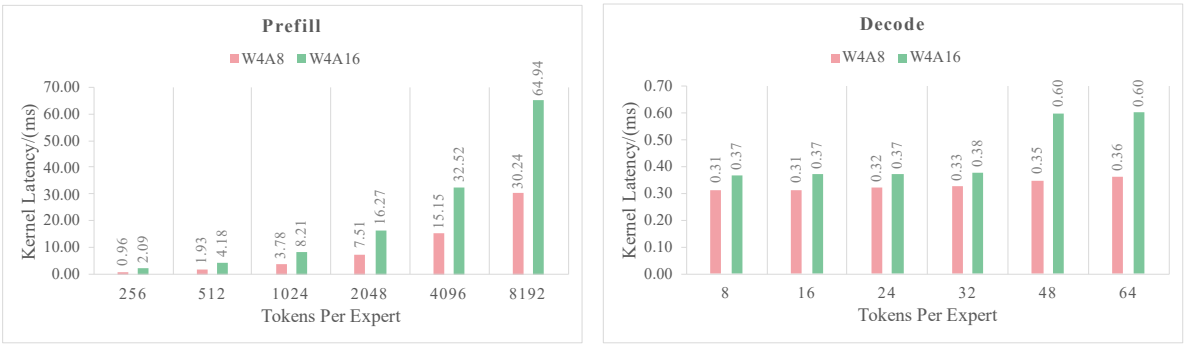
$$x_{\text{quantized}} = \text{float8\_e4m3}\left(\text{clip}\left(\left\lfloor \frac{x_{\text{bf16}} \times \text{fp8\_max}}{scale} \right\rceil, -448.0, 448.0\right)\right). \tag{6}$$

6.2 Inference Acceleration

Having fully considered the collaboration with quantization algorithms and hardware architectures, we develop a series of efficiently optimized inference kernels. For the MoE W4A8 quantization, an efficient CUTLASS kernel, with fast bit-shift dequantization, achieves significant improvements in memory throughput and inference speed. For attention and KV quantization, we design different polynomial approximations of the exponential function for INT8-formatted and FP8-formatted softmax calculation on different GPU architectures, delivering 1.68x speedup over FlashAttention-2 and 1.48x speedup over FlashAttention-3.

6.2.1 W4A8 Kernel Acceleration

In LLM inference, 4-bit weight-only and INT8/FP8 quantization are commonly employed acceleration techniques, but W4A8-based GEMM on the MoE module delivers two key advantages compared with them: (1) a 50% memory reduction compared to FP8/INT8 quantization through 4-bit weight representa-



(a) (b)

Figure 17: Kernel execution time on A800, N=1792, K=8192. (a) Comparison of prefill kernel execution time, (b) Comparison of decode kernel execution time.

tion, and (2) 2 times improvement in theoretical computational throughput over FP16 or BF16 baseline by leveraging INT8 Tensor Core.

To optimize computational efficiency, we streamline the fast conversion process from INT4 to INT8 weights which is co-designed with quantization algorithms. The technical details are as follows:

- **Range Mapping:** The representable range of INT4 is [-8, 7]. We constrain the mapping range to [-7, 7].
- **Bit-Shift Conversion:** The INT4 weights are converted to INT8 via a 4-bit left shift operation, resulting in an effective mapped range of [-112, 112] for the INT8 weights.
- **Weight Layout Optimization:** To minimize operations required for achieving the Tensor Core-compatible INT8 weight layout, we employ a weight prepacking strategy that arranges weights in an interleaved format. This design enables the conversion of every eight INT8 data elements using only 3 instructions (via efficient utilization of LOP3 and bit-shift operations).

We leverage Tensor Core units for multiply-accumulate operations and implement a high-efficiency kernel via CUTLASS, with dequantization in the epilogue phase to maximize pipeline throughput. A problem visitor module dynamically balances expert workloads, while systematic hyperparameter pre-tuning enables real-time adaptive configuration selection during inference.

The aforementioned optimizations yield the following performance gains for W4A8 MoE GEMM kernel:

- Achieves 70% to 80% of memory bandwidth for decoder-shaped workloads.
- Delivers over 100% speedup on encoder-shaped workloads compared to W4A16 and over 20% speedup on decoder-shaped workloads.

To mitigate precision loss during intermediate computations, we maintain INT32 accumulation precision throughout the MAC phase. For activations, we support flexible quantization granularity (per-token or per-expert), while for weights, we adopt per-channel quantization to achieve optimal accuracy.

6.2.2 Efficient Attention Kernel

To balance accuracy with hardware capabilities in attention quantization, we leverage FP8 computation on Hopper architecture GPU and INT8 on Ampere architecture GPU. When batched GEMM computations leverage Tensor Core-based INT8 or FP8, their execution time is significantly reduced. Consequently, softmax and dequantization operations computed on CUDA Cores emerge as the performance bottleneck. To address this, we employ a novel implementation for exponential function and dequantization to minimize computation overhead on the CUDA Cores.

INT8-Formatted Attention. We implement a fast exponentiation computation, building upon the findings presented in Schraudolph (1999). The exponential function for floating-point data x can be expressed as:

$$e^x \approx F_{float}\left(2^{23} \times \lfloor \text{scale} \times x + \text{bias} \rfloor\right) \tag{7}$$

where scale and bias are predefined constants, and $\lfloor\ \rfloor$ is calculated by integer casting, F_{float} represents *reinterpret_cast* in programing. After the INT8 quantization of $Q * K^T$ operation, the dequantization step (with a coefficient denoted as S_{qk}) and the subsequent exponential operation can be merged into a single Fused Multiply-Add (FMA) instruction, as described by the following formula:

$$e^{x \times S_{qk}} \approx F_{float}\Big(\lfloor 2^{23} \times \text{scale} \times S_{qk} \rfloor \times x + \lfloor 2^{23} \times \text{bias} \rfloor\Big) \tag{8}$$

Experimental results demonstrate that this approach achieves a **68%** performance improvement over FlashAttention-2 in the prefilling stage with 128K as the input sequence length on A800 GPUs.

FP8-Formatted Attention. For the attention module using C4 (asymmetrical channel-wise 4-bit quantization for KV cache), we quantize the two GEMM operations into the FP8E4M3 format and implement the following optimizations.

- **Fast Conversion:** By placing 4-bit KV data in the lower 4 bits of FP8E4M3 data, UINT4-to-FP8E4M3 conversion is established:

$$Y = 2^{-9} \times X \qquad Y \textit{ in } \text{FP8E4M3} \qquad X \textit{ in } \text{UINT4} \tag{9}$$

  Similarly, UINT8-to-BF16 conversion can also adopt this method. For CacheKV storage, the original data sequence [0,1,2,3,4,5,6,7] is interleaved into [0,4,1,5,2,6,3,7], enabling the conversion of eight 4-bit values with a single bit operation and LOP3 instruction. a single SHIFT instruction and a single LOP3 instruction. This reduces the cost to 0.25 instructions per value conversion.
- **Fast Dequantization:** For the dequantization of K, the following equivalence holds:

$$P = \text{softmax}\left(\frac{Q \times (K^T - Z_k) \times S_{qk}}{\sqrt{d_k}}\right) \tag{10}$$

$$= \text{softmax}\left(\frac{(Q \times S_{qk}) \times K^T}{\sqrt{d_k}}\right) \tag{11}$$

  where Z_k is the dequantization zero point of K, and S_{qk} is the product of dequantization scales of Q and K. The fast UINT4-to-FP8E4M3 conversion can be applied to K, which only requires shift and bitwise AND operations, significantly reducing FMA instructions and data type conversion overhead. For V dequantization, the computation follows:

$$O = P \times V * S_v - P \times Z_v \times S_v \tag{12}$$

  where Z_v and S_v are the quantization zero point and coefficient of V. Since the matrix multiplication of $P \times V$ accumulates along the sequence length dimension while S_v is a channel-wise parameter, the dequantization of V can be deferred until after $P \times V$. Additionally, the term $P \times Z_v \times S_v$ can be efficiently computed using pre-stored softmax row summations.
- **Matrix Transposition:** The hopper architecture GPUs require matrix B in A*B to be loaded from shared memory for GEMM. Therefore, we restructure the computation flow for 4-bit quantized K and V, transformed into following formula to avoid moving quantized K and V from registers to shared memory.

$$P = (K \times Q^T)^T \quad \text{and} \quad O = (V^T \times P^T)^T \tag{13}$$

These optimizations generally make the input 128k sequence length of C4 decode attention reach more than 80% of the theoretical bandwidth utilization. In the FP8 attention, the exponential function is approximated using its limit. Ultimately, compared to FlashAttention-3's FP8 implementation, this approach delivers an average **50%** speedup in the prefilling stage with 128K as the input sequence length.

6.2.3 Speculative Decoding

ERNIE 4.5 is equipped with a Multi-Token Prediction (MTP) module, so speculative decoding is used during the inference phase. We implement a unified speculative decoding framework, which features a concurrent proposer architecture, enabling seamless integration with MTP schemes. By leveraging parallel sampling and verification, as well as deeply customized attention kernel, MTP achieves a 60% increase in output throughput while maintaining TPOT (Time per output token) comparable to autoregressive decoding.

6.3 Deployment

For the ERNIE-4.5-300B-A47B model, we optimize system throughput and latency through a PD disaggregation deployment combined with expert parallelism. Specifically, the prefilling stage employs 8-way expert parallelism (EP8) without tensor parallelism for the attention module. During the decoding stage, the system supports flexible parallelization ranging from EP8 to EP64 configurations. Given the distinct computational characteristics of prefilling and decoding stages, we adopt different quantization schemes tailored to each stage's requirements. The prefilling phase utilizes block-wise FP8 quantization, while the decoding stage employs W4A8 quantization to optimize memory bandwidth and reduce latency.

The efficacy of the PD disaggregated system critically depends on three key factors: (1) inter-node transmission efficiency of KV Cache, (2) all-to-all communication efficiency for expert routing, and (3) load balancing across distributed compute resources. These components collectively determine the overall system performance and scalability of the disaggregated architecture.

KV Cache Transfer. We develop a cross-node KV cache transfer module based on Remote Direct Memory Access (RDMA) with minimal dependencies. It only requires a basic RDMA runtime environment such as OFED, making it lightweight and easy to deploy. Moreover, to achieve better performance, our implementation has some detailed differences with existent solutions, including reducing the Completion Queue Entry (CQE) number and enabling PCIe relaxed ordering. Furthermore, building on an unified KV Cache transfer design, our system intelligently routes KV cache traffic through the optimal channel, utilizing NVLink intra-node and RDMA inter-node.

All-to-All Communication. For all-to-all communication, we utilize NVLink P2P copy for intra-node data transfer in low-latency scenarios. This enhancement improves decoding throughput in smaller-scale decoding instances, with EP8 deployment efficiency increased by 70% compared to EP16.

Multi-level Load Balancing. Load balancing is extremely important in a large-scale PD disaggregation system that incorporates data parallelism and expert parallelism. We implement multi-level load balancing in our system.

- **Load Balancing of Data Parallelism:** A global load-aware scheduler dispatches queries based on the KV cache hit rate and token count, ensuring that the number of tokens processed each DP rank is relatively balanced in the prefilling and decoding stage.
- **Load Balancing of Expert Parallelism:** Our system incorporates a dynamic expert redundancy strategy (DeepSeek-AI et al., 2024b) and enforces global expert rescheduling to optimize load balancing in MoE computations. To reduce the impact of expert rearrangement on online services, we employ a scheme of weight prefetching and asynchronous loading to achieve extremely low latency service stagnation. A gray-level rearrangement strategy is used to avoid the occurrence of the thundering herd problem and to ensure the stability of the overall service.
- **Load Balancing of PD Disaggregation:** In the PD disaggregation system, a mismatched PD ratio can make it challenging to meet Service-Level Objective (SLO) and may result in low GPU utilization. In online service, dynamically varying input and output lengths result in an imbalance between the prefilling and decoding stages. To mitigate this, we continuously monitor workload across all instances and dynamically adjust the ratio between prefill and decode to achieve relative balance. However, we observe that system throughput remains frequently bottlenecked by resource constraints in the prefilling stage. To address this limitation, our system supports dynamic role switching for instances. This capability enables decoding instances to intelligently handle prefilling for short-input requests based on real-time load conditions. Critically, this approach incurs virtually no impact on TPOT while simultaneously boosting resource utilization within the Decode stage.

Leveraging the aforementioned optimization strategies, ERNIE-4.5-300B-A47B achieves an inference performance of 56k input TPS and 18k output TPS per H800 node without prompt caching for an input length of 2K and an output length of 400, under the constraint of a 50ms Time Per Output Token (TPOT) latency.

It is worth noting that, under our PD disaggregation deployment solution for the largest ERNIE 4.5 language model, both prefilling and decoding demand a minimum deployment unit of one node with 8 GPUs which significantly enhances operational efficiency and reduces the complexity of cluster management in production environments.

While the PD disaggregation on large-scale GPU clusters delivers outstanding performance, streamlined single-node deployment provides advantages for rapid application in diverse scenarios. Even the largest

model, ERNIE-4.5-A47B, can be deployed on a single node, such as 4x80GB A800 or H800 with 4-bit precision and 1x141G H20 with 2-bit precision as mentioned above. ERNIE 4.5 can not only be efficiently deployed on NVIDIA GPUs but also on a wide range of hardware platforms based on PaddlePaddle, including Kunlunxin XPU, Hygon DCU, Ascend NPU, and beyond.

7 Open-Source Development Tools

We open-source ERNIEKit[1] and FastDeploy[2] based on PaddlePaddle framework to streamline model training and deployment for ERNIE 4.5. These toolkits feature industrial-grade capabilities, resource-efficient training and inference workflows, and multi-hardware compatibility.

7.1 ERNIEKit

ERNIEKit is an industrial-grade development toolkit for ERNIE 4.5. It provides model training and compression capabilities, including pre-training, Supervised Fine-Tuning (SFT), Low-Rank Adaptation (LoRA), Direct Preference Optimization (DPO), Quantization-Aware Training (QAT) and Post-Training Quantization (PTQ) techniques. To enable developers to fully experience the capabilities of ERNIE 4.5, we have introduced the following technical innovations:

- **Industrial-Grade High-Performance Pre-Training:** The toolkit has provided a high-performance implementation of our largest ERNIE 4.5 language model pre-training, including the hybrid parallelism training strategy and FP8 mixed precision optimization.
- **Low-Bit Quantization-Aware Fine-tuning:** To significantly reduce fine-tuning and deployment resources for ERNIE 4.5, we introduce a novel FP8-QAT solution integrating low-precision training with optimizer offloading. The resulting models match the quality of BF16 fine-tuned (SFT) models while reducing the minimum GPU requirement of our largest ERNIE 4.5 language model from 96 GPUs to 16 GPUs. Crucially, unlike pre-training FP8 schemes requiring dynamic (block/tile-wise) quantization, models from our FP8-QAT support efficient offline tensor-wise FP8 quantization for inference, eliminating runtime quantization overhead.
- **Visual Training & Inference Interface:** The integrated Gradio-based WebUI supports zero-code fine-tuning, alignment, and inference operations on ERNIE 4.5 with out-of-the-box functionality.

7.2 FastDeploy

FastDeploy is an inference and deployment toolkit for large language models and vision language models. It is designed for ease of use, offering out-of-the-box compatibility with vLLM interfaces. To address the requirements of both enterprise users and individual developers, we have introduced the following technical features:

- **PD Disaggregation with Multi-level Load Balancing:** We open-source an industrial-grade Prefill-Decode Disaggregation Deployment with context caching. It offers optimal distributed inference configurations for NVIDIA GPU based on ERNIE 4.5's architecture characteristics. Benefiting from the unified KV Cache transfer design, our system automatically selects the most efficient communication link between NVLink and RDMA. During multi-machine deployment, instances dynamically switch between prefill and decode roles according to load conditions to improve throughput.
- **Comprehensive Low-Bit Quantized Inference Support:** FastDeploy supports a variety of quantization precisions, including W8A8, W8A16, W4A8, W4A16, and W2A16, with data types such as INT4, INT8, FP8, and BF16. Notably, we provide a built-in 2-bit weight-only quantized model to lower the deployment resource requirements for ERNIE 4.5. This model delivers nearly lossless performance compared to FP8 precision across multiple benchmarks, while enabling single-card inference on NVIDIA H20 GPU with 141GB of memory.
- **Multi-hardware Support:** Benefiting from PaddlePaddle's multi-hardware adaptation capabilities, in addition to NVIDIA GPU, ERNIE 4.5 also supports inference deployment on various chips including Kunlunxin XPU, Hygon DCU and Ascend NPU.

[1] https://github.com/PaddlePaddle/ERNIE
[2] https://github.com/PaddlePaddle/FastDeploy

8 Evaluation and Results

To comprehensively demonstrate the capabilities of ERNIE 4.5, we conducte extensive evaluations on a wide variety of text and vision benchmarks. The comparative results of ERNIE 4.5's language models and multimodal models against other leading models are presented in Sections 8.1 and 8.2, respectively.

