HunyuanOCR-1.5: Making Lightweight OCR VLMs Faster and Better

Gengluo Li[1,*] **Xingyu Wan**[2,*] **Shangpin Peng**[2,*] **Weinong Wang**[2,*] **Hao Feng**[2,*]
Yongkun Du[2,*] **Binghong Wu**[2] **Zheng Ruan**[2] **Zhiqiong Lu**[2] **Liang Wu**[2] **Pengyuan Lyu**[2]
Huawen Shen[2] **Zibin Lin**[2] **Shijing Hu**[2] **Jieneng Yang**[2] **Hongbing Wen**[2] **Guanghua Yu**[2]
Hong Liu[2] **Bochao Wang**[2] **Can Ma**[1] **Han Hu**[2] **Chengquan Zhang**[2,†,✉] **Yu Zhou**[3,✉]

[1]Institute of Information Engineering, Chinese Academy of Sciences
[2]Large Language Model Department, Tencent [3]Nankai University

ligengluo@iie.ac.cn zchengquan@gmail.com yzhou@nankai.edu.cn

https://huggingface.co/tencent/HunyuanOCR

https://github.com/Tencent-Hunyuan/HunyuanOCR

Abstract

We present HunyuanOCR-1.5, a lightweight and end-to-end OCR-specialized vision-language model. HunyuanOCR targets a broad range of text-centric visual tasks, unifying document parsing, text spotting, information extraction, text-image translation, and multi-image document understanding within a single end-to-end VLM. Building upon the validated lightweight architecture of HunyuanOCR-1.0, HunyuanOCR-1.5 does not redesign the model backbone, but instead performs a systematic upgrade around two goals: making the model faster and better. For efficiency, we adapt *DFlash* inference acceleration to OCR decoding, significantly reducing the decoding latency of long structured outputs such as dense documents, tables, and formulas while preserving the output distribution. Powered by *DFlash*, HunyuanOCR-1.5 achieves a **6.37**× speedup in Transformer inference and a **2.14**× speedup under vLLM, delivering the fastest inference speed among all lightweight OCR VLMs. For capability, we propose *Agentic Data Flow*, an agent-driven data construction system that transforms model weaknesses into executable data requirements and autonomously performs material search, quality verification, and data pipeline development. Through this framework, we significantly enhance the model's long-tail capabilities across ancient-script OCR, fine-grained chart and table parsing, multi-image text-centric QA, low-resource multilingual parsing, and document hallucination evaluation. Crucially, HunyuanOCR-1.5 stands as **the top-tier end-to-end OCR solution** on OmniDocBench v1.6, paired with unrivaled inference efficiency and new performance milestones across the aforementioned long-tail domains. Combined with an upgraded pretraining and post-training recipe, HunyuanOCR-1.5 further extends the capability boundary of the model in high-resolution, long-context, and multi-task scenarios. We characterize these upgrades through a capability-oriented evaluation, and experiments show that HunyuanOCR-1.5 achieves both faster inference and broader OCR capability coverage while retaining the deployment advantages of a lightweight end-to-end model. We will release the model weights and training code to the community to promote the research, reproduction, and real-world application of OCR-specialized vision-language models.

1 Introduction

Visual text serves as the most ubiquitous and dense carrier of human knowledge. For decades, Optical Character Recognition (OCR) [1, 2] has been the foundational technology for digitizing this information, traditionally functioning as a simple text transcription tool. However, as the demand for machine intelligence grows, this narrow definition is no longer sufficient. Modern applications require a comprehensive interface capable of supporting diverse text-centric visual tasks, ranging from document parsing [3–5] and information extraction to visual question answering [6], text-image translation [7, 8], and multi-image document understanding [9, 10].

*Equal contribution. †Project leader. ✉Corresponding author.

Preprint.

To tackle these complex tasks, traditional cascaded pipelines relying on disjointed modules for detection, recognition, and downstream processing often struggle with error propagation and architectural redundancy. In contrast, the development of vision-language models (VLMs) [11–27] has paved the way for an elegant, end-to-end alternative. Driven by this trend, the community is developing OCR-specialized VLMs. These models are expected to tackle OCR as a unified visual-text understanding problem, where fine-grained perception, layout modeling, structured generation, and semantic reasoning are jointly performed within a single architecture.

However, most existing OCR-specialized VLMs are still primarily designed around document parsing, with the objective often restricted to converting a single page into structured outputs such as Markdown, HTML, or LaTeX. While document parsing is indeed one of the core OCR capabilities, the demands of real-world OCR go far beyond it, including text spotting in open scenarios, structured field extraction, question answering grounded in textual images, multilingual text-image translation, and reasoning across multiple pages or images. Fundamentally, a true OCR-specialized model should not be reduced to a mere document parser, but should be a unified end-to-end model covering diverse OCR tasks.

HunyuanOCR-1.0 [28] has validated the feasibility of this philosophy: a lightweight, end-to-end OCR-specialized VLM that achieves leading performance across document parsing, text spotting, information extraction, visual question answering, and text-image translation. It demonstrated the effectiveness of unifying OCR capabilities within a compact architecture and highlighted the crucial role of high-quality OCR data and task-oriented training strategies in building practical OCR systems. Building on this foundation, HunyuanOCR-1.5 does not pursue a redesign of the model architecture, but instead addresses a more deployment-oriented question: *on top of the validated HunyuanOCR framework, how can the model become faster and better*?

Faster: DFlash-based inference acceleration. End-to-end OCR is often accompanied by long autoregressive decoding [29]. In scenarios such as dense documents, tables, formulas, and long structured outputs, the decoding overhead becomes a major bottleneck in practical deployment [30]. To this end, HunyuanOCR-1.5 introduces a speculative decoding [31–35] framework based on DFlash [36] for inference acceleration: a lightweight block-diffusion [37–39] draft model drafts multiple candidate tokens in parallel, which are then verified by the target model in a single pass. While preserving the output distribution of the target model, this significantly improves decoding efficiency for long outputs, achieving a $6.37\times$ speedup with Transformers and a $2.14\times$ speedup with vLLM in our evaluation, making the model more practical in real-world deployment environments that demand both accuracy and speed. Beyond server-grade deployment with vLLM [40], HunyuanOCR-1.5 also supports PC-side inference through llama.cpp [41], enabling deployment on CPUs, consumer GPUs, and laptops.

Better: Agentic data flow and refined training recipes. Driven by comprehensive upgrades on both the data and training sides, HunyuanOCR-1.5 establishes itself as the **SOTA end-to-end OCR solution** on OmniDocBench v1.6. To achieve this capability boundary extension, we propose *Agentic Data Flow* on the data side, an agent-driven data-construction system that translates model weaknesses into executable data requirements [42–44]. Different from conventional pipelines that rely entirely on manually written scripts and manually collected materials, Agentic Data Flow allows agents to deeply participate in material search, tool-based verification, sample cleaning, and data pipeline development, and to iterate in a closed loop with algorithm engineers. In HunyuanOCR-1.5, this system is used for targeted data construction of long-tail capabilities such as low-resource OCR, ancient-script OCR [4], and multi-image QA [9, 10].

On the training side, we systematically upgrade the training recipe around capability boundary extension. In the pretraining stage, we revisit and re-plan Stage3 of HunyuanOCR-1.0, incorporating the new capability data produced by Agentic Data Flow, multi-image data, and historical OCR data, while increasing the maximum image resolution to 4K and extending the context window to 128K, so that the model can robustly adapt to high-resolution documents, long contexts, and multi-page or multi-image inputs. In the post-training stage, we refine the SFT data and introduce new high-quality training data and further explore RL across different OCR tasks to amplify the gains brought by reinforcement learning.

To systematically characterize the practical benefits of these upgrades, HunyuanOCR-1.5 is evaluated from a capability-oriented perspective rather than relying on a single benchmark. The evaluation covers both inherited and newly added capabilities, including end-to-end document parsing [3], text spotting, multilingual OCR, ancient-script recognition [4], text-image translation [7, 8], multi-image QA [9, 10], information extraction, and hallucination-related reliability. This evaluation perspective is aligned with the design goal of HunyuanOCR-1.5: extending HunyuanOCR into a faster and more comprehensive unified end-to-end OCR-specialized VLM. In addition, we plan to release all the model weights and training code of HunyuanOCR-1.5 to the community, providing infrastructure for reproducing, fine-tuning, and extending OCR-specialized VLMs, and further promoting research and applications in OCR perception, document understanding, and multi-task modeling.

The main contributions of this report are summarized as follows:

- We present **HunyuanOCR-1.5**, an upgraded lightweight end-to-end OCR-specialized VLM that further extends diverse OCR task capabilities on top of HunyuanOCR-1.0, and we plan to release the model weights and training code to support community reproduction, fine-tuning, and capability extension.
- We adapt **DFlash** to HunyuanOCR inference and support PC-side deployment through llama.cpp, significantly improving the decoding efficiency of long structured OCR outputs while enabling both server-grade and local OCR deployment.
- We propose **Agentic Data Flow** and systematically upgrade the training recipe: an agent-driven data system produces long-tail capability data such as low-resource OCR, ancient-script OCR, and multi-image QA; in pretraining, we re-plan Stage3 and extend to 4K resolution and a context of 128K; and in post-training, we improve the capability ceiling through high-quality SFT data and task-specific RL exploration.