8.1 Evaluation of Language Models

8.1.1 Results of Pre-Trained Language Models

In this section, we present a comprehensive evaluation of ERNIE-4.5-Base against state-of-the-art models, including DeepSeek-V3-Base (Wang et al., 2024b) and Qwen3-30B-A3B-Base (Yang et al., 2025a). We conduct systematic evaluations across a diverse set of benchmarks encompassing five fundamental capabilities for these pre-trained models. The evaluation benchmarks and their respective methodologies are described in detail below:

- **General Tasks:** C-Eval (5-shot) (Huang et al., 2023), CMMLU (5-shot) (Li et al., 2024a), MMCU (5-shot) (Zeng, 2023), AGIEval (5-shot) (Zhong et al., 2024), MMLU (5-shot) (Hendrycks et al., 2021a), MMLU-Redux (5-shot) (Gema et al., 2025), and MMLU-Pro (5-shot) (Wang et al., 2024c), assessing comprehensive knowledge and understanding across diverse academic and professional domains.
- **Factual Knowledge:** SimpleQA (10-shot) (Wei et al., 2024a) and ChineseSimpleQA (10-shot) (He et al., 2024a), testing world knowledge capabilities across different languages and domains.
- **Reasoning:** BBH (3-shot) (Sanh & Raffel, 2023), DROP (3-shot) (Dua et al., 2019), ARC-Easy & Challenge (5-shot) (Clark et al., 2018), HellaSwag (5-shot) (Zellers et al., 2019), PIQA (5-shot) (Bisk et al., 2020), WinoGrande (5-shot) (Sakaguchi et al., 2021), and CLUEWSC (5-shot) (Xu et al., 2020), evaluating logical reasoning, reading comprehension, and commonsense inference capabilities.
- **Code Generation & Comprehension:** Evalplus (0-shot, including HumanEval+ and MBPP+) (Liu et al., 2023), and MultiPL-E (0-shot) (Cassano et al., 2023), assessing programming abilities, code understanding across multiple programming languages.
- **Mathematical Reasoning:** GSM8K (5-shot, we removed the calculator tag to ensure stable model performance) (Cobbe et al., 2021), MATH (8-shot, chain-of-thought and direct-answer) (Hendrycks et al., 2021b), and CM17K (5-shot) (Qin et al., 2021), evaluating mathematical problem-solving across various mathematical domains.

The results are summarized in Table 4. Based on the overall evaluation, we highlight several key findings regarding the performance of ERNIE-4.5-Base models.

ERNIE-4.5-300B-A47B-Base. ERNIE-4.5-300B-A47B-Base surpasses DeepSeek-V3-671B-A37B-Base on 22 out of 28 benchmarks, demonstrating leading performance across all major capability categories. This underscores the substantial improvements in generalization, reasoning, and knowledge-intensive tasks brought about by scaling up the ERNIE-4.5-Base model relative to other state-of-the-art large models. The model is particularly strong on general and knowledge tasks in Chinese, including CMMLU and ChineseSimpleQA. This superior performance in Chinese can be attributed to the iterative refinement of pre-training data, which enhances data quality and learning efficiency. In addition, the model's outstanding QA performance benefits from optimized common-sense data formats and the incorporation of high-quality synthetic data, further strengthening its ability to handle factual question answering and complex Chinese linguistic contexts.

ERNIE-4.5-21B-A3B-Base. With a total parameter size of 21B (approximately 70% that of Qwen3-30B), ERNIE-4.5-21B-A3B-Base outperforms Qwen3-30B-A3B-Base on several math and reasoning benchmarks, including BBH and CMATH. ERNIE-4.5-21B-A3B-Base remains highly competitive given its significantly smaller model size, demonstrating notable parameter efficiency and favorable performance trade-offs.

Both ERNIE-4.5-21B-A3B-Base and ERNIE-4.5-300B-A47B-Base achieve impressive results on knowledge-centric tasks, attaining high scores on SimpleQA and ChineseSimpleQA. These outcomes reflect effective knowledge memorization and language understanding, primarily due to large-scale pre-training on massive, high-quality text data. This comprehensive pre-training establishes a solid foundation for subsequent post-training, resulting in robust capabilities in factual knowledge acquisition.

Capability	Benchmark	ERNIE-4.5-0.3B-Base	Qwen3-30B-A3B-Base	ERNIE-4.5-21B-A3B-Base	DeepSeek-V3 671B-A37B-Base	ERNIE-4.5-300B-A47B-Base
General	C-Eval	40.7	87.2	**88.0**	90.2	**91.5**
	CMMLU	39.8	86.0	**86.2**	88.2	**91.2**
	MMCU	37.2	88.8	**94.0**	94.0	**95.9**
	AGIEVAL	28.5	**72.8**	68.4	75.8	**78.4**
	MMLU	41.9	**81.0**	78.9	**87.9**	87.4
	MMLU-Redux	43.2	**84.6**	80.7	**89.4**	89.2
	MMLU-Pro	16.0	**56.7**	51.2	66.7	**69.5**
Knowledge	SimpleQA	1.8	7.1	**30.4**	24.9	**38.4**
	ChineseSimpleQA	7.4	52.0	**54.8**	64.8	**72.2**
Reasoning	BBH	30.4	72.7	**77.5**	87.5	**89.4**
	DROP	28.6	39.6	**70.8**	**84.6**	82.8
	ARC-Easy	60.7	**98.6**	96.9	98.7	**99.2**
	ARC-Challenge	40.6	**93.7**	90.7	95.1	**96.3**
	HellaSwag	33.0	87.7	**92.1**	91.3	**96.6**
	PIQA	55.2	**91.0**	80.6	94.1	**94.6**
	WinoGrande	51.3	**76.3**	70.6	88.2	**88.3**
	CLUEWSC	48.6	**76.8**	76.2	**81.4**	79.9
Math	GSM8K	25.2	70.8	**81.0**	90.6	**91.8**
	MATH	12.4	**61.0**	50.8	63.0	**69.1**
	CM17K	4.6	**69.4**	64.9	79.3	**86.4**
	MGSM	2.7	**71.2**	69.2	79.8	**88.6**
	ASDIV	29.8	82.8	**83.5**	89.7	**90.2**
	SVAMP	24.0	**86.3**	79.7	**92.7**	90.0
	MATHQA	21.6	39.4	**56.1**	76.9	**83.0**
	CMATH	52.2	88.9	**93.7**	90.7	**96.2**
Coding	HumanEval+	25.0	83.5	**86.0**	76.0	**84.8**
	MBPP+	40.2	**76.2**	75.1	**76.7**	74.9
	MultiPL-E	14.2	**66.1**	65.4	63.0	**68.6**

Table 4: Performace of ERNIE-4.5-Base pre-trained models. Here, we report our own evaluation results for Qwen3-30B-A3B-Base and DeepSeek-V3-671B-A37B-Base models rather than scores reported in their original papers. It should be noted that the **base** version of Qwen3-235B-A22B model has not been open-sourced, and therefore could not be evaluated. However, we include comparisons against its post-trained version in Table 5. Bold values indicate the best scores within each group of comparable parameter sizes.

8.1.2 Results of Post-Trained Language Models

In this section, we present a comprehensive evaluation of ERNIE-4.5 against state-of-the-art models, including DeepSeek-V3-0324 (DeepSeek-AI et al., 2024b), GPT-4.1 (OpenAI, 2025a) and Qwen3-235B-A22B (Yang et al., 2025a) under non-thinking mode. We conduct systematic evaluations across a diverse set of benchmarks encompassing six fundamental capabilities as shown in Table 5. The evaluation benchmarks and their respective methodologies are described in detail below:

- **General Tasks:** MMLU (Hendrycks et al., 2021a), MMLU-Pro (Wang et al., 2024c), C-Eval (Huang et al., 2023), and CMMLU (Li et al., 2024a), LiveBench (2024-11-25) (White et al., 2025), assessing broad knowledge and understanding across multi-domain question answering.
- **Knowledge Tasks:** ChineseSimpleQA (Li et al., 2022) and SimpleQA (Wei et al., 2024a), testing factual retrieval and multilingual knowledge accuracy.
- **Instruction Following:** IFEval (Zhou et al., 2023), Multi-IF (He et al., 2024b), Sysbench (Kim et al., 2023) evaluating complex instruction following abilities.
- **Math Tasks:** MATH-500 (Zhou & Li, 2023), GSM8K (Cobbe et al., 2021), CMATH (Wei et al., 2023), OlympicArena (Huang et al., 2024), AIME'24 and AIME'25 (AIME, 2025) , evaluating complex mathematical reasoning and problem solving.
- **Reasoning Tasks:** BBH (Sanh & Raffel, 2023), MUSR (Gao & Chen, 2023), DROP (Dua et al., 2019), Zebralogic (Tan et al., 2023), evaluating logical reasoning, commonsense inference, and

Capability	Benchmark	Qwen3-235B-A22B	DeepSeek-V3-0324	GPT-4.1	ERNIE-4.5-300B-A47B
General	C-Eval	86.1	87.3	77.9	**90.6**
	CMMLU	87.5	88.2	78.1	**90.2**
	MMLU	85.5	86.5	**90.2**	86.5
	MMLU-Pro	72.9	**81.2**	80.7	78.4
	LiveBench	62.5	66.3	67.7	**68.9**
Knowledge	ChineseSimpleQA	63.0	72.0	64.0	**77.1**
	SimpleQA	12.4	27.3	40.2	**45.9**
Instruction Following	IFEval	83.2	82.3	87.4	**88.0**
	Multi-IF	70.2	66.9	70.8	**76.6**
	Sysbench	51.3	71.3	**74.1**	69.8
Math	MATH-500	91.2	94.0	92.8	**96.4**
	GSM8K	96.4	96.3	95.9	**96.6**
	CMath	95.7	94.8	92.8	**96.7**
	OlympicArena	63.7	**76.2**	69.5	75.3
	AIME'24	40.1	**59.4**	48.1	54.8
	AIME'25	24.7	**47.7**	36.7	35.1
Reasoning	BBH	84.5	91.0	84.1	**94.3**
	MUSR	66.8	65.9	62.6	**69.9**
	DROP	88.7	89.7	89.2	**91.1**
	Zebralogic	37.7	**81.8**	56.3	58.1
Coding	LiveCodeBench	36.1	**45.4**	40.5	38.8
	Humaneval+	89.6	89.6	**92.1**	**92.1**
	MBPP+	77.0	77.8	79.9	**81.0**
	FullStackBench	57.1	64.7	**68.5**	67.1

Table 5: Performance comparison among post-trained language model ERNIE-4.5-300B-A47B and other baselines on text benchmarks. Bold values indicate the best scores.

real-time problem solving.

- **Code Tasks:** HumanEval+ (Liu et al., 2023), MBPP+ (Liu et al., 2023), LiveCodeBench (v6, 20240801-20250501) (Jain et al., 2025), FullStackBench (Cheng et al., 2024), measuring code generation correctness and code comprehension across multiple programming languages.

ERNIE-4.5-300B-A47B. Table 5 reveals the comparative performance of ERNIE-4.5-300B-A47B against other leading open-source models across various benchmarks. Notably, it demonstrates significant strengths in instruction following and knowledge tasks, as evidenced by the state-of-the-art scores on benchmarks such as IFEval, Multi-IF, SimpleQA, and ChineseSimpleQA. The model's strong capabilities in instruction following and knowledge utilization in single-turn, multi-turn, and multilingual scenarios may be attributed to our unified rewarding system, which incorporates carefully designed reward mechanisms to guide the model in better interpreting and adhering to diverse user instructions and internal knowledge.

In mathematics and coding tasks, ERNIE-4.5-300B-A47B significantly outperforms Qwen3-235B-A22B on most benchmarks. The results demonstrate the model's advanced proficiency in mathematical reasoning, accurate computation, and code generation, making it particularly well-suited for applications in mathematical computing and programming-related tasks. However, on some challenging math and coding tasks, there remains room for further improvement.

Lightweight Language Models. The benchmark results for lightweight language models are summarized in Table 6. The experimental results indicate that the lightweight model ERNIE-4.5-21B-A3B achieves competitive performance compared to Qwen3-30B-A3B, despite having approximately 30% fewer total parameters. Furthermore, the ultra-lightweight ERNIE-4.5-0.3B demonstrates reasonable performance given its extremely compact size, enabling efficient inference even on standard laptops.

Capability	Benchmark	ERNIE-4.5-0.3B	Qwen3-30B-A3B	ERNIE-4.5-21B-A3B
General	MMLU	34.3	**77.2**	76.0
	MMLU-Pro	18.9	**69.7**	66.0
	C-Eval	32.9	85.0	**85.4**
	CMMLU	34.5	82.9	**84.8**
	LiveBench	16.7	46.6	**51.3**
Knowledge	ChineseSimpleQA	3.5	45.6	**48.9**
	SimpleQA	0.7	4.7	**24.2**
Instruction Following	IFEval	43.8	**82.3**	80.0
	Multi-IF	23.0	**67.0**	61.0
	Sysbench	2.9	35.5	**48.0**
Math	MATH-500	19.4	**87.4**	87.2
	GSM8K	35.1	**94.8**	93.3
	CMath	49.0	**94.8**	**94.8**
	OlympicArena	8.8	54.9	**58.8**
	AIME'24	0.4	**32.9**	27.9
	AIME'25	0.2	**23.2**	19.7
Reasoning	BBH	30.0	**85.8**	83.0
	MUSR	42.6	58.4	**67.7**
	DROP	36.5	**88.2**	84.4
	Zebralogic	2.5	29.2	**34.1**
Coding	LiveCodeBench	2.6	**31.7**	26.4
	Humaneval+	36.6	**90.2**	89.6
	MBPP+	40.2	75.4	**77.5**
	FullStackBench	11.5	**51.8**	48.2

Table 6: Text benchmark comparison between ERNIE-4.5-21B-A3B and Qwen3-30B-A3B, with ERNIE-4.5-0.3B provided for reference. Bold values indicate the best scores.

8.2 Evaluation of Multimodal Models

In this section, we present a comprehensive multimodal evaluation of ERNIE-4.5-VL against leading baseline models, including OpenAI-o1 (OpenAI, 2025c) and QWen2.5-VL (Bai et al., 2025). We conducted a comprehensive evaluation across multimodal benchmarks designed to assess five fundamental capabilities: general visual knowledge comprehension, document and chart understanding, multimodal reasoning, visual perception, and video understanding. All models are evaluated using a standard zero-shot protocol across all multimodal tasks. The evaluation benchmarks are detailed below:

- **Visual Knowledge:** CCBench (Liu et al., 2024b), SimpleVQA (Wei et al., 2024a), and MMBench-v1.1 (Liu et al., 2024b), assessing general visual knowledge.
- **Document & Chart:** OCRBench (Liu et al., 2024c), TableVQA (Kim et al., 2024), ChartQA (Masry et al., 2022), AI2D (Kembhavi et al., 2016), and DocVQA (validation set) (Mathew et al., 2021), evaluating text recognition, document and chart interpretation on structured information.
- **Multimodal Reasoning:** VisualPuzzle (Song et al., 2025), Zerobench (Roberts et al., 2025), MMMU (Yue et al., 2024a), MMMU-Pro (Yue et al., 2024b), and MathVista (Lu et al., 2023), testing logical reasoning and visual-related problem-solving capabilities through complex visual scenarios.
- **Visual Perception:** CV-Bench (Tong et al., 2024), CountBench (Paiss et al., 2023), and RealWorldQA (xAI, 2024), evaluating model's spatial reasoning, counting, and other basic visual perception capabilities.
- **Video Understanding:** MVBench (Li et al., 2024c), VideoMME with/without subtitles (Fu et al., 2024), and LongVideoBench (Wu et al., 2024), evaluating temporal understanding, video comprehension, and long-form video analysis capabilities.

All image benchmark results are re-evaluated using our own evaluation infrastructure. The results for Qwen2.5-VL are based on evaluations using the open-source model weights. The results for OpenAI-o1 are collected using its official API in June 2025. For OCRBench, RealWorldQA and DocVQA (Mathew et al., 2021), we use only the evaluation metrics provided by their official implementations. For all

Capability	Benchmark	OpenAI-o1	ERNIE-4.5-VL-424B-A47B	ERNIE-4.5-VL-28B-A3B
Visual Knowledge	CCBench	79.7	**83.9**	81.5
	SimpleVQA	**64.1**	59.8	50.8
	MMBench-cn	86.8	**90.9**	87.5
	MMBench-en	85.9	**92.0**	88.8
Doc and Chart	OCRBench	761	**872**	805
	TableVQA	84.5	**86.7**	60.2
	ChartQA	80.4	**86.8**	79.2
	AI2D	94.2	**95.3**	93.5
	DocVQA	81.0	**93.1**	89.1
Multimodal Reasoning	VisualPuzzle	**48.5**	46.3	36.9
	ZeroBench(sub)	20.2	**22.5**	15.3
	Visulogic	**29.0**	27.3	26.0
	MMMU	**77.7**	70.0	61.8
	MMMU-Pro	**65.8**	58.8	50.8
	MathVista	71.4	**78.9**	72.6
Visual Perception	CV-Bench	81.3	**84.8**	76.7
	CountBench	87.2	**89.0**	85.3
	RealWorldQA	72.5	**73.2**	71.8

Table 7: Performance comparison of VLMs in the thinking mode on multimodal benchmarks. QVQ-72B-Preview was excluded from this comparison because it is a preview version, not optimized well on the visual understanding benchmarks we evaluated. Bold values indicate the best scores.