2 Related Work

General vision-language models. Recent general VLMs have demonstrated strong multimodal perception and reasoning abilities and have shown promising OCR-related capabilities in diverse visual scenarios. Representative models, such as GPT-4o [11], Gemini [12–16], Qwen-VL [17–20], and InternVL [21–27], can recognize text in natural images, documents, charts, and screenshots and further perform text-centric question answering or reasoning based on visual content. However, these models are primarily designed as general-purpose multimodal assistants rather than OCR-specialized systems. As a result, they often require large model sizes and high inference costs, and their performance may become unstable in OCR-intensive scenarios that require fine-grained text perception, dense document parsing, strict reading order preservation, or faithful structured output. In addition, general VLMs are not explicitly optimized for deployment-oriented OCR workloads, especially high-resolution long-document parsing and large-scale production serving.

OCR-specific vision-language models. To address the limitations of general VLMs in OCR-centric scenarios, recent works have explored OCR-specific vision-language models [28, 45, 46]. Most existing OCR expert VLMs are primarily designed for document parsing [47–50], aiming to convert page-level document images into structured outputs such as Markdown, HTML, or LaTeX. In addition to large OCR expert models, recent lightweight designs have also shown promising results. For example, UniRec-0.1B [51] optimizes a compact 0.1B-parameter model for text blocks and formula blocks, demonstrating competitive OCR performance under a highly lightweight setting. According to their modeling paradigm, OCR-specific VLMs can be roughly divided into modular and end-to-end approaches. Modular methods usually cascade a layout analysis model before OCR recognition: a page-level document is first decomposed into block-level regions, and each block is then parsed by an OCR VLM. Such designs can reduce the difficulty of local parsing through region-level cropping, but the overall pipeline still depends on the preceding layout analysis results and may suffer from errors in region detection, reading-order recovery, and cross-block relation modeling. In contrast, end-to-end OCR-specific models directly model page-level documents and parse the entire page image within a unified framework, with representative examples including dots.ocr [52] and DeepSeek-OCR [53]. This paradigm avoids explicit layout splitting and the associated error propagation, allowing the model to jointly capture text, tables, formulas, charts, and reading order in the full-page context. Therefore, end-to-end modeling is more beneficial for improving the native OCR capability of VLMs. We argue that an OCR-specialized VLM should not be defined only as a document parsing model but should support a broader range of OCR-related tasks, including text spotting, information extraction, document question answering, text-image translation, and multi-image understanding. HunyuanOCR [28] follows the lightweight end-to-end OCR-specific VLM paradigm, and HunyuanOCR-1.5 further extends this direction by keeping the validated architecture unchanged while improving capability boundaries through data construction, training recipe upgrades, and system-level inference acceleration.

Multi-token prediction. Autoregressive decoding [29] is a key latency bottleneck for long-output OCR scenarios such as document parsing, table reconstruction, and formula transcription. Speculative decoding accelerates generation through a draft-then-verify paradigm [54], where candidate tokens proposed by a lightweight draft model are verified by the target model while preserving the original output distribution. However, many speculative methods still rely on autoregressive drafting, so the draft cost grows with the number of proposed tokens. Recent parallel drafting methods, such as multi-head prediction [35, 55] and diffusion-based generation, aim to predict multiple future tokens simultaneously. Among them, DFlash [36] trains a block-diffusion draft model conditioned on target-model hidden states, enabling an entire candidate block to be proposed in one parallel forward pass and then verified by the target model. This makes DFlash well-suited for HunyuanOCR-1.5, where OCR-centric generation often produces long and structured outputs.

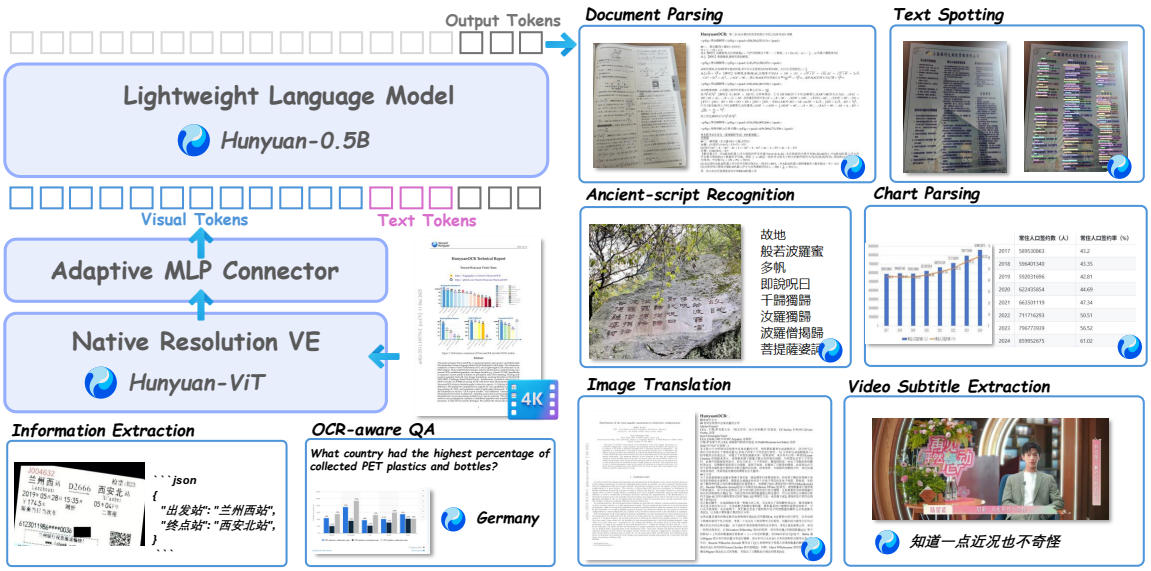

Figure 1: **Overview of the HunyuanOCR-1.5 architecture.** A compact end-to-end model that unifies diverse OCR-centric capabilities, including document parsing, text spotting, information extraction, OCR-aware QA, ancient-script recognition, chart parsing, image translation, and video subtitle extraction.

3 Model Design

3.1 Model Architecture

HunyuanOCR-1.5 follows the compact, fully end-to-end architecture of HunyuanOCR-1.0 [28], comprising a native-resolution visual encoder, an adaptive MLP connector, and a lightweight language model (Fig. 1). The pivotal upgrade in the model backbone lies in the visual encoder: built upon Hunyuan-ViT [56, 57], we extend the maximum input image resolution from 2K to 4K. This crucial scaling allows the model to preserve native aspect ratios and spatial layouts while capturing finer structural details, which is instrumental for processing highly dense documents, over-sized tables, and complex charts.

The remaining components maintain their validated lightweight configurations to ensure deployment efficiency. The adaptive MLP connector compresses high-resolution visual features into compact tokens while preserving layout sensitivity. Concurrently, the language component, a lightweight Hunyuan-0.5B model with XD-RoPE [58], processes these tokens to autoregressively generate structured OCR outputs. Through this streamlined formulation, HunyuanOCR-1.5 directly maps multi-modal inputs into diverse OCR-centric outputs (e.g., Markdown documents, HTML tables, LaTeX formulas, and chart descriptions) without relying on any task-specific post-processing modules.

3.2 Multi-token Prediction

Autoregressive decoding [29] is a major efficiency bottleneck for end-to-end OCR-centric VLMs, especially in document parsing scenarios [47–50] that require long structured outputs, such as dense tables, multi-column documents, and long formulas. Although speculative decoding [31–35] reduces latency by drafting multiple candidate tokens and verifying them with the target model, many existing methods still generate draft tokens autoregressively, making the draft cost grow with the number of candidates. To address this limitation, HunyuanOCR-1.5 adopts DFlash [36], which uses a lightweight block-diffusion draft model to predict a block of candidate tokens in one parallel forward pass. Given a block size B, the draft model proposes $\hat{\mathbf{y}}_{1:B}$ at once, and the target model verifies the block in parallel and accepts the longest valid prefix, preserving the output distribution of the target model.

During training, the HunyuanOCR-1.5 target model is frozen, and only the DFlash draft model is optimized. For each training sequence, we first run the target model once and cache its hidden states as conditional representations. We then randomly sample n anchor positions, each corresponding to an independent block-drafting task. These n blocks are concatenated and trained in a single forward pass with a FlexAttention [59]

Figure 2: **Overview of DFlash training with a joint FlexAttention mask.** One target forward is performed, K anchors are sampled at random positions, and all K blocks attend in a single pass. Rows denote Query tokens, and columns denote Key/Value tokens.

block-diagonal mask, where each block can attend to the target hidden states before its anchor and to the mask tokens within the same block, while different blocks remain isolated, as shown in Fig. 2. We use ground-truth continuation tokens as labels and optimize a position-weighted next-token cross-entropy loss:

$$w_k^{(j)} = \mathbb{I}[k > 0] \cdot \mathbb{I}[\text{valid}] \cdot \exp(-\max(k-1, 0)/\gamma)\,, \tag{1}$$

$$\mathcal{L}_{\text{DFlash}} = \frac{1}{Z} \sum_{j=1}^{n} \sum_{k=1}^{B-1} w_k^{(j)} \left[-\log p_\theta \left(y_k^{(j)} \mid \mathbf{h}_{<a_j}, \mathbf{m}_{1:B}^{(j)} \right) \right], \quad Z = \sum_{j=1}^{n} \sum_{k=1}^{B-1} w_k^{(j)}, \tag{2}$$

where a_j is the j-th anchor position, $\mathbf{h}_{<a_j}$ denotes the target hidden states before the anchor, $\mathbf{m}_{1:B}^{(j)}$ denotes the mask-token queries of the draft block, and $y_k^{(j)}$ is the ground-truth continuation token. The weight excludes the anchor token and invalid positions, while the exponential decay reduces the loss weight for farther positions that are harder to predict. In our implementation, the DFlash draft model has approximately 90.7M parameters, uses block size $B = 16$, samples $n = 16$ anchors per sequence, sets $\gamma = 7.0$, and is implemented as a 5-layer Transformer initialized from the last 5 decoder layers of the target model.