Capability	Benchmark	Qwen2.5-VL-7B	Qwen2.5-VL-32B	ERNIE-4.5-VL-28B-A3B	Qwen2.5-VL-72B	ERNIE-4.5-VL-424B-A47B
Visual Knowledge	CCBench	70.4	73.0	81.0	76.5	**84.4**
	SimpleVQA	51.7	53.6	46.0	**60.1**	58.3
	MMBench-cn	85.7	89.7	85.8	**91.0**	89.8
	MMBench-en	87.0	89.9	86.9	**91.1**	90.8
Document and Chart	OCRBench	868	815	**885**	872	883
	TableVQA	70.6	69.7	70.0	78.1	**79.6**
	ChartQA	80.3	82.9	82.2	84.0	**86.4**
	AI2D	93.3	93.6	95.4	95.4	**96.0**
	DocVQA	93.3	92.2	94.1	92.4	**94.3**
Multimodal Reasoning	VisualPuzzle	31.8	41.1	34.6	**45.0**	41.0
	ZeroBench(sub)	14.1	14.4	16.2	16.2	**20.1**
	Visulogic	26.1	26.9	27.3	25.6	**28.7**
	MMMU	51.8	64.4	58.9	67.1	**67.3**
	MMMU-Pro	38.8	50.4	**53.0**	52.8	52.6
	MathVista	68.5	75.5	74.9	77.6	**78.8**
Visual Perception	CV-Bench	79.7	82.1	76.1	82.6	**85.5**
	CountBench	86.8	83.1	87.6	93.1	**93.3**
	RealWorldQA	70.6	69.7	69.2	72.6	**75.2**
Video Understanding	MVBench	69.6	-	**72.0**	70.4	70.6
	VideoMME w/subs	71.6	77.9	74.4	79.1	**79.7**
	VideoMME wo/subs	65.1	70.5	66.8	**73.3**	70.0
	LongVideoBench	56.0	-	62.1	60.7	**66.2**

Table 8: Performance comparison of VLMs in the non-thinking mode on multimodal benchmarks. Bold values indicate the best scores.

other benchmarks, we follow the official implementation guidelines and additionally incorporate a LLM-as-judge pipeline to ensure the accuracy and reliability of our evaluations. In particular, we use GPT-4.1 (OpenAI, 2025a) and a distilled smaller model that inherits its judging capability as the evaluator. For the CCBench (Liu et al., 2024b) benchmark, we employ a "Caption Then Answer" prompting strategy to improve model performance. For AI2D, we adopt the transparent setting and carefully align our

implementation with Molmo (Deitke et al., 2024).

For the video benchmarks, the results for Qwen2.5-VL are taken from its original paper or huggingface post. To ensure a fair comparison, we adopt a sequence length of 32,768 similar to Qwen2.5-VL. The hyperparameters for dynamic frame-resolution sampling are set as follows: a frame rate of 2 FPS, a maximum of 480 frames, a minimum of 16 frames, and a minimum resolution of 360p (or the original size if it is less than 360p). The actual frame resolution and the number of frames are determined dynamically according to this method, ensuring that the combined token count of the video and text input fits within the sequence length constraint. The answers are automatically graded based on letter matching.

We report results for ERNIE-4.5-VL in both thinking mode and non-thinking mode in Table 7 and Table 8. The detailed performance across different benchmarks is discussed below.

Non-Thinking Mode: Robust Visual Perception and Knowledge Factuality. In the non-thinking mode, ERNIE-4.5-VL demonstrates robust capabilities in perceiving image details and recalling relevant knowledge. It achieves strong performance across multiple visual understanding tasks. For example, ERNIE-4.5-VL-424B-A47B attains high scores on visual perception benchmarks such as CountBench, CV-Bench, and RealWorldQA. ERNIE-4.5-VL also shows strong results in OCR and chart understanding tasks, demonstrates effective scientific diagram comprehension, and delivers solid performance in document understanding, with further examples highlighted in Appendices B.1, B.2 and B.3. In addition, ERNIE-4.5-VL displays impressive video analysis capabilities on video comprehension benchmarks, reflecting its coherent understanding of both short and long video sequences. Appendix B.4 presents an example illustrating its temporal grounding ability, which benefits from the adopted timestamp rendering method.

Beyond perceptual ability, ERNIE-4.5-VL also demonstrates a deep understanding of visual knowledge, as evidenced by its strong performance on both Chinese-specific and general benchmarks. Notably, it excels on CCBench, underscoring its comprehensive grasp of Chinese knowledge and culture. This capability can be attributed to extensive pre-training of vision encoder on visual concepts and further enhancement through multimodal joint training. The model's performance also benefits from the incorporation of high-quality Chinese textual and visual data, further reinforcing its cultural comprehension. Appendix B.5 provides an illustrative case of Chinese ancient character recognition.

Overall, ERNIE-4.5-VL exhibits outstanding proficiency in visual perception, document and chart understanding, and visual knowledge, performing strongly across a range of established benchmarks.

Thinking Mode: Enhanced Multimodal Reasoning Capabilities. Under the thinking mode, ERNIE-4.5-VL not only demonstrates enhanced reasoning abilities compared to the non-thinking mode, but also retains the strong perception capabilities of the latter. ERNIE-4.5-VL-424B-A47B delivers consistently strong results across the various multimodal evaluation benchmarks. Its thinking mode offers a distinct advantage on challenging benchmarks such as MathVista, MMMU, and VisualPuzzle, while maintaining competitive performance on perception-focused datasets like CV-Bench and RealWorldQA. Appendix B.6 includes illustrative examples that highlight the model's capacity for reflective reasoning when solving visual puzzles.

The strong performance of the thinking mode on multimodal reasoning tasks results not only from its advanced reasoning capabilities, but also from its improved understanding of STEM-related images acquired during the SFT stage. At the same time, the non-thinking mode is also improved through the joint training of both modes, as reflected in the model's solid mathematical problem-solving skills and competent multimodal understanding in both thinking and non-thinking scenarios. Appendices B.7 and B.8 provide additional reasoning examples under the thinking mode.

Lightweight VLM. The experimental results show that the lightweight vision-language model ERNIE-4.5-28B-A3B achieves competitive or even superior performance compared to Qwen2.5-VL-7B and Qwen2.5-VL-32B across most benchmarks, despite using significantly fewer activation parameters. Notably, our lightweight model also supports both thinking and non-thinking modes, offering functionalities consistent with ERNIE-4.5-VL-424B-A47B.

9 Conclusion

In this work, we present ERNIE 4.5, a family of large-scale multimodal models. Our models employ a novel heterogeneous MoE structure, which supports parameter sharing as well as modality-specific expert specialization, allowing for more flexible and effective multimodal joint training and knowledge fusion. Extensive evaluations demonstrate that our models achieve state-of-the-art performance across multiple text and multimodal benchmarks, especially in text and vision knowledge factuality, instruction following, visual understanding, and multimodal reasoning.

The development of ERNIE 4.5 has benefited greatly from the collective wisdom of the research community, drawing on best practices and recent advances in large-scale model training. To facilitate future research and practical deployment, we have released our innovation details in distributed training and quantization techniques, and open-sourced our development toolkits ERNIEKit and FastDeploy. By sharing these insights and tools, we aim to make a meaningful contribution to encourage innovation in efficient large-model training and deployment. In the future, we look forward to collaborating with the research and development community to further advance progress in this rapidly evolving field.

A Appendix

A.1 Ablation Study for Router Orthogonalization Loss

To verify the efficiency of router orthogonalization loss, we conduct ablation experiments on a MoE model with 28B total parameters and 3B activated parameters. Both baseline and experimental models are trained on same amout of tokens using identical data, hyperparameters, and exponential moving average (EMA) parameter accumulation. The coefficient for the orthogonalization loss is set to 1×10^{-2}, and unlike weight decay, it is not scaled by the learning rate. Table 9 presents the results of few-shot evaluation on downstream benchmarks. Although the training loss remains similar, MoE training with the router orthogonalization loss outperforms the baseline, indicating that constraining the router weights promotes better OOD performance.

Model	Mean	MMLU	CEval	CMMLU	MMCU	HumanEval	AGIEval	BBH	MBPP	GSM8K	Math
Baseline	50.2	46.1	55.4	51.7	66.8	50.6	47.1	37.7	65.6	**63.3**	**21.5**
+ Router Orthogonal Loss	**53.1**	**51.6**	**63.5**	**63.4**	**68.4**	**58.5**	**47.3**	**38.9**	**66.1**	60.7	21.3

Table 9: Performance comparison between the baseline and the model trained with Router Orthogonal Loss across multiple benchmarks.

A.2 EMA in Terms of Update Deltas

Let $\theta_t \in \mathbb{R}^d$ denote the model parameters at training step t and let $\delta_t = \theta_{t+1} - \theta_t$ represent the parameter update delta applied at step t. The EMA model parameters, denoted by θ_t^{EMA}, is defined recursively as:

$$\theta_t^{\text{EMA}} = \alpha \theta_{t-1}^{\text{EMA}} + (1-\alpha)\theta_t, \quad \text{with } \theta_0^{\text{EMA}} = \theta_0, \tag{14}$$

where $\alpha \in (0,1)$ is the EMA decay coefficient.

Let $\delta_t = \theta_{t+1} - \theta_t$ represent the parameter update vector applied at step t. Unfolding the recursion in equation 14 yields:

$$\begin{aligned}
\theta_1^{\text{EMA}} &= \alpha\theta_0 + (1-\alpha)\theta_1 && (15)\\
&= \theta_0 + (1-\alpha)\delta_0, && (16)\\
\theta_2^{\text{EMA}} &= \alpha\theta_1^{\text{EMA}} + (1-\alpha)\theta_2 \\
&= \theta_0 + (1-\alpha^2)\delta_0 + (1-\alpha)\delta_1, && (17)\\
\theta_n^{\text{EMA}} &= \theta_0 + \sum_{i=0}^{n-1}(1-\alpha^{n-i}))\delta_i. && (18)
\end{aligned}$$

A.3 Effective Decay Window of EMA

Since effective learning rate $\eta_i^{(\alpha)} = 1 - \alpha^{n-i}$ is monotonically decreasing with respect to i, we formally define the effective decay window size as:

$$W := n - j, \tag{19}$$

where $j \in \{0, \ldots, n\}$ is the *smallest* index such that:

$$\eta_j^{(\alpha)} < 1 - \epsilon \quad \text{and} \quad \eta_{j-1}^{(\alpha)} \geq 1 - \epsilon. \tag{20}$$

All updates δ_i with $i > j$ satisfy $\eta_i^{(\alpha)} < 1 - \epsilon$ and are therefore considered to lie within the effective decay window.

In practice, we may wish to select α to induce a desired effective decay window size, denoted by $\hat{W}$. We enforce the boundary condition at position $j = n - \hat{W}$ to construct the following equation:

$$\eta_j^{(\hat{\alpha})} = 1 - \hat{\alpha}^{\hat{W}} = 1 - \epsilon. \tag{21}$$

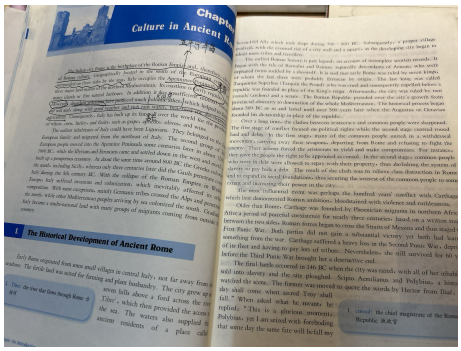
Solving for $\hat{\alpha}$ yields:

$$\hat{\alpha} = \exp\left(\frac{1}{\hat{W}} \log \epsilon\right). \tag{22}$$

B Qualitative examples

This section presents a selection of qualitative examples illustrating the capabilities of the ERNIE-4.5-VL model through responses generated from various prompts. Document understanding and OCR capabilities are highlighted in Figures 18, 19, and 20. Extending beyond static images, Figure 21 presents an example of generating video subtitles and locating specific events based on user instructions. Figures 22 highlight the model's ability to recognize ancient Chinese character. Figures 23, 24, 25 and 26 demonstrate the reasoning ability of the model in different disciplines. The model's proficiency in solving visual reasoning problems is presented in Figures 27, 28, 29, and 30. Examples demonstrating the model's common sense reasoning are provided in Figure 31. Figure 32 illustrates the model's ability for code synthesis. Figure 33 showcases an instance of image-conditioned creative writing generated by our model.

B.1 OCR Parsing and Document Understanding I



Prompts: Using the Roman history depicted in the image, create a mind map about the development of Rome. Requirements: in English, with conciseness and logical structure.

Model Responses: `<think>...</think>` Based on the provided text, here is a concise and logically structured mind map outlining the development of Ancient Rome:

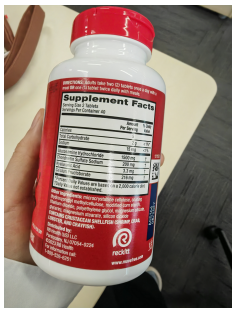
Central Theme: Development of Ancient Rome

1. **Geography & Foundation:**
   - Located in Italy (south of Europe, surrounded by seas).
   - Occupies Apennine Peninsula with a fairly regular coastline.
   - Rich volcanic soil, good weather, and rainwater benefited agriculture.
   - Early inhabitants: Potentially Ligurians; later influxes: Europeans (Apennines 1000 BC), Illyrians/Etruscans (1200 BC West/North), Greeks (800 BC South/Sicily), Gauls (5th Century BC North).
   - Became a multi-national land due to invasions, colonization, and migration.
2. **Historical Timeline:**
   - **Early Settlements (c. 1000 BC):** Originated from small villages in central Italy near the Tiber River.
   - **Etruscan Influence:** Ruled by seven kings (last three possibly Etruscan), including Tarquinius Superbus.
   - **Roman Republic (c. 509 BC):** Founded after expelling the last king. Governed by consuls (archons) and a senate.
   - **Expansion:** Grew from a city-state obscured by provinces to dominate the Mediterranean.
   - **Transition to Empire (End of Republic):** Augustus/Octavian founded dictatorship, replacing the Republic (over 500 years after its founding).
3. **Social Conflicts (Aristocracy vs. Common People):**
   - **Stage 1 (Political Rights):** Common people united in withdrawal, refusing to fight. Forced aristocrats to yield, gaining rights (e.g., appointment as consuls).
   - **Stage 2 (Debt & Land):** Common people in debt allowed to repay with property, abolishing debt slavery. Result: Relieved class distinction, expanded social foundations, secured common people's interest/power.
4. **Major Conflict: Punic Wars with Carthage:**
   - **Cause:** Demonstrated Roman ambition; conflict lasted 100 years, marked by violence and ruthlessness.
   - **Carthage:** Founded by Phoenician migrants in North Africa; coexisted peacefully for 3 centuries before conflict.
   - **First Punic War:** Roman forces crossed Straits of Messina. No decisive victory; Carthage suffered heavy losses in the Second Punic War.
   - **Final Battle (146 BC):** Rome razed Carthage, enslaved inhabitants, ploughed the site. Symbolized Rome's destructive end for Carthage.

This mind map captures the key geographical, chronological, social, and conflict-driven elements of Rome's development as described in the text.

Figure 18: This case demonstrates the document comprehension ability of ERNIE-4.5-VL.

B.2 OCR Parsing and Document Understanding II



Prompts: What is the purpose of this medication? How should I take it?
Model Responses: <think>...</think> Based on the supplement facts label, here's the information you need:
1. **Purpose:** This supplement is formulated for **joint health and support.**

- **Key Ingredients & Their Roles:**
  - **Glucosamine Hydrochloride & Chondroitin Sulfate Sodium:** These are the primary ingredients. They are building blocks of cartilage and are commonly used to support joint comfort, flexibility, and mobility, often targeted at osteoarthritis (especially knee osteoarthritis).
  - **Hyaluronic Acid:** Helps lubricate joints and may support joint cushioning.
  - **Calcium Fructoborate:** A form of boron that may support bone health and joint function.
- **Overall Goal:** The combination aims to support the structure and function of joints, potentially reducing discomfort and improving mobility.

2. **How to Take It (Directions):**

- **Adults take two (2) tablets once a day with a meal OR one (1) tablet twice daily with meals.**
- **Key Points:**
  - **Consistency:** Take it regularly as directed.
  - **With Food:** Always take it with a meal to help with absorption and minimize potential stomach upset.
  - **Dosage:** Follow the label instructions (2 tabs/day total, either all at once or split into two doses). Do not exceed the recommended dosage unless directed by a healthcare professional.
  - **Not for Children:** The label specifies "Adults," so it's not intended for children.

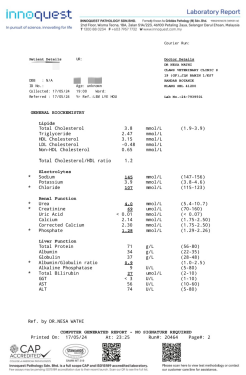
Important Considerations:

- **Not FDA-Approved:** Remember, this is a dietary supplement, not a drug. It is not intended to diagnose, treat, cure, or prevent any disease. Its effectiveness can vary between individuals.
- **Consult Your Doctor:**
  - **Before Starting:** Always talk to your doctor or a qualified healthcare provider before starting any new supplement, especially if you have underlying health conditions (like diabetes, shellfish allergies - see below), are pregnant or breastfeeding, or are taking other medications.
  - **Shellfish Allergy:** This product contains shellfish (shrimp, crab, lobster, crayfish). Do not take it if you have a shellfish allergy.
  - **Interactions:** Supplements can interact with medications (e.g., blood thinners like warfarin). Your doctor can advise on potential interactions.
  - **Underlying Conditions:** If you have severe joint pain or a diagnosed condition, your doctor can determine the best treatment plan, which may or may not include this supplement.
- **Sodium:** While low (<1% DV), if you are on a strict sodium-restricted diet, be aware of the sodium content (2g per serving).
- **Other Ingredients:** The label lists other components like microcrystalline cellulose, coatings, and fillers. If you have sensitivities to any of these, check the "Other Ingredients" list carefully.

In summary: This supplement is designed to support joint health. Take **2 tablets daily**, either all at once with a meal or split into two doses (1 tablet twice daily with meals). **Crucially, consult your doctor before starting it**, particularly if you have health conditions, take medications, or have a shellfish allergy.

Figure 19: Demonstration of ERNIE-4.5-VL's ability of accurately extracting texts from an image, understanding the user's query, and retrieving the answer from the document. Source: photo taken by an author from the Baidu ERNIE team.

B.3 Multilingual OCR Parsing



Prompts: Extract the medical test guidelines and create an Excel sheet (including the items and reference values with complete condition ranges and units).
Model Responses: `<think>...</think>` Here is the extracted medical testing data structured into an Excel-style table:

Item	Test Result	Reference Range	Unit
Lipids			
Total Cholesterol	3.8	1.9–3.9	mmol/L
Triglyceride	2.47	—	mmol/L
HDL Cholesterol	3.15	—	mmol/L
LDL Cholesterol	-0.48	—	mmol/L
Non-HDL Cholesterol	0.65	—	mmol/L
Total Cholesterol/HDL ratio	1.2	—	(Ratio)
Electrolytes			
Sodium	145	147–156	mmol/L
Potassium	3.9	3.8–4.6	mmol/L
Chloride	107	115–123	mmol/L
Renal Function			
Urea	4.0	5.4–10.7	mmol/L
Creatinine	69	70–160	μmol/L
Uric Acid	<0.01	<0.07	mmol/L
Calcium	2.14	1.75–2.50	mmol/L
Corrected Calcium	2.30	1.75–2.50	mmol/L
Phosphate	1.28	1.29–2.26	mmol/L
Liver Function			
Total Protein	71	56–80	g/L
Albumin	34	22–35	g/L
Globulin	37	28–48	g/L
Albumin/Globulin ratio	0.9	1.0–2.5	(Ratio)
Alkaline Phosphatase	9	5–80	U/L
Total Bilirubin	27	2–10	μmol/L
GGT	<3	1–10	U/L
AST	56	10–60	U/L
ALT	74	5–80	U/L

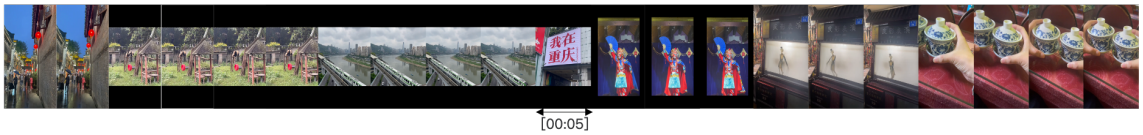
Notes:

- ”—” indicates no reference range provided in the report.
- **Units** are preserved as listed (e.g., μmol/L, mmol/L, g/L).
- **Inequalities** (e.g., ¡0.01) are retained from the original report.
- **Ratios** (e.g., Total Cholesterol/HDL) lack units but are labeled as ”Ratio” for clarity.

Figure 20: This example demonstrates the OCR parsing and instruction following capability of ERNIE-4.5-VL.

B.4 Video Temporal Grounding

Input video



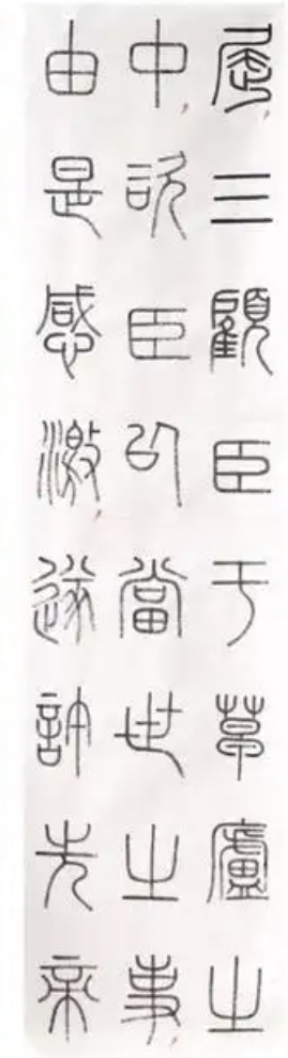
Prompts: Based on the video, could you help me create some engaging subtitles to put below it?
Model Responses:

- Exploring the vibrant streets of Chongqing at dusk, where tradition meets modernity.
- A glimpse of the adorable pandas in their natural habitat, a true symbol of China.
- The sleek monorail glides above the river, offering stunning views of the cityscape.
- "I am in Chongqing" - A bold declaration on a city building, embracing the local spirit.
- The ancient art of shadow play comes to life, a captivating cultural experience.
- Savoring the elegance of traditional Chinese tea sets, a moment of tranquility.

Prompts: At which second does "CHONGQING" appear in the video?
Model Responses: At 5 seconds, the word 'CHONGQING' appears in red on a large white screen.

Figure 21: The first query tests the model's skill in breaking down a video into its important events and describing them concisely (like creating a detailed caption for different parts of a video). The second query checks the model's capability to pinpoint and identify specific events within a video (like finding a particular scene in a video). Source: video taken by an author from the Baidu ERNIE team.

B.5 OCR Ancient Chinese Character Recognition



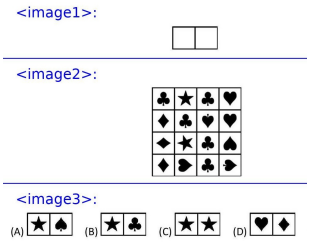
Prompts: 请问这张图片中的文字是什么？出自哪里？
Model Responses: <think>...</think> 图片中的文字为篆书字体，内容是诸葛亮《前出师表》中的经典段落：
「（先帝）三顾臣于草庐之中，咨臣以当世之事，由是感激，遂许先帝以驱驰。」
解析：
1. 文字识别：篆书逐字转写为简体字后，内容为：「（臣）三顾臣于草庐之中，咨臣以当世之事，由是感激，遂许先帝」（图片文字略有省略，但核心句意完整）。
2. 出处：出自三国时期诸葛亮《前出师表》。此段描述刘备（先帝）三次到诸葛亮隐居的草庐拜访，询问天下大事，诸葛亮感其诚意，决心出山辅佐。
3. 书法特点：篆书以线条圆转、结构对称为特征，此作体现了古朴典雅的篆书风格，符合汉代以后篆书用于铭刻、庄重文本的传统。

背景意义：
《出师表》是诸葛亮北伐前写给后主刘禅的奏章，既表达忠君报国之志，又蕴含对先帝刘备的深切追念。此段文字因体现君臣知遇之恩，成为千古名句。

Translated prompt: What is the text in this picture? Where did it come from?
Translated model response: <think>...</think> The characters in the image are written in Seal Script, featuring a classic passage from Zhuge Liang's "Former Memorial to the Emperor on Sending Troops": "The Late Emperor thrice honored me with his visit to my humble thatched cottage, consulting me on matters governing our tumultuous age. Moved by such magnanimous courtesy and profound trust, I thus pledged my unwavering service to His Majesty"s cause."
Analysis:
1. Character Recognition: After transcribing the Seal Script characters into Simplified Chinese, the content reads:
"The Late Emperor thrice honored me with his visit to my humble thatched cottage, consulting me on matters governing our tumultuous age. Moved by such magnanimous courtesy and profound trust, I thus pledged my unwavering service to His Majesty"s cause." (Some wording is omitted in the image, but the core meaning remains intact.)
2. Source: This passage is from Zhuge Liang's "Former Memorial to the Emperor on Sending Troops" ("Qian Chu Shi Biao") of the Three Kingdoms period. It describes how Liu Bei (the late emperor) visited Zhuge Liang three times in his humble abode to seek his counsel on state affairs. Touched by Liu Bei's sincerity, Zhuge Liang resolved to leave seclusion and assist him.
3. Calligraphic Features: Seal Script is characterized by its rounded strokes and symmetrical structure. This work exemplifies the ancient and elegant style of Seal Script, in keeping with the tradition of using this script for inscriptions and formal texts after the Han Dynasty.
Background Significance:
"Former Memorial to the Emperor on Sending Troops" is a memorial written by Zhuge Liang to Emperor Liu Shan before his northern expedition. It not only expresses Zhuge Liang's loyalty and dedication to his country but also conveys his deep respect and remembrance for the late Emperor Liu Bei. This passage, in particular, has become a timeless quote for its portrayal of the profound bond between sovereign and minister.

Figure 22: This example highlight the model's ability to recognize ancient Chinese character. Our model is able to accurately recognize Small Seal Script and understand the historical allusions behind the characters. We omit thinking tokens for brevity.

B.6 Reasoning Cases: Deductive Visual Puzzle



Prompts: Karina cuts out a piece of form in from the diagram in. Which one of the following pieces can she cut out in ? Solve the multiple-choice question and then answer with the option letter from the given choices. The last line of your response should be of the following format: Final Answer: $LETTER' (without quotes) where LETTER is one of options. Think step by step before answering. IMPORTANT: Remember, to end your answer with Final Answer: .
Model Responses: `<think>`
I need to figure out which piece Karina can cut out from the diagram in image2 using the shape in image1. Let me start by understanding the shapes involved.

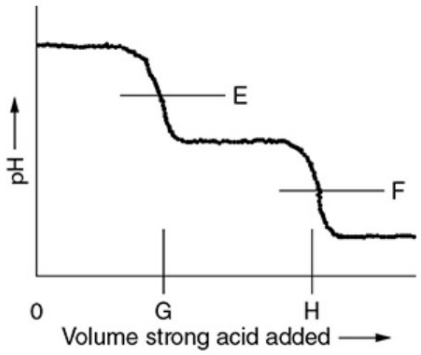
First, image1 shows a simple rectangle divided into two equal squares side by side. So, the shape is a 1x2 rectangle. That means whatever piece Karina cuts out from image2 must be a 1x2 rectangle, right? But wait, image2 is a 4x4 grid of different symbols: clubs, stars, diamonds, and hearts. Each symbol is in a square, so the grid is 4 rows by 4 columns.
Karina is cutting out a piece that's two squares wide and one square tall, or maybe two squares tall and one square wide? Wait, image1 is two squares adjacent horizontally. So the shape is two squares in a row. So the piece she cuts out from image2 must be two adjacent squares in a horizontal line. But looking at image3, the options are all pairs of symbols. Let me check the options:
...
Wait, maybe I'm miscounting. Wait, each row has four symbols. Let me list them properly.
Row 1: ♣, ★, ♣, ♥
Row 2: ♦, ♣, ♥, ♥
Row 3: ♦, ★, ♣, ♥
Row 4: ♦, ♥, ♣, ♥
...
Therefore, the correct answer is option B.
`</think>`
Karina needs to cut out a 1×2 horizontal piece from the 4×4 grid shown in **image2** using the shape in **image1**. The valid symbol pairs must match one of the options in **image3**.

- **Option A (★ and ♥)**: Spades (♥) do not appear in **image2**, so this option is invalid.
- **Option B (★ and ♣)**: In row 1, columns 2 (★) and 3 (♣) form a valid horizontal pair. Similarly, row 3, columns 2 (★) and 3 (♣) also form a valid match. This matches Option B.
- **Option C (★ and ★)**: There are no two adjacent stars (★) in the grid, so this option is invalid.
- **Option D (♠ and ♦)**: The only heart-diamond pair appears in row 4, columns 1 (♦) and 2 (♠), which is the reverse order (♦ then ♠), not matching the required order ♠ followed by ♦ in Option D.

Final Answer: Option B

Figure 23: An example of ERNIE-4.5-VL that briefly demonstrates the reflective reasoning ability through solving a deductive visual puzzle.

B.7 Reasoning Cases: Chemistry



Prompts: The diagram above represents the idealized titration curve for the reaction of sodium oxalate ($Na_2C_2O_4$) with a strong acid like hydrochloric acid (HCl). E and F represent the pH at the endpoints. G and H will depend on the composition of the sample with the possibility that one may not be present. In addition to water, what are the predominant species in solution at E?
Options:
(A) $Na_2C_2O_4$ and HCl
(B) Na^+, Cl^-, and $HC_2O_4^-$
(C) $C_2O_4^{2-}$ and H^+
(D) Na^+, H^+, and $C_2O_4^{2-}$

Model Responses: `<think>...</think>` At point E on the titration curve, the first equivalence point is reached. Sodium oxalate ($Na_2C_2O_4$) dissociates completely into $2Na^+$ and $C_2O_4^{2-}$. During titration with HCl, H^+ ions react with $C_2O_4^{2-}$ to form $HC_2O_4^-$. At the first equivalence point, all $C_2O_4^{2-}$ is converted to $HC_2O_4^-$. The solution also contains Na^+ (from the original salt) and Cl^- (from HCl).

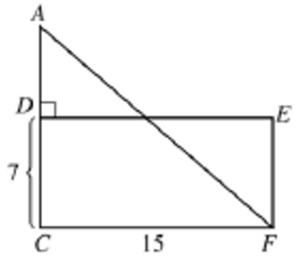
Analysis of options:

- **(A)** Incorrect. $Na_2C_2O_4$ and HCl are not present at equivalence; the reaction is complete, and HCl is neutralized.
- **(B)** Correct. Na^+ (spectator ion), Cl^- (from HCl), and $HC_2O_4^-$ (product of the reaction) are the predominant species.
- **(C)** Incorrect. $C_2O_4^{2-}$ is fully converted to $HC_2O_4^-$, and H^+ is not a dominant species at this stage.
- **(D)** Incorrect. While Na^+ is present, H^+ and $C_2O_4^{2-}$ are not predominant; $HC_2O_4^-$ dominates instead.

Answer: (B) (Na^+, Cl^-, and $HC_2O_4^-$)

Figure 24: An example of ERNIE-4.5-VL solving a chemistry problem, which highlights the model's logical reasoning ability. We omit thinking tokens for brevity.