DFlash accelerates inference by trading otherwise underutilized computation for fewer autoregressive decoding steps. In single-request or low-concurrency scenarios, standard AR decoding is often memory-bandwidth-bound, leaving substantial compute idle. DFlash uses this idle compute to draft a token block in parallel, so that each target-model verification step can advance multiple tokens. This is particularly effective for OCR outputs with strong local regularity, such as HTML tables, formulas, and structured Markdown parsing results, where future tokens are more predictable and the effective accepted length is longer. As shown in Sec. 7.1, DFlash substantially improves inference speed, especially for long and structured OCR outputs.

The acceleration of DFlash [36] can be interpreted as trading idle computation for lower decoding latency. In single-request or low-concurrency inference scenarios, target-model autoregressive decoding is often constrained by memory bandwidth, leaving a considerable amount of compute underutilized. DFlash leverages this idle compute to produce a block of draft tokens through a lightweight parallel forward pass, allowing the target model to verify multiple candidate tokens within one forward pass. Therefore, the effective number of accepted tokens per target forward pass increases. Empirically, this property is particularly beneficial for OCR outputs with strong local regularity, such as tables, formulas, and structured document parsing results, where candidate tokens are more predictable, and the accepted prefix length tends to be longer. As shown in the Experiment section, the acceleration ratio increases with output length and becomes more significant for highly structured OCR generation tasks.

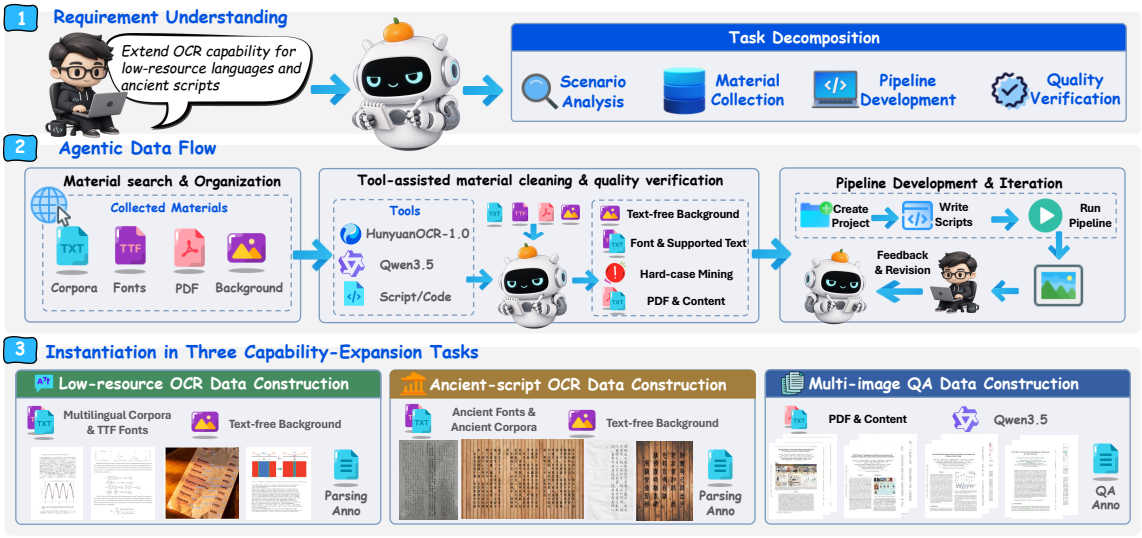

Figure 3: **Overview of *Agentic Data Flow*.** An agent-driven data construction system, instantiated in three capability-expansion tasks: low-resource OCR, ancient-script OCR, and multi-image QA.

4 Agentic Data Flow

The capability boundary extension of HunyuanOCR-1.5 is not achieved by simply scaling up data volume, but is driven by a data construction system oriented toward model weaknesses. We refer to this system as *Agentic Data Flow*. It takes concrete capability gaps, such as insufficient coverage of low-resource languages, weak perception of ancient scripts, lack of multi-image document understanding, and insufficient hard cases in complex scenarios, and converts them into executable data requirements that further drive the subsequent data production loop.

4.1 Agentic Data Flow Pipeline

We equip the agent with tool-calling structures and usage instructions, enabling it to access web search, OCR services, vision-language model services, file processing scripts, image cleaning tools, and data generation tools. Algorithm engineers provide target capability requirements to the agent in natural language, such as constructing synthetic data for low-resource OCR, generating ancient-script parsing samples, mining failure cases of HunyuanOCR-1.0, or constructing multi-image QA data. The agent then autonomously decomposes the task, determines the required materials, tool calls, scripts, and quality criteria, and continuously interacts with algorithm engineers during data production. By inspecting intermediate samples, pointing out quality issues, and adding constraints, the algorithm engineers guide the agent to iteratively refine the data pipeline, eventually forming a reusable data production workflow for the target weakness. Within this loop, as illustrated in Fig. 3, the agent mainly undertakes three types of key operations.

Material search and organization. The agent autonomously invokes web search and other tools to collect materials required for data construction. For low-resource OCR, it searches for multilingual text corpora, TTF font files, and rendering backgrounds. For ancient-script OCR, it searches for fonts related to the seven historical forms of Chinese characters, as well as backgrounds with ancient-book or historical-document styles. For multi-image QA, it mainly searches and organizes multi-page PDF documents and uses PDF tools to extract page-level text and structural information as the basic context for subsequent QA generation. Compared with manual collection, the agent can organize scattered resources into structured material directories and maintain mappings among corpora, fonts, backgrounds, PDF documents, and other resources.

Tool-assisted material cleaning and quality verification. Beyond material collection, the agent further invokes tools to clean and refine the collected resources. For image backgrounds, it can call HunyuanOCR-1.0 [28] and Qwen3.5 [60] services for automatic inspection, filtering out candidate images that contain interfering text, overly complex foreground objects, or unstable visual quality, thereby maintaining high-quality text-free backgrounds suitable for synthesis. For fonts, the agent tests the rendering compatibility of candidate TTF files for the corresponding languages or scripts and maintains the supported rendering

vocabulary of each font. For hard-case mining, the agent can run HunyuanOCR-1.0 inference on candidate images in batch and automatically maintain hard-sample sets according to parsing results such as missed recognition, structural disorder, table parsing failure, and incorrect multi-column reading order.

Weakness-oriented data pipeline development and iteration. After the materials are prepared, the agent autonomously develops data production pipelines for specific weakness topics. It creates data projects, writes rendering or QA generation scripts, organizes material paths, defines task formats, and progressively supports different layout renderings, background combinations, degradation augmentations, and output schemas. During development, the agent continuously interacts with algorithm engineers: it first generates initial demos and then performs multiple rounds of revision based on feedback regarding layout quality, visual realism, task difficulty, label format, and data diversity. As iteration proceeds, the pipeline gradually evolves from a single-template prototype into a data production system supporting multiple layouts, augmentations, and task formats.

4.2 Instantiation on Capability-Expansion Tasks

Through the above mechanism, Agentic Data Flow connects model weakness identification, material construction, data cleaning, pipeline development, and training data injection into a closed loop. In HunyuanOCR-1.5, we instantiate this system in three representative capability-expansion tasks: low-resource OCR, ancient-script OCR, and multi-image QA.

Low-resource OCR data construction. For low-resource OCR, the agent automatically collects multilingual text corpora and corresponding TTF font files from the web. Since different fonts vary significantly in character coverage, the agent tests the rendering compatibility of candidate fonts for each language and maintains a mapping among languages, fonts, and supported rendering vocabularies. Based on the design philosophy of SynthText [61] and SynthDoG [62], the agent develops a multilingual synthetic data production pipeline that renders texts from different languages onto diverse backgrounds with controllable layouts and visual styles. Through this process, we maintain parsing data covering 331 languages, providing pretraining supervision to improve the multilingual perception capability of HunyuanOCR-1.5.

Ancient-script OCR data construction. For ancient-script OCR, we focus on the seven historical forms of Chinese characters. For each historical script, the agent autonomously searches for multiple TTF font files with different rendering styles, and maintains diverse text-free background materials through both autonomous collection and tool-assisted verification. For example, when collecting background images, the agent can invoke HunyuanOCR-1.0 and Qwen3.5 services for multi-model validation, filtering out candidate images that contain interfering text or unstable visual quality. The agent then develops an ancient-script parsing data synthesis pipeline according to the writing directions, layout patterns, and visual styles of historical documents, supporting different backgrounds, fonts, layouts, and degradation augmentations. The generated data are mainly used in the pretraining stage to supplement rare historical character forms and improve the model's fundamental perception of ancient documents.