B.8 Reasoning Cases: Math



Prompts: In the figure above if the area of triangle CAF is equal to the area of rectangle CDEF, what is the length of segment AD?
Choices:
A. 7/2
B. 5
C. 7
D. 15/2
E. 15

Model Responses: `<think>...</think>` The rectangle CDEF has CD = 7. The area of rectangle CDEF is given by the product of its adjacent sides, CD and CF (where CF is the same as FC, a side of the rectangle). Let the length of CF be l. Thus, the area of rectangle CDEF is

$$7 \times l = 7l$$

Triangle CAF is a right triangle with the right angle at C. The legs are CA and CF. Since D is on CA and CD = 7, and assuming A is beyond D so CA = CD + AD, let AD = d. Then CA = $7 + d$. The area of triangle CAF is

$$\frac{1}{2} \times CA \times CF = \frac{1}{2} \times (7 + d) \times l$$

Given that the area of triangle CAF equals the area of rectangle CDEF:

$$\frac{1}{2}(7 + d)l = 7l$$

Assuming $l \neq 0$ (since CF is a side of the rectangle and must have positive length), divide both sides by l:

$$\frac{1}{2}(7 + d) = 7$$

Multiply both sides by 2:

$$7 + d = 14$$

Solve for d:

$$d = 7$$

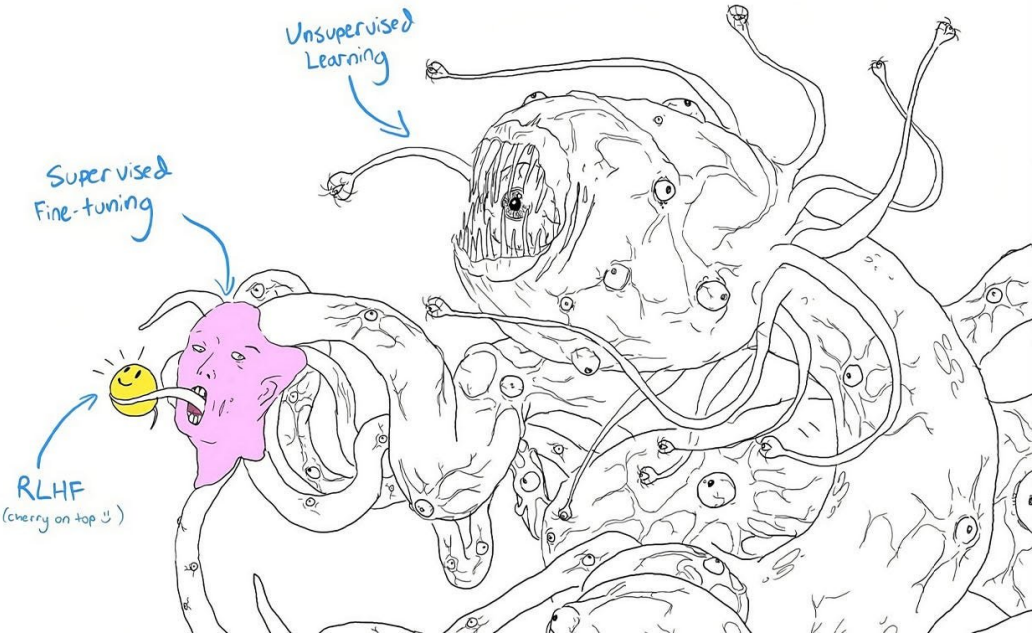
Thus, the length of segment AD is 7.

This result is independent of the length of CF, as l cancels out in the equation.

Answer: C

Figure 25: An example of solving a mathematical analytical geometry problem, demonstrating logical reasoning ability and mathematical computation skills in ERNIE-4.5-VL. We omit thinking tokens for brevity.

B.9 Reasoning Cases: Deep Semantic Image Understanding



Prompts: What is this figure trying to convey?
Model Responses: `<think>...</think>` This image is a satirical, metaphorical illustration of machine learning concepts, using grotesque and humorous visuals to convey key ideas:

- **Unsupervised Learning** (large creature):
  - Represented as a chaotic, multi-eyed monster with tentacles.
  - Symbolizes the unstructured, exploratory nature of unsupervised learning—finding patterns in unlabeled data without predefined rules.
  - The creature's complexity reflects how unsupervised methods (e.g., clustering, generative models) can be messy but powerful.
- **Supervised Fine-tuning** (pink humanoid):
  - Depicted as a smaller, vulnerable figure being manipulated by the monster.
  - Represents the process of refining a pre-trained model (e.g., a large language model) using labeled data for specific tasks.
  - The humanoid's subservience highlights how fine-tuning builds upon unsupervised foundations but is more controlled.
- **RLHF (Reinforcement Learning from Human Feedback)** (tennis ball):
  - Labeled as the "cherry on top" with a smiley face.
  - Symbolizes RLHF as a "reward" or refinement step where human feedback aligns model outputs with human preferences (e.g., making responses helpful/harmless).
  - The tennis ball (a treat) suggests RLHF adds a layer of polish or user-friendliness.

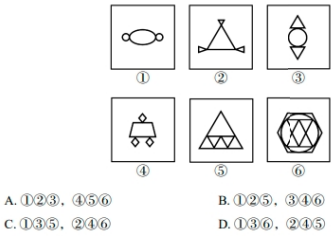
Overall Message:

- The image humorously critiques the machine learning pipeline:
  - **Unsupervised learning** is the chaotic "base" that generates raw capability.
  - **Supervised fine-tuning** refines this base for practical use.
  - **RLHF** is the final touch that makes outputs more aligned with human values.
- The grotesque style satirizes the complexity and potential "monstrosity" of AI systems, while the smiley face on the tennis ball adds irony—RLHF is portrayed as a superficial "fix" for deeper issues.

This metaphor emphasizes how modern AI combines unstructured exploration (unsupervised), task-specific refinement (supervised), and human-centric adjustments (RLHF) to create functional systems.

Figure 26: This showcases the advanced image understanding capabilities of the ERNIE-4.5-VL. We omit thinking tokens for brevity.

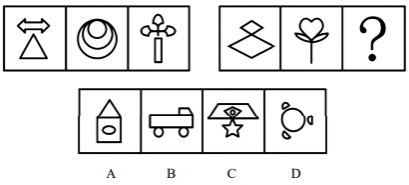
B.10 Visual Reasoning: Visual Pattern Recognition



Prompts: Choose the most appropriate option from the four given choices to fill in the question mark, so that the figures follow a pattern.
Model Responses: `<think>...</think>` The figures can be categorized based on their structural composition:

- **Category 1 (Figures 1, 3, 6):** These figures consist of **distinct elements connected or overlaid** (e.g., separate circles linked by a line, a triangle with a superimposed circle, or a hexagon with intersecting internal lines).
- **Category 2 (Figures 2, 4, 5):** These figures feature **integrated structures** (e.g., a triangle subdivided by lines, a diamond with attached smaller diamonds, or a triangle partitioned into smaller triangles).

Answer: D



Prompts: Choose the most appropriate option from the four given choices to fill in the question mark, so that the figures follow a pattern.
Model Responses: The sequence of figures follows a pattern based on symmetry, complexity, and the introduction of new elements. Here's the analysis:

- **Symmetry Progression:** The figures exhibit varying symmetries—vertical (triangle), rotational (circle with spiral), vertical/horizontal (cross), diagonal (overlapping parallelograms), and radial (flower). The next figure should align with this progression. Option C (star in a parallelogram) introduces diagonal symmetry, complementing the earlier parallelograms, while D (circle with a dot) maintains radial symmetry. However, the sequence does not strictly repeat symmetry types, favoring structural evolution instead.
- **Complexity and Layering:** The figures grow more complex, from simple shapes (triangle, circle) to combined forms (cross with circles, overlapping parallelograms) and organic forms (flower). Option C adds a star (a new element) within a parallelogram, reflecting this layering trend.
- **Element Introduction:** Each figure introduces a new component (e.g., spiral, cross arms, overlapping shapes, petals). Option C continues this by combining a star (new) with a parallelogram (previously seen).

Answer: C

Figure 27: Two examples of ERNIE-4.5-VL using its visual inductive reasoning ability to correctly find pattern and predict the missing patterns.

B.11 Visual Reasoning: Emoji Quiz

Prompts: 你能根据图中的emoji猜出著名电影吗?
Model Responses: <think>...</think> 根据图中的emoji，对应的著名电影是《飞屋环游记》（Up）。以下是解析过程：
1. 飞机：象征“飞”，对应电影中的飞行元素。
2. 房子：代表“屋”，即电影中飞行的房子。
3. 红色圆圈：象征“环”（如“环游”中的“环”）。
4. 游泳的人：可能隐含“游”（环游的“游”）。
5. 公鸡：谐音“记”（鸡→记），呼应电影名的结尾字。
组合后即为《飞屋环游记》（英文原名Up），完美契合emoji的隐喻与谐音逻辑。

Translated prompt: Can you guess the famous movie based on the emojis in the picture?
Translated model response: <think>...</think> The famous movie corresponding to the emojis in the picture is 'Up'. Here is the analysis process:
1. Airplane: symbolizes 'fly', corresponding to the flying element in the movie.
2. House: represents 'house', referring to the flying house in the movie.
3. Red Circle: symbolizes 'circle' (as in the 'circle' in 'circumnavigate').
4. Swimming Person: might imply 'travel' (as in the 'travel' in 'circumnavigate').
5. Rooster: phonetically similar to 'remember' (chicken → remember), echoing the last character of the movie title.
When combined, it forms "Up" (the original English title), perfectly fitting the metaphor and phonetic logic of the emoji.

Prompts: 图中是一个什么梗?
Model Responses: <think>...</think> 图中是一个结合视觉与谐音的梗。上半部分正常标注“这是马”，而下半部分将马周围布满数字“2”，利用以下逻辑制造笑点：
1. 视觉混淆：数字“2”密集排列形成类似“马赛克”或“二维码”的视觉效果。
2. 谐音梗：中文“马”与“码”同音，数字“2”在中文网络语境中常代指“二”。当“马”被大量“2”包围时，可解读为“码（马）的二维码”（即“二+马”的重复组合）。
3. 最终答案：填空处应为“二维码”，既呼应数字排列的视觉效果，又通过“码=马”的谐音完成双关。
该梗通过简单的元素重组与语言游戏，制造出“意料之外又情理之中”的幽默效果。

Translated prompt: What kind of joke is it in the picture?
Translated model response: <think>...</think> The image is a joke that combines visual and phonetic elements. The upper part is normally annotated as "this is a horse", while the lower part fills the area around the horse with the number "2", creating humor through the following logic:
1. Visual confusion: The densely arranged number "2" forms a visual effect similar to a "mosaic" or a "QR code".
2. Phonetic joke: In Chinese, "horse" sounds the same as "code", and the number "2" often represents "two" in the Chinese internet context. When the "horse" is surrounded by a large number of "2", it can be interpreted as "the QR code of the horse" (i.e., a repeated combination of "two" and "horse").
3. Final answer: The blank should be filled with "QR code", which not only echoes the visual effect of the number arrangement, but also completes the pun through the homophonic "code=horse".
This joke, through simple element recombination and language play, creates a humorous effect that is 'unexpected yet makes perfect sense'.

Figure 28: These two examples successfully demonstrate how the ERNIE-4.5-VL solves problems based on the given emoji inputs. These cases highlight the model's ability to recognize emoji icons, understand movie plot contents, and analyze internet meme images.

B.12 Visual Reasoning: Depth Sorting

Prompts: 请列出图中除了小熊以外的所有物品，并且以与小熊的距离为依据来升序排序。
Model Responses: `<think>...</think>` 根据与小熊的距离由近到远排序如下：
1. 黄色小鸭玩具（LINLEE）：紧邻小熊右下方，距离最近。
2. 海苔圈零食（蓝色包装）：位于小熊正前方偏左，稍远于小鸭。
3. 维达抽纸（蓝色包装）：在零食左侧，距离小熊略远于零食本身。
4. 雀巢茶萃（绿色饮料盒）与太魔性酸梅汤（紫色饮料盒）：叠放于小熊左后方，为最远物品（两者距离相近，并列最后）。
理由：小鸭与小熊直接接触，最近；零食和纸巾位于前方但分散左右；饮料盒因处于后方且叠放，距离最远。

Translated prompt: Please list all the items in the picture except the little bear and sort them in ascending order based on their distance from the little bear.
Translated model response: `<think>...</think>` The order from the closest to the farthest distance from the little bear is as follows: 1. Yellow duck toy (LINLEE): It is right next to the lower right of the little bear, the closest in distance.
2. Seaweed ring snack (blue packaging): It is located just in front and slightly to the left of the little bear, a bit farther than the duck.
3. Vinda tissue paper (blue packaging): It is on the left side of the snack, slightly farther from the bear than the snack itself.
4. Nestle tea extract (green beverage box) and Taimo Plum Soup (purple beverage box): They are stacked at the back left of the bear and are the farthest items (both are at similar distances, ranked last).
Reason: The duck is in direct contact with the bear and is the closest; the snack and tissue are in front but scattered left and right; the beverage boxes are the farthest because they are at the back and stacked.

Figure 29: An illustration of the model's capability to deduce the spatial arrangement of objects in terms of depth from a solitary photograph. Source: photo taken by an author from the Baidu ERNIE team.

B.13 Visual Reasoning: Counting

Prompts: Can you tell me how many different colors of pens are in the picture?

Model Responses: `<think>...</think>` There are **5 different colors** of pens in the picture:

1. Red (two pens, but same color)
2. Blue
3. Green
4. Pink
5. Brown

These are all visible in the black mesh pen holder.

Figure 30: An example of ERNIE-4.5-VL recognizing queried objects and counting their occurrences. Source: photo taken by an author from the Baidu ERNIE team.

B.14 Common Sense Reasoning

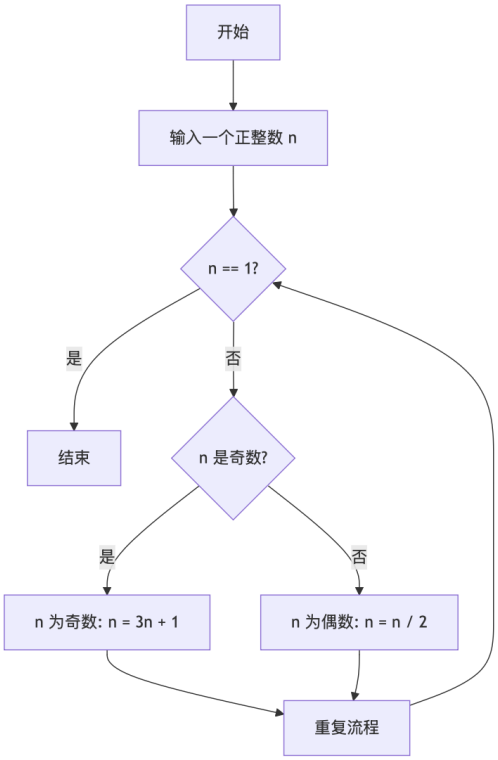
Prompts: Where were the cultural relics in the picture unearthed?
Model Responses: The cultural relic shown in the picture is a bronze mask unearthed from Sanxingdui. The Sanxingdui site is located in Guanghan City, Sichuan Province, and is one of the important archaeological discoveries in China in the 20th century. It provides important physical materials for studying the ancient history, culture, and society of the Shu (Sichuan) region.

Prompts: What is his occupation?
Model Responses: The character in the image is from the anime "Detective Conan" (Case Closed). He is Kogoro Mori, who is a private detective. Despite his sometimes comical and lazy demeanor, he runs a detective agency and often finds himself involved in solving cases, sometimes with the help of Conan Edogawa.

Prompts: What's in the picture?
Model Responses: The picture shows a small, yellow LEGO or building block figure resembling Pikachu, a popular Pokémon character. The figure has black-tipped ears, black eyes, a black nose, and red cheeks. It is positioned on a white surface next to a white electronic device with black buttons and a black cable.

Figure 31: Three illustrative instances of ERNIE-4.5-VL successfully identifying famous cultural relics and famous celebrities from their corresponding inputs. These examples highlight the model's common sense ability.

B.15 Code Synthesis



Prompts: In python, using the process as shown in the figure, how many steps are needed to calculate 10,000,000 before it becomes 1?

Model Responses: To determine how many steps are needed to reduce 10,000,000 to 1 using the process described in the flowchart (Collatz conjecture), we can write a Python function to simulate the process. Here's how you can implement it:

```
def collatz_steps(n):
    steps = 0
    while n != 1:
        if n % 2 == 0:
            n = n // 2
        else:
            n = 3 * n + 1
        steps += 1
    return steps

# Calculate the number of steps for 10,000,000
n = 10000000
steps_needed = collatz_steps(n)
print(f"Number of steps to reduce {n} to 1: {steps_needed}")
```

Running this code will give you the number of steps required to reduce 10,000,000 to 1. The Collatz conjecture is the basis for this process, and while it's proven for many numbers, it hasn't been proven for all integers. However, for 10,000,000, the process will eventually reach 1.

When you run the code, it will output the number of steps needed. The exact number might be large, as the process can take many iterations for large numbers.

Figure 32: This example validates ERNIE-4.5-VL's capability to convert flowcharts into Python code based on user commands, showcasing its strength in both adherence to instructions and code generation.

B.16 Image Conditioned Creative Writing

Prompts: Besides the ingredients in the fridge, there are also flour, rice and other materials at home. Please help me design a lunch for two people based on the ingredients. It requires three dishes and one soup. Please list it in the form of a table.

Model Responses: I see a well-stocked fridge with various ingredients. Let me design a balanced lunch menu for two people with three dishes and one soup, utilizing what's available. Here's a table with the menu:

Category	Dish Name	Main Ingredients Used	Description
Main Dish 1	Grilled Mackerel	Mackerel fillets	Simple seasoned grilled mackerel - a healthy protein source
Main Dish 2	Stir-fried Chicken with Vegetables	Chicken slices, cabbage, green peppers, other available veggies	Quick stir-fry with soy sauce seasoning
Side Dish	Steamed Egg Custard	Eggs, with optional dashi or soy milk	Silky smooth egg custard, can add some green onions on top
Soup	Miso Soup with Tofu and Wakame	Tofu, miso paste, wakame seaweed (if available)	Classic Japanese soup to complete the meal

For the stir-fry, I'd use the chicken slices along with whatever vegetables look freshest - probably cabbage, green peppers, and maybe some broccoli if available. The egg custard is simple but comforting, and miso soup ties everything together nicely.