Multi-image QA data construction. For multi-image document understanding, we extend Agentic Data Flow to a QA data production process based on multi-page PDFs. The agent first collects and organizes multi-page PDF documents, and invokes PDF tools to extract page-level text and basic structural information. The extracted text is then organized into cross-page contexts according to page order and provided to a strong text model to generate multi-image QA samples, including cross-page information retrieval, multi-page content comparison, evidence aggregation, and document-level reasoning questions. To ensure that the generated data truly require multi-page understanding, we further filter out questions that can be answered from a single page, samples whose answers are inconsistent with the extracted PDF context, and questions without explicit textual evidence. This pipeline extends the capability boundary of HunyuanOCR-1.5 from single-image OCR and single-page document parsing to multi-page and multi-image document understanding.

Overall, Agentic Data Flow serves as a capability-expansion data system for HunyuanOCR-1.5. It is not limited to a specific data type, but provides a reusable data construction paradigm: defining data requirements around model weaknesses, automatically completing material search, tool-based verification, sample cleaning, pipeline development, and human-agent iteration through agents, and injecting the resulting data into subsequent training stages. This system supports the improvement of HunyuanOCR-1.5 in long-tail directions such as low-resource languages, ancient scripts, multi-image understanding, and hard-case robustness.

5 Training Recipe

The training recipe of HunyuanOCR-1.5 follows the staged training paradigm of HunyuanOCR [28], while shifting the objective from building general OCR capabilities to extending capability boundaries and improving task ceilings. Overall, we structure the training pipeline of HunyuanOCR-1.5 into three main phases: *pretraining* (Sec. 5.1), *supervised fine-tuning* (SFT, Sec. 5.2), and *reinforcement learning* (RL, Sec. 5.3).

The pretraining stage mainly injects newly constructed capability-expansion data into the model and improves its adaptation to complex inputs through resolution and context-window extension. The subsequent SFT and RL stages collaboratively focus on **capability ceiling improvement**, pushing the upper bound of each OCR task while enhancing output stability and mitigating hallucinations in document scenarios. Specifically, SFT establishes a clean and highly structured foundation by refining the training data and unifying the prompt interface. Building upon this high-quality basis, RL further pushes the capability ceilings using verifiable rewards and judge-based supervision, forming a complementary optimization pipeline.

5.1 Pretraining: Revisiting Stage3 for Capability Boundary Extension

In the pretraining stage, we do not redesign the full pretraining procedure of HunyuanOCR-1.0. Instead, we reuse its first two stages and only re-plan the third stage (Stage3). Two upgrades are applied to Stage3: (a) we inject new capability-expansion data to broaden what the model can recognize, and (b) we enlarge the input specification so that the model can handle high-resolution and long-context inputs.

Data upgrade. Stage3 now mixes three sources: the new capability data produced by *Agentic Data Flow* (Sec. 4), multi-image understanding data, and historical OCR data from HunyuanOCR-1.0. The new data target the model's weak spots, covering low-resource OCR, ancient-script OCR, multi-image document understanding, hard cases, and long-tail layouts, and thus drive capability expansion. The historical OCR data are kept to preserve existing strengths in general OCR, document parsing, and structured output. Training on both jointly lets the re-planned Stage3 expand new capabilities without regressing on old ones.

Input specification. We also raise the maximum image resolution to 4K and extend the context window to 128K. This lets the model take in far more demanding inputs, such as dense documents, multi-page and multi-image contexts, and long structured outputs. As a result, HunyuanOCR-1.5 pushes its capability boundary toward high-resolution, long-context, and multi-image scenarios while keeping the architecture unchanged.

5.2 SFT: Building a High-Quality Foundation for the RL Stage

As the first step toward capability ceiling improvement, the SFT stage prepares a clean, well-organized, and interface-consistent training set for the subsequent RL stage. This preparation consists of three parts: refining the data quality, splitting the data between SFT and RL, and unifying the prompt design across tasks.

Data refinement. We start from the post-training data of HunyuanOCR-1.0 and clean it thoroughly, removing annotation errors, format inconsistencies, image-text mismatches, ambiguous task objectives, and low-quality duplicated samples. We then enrich the data pool with the new capability data produced by Agentic Data Flow, user-provided hard cases, and high-quality data for the newly introduced capabilities, with careful manual annotation and verification for key samples. Through this process, the SFT data are upgraded from the general task coverage of HunyuanOCR-1.0 to a high-quality training set oriented toward capability ceiling improvement and robustness in complex scenarios.

Data splitting for SFT and RL. We divide the curated data into two disjoint portions. One portion is used for supervised fine-tuning in the current stage, while the other, consisting mainly of high-difficulty samples, is reserved for the subsequent RL stage. This split lets SFT establish broad task competence while keeping the most challenging samples for RL to push the capability ceiling.

Unified prompt design. Finally, we unify the prompt design across tasks, routing each task capability to its own specialized prompt. This reduces instruction ambiguity, gives each task a clear and consistent interface, and thereby provides well-defined task boundaries for the subsequent RL stage.