This menu provides a good balance of protein (fish, chicken, tofu, eggs), vegetables, and carbohydrates (from the rice that's mentioned as being available at home). The flavors are complementary and it's all quite easy to prepare.

Figure 33: An example of image-conditioned creative text generation. The model processes an input image and generates a menu semantically related to the visual content.

References

Amro Abbas, Kushal Tirumala, Dániel Simig, Surya Ganguli, and Ari S. Morcos. Semdedup: Data-efficient learning at web-scale through semantic deduplication, 2023. URL https://arxiv.org/abs/2303.09540.

AIME. Aime problems and solutions, 2025. URL https://artofproblemsolving.com/wiki/index.php/AIME_Problems_and_Solutions.

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob Menick, Sebastian Borgeaud, Andrew Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikolaj Binkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karen Simonyan. Flamingo: a visual language model for few-shot learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, pp. 23716–23736, December 2022. URL https://arxiv.org/abs/2204.14198.

Anthropic. Introducing claude 4, 2025. URL https://www.anthropic.com/news/claude-4.

Saleh Ashkboos, Amirkeivan Mohtashami, Maximilian L Croci, Bo Li, Pashmina Cameron, Martin Jaggi, Dan Alistarh, Torsten Hoefler, and James Hensman. Quarot: Outlier-free 4-bit inference in rotated llms. *Advances in Neural Information Processing Systems*, 37:100213–100240, 2024.

Anas Awadalla, Le Xue, Oscar Lo, Manli Shu, Hannah Lee, Etash Kumar Guha, Matt Jordan, Sheng Shen, Mohamed Awadalla, Silvio Savarese, Caiming Xiong, Ran Xu, Yejin Choi, and Ludwig Schmidt. Mint-1t: Scaling open-source multimodal data by 10x: A multimodal dataset with one trillion tokens, 2024. URL https://arxiv.org/abs/2406.11271.

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.

James Betker, Gabriel Goh, Li Jing, † TimBrooks, Jianfeng Wang, Linjie Li, † LongOuyang, † JuntangZhuang, † JoyceLee, † YufeiGuo, † WesamManassra, † PrafullaDhariwal, † CaseyChu, † YunxinJiao, and Aditya Ramesh. Improving image generation with better captions. URL https://api.semanticscholar.org/CorpusID:264403242.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. PIQA: reasoning about physical commonsense in natural language. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pp. 7432–7439. AAAI Press, 2020. doi: 10.1609/AAAI.V34I05.6239. URL https://doi.org/10.1609/aaai.v34i05.6239.

Federico Cassano, John Gouwar, Daniel Nguyen, Sydney Nguyen, Luna Phipps-Costin, Donald Pinckney, Ming-Ho Yee, Yangtian Zi, Carolyn Jane Anderson, Molly Q. Feldman, Arjun Guha, Michael Greenberg, and Abhinav Jangda. Multipl-e: A scalable and polyglot approach to benchmarking neural code generation. *IEEE Trans. Software Eng.*, 49(7):3675–3691, 2023. doi: 10.1109/TSE.2023.3267446. URL https://doi.org/10.1109/TSE.2023.3267446.

Yaoyao Chang, Lei Cui, Li Dong, Shaohan Huang, Yangyu Huang, Yupan Huang, Scarlett Li, Tengchao Lv, Shuming Ma, Qinzheng Sun, Wenhui Wang, Furu Wei, Ying Xin, Mao Yang, Qiufeng Yin, and Xingxing Zhang. Redstone: Curating general, code, math, and QA data for large language models. *CoRR*, abs/2412.03398, 2024. doi: 10.48550/ARXIV.2412.03398. URL https://doi.org/10.48550/arXiv.2412.03398.

Chunyang Chen, Ting Su, Guozhu Meng, Zhenchang Xing, and Yang Liu. From ui design image to gui skeleton: a neural machine translator to bootstrap mobile gui implementation. In *Proceedings of the 40th International Conference on Software Engineering*, pp. 665–676, 2018.

Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Jiaqi Wang, Yu Qiao, Dahua Lin, and Feng Zhao. Are we on the right way for evaluating large vision-language models? In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang (eds.), *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/2f8ee6a3d766b426d2618e555b5aeb39-Abstract-Conference.html.

Yao Cheng, Jianfeng Chen, Jie Chen, Li Chen, Liyu Chen, Wentao Chen, Zhengyu Chen, Shijie Geng, Aoyan Li, Bo Li, Bowen Li, Linyi Li, Boyi Liu, Jerry Liu, Kaibo Liu, Qi Liu, Shukai Liu, Siyao Liu, Tianyi Liu, Tingkai Liu, Yongfei Liu, Rui Long, Jing Mai, Guanghan Ning, Z. Y. Peng, Kai Shen, Jiahao Su, Jing Su, Tao Sun, Yifan Sun, Yunzhe Tao, Guoyin Wang, Siwei Wang, Xuwu Wang, Yite Wang, Zihan Wang, Jinxiang Xia, Liang Xiang, Xia Xiao, Yongsheng Xiao, Chenguang Xi, Shulin Xin, Jingjing Xu, Shikun Xu, Hongxia Yang, Jack Yang, Yingxiang Yang, Jianbo Yuan, Jun Zhang, Yufeng Zhang, Yuyu Zhang, Shen Zheng, He Zhu, and Ming Zhu. Fullstack bench: Evaluating llms as full stack coders. *CoRR*, abs/2412.00535, 2024. doi: 10.48550/ARXIV.2412.00535. URL https://doi.org/10.48550/arXiv.2412.00535.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the AI2 reasoning challenge. *CoRR*, abs/1803.05457, 2018. URL http://arxiv.org/abs/1803.05457.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *CoRR*, abs/2110.14168, 2021. URL https://arxiv.org/abs/2110.14168.

Google DeepMind. Gemini 2.5, 2025. URL https://blog.google/technology/google-deepmind/gemini-model-thinking-updates-march-2025/.

DeepSeek-AI. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025. URL https://arxiv.org/abs/2501.12948.

DeepSeek-AI, Aixin Liu, Bei Feng, Bin Wang, Bingxuan Wang, Bo Liu, Chenggang Zhao, Chengqi Deng, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, Hao Zhang, Hanwei Xu, Hao Yang, Haowei Zhang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Li, Hui Qu, J. L. Cai, Jian Liang, Jianzhong Guo, Jiaqi Ni, Jiashi Li, Jin Chen, Jingyang Yuan, Junjie Qiu, Junxiao Song, Kai Dong, Kaige Gao, Kang Guan, Lean Wang, Lecong Zhang, Lei Xu, Leyi Xia, Liang Zhao, Liyue Zhang, Meng Li, Miaojun Wang, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingming Li, Ning Tian, Panpan Huang, Peiyi Wang, Peng Zhang, Qihao Zhu, Qinyu Chen, Qiushi Du, R. J. Chen, R. L. Jin, Ruiqi Ge, Ruizhe Pan, Runxin Xu, Ruyi Chen, S. S. Li, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shaoqing Wu, Shengfeng Ye, Shirong Ma, Shiyu Wang, Shuang Zhou, Shuiping Yu, Shunfeng Zhou, Size Zheng, Tao Wang, Tian Pei, Tian Yuan, Tianyu Sun, W. L. Xiao, Wangding Zeng, Wei An, Wen Liu, Wenfeng Liang, Wenjun Gao, Wentao Zhang, X. Q. Li, Xiangyue Jin, Xianzu Wang, Xiao Bi, Xiaodong Liu, Xiaohan Wang, Xiaojin Shen, Xiaokang Chen, Xiaosha Chen, Xiaotao Nie, Xiaowen Sun, Zihan Wang, and et al. Deepseek-v2: A strong, economical, and efficient mixture-of-experts language model. *CoRR*, abs/2405.04434, 2024a. doi: 10.48550/ARXIV.2405.04434. URL https://doi.org/10.48550/arXiv.2405.04434.

DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Haowei Zhang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Li, Hui Qu, J. L. Cai, Jian Liang, Jianzhong Guo, Jiaqi Ni, Jiashi Li, Jiawei Wang, Jin Chen, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, Junxiao Song, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Lei Xu, Leyi Xia, Liang Zhao, Litong Wang, Liyue Zhang, Meng Li, Miaojun Wang, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingming Li, Ning Tian, Panpan Huang, Peiyi Wang, Peng Zhang, Qiancheng Wang, Qihao Zhu, Qinyu Chen, Qiushi Du, R. J. Chen, R. L. Jin, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, Runxin Xu, Ruoyu Zhang, Ruyi Chen, S. S. Li, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shaoqing Wu, Shengfeng Ye, Shengfeng Ye, Shirong Ma, Shiyu Wang, Shuang Zhou, Shuiping Yu, Shunfeng Zhou, Shuting Pan, T. Wang, Tao Yun, Tian Pei, Tianyu Sun, W. L. Xiao, and Wangding Zeng. Deepseek-v3 technical report. *CoRR*, abs/2412.19437, 2024b. doi: 10.48550/ARXIV.2412.19437. URL https://doi.org/10.48550/arXiv.2412.19437.

Mostafa Dehghani, Basil Mustafa, Josip Djolonga, Jonathan Heek, Matthias Minderer, Mathilde Caron, Andreas Steiner, Joan Puigcerver, Robert Geirhos, Ibrahim M. Alabdulmohsin, Avital Oliver, Piotr Padlewski, Alexey A. Gritsenko, Mario Lucic, and Neil Houlsby. Patch n' pack: Navit, a vision transformer for any aspect ratio and resolution. 2023a. URL http://papers.nips.cc/paper_files/paper/2023/hash/06ea400b9b7cfce6428ec27a371632eb-Abstract-Conference.html.

Mostafa Dehghani, Basil Mustafa, Josip Djolonga, Jonathan Heek, Matthias Minderer, Mathilde Caron, Andreas Steiner, Joan Puigcerver, Robert Geirhos, Ibrahim M. Alabdulmohsin, Avital Oliver, Piotr Padlewski, Alexey A. Gritsenko, Mario Lucic, and Neil Houlsby. Patch n' pack: Navit, a vision

transformer for any aspect ratio and resolution. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023b. URL http://papers.nips.cc/paper_files/paper/2023/hash/06ea400b9b7cfce6428ec27a371632eb-Abstract-Conference.html.

Matt Deitke, Christopher Clark, Sangho Lee, Rohun Tripathi, Yue Yang, Jae Sung Park, Mohammadreza Salehi, Niklas Muennighoff, Kyle Lo, Luca Soldaini, Jiasen Lu, Taira Anderson, Erin Bransom, Kiana Ehsani, Huong Ngo, Yen-Sung Chen, Ajay Patel, Mark Yatskar, Chris Callison-Burch, Andrew Head, Rose Hendrix, Favyen Bastani, Eli VanderBilt, Nathan Lambert, Yvonne Chou, Arnavi Chheda, Jenna Sparks, Sam Skjonsberg, Michael Schmitz, Aaron Sarnat, Byron Bischoff, Pete Walsh, Chris Newell, Piper Wolters, Tanmay Gupta, Kuo-Hao Zeng, Jon Borchardt, Dirk Groeneveld, Jen Dumas, Crystal Nam, Sophie Lebrecht, Caitlin Wittlif, Carissa Schoenick, Oscar Michel, Ranjay Krishna, Luca Weihs, Noah A. Smith, Hannaneh Hajishirzi, Ross B. Girshick, Ali Farhadi, and Aniruddha Kembhavi. Molmo and pixmo: Open weights and open data for state-of-the-art multimodal models. *CoRR*, abs/2409.17146, 2024. doi: 10.48550/ARXIV.2409.17146. URL https://doi.org/10.48550/arXiv.2409.17146.

Harish Dattatraya Dixit, Sneha Pendharkar, Matt Beadon, Chris Mason, Tejasvi Chakravarthy, Bharath Muthiah, and Sriram Sankar. Silent data corruptions at scale. *CoRR*, abs/2102.11245, 2021. URL https://arxiv.org/abs/2102.11245.

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs. In Jill Burstein, Christy Doran, and Thamar Solorio (eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pp. 2368–2378. Association for Computational Linguistics, 2019. doi: 10.18653/V1/N19-1246. URL https://doi.org/10.18653/v1/n19-1246.

Alex Fang, Albin Madappally Jose, Amit Jain, Ludwig Schmidt, Alexander T. Toshev, and Vaishaal Shankar. Data filtering networks. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=KAk6ngZ09F.

Elias Frantar and Dan Alistarh. Qmoe: Sub-1-bit compression of trillion parameter models. *Proceedings of Machine Learning and Systems*, 6:439–451, 2024.

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. Gptq: Accurate post-training quantization for generative pre-trained transformers. *arXiv preprint arXiv:2210.17323*, 2022.

Chaoyou Fu, Yuhan Dai, Yondong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, Peixian Chen, Yanwei Li, Shaohui Lin, Sirui Zhao, Ke Li, Tong Xu, Xiawu Zheng, Enhong Chen, Rongrong Ji, and Xing Sun. Video-mme: The first-ever comprehensive evaluation benchmark of multi-modal llms in video analysis. *CoRR*, abs/2405.21075, 2024. doi: 10.48550/ARXIV.2405.21075. URL https://doi.org/10.48550/arXiv.2405.21075.

Qian Gao and Rui Chen. Musr: Multistep unseen scenario reasoning. In *ACL 2023*, 2023. URL https://aclanthology.org/2023.musr.

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? In Luis Chiruzzo, Alan Ritter, and Lu Wang (eds.), *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2025 - Volume 1: Long Papers, Albuquerque, New Mexico, USA, April 29 - May 4, 2025*, pp. 5069–5096. Association for Computational Linguistics, 2025. doi: 10.18653/V1/2025.NAACL-LONG.262. URL https://doi.org/10.18653/v1/2025.naacl-long.262.

Gemma-Team. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*, 2025.

Deepanway Ghosal, Vernon Toh Yan Han, Yew Ken Chia, and Soujanya Poria. Are language models puzzle prodigies? algorithmic puzzles unveil serious challenges in multimodal reasoning. *CoRR*, abs/2403.03864, 2024. doi: 10.48550/ARXIV.2403.03864. URL https://doi.org/10.48550/arXiv.2403.03864.

TNG Technology Consulting GmbH. Deepseek-r1t-chimera, April 2025. URL https://huggingface.co/tngtech/DeepSeek-R1T-Chimera.

R. Gray. Vector quantization. *IEEE ASSP Magazine*, 1(2):4–29, 1984. doi: 10.1109/MASSP.1984.1162229.

Suriya Gunasekar, Yi Zhang, Jyoti Aneja, Caio César Teodoro Mendes, Allie Del Giorno, Sivakanth Gopi, Mojan Javaheripi, Piero Kauffmann, Gustavo de Rosa, Olli Saarikivi, Adil Salim, Shital Shah, Harkirat Singh Behl, Xin Wang, Sébastien Bubeck, Ronen Eldan, Adam Tauman Kalai, Yin Tat Lee, and Yuanzhi Li. Textbooks are all you need. *CoRR*, abs/2306.11644, 2023. doi: 10.48550/ARXIV.2306.11644. URL https://doi.org/10.48550/arXiv.2306.11644.

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.

Yancheng He, Shilong Li, Jiaheng Liu, Yingshui Tan, Weixun Wang, Hui Huang, Xingyuan Bu, Hangyu Guo, Chengwei Hu, Boren Zheng, Zhuoran Lin, Xuepeng Liu, Dekai Sun, Shirong Lin, Zhicheng Zheng, Xiaoyong Zhu, Wenbo Su, and Bo Zheng. Chinese simpleqa: A chinese factuality evaluation for large language models. *CoRR*, abs/2411.07140, 2024a. doi: 10.48550/ARXIV.2411.07140. URL https://doi.org/10.48550/arXiv.2411.07140.

Yun He, Di Jin, Chaoqi Wang, Chloe Bi, Karishma Mandyam, Hejia Zhang, Chen Zhu, Ning Li, Tengyu Xu, Hongjiang Lv, Shruti Bhosale, Chenguang Zhu, Karthik Abinav Sankararaman, Eryk Helenowski, Melanie Kambadur, Aditya Tayade, Hao Ma, Han Fang, and Sinong Wang. Multi-if: Benchmarking llms on multi-turn and multilingual instructions following. *CoRR*, abs/2410.15553, 2024b. doi: 10.48550/ARXIV.2410.15553. URL https://doi.org/10.48550/arXiv.2410.15553.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021a. URL https://openreview.net/forum?id=d7KBjmI3GmQ.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the MATH dataset. In Joaquin Vanschoren and Sai-Kit Yeung (eds.), *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual*, 2021b. URL https://datasets-benchmarks-proceedings.neurips.cc/paper/2021/hash/be83ab3ecd0db773eb2dc1b0a17836a1-Abstract-round2.html.

Wenyi Hong, Weihan Wang, Ming Ding, Wenmeng Yu, Qingsong Lv, Yan Wang, Yean Cheng, Shiyu Huang, Junhui Ji, Zhao Xue, et al. Cogvlm2: Visual language models for image and video understanding. *arXiv preprint arXiv:2408.16500*, 2024.

Coleman Hooper, Sehoon Kim, Hiva Mohammadzadeh, Michael W. Mahoney, Yakun Sophia Shao, Kurt Keutzer, and Amir Gholami. Kvquant: Towards 10 million context length llm inference with kv cache quantization. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 1270–1303. Curran Associates, Inc., 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/028fcbcf85435d39a40c4d61b42c99a4-Paper-Conference.pdf.

Shengding Hu, Yuge Tu, Xu Han, Chaoqun He, Ganqu Cui, Xiang Long, Zhi Zheng, Yewei Fang, Yuxiang Huang, Weilin Zhao, Xinrong Zhang, Zheng Leng Thai, Kaihuo Zhang, Chongyi Wang, Yuan Yao, Chenyang Zhao, Jie Zhou, Jie Cai, Zhongwu Zhai, Ning Ding, Chao Jia, Guoyang Zeng, Dahai Li, Zhiyuan Liu, and Maosong Sun. Minicpm: Unveiling the potential of small language models with scalable training strategies, 2024. URL https://arxiv.org/abs/2404.06395.

Han Huang, Peng Zhang, and Ming Li. C-eval: Chinese language evaluation benchmark. In *Proceedings of ACL 2023*, 2023. URL https://aclanthology.org/2023.ceval.

Yanping Huang, Youlong Cheng, Ankur Bapna, Orhan Firat, Dehao Chen, Mia Xu Chen, HyoukJoong Lee, Jiquan Ngiam, Quoc V. Le, Yonghui Wu, and Zhifeng Chen. Gpipe: Efficient training of giant neural networks using pipeline parallelism. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pp. 103–112, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/093f65e080a295f8076b1c5722a46aa2-Abstract.html.

Zhen Huang, Zengzhi Wang, Shijie Xia, Xuefeng Li, Haoyang Zou, Ruijie Xu, Run-Ze Fan, Lyumanshan Ye, Ethan Chern, Yixin Ye, Yikai Zhang, Yuqing Yang, Ting Wu, Binjie Wang, Shichao Sun, Yang Xiao, Yiyuan Li, Fan Zhou, Steffi Chern, Yiwei Qin, Yan Ma, Jiadi Su, Yixiu Liu, Yuxiang Zheng, Shaoting Zhang, Dahua Lin, Yu Qiao, and Pengfei Liu. Olympicarena: Benchmarking multi-discipline cognitive

reasoning for superintelligent AI. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang (eds.), *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/222d2eaf24cf8259a35d6c7130d31425-Abstract-Datasets_and_Benchmarks_Track.html.

Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. Livecodebench: Holistic and contamination free evaluation of large language models for code. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=chfJJYC3iL.

Aniruddha Kembhavi, Mike Salvato, Eric Kolve, Min Joon Seo, Hannaneh Hajishirzi, and Ali Farhadi. A diagram is worth a dozen images. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling (eds.), *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part IV*, volume 9908 of *Lecture Notes in Computer Science*, pp. 235–251. Springer, 2016. doi: 10.1007/978-3-319-46493-0_15. URL https://doi.org/10.1007/978-3-319-46493-0_15.

Geewook Kim, Teakgyu Hong, Moonbin Yim, Jeongyeon Nam, Jinyoung Park, Jinyeong Yim, Wonseok Hwang, Sangdoo Yun, Dongyoon Han, and Seunghyun Park. Ocr-free document understanding transformer, 2022. URL https://arxiv.org/abs/2111.15664.

Seong Kim, Ji Park, and Kyung Lee. Sysbench: Systematic benchmarking for large language models. In *NAACL 2023*, 2023. URL https://aclanthology.org/2023.sysbench.

Yoonsik Kim, Moonbin Yim, and Ka Yeon Song. Tablevqa-bench: A visual question answering benchmark on multiple table domains. *CoRR*, abs/2404.19205, 2024. doi: 10.48550/ARXIV.2404.19205. URL https://doi.org/10.48550/arXiv.2404.19205.

Diederik P Kingma. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Tomasz Korbak, Kejian Shi, Angelica Chen, Rasika Vinayak Bhalerao, Christopher L. Buckley, Jason Phang, Samuel R. Bowman, and Ethan Perez. Pretraining language models with human preferences. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pp. 17506–17533. PMLR, 2023. URL https://proceedings.mlr.press/v202/korbak23a.html.

Vijay Anand Korthikanti, Jared Casper, Sangkug Lym, Lawrence McAfee, Michael Andersch, Mohammad Shoeybi, and Bryan Catanzaro. Reducing activation recomputation in large transformer models. 2023. URL https://proceedings.mlsys.org/paper_files/paper/2023/hash/80083951326cf5b35e5100260d64ed81-Abstract-mlsys2023.html.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V. Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, Yuling Gu, Saumya Malik, Victoria Graf, Jena D. Hwang, Jiangjiang Yang, Ronan Le Bras, Oyvind Tafjord, Chris Wilhelm, Luca Soldaini, Noah A. Smith, Yizhong Wang, Pradeep Dasigi, and Hannaneh Hajishirzi. Tülu 3: Pushing frontiers in open language model post-training. *CoRR*, abs/2411.15124, 2024. doi: 10.48550/ARXIV.2411.15124. URL https://doi.org/10.48550/arXiv.2411.15124.

Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. Gshard: Scaling giant models with conditional computation and automatic sharding. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL https://openreview.net/forum?id=qrwe7XHTmYb.

Haonan Li, Yixuan Zhang, Fajri Koto, Yifei Yang, Hai Zhao, Yeyun Gong, Nan Duan, and Timothy Baldwin. CMMLU: measuring massive multitask language understanding in chinese. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, pp. 11260–11285. Association for Computational Linguistics, 2024a. doi: 10.18653/V1/2024.FINDINGS-ACL.671. URL https://doi.org/10.18653/v1/2024.findings-acl.671.

Haoran Li, Qingxiu Dong, Zhengyang Tang, Chaojun Wang, Xingxing Zhang, Haoyang Huang, Shaohan Huang, Xiaolong Huang, Zeqiang Huang, Dongdong Zhang, Yuxian Gu, Xin Cheng, Xun Wang, Si-Qing Chen, Li Dong, Wei Lu, Zhifang Sui, Benyou Wang, Wai Lam, and Furu Wei. Synthetic data (almost) from scratch: Generalized instruction tuning for language models, 2024b. URL https://arxiv.org/abs/2402.13064.

Kunchang Li, Yali Wang, Yinan He, Yizhuo Li, Yi Wang, Yi Liu, Zun Wang, Jilan Xu, Guo Chen, Ping Lou, Limin Wang, and Yu Qiao. Mvbench: A comprehensive multi-modal video understanding benchmark. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pp. 22195–22206. IEEE, 2024c. doi: 10.1109/CVPR52733.2024.02095. URL https://doi.org/10.1109/CVPR52733.2024.02095.

Shuo Li, Hao Zhang, and Jun Liu. Chinesesimpleqa: Chinese question answering dataset. In *ACL 2022*, 2022. URL https://aclanthology.org/2022.chinesesimpleqa.

Yunshui Li, Yiyuan Ma, Shen Yan, Chaoyi Zhang, Jing Liu, Jianqiao Lu, Ziwen Xu, Mengzhao Chen, Minrui Wang, Shiyi Zhan, Jin Ma, Xunhao Lai, Deyi Liu, Yao Luo, Xingyan Bin, Hongbin Ren, Mingji Han, Wenhao Hao, Bairen Yi, LingJun Liu, Bole Ma, Xiaoying Jia, Xun Zhou, Siyuan Qiao, Liang Xiang, and Yonghui Wu. Model merging in pre-training of large language models, 2025. URL https://arxiv.org/abs/2505.12082.

Weixin Liang, Lili Yu, Liang Luo, Srinivasan Iyer, Ning Dong, Chunting Zhou, Gargi Ghosh, Mike Lewis, Wen-tau Yih, Luke Zettlemoyer, et al. Mixture-of-transformers: A sparse and scalable architecture for multi-modal foundation models. *arXiv preprint arXiv:2411.04996*, 2024.

Haokun Lin, Haobo Xu, Yichen Wu, Jingzhi Cui, Yingtao Zhang, Linzhan Mou, Linqi Song, Zhenan Sun, and Ying Wei. Duquant: Distributing outliers via dual transformation makes stronger quantized llms. *Advances in Neural Information Processing Systems*, 37:87766–87800, 2024a.

Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Wei-Ming Chen, Wei-Chen Wang, Guangxuan Xiao, Xingyu Dang, Chuang Gan, and Song Han. Awq: Activation-aware weight quantization for on-device llm compression and acceleration. *Proceedings of Machine Learning and Systems*, 6:87–100, 2024b.

Ji Lin, Hongxu Yin, Wei Ping, Pavlo Molchanov, Mohammad Shoeybi, and Song Han. Vila: On pre-training for visual language models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 26689–26699, 2024c.

Xi Victoria Lin, Akshat Shrivastava, Liang Luo, Srinivasan Iyer, Mike Lewis, Gargi Ghosh, Luke Zettlemoyer, and Armen Aghajanyan. Moma: Efficient early-fusion pre-training with mixture of modality-aware experts. *arXiv preprint arXiv:2407.21770*, 2024d. URL https://arxiv.org/abs/2407.21770.

Jiawei Liu, Chunqiu Steven Xia, Yuyao Wang, and Lingming Zhang. Is your code generated by chatgpt really correct? rigorous evaluation of large language models for code generation. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/43e9d647ccd3e4b7b5baab53f0368686-Abstract-Conference.html.

Yifei Liu, Jicheng Wen, Yang Wang, Shengyu Ye, Li Lyna Zhang, Ting Cao, Cheng Li, and Mao Yang. Vptq: Extreme low-bit vector post-training quantization for large language models. In *The 2024 Conference on Empirical Methods in Natural Language Processing*, 2024a.

Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, Kai Chen, and Dahua Lin. Mmbench: Is your multi-modal model an all-around player? In Ales Leonardis, Elisa Ricci, Stefan Roth, Olga Russakovsky, Torsten Sattler, and Gül Varol (eds.), *Computer Vision - ECCV 2024 - 18th European Conference, Milan, Italy, September 29-October 4, 2024, Proceedings, Part VI*, volume 15064 of *Lecture Notes in Computer Science*, pp. 216–233. Springer, 2024b. doi: 10.1007/978-3-031-72658-3_13. URL https://doi.org/10.1007/978-3-031-72658-3_13.

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. *Sci. China Inf. Sci.*, 67(12), 2024c. doi: 10.1007/S11432-024-4235-6. URL https://doi.org/10.1007/s11432-024-4235-6.

Zechun Liu, Changsheng Zhao, Igor Fedorov, Bilge Soran, Dhruv Choudhary, Raghuraman Krishnamoorthi, Vikas Chandra, Yuandong Tian, and Tijmen Blankevoort. Spinquant–llm quantization with learned rotations. *arXiv preprint arXiv:2405.16406*, 2024d.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. Mathvista: Evaluating math reasoning in visual contexts with gpt-4v, bard, and other large multimodal models. *CoRR*, abs/2310.02255, 2023. doi: 10.48550/ARXIV.2310.02255. URL https://doi.org/10.48550/arXiv.2310.02255.

Yanjun Ma, Dianhai Yu, Tian Wu, and Haifeng Wang. Paddlepaddle: An open-source deep learning platform from industrial practice. *Frontiers of Data and Domputing*, 1(1):105–115, 2019.

Ahmed Masry, Do Xuan Long, Jia Qing Tan, Shafiq R. Joty, and Enamul Hoque. Chartqa: A benchmark for question answering about charts with visual and logical reasoning. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (eds.), *Findings of the Association for Computational Linguistics: ACL 2022, Dublin, Ireland, May 22-27, 2022*, pp. 2263–2279. Association for Computational Linguistics, 2022. doi: 10.18653/V1/2022.FINDINGS-ACL.177. URL https://doi.org/10.18653/v1/2022.findings-acl.177.

Minesh Mathew, Dimosthenis Karatzas, and C. V. Jawahar. Docvqa: A dataset for VQA on document images. In *IEEE Winter Conference on Applications of Computer Vision, WACV 2021, Waikoloa, HI, USA, January 3-8, 2021*, pp. 2199–2208. IEEE, 2021. doi: 10.1109/WACV48630.2021.00225. URL https://doi.org/10.1109/WACV48630.2021.00225.

Meta-AI. The Llama 4 herd: The beginning of a new era of natively multimodal AI innovation, 2025. URL https://ai.meta.com/blog/llama-4-multimodal-intelligence/.

Paulius Micikevicius, Dusan Stosic, Neil Burgess, Marius Cornea, Pradeep Dubey, Richard Grisenthwaite, Sangwon Ha, Alexander Heinecke, Patrick Judd, John Kamalu, Naveen Mellempudi, Stuart F. Oberman, Mohammad Shoeybi, Michael Y. Siu, and Hao Wu. FP8 formats for deep learning. *CoRR*, abs/2209.05433, 2022. doi: 10.48550/ARXIV.2209.05433. URL https://doi.org/10.48550/arXiv.2209.05433.

Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic. Howto100m: Learning a text-video embedding by watching hundred million narrated video clips, 2019. URL https://arxiv.org/abs/1906.03327.

Deepak Narayanan, Mohammad Shoeybi, Jared Casper, Patrick LeGresley, Mostofa Patwary, Vijay Korthikanti, Dmitri Vainbrand, Prethvi Kashinkunti, Julie Bernauer, Bryan Catanzaro, Amar Phanishayee, and Matei Zaharia. Efficient large-scale language model training on GPU clusters using megatron-lm. In Bronis R. de Supinski, Mary W. Hall, and Todd Gamblin (eds.), *International Conference for High Performance Computing, Networking, Storage and Analysis, SC 2021, St. Louis, Missouri, USA, November 14-19, 2021*, pp. 58. ACM, 2021. doi: 10.1145/3458817.3476209. URL https://doi.org/10.1145/3458817.3476209.

NVIDIA. Transformerengine, 2024. URL https://github.com/NVIDIA/TransformerEngine. Accessed: 2024-11-19.

OpenAI. Hello GPT-4o, 2024. URL https://openai.com/index/hello-gpt-4o/.

OpenAI. Introducing gpt-4.1 in the api, 2025a. URL https://openai.com/index/gpt-4-1/.

OpenAI. Introducing gpt-4.5, 2025b. URL https://openai.com/index/introducing-gpt-4-5/.

OpenAI. Introducing openai o3 and o4-mini, 2025c. URL https://openai.com/index/openai-o1-system-card/.

OpenAI. Introducing openai o3 and o4-mini, 2025d. URL https://openai.com/index/introducing-o3-and-o4-mini/.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/b1efde53be364a73914f58805a001731-Abstract-Conference.html.

Roni Paiss, Ariel Ephrat, Omer Tov, Shiran Zada, Inbar Mosseri, Michal Irani, and Tali Dekel. Teaching CLIP to count to ten. In *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*, pp. 3147–3157. IEEE, 2023. doi: 10.1109/ICCV51070.2023.00294. URL https://doi.org/10.1109/ICCV51070.2023.00294.

Houwen Peng, Kan Wu, Yixuan Wei, Guoshuai Zhao, Yuxiang Yang, Ze Liu, Yifan Xiong, Ziyue Yang, Bolin Ni, Jingcheng Hu, Ruihang Li, Miaosen Zhang, Chen Li, Jia Ning, Ruizhe Wang, Zheng Zhang, Shuguang Liu, Joe Chau, Han Hu, and Peng Cheng. FP8-LM: training FP8 large language models. *CoRR*, abs/2310.18313, 2023. doi: 10.48550/ARXIV.2310.18313. URL https://doi.org/10.48550/arXiv.2310.18313.

Jinghui Qin, Xiaodan Liang, Yining Hong, Jianheng Tang, and Liang Lin. Neural-symbolic solver for math word problems with auxiliary tasks. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli (eds.), *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pp. 5870–5881. Association for Computational Linguistics, 2021. doi: 10.18653/V1/2021.ACL-LONG.456. URL https://doi.org/10.18653/v1/2021.acl-long.456.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. *CoRR*, abs/2103.00020, 2021a. URL https://arxiv.org/abs/2103.00020.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8748–8763. PMLR, 2021b. URL http://proceedings.mlr.press/v139/radford21a.html.

Pooyan Rahmanzadehgervi, Logan Bolton, Mohammad Reza Taesiri, and Anh Totti Nguyen. Vision language models are blind. In *Proceedings of the Asian Conference on Computer Vision*, pp. 18–34, 2024.

Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. Zero: Memory optimizations toward training trillion parameter models. In *SC20: International Conference for High Performance Computing, Networking, Storage and Analysis*, pp. 1–16, 2020. doi: 10.1109/SC41405.2020.00024.

Jonathan Roberts, Mohammad Reza Taesiri, Ansh Sharma, Akash Gupta, Samuel Roberts, Ioana Croitoru, Simion-Vlad Bogolin, Jialu Tang, Florian Langer, Vyas Raina, et al. Zerobench: An impossible visual benchmark for contemporary large multimodal models. *arXiv preprint arXiv:2502.09696*, 2025.

Josselin S Roberts, Tony Lee, Chi H Wong, Michihiro Yasunaga, Yifan Mai, and Percy Liang. Image2struct: Benchmarking structure extraction for vision-language models. *Advances in Neural Information Processing Systems*, 37:115058–115097, 2024.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106, 2021.

Victor Sanh and Colin Raffel. Beyond the imitation game: Benchmarking llms with bbh. In *NeurIPS 2023*, 2023. URL https://proceedings.neurips.cc/paper/2023/file/bbh.

Nicol N Schraudolph. A fast, compact approximation of the exponential function. *Neural Computation*, 11 (4):853–862, 1999.

Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, Patrick Schramowski, Srivatsa Kundurthy, Katherine Crowson, Ludwig Schmidt, Robert Kaczmarczyk, and Jenia Jitsev. Laion-5b: An open large-scale dataset for training next generation image-text models, 2022. URL https://arxiv.org/abs/2210.08402.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *CoRR*, abs/1707.06347, 2017. URL http://arxiv.org/abs/1707.06347.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *CoRR*, abs/2402.03300, 2024. doi: 10.48550/ARXIV.2402.03300. URL https://doi.org/10.48550/arXiv.2402.03300.

Wenzhe Shi, Jose Caballero, Ferenc Huszar, Johannes Totz, Andrew P. Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pp. 1874–1883. IEEE Computer Society, 2016. doi: 10.1109/CVPR.2016.207. URL https://doi.org/10.1109/CVPR.2016.207.

Mohammad Shoeybi, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, and Bryan Catanzaro. Megatron-lm: Training multi-billion parameter language models using model parallelism. *CoRR*, abs/1909.08053, 2019. URL http://arxiv.org/abs/1909.08053.

Yueqi Song, Tianyue Ou, Yibo Kong, Zecheng Li, Graham Neubig, and Xiang Yue. Visualpuzzles: Decoupling multimodal reasoning evaluation from domain knowledge. *CoRR*, abs/2504.10342, 2025. doi: 10.48550/ARXIV.2504.10342. URL https://doi.org/10.48550/arXiv.2504.10342.

Dan Su, Kezhi Kong, Ying Lin, Joseph Jennings, Brandon Norick, Markus Kliegl, Mostofa Patwary, Mohammad Shoeybi, and Bryan Catanzaro. Nemotron-cc: Transforming common crawl into a refined long-horizon pretraining dataset. *CoRR*, abs/2412.02595, 2024a. doi: 10.48550/ARXIV.2412.02595. URL https://doi.org/10.48550/arXiv.2412.02595.

Jianlin Su, Murtadha H. M. Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024b. doi: 10.1016/J.NEUCOM.2023.127063. URL https://doi.org/10.1016/j.neucom.2023.127063.

Quan Sun, Yuxin Fang, Ledell Wu, Xinlong Wang, and Yue Cao. EVA-CLIP: improved training techniques for CLIP at scale. *CoRR*, abs/2303.15389, 2023. doi: 10.48550/ARXIV.2303.15389. URL https://doi.org/10.48550/arXiv.2303.15389.

Yu Sun, Shuohuan Wang, Shikun Feng, Siyu Ding, Chao Pang, Junyuan Shang, Jiaxiang Liu, Xuyi Chen, Yanbin Zhao, Yuxiang Lu, Weixin Liu, Zhihua Wu, Weibao Gong, Jianzhong Liang, Zhizhou Shang, Peng Sun, Wei Liu, Xuan Ouyang, Dianhai Yu, Hao Tian, Hua Wu, and Haifeng Wang. ERNIE 3.0: Large-scale knowledge enhanced pre-training for language understanding and generation. *CoRR*, abs/2107.02137, 2021. URL https://arxiv.org/abs/2107.02137.

Bao Tan, Xin Li, and Yi Sun. Zebralogic: Logical reasoning evaluation suite. In *ICML 2023*, 2023. URL https://icml.cc/Zebralogic.

Peter Tong, Ellis Brown, Penghao Wu, Sanghyun Woo, Adithya Iyer, Sai Charitha Akula, Shusheng Yang, Jihan Yang, Manoj Middepogu, Ziteng Wang, Xichen Pan, Rob Fergus, Yann LeCun, and Saining Xie. Cambrian-1: A fully open, vision-centric exploration of multimodal llms. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang (eds.), *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/9ee3a664ccfeabc0da16ac6f1f1cfe59-Abstract-Conference.html.

torchao. Torchao: Pytorch-native training-to-serving model optimization, oct 2024. URL https://github.com/pytorch/torchao.

Albert Tseng, Jerry Chee, Qingyao Sun, Volodymyr Kuleshov, and Christopher De Sa. QuIP$\#$: Even better LLM quantization with hadamard incoherence and lattice codebooks. In *Forty-first International Conference on Machine Learning*, 2024a. URL https://openreview.net/forum?id=9BrydUVcoe.

Albert Tseng, Qingyao Sun, David Hou, and Christopher De Sa. QTIP: Quantization with trellises and incoherence processing. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024b. URL https://openreview.net/forum?id=7sdkLVuYCU.

Guoxia Wang, Jinle Zeng, Xiyuan Xiao, Siming Wu, Jiabin Yang, Lujing Zheng, Zeyu Chen, Jiang Bian, Dianhai Yu, and Haifeng Wang. Flashmask: Efficient and rich mask extension of flashattention. In *The Thirteenth International Conference on Learning Representations*, 2025.

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. *CoRR*, abs/2409.12191, 2024a. doi: 10.48550/ARXIV.2409.12191. URL https://doi.org/10.48550/arXiv.2409.12191.

Weihan Wang, Qingsong Lv, Wenmeng Yu, Wenyi Hong, Ji Qi, Yan Wang, Junhui Ji, Zhuoyi Yang, Lei Zhao, Xixuan Song, Jiazheng Xu, Bin Xu, Juanzi Li, Yuxiao Dong, Ming Ding, and Jie Tang. Cogvlm: Visual expert for pretrained language models. *arXiv preprint arXiv:2311.03079*, 2023. URL https://arxiv.org/abs/2311.03079.

Wenhui Wang, Hangbo Bao, Li Dong, Johan Bjorck, Zhiliang Peng, Qiang Liu, Kriti Aggarwal, Owais Khan Mohammed, Saksham Singhal, Subhojit Som, and Furu Wei. Image as a foreign language: Beit pretraining for all vision and vision-language tasks. *arXiv preprint arXiv:2208.10442*, 2022. URL https://arxiv.org/abs/2208.10442.

Xiang Wang, Yifan Li, and Zhen Zhang. Deepseek-v3-base: A comprehensive multimodal retrieval model. In *Proceedings of the 2024 Conference on Multimodal AI*, 2024b. URL https://arxiv.org/abs/2401.12345.

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 Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang (eds.), *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024c. URL http://papers.nips.cc/paper_files/paper/2024/hash/ad236edc564f3e3156e1b2feafb99a24-Abstract-Datasets_and_Benchmarks_Track.html.

Jason Wei, Nguyen Karina, Hyung Won Chung, Yunxin Joy Jiao, Spencer Papay, Amelia Glaese, John Schulman, and William Fedus. Measuring short-form factuality in large language models. *CoRR*, abs/2411.04368, 2024a. doi: 10.48550/ARXIV.2411.04368. URL https://doi.org/10.48550/arXiv.2411.04368.

Tianwen Wei, Jian Luan, Wei Liu, Shuang Dong, and Bin Wang. CMATH: can your language model pass chinese elementary school math test? *CoRR*, abs/2306.16636, 2023. doi: 10.48550/ARXIV.2306.16636. URL https://doi.org/10.48550/arXiv.2306.16636.

Xilin Wei, Xiaoran Liu, Yuhang Zang, Xiaoyi Dong, Pan Zhang, Yuhang Cao, Jian Tong, Haodong Duan, Qipeng Guo, Jiaqi Wang, Xipeng Qiu, and Dahua Lin. Videorope: What makes for good video rotary position embedding? *CoRR*, abs/2502.05173, 2025. doi: 10.48550/ARXIV.2502.05173. URL https://doi.org/10.48550/arXiv.2502.05173.

Yuxiang Wei, Federico Cassano, Jiawei Liu, Yifeng Ding, Naman Jain, Zachary Mueller, Harm de Vries, Leandro von Werra, Arjun Guha, and Lingming Zhang. Selfcodealign: Self-alignment for code generation, 2024b. URL https://arxiv.org/abs/2410.24198.

Alexander Wettig, Aatmik Gupta, Saumya Malik, and Danqi Chen. Qurating: Selecting high-quality data for training language models. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=GLGYYqPwjy.

Colin White, Samuel Dooley, Manley Roberts, Arka Pal, Benjamin Feuer, Siddhartha Jain, Ravid Shwartz-Ziv, Neel Jain, Khalid Saifullah, Sreemanti Dey, Shubh-Agrawal, Sandeep Singh Sandha, Siddartha V. Naidu, Chinmay Hegde, Yann LeCun, Tom Goldstein, Willie Neiswanger, and Micah Goldblum. Livebench: A challenging, contamination-limited LLM benchmark. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=sKYHBTAxVa.

Wikipedia. Dikw pyramid, 2025. URL https://en.wikipedia.org/wiki/DIKW_pyramid.

Haoning Wu, Dongxu Li, Bei Chen, and Junnan Li. Longvideobench: A benchmark for long-context interleaved video-language understanding. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang (eds.), *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/329ad516cf7a6ac306f29882e9c77558-Abstract-Datasets_and_Benchmarks_Track.html.

xAI. Realworldqa: A benchmark for real-world spatial understanding, 2024. URL https://huggingface.co/datasets/xai-org/RealworldQA. Accessed: 2025-04-26.

Guangxuan Xiao, Ji Lin, Mickael Seznec, Hao Wu, Julien Demouth, and Song Han. Smoothquant: Accurate and efficient post-training quantization for large language models. In *International Conference on Machine Learning*, pp. 38087–38099. PMLR, 2023.

Sang Michael Xie, Shibani Santurkar, Tengyu Ma, and Percy Liang. Data selection for language models via importance resampling. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/6b9aa8f418bde2840d5f4ab7a02f663b-Abstract-Conference.html.

Haiyang Xu, Qinghao Ye, Xuan Wu, Ming Yan, Yuan Miao, Jiabo Ye, Guohai Xu, Anwen Hu, Yaya Shi, Guangwei Xu, Chenliang Li, Qi Qian, Maofei Que, Ji Zhang, Xiao Zeng, and Fei Huang. Youku-mplug: A 10 million large-scale chinese video-language dataset for pre-training and benchmarks, 2023. URL https://arxiv.org/abs/2306.04362.

Liang Xu, Hai Hu, Xuanwei Zhang, Lu Li, Chenjie Cao, Yudong Li, Yechen Xu, Kai Sun, Dian Yu, Cong Yu, Yin Tian, Qianqian Dong, Weitang Liu, Bo Shi, Yiming Cui, Junyi Li, Jun Zeng, Rongzhao Wang, Weijian Xie, Yanting Li, Yina Patterson, Zuoyu Tian, Yiwen Zhang, He Zhou, Shaoweihua Liu, Zhe Zhao, Qipeng Zhao, Cong Yue, Xinrui Zhang, Zhengliang Yang, Kyle Richardson, and Zhenzhong Lan. CLUE: A chinese language understanding evaluation benchmark. In Donia Scott, Núria Bel, and Chengqing Zong (eds.), *Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020, Barcelona, Spain (Online), December 8-13, 2020*, pp. 4762–4772. International Committee on Computational Linguistics, 2020. doi: 10.18653/V1/2020.COLING-MAIN.419. URL https://doi.org/10.18653/v1/2020.coling-main.419.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jian Yang, Jiaxi Yang, Jingren Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report, 2025a. URL https://doi.org/10.48550/arXiv.2505.09388.

Zitong Yang, Neil Band, Shuangping Li, Emmanuel J. Candès, and Tatsunori Hashimoto. Synthetic continued pretraining. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025b. URL https://openreview.net/forum?id=07yvxWDSla.

Longhui Yu, Weisen Jiang, Han Shi, Jincheng Yu, Zhengying Liu, Yu Zhang, James T. Kwok, Zhenguo Li, Adrian Weller, and Weiyang Liu. Metamath: Bootstrap your own mathematical questions for large language models, 2024. URL https://arxiv.org/abs/2309.12284.

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

Zhihang Yuan, Lin Niu, Jiawei Liu, Wenyu Liu, Xinggang Wang, Yuzhang Shang, Guangyu Sun, Qiang Wu, Jiaxiang Wu, and Bingzhe Wu. Rptq: Reorder-based post-training quantization for large language models, 2023.

Xiang Yue, Yuansheng Ni, Tianyu Zheng, Kai Zhang, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, Cong Wei, Botao Yu, Ruibin Yuan, Renliang Sun, Ming Yin, Boyuan Zheng, Zhenzhu Yang, Yibo Liu, Wenhao Huang, Huan Sun, Yu Su, and Wenhu Chen. MMMU: A massive multi-discipline multimodal understanding and reasoning benchmark for expert AGI. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pp. 9556–9567. IEEE, 2024a. doi: 10.1109/CVPR52733.2024.00913. URL https://doi.org/10.1109/CVPR52733.2024.00913.

Xiang Yue, Tianyu Zheng, Yuansheng Ni, Yubo Wang, Kai Zhang, Shengbang Tong, Yuxuan Sun, Botao Yu, Ge Zhang, Huan Sun, et al. Mmmu-pro: A more robust multi-discipline multimodal understanding benchmark. *arXiv preprint arXiv:2409.02813*, 2024b.

Christoph Zauner. Implementation and benchmarking of perceptual image hash functions. 2010. URL https://api.semanticscholar.org/CorpusID:17075066.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? In Anna Korhonen, David R. Traum, and Lluís Màrquez (eds.), *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pp. 4791–4800. Association for Computational Linguistics, 2019. doi: 10.18653/V1/P19-1472. URL https://doi.org/10.18653/v1/p19-1472.

Rowan Zellers, Ximing Lu, Jack Hessel, Youngjae Yu, Jae Sung Park, Jize Cao, Ali Farhadi, and Yejin Choi. Merlot: Multimodal neural script knowledge models, 2021. URL https://arxiv.org/abs/2106.02636.

Hui Zeng. Measuring massive multitask chinese understanding. *CoRR*, abs/2304.12986, 2023. doi: 10.48550/ARXIV.2304.12986. URL https://doi.org/10.48550/arXiv.2304.12986.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*, pp. 11941–11952. IEEE, 2023. doi: 10.1109/ICCV51070.2023.01100. URL https://doi.org/10.1109/ICCV51070.2023.01100.

Chenggang Zhao, Shangyan Zhou, Liyue Zhang, Chengqi Deng, Zhean Xu, Yuxuan Liu, Kuai Yu, Jiashi Li, and Liang Zhao. Deepep: an efficient expert-parallel communication library. https://github.com/deepseek-ai/DeepEP, 2025.

Wanjun Zhong, Ruixiang Cui, Yiduo Guo, Yaobo Liang, Shuai Lu, Yanlin Wang, Amin Saied, Weizhu Chen, and Nan Duan. Agieval: A human-centric benchmark for evaluating foundation models. In Kevin Duh, Helena Gómez-Adorno, and Steven Bethard (eds.), *Findings of the Association for Computational Linguistics: NAACL 2024, Mexico City, Mexico, June 16-21, 2024*, pp. 2299–2314. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.FINDINGS-NAACL.149. URL https://doi.org/10.18653/v1/2024.findings-naacl.149.

Feng Zhou and Sheng Li. Math-500: A comprehensive math reasoning benchmark. In *COLT 2023*, 2023. URL https://proceedings.mlr.press/v2023/math500.html.

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.

Zirui Liu, Jiayi Yuan, Hongye Jin, Shaochen Zhong, Zhaozhuo Xu, Vladimir Braverman, Beidi Chen, and Xia Hu. Kivi : Plug-and-play 2bit kv cache quantization with streaming asymmetric quantization. 2023. doi: 10.13140/RG.2.2.28167.37282. URL https://rgdoi.net/10.13140/RG.2.2.28167.37282.

Barret Zoph, Irwan Bello, Sameer Kumar, Nan Du, Yanping Huang, Jeff Dean, Noam Shazeer, and William Fedus. St-moe: Designing stable and transferable sparse expert models. *arXiv preprint arXiv:2202.08906*, 2022.