5.3 Reinforcement Learning

Reinforcement learning (RL) has emerged as a powerful paradigm for large language models (LLMs) and multimodal large language models (MLLMs), with success in mathematical reasoning [63] and image segmentation [64]. This is largely attributed to RL's ability to align model outputs with verifiable metrics [65]

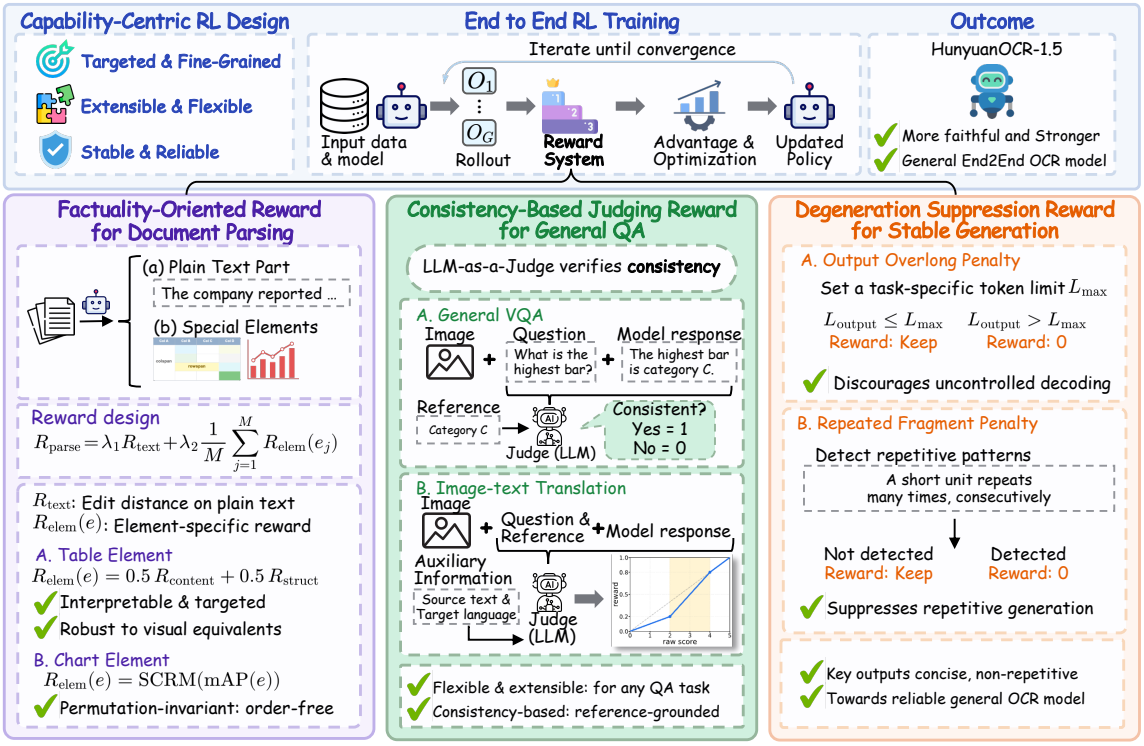

Figure 4: **Overview of the RL framework.** The RL framework that optimizes the general OCR model toward more faithful, stronger, and more comprehensive behavior through three complementary reward components.

or human preferences [66, 67]. HunyuanOCR [28] has already validated this potential in the OCR domain: through high-quality RL data and an ability-adaptive reward design, it achieves stable and effective training, showing that RL can substantially improve lightweight OCR models across diverse tasks.

HunyuanOCR-1.5 pushes this direction further. As shown in Fig. 4, we build a reward system tailored to the capabilities an OCR model must acquire, providing fine-grained, discriminative signals. Concretely, it consists of three complementary components: (i) a *capability-routed, structure-aware rule reward* that targets the factual fidelity of document parsing; (ii) a *consistency-based judging reward* for general question answering that flexibly scores arbitrary QA tasks and can be easily extended; and (iii) a *degeneration-suppression reward* that detects overlong and repetitive outputs to keep generation stable. Together, these components drive HunyuanOCR-1.5 toward a more faithful, stronger, and more comprehensive general OCR model.

5.3.1 Training Strategy

Data curation. Following HunyuanOCR [28], we curate the RL data with an emphasis on quality, diversity, and difficulty balance. Starting from the SFT policy, we perform N=16 on-policy rollouts for each candidate query and estimate its difficulty based on the rollout outcomes. Queries that are already solved consistently by the policy are discarded, retaining only informative examples with non-trivial reward variance for RL training.

Optimization. We adopt IcePop [68], a GRPO-style policy optimization variant, as our main reinforcement learning framework to mitigate the training–inference mismatch. Let π_{infer} and π_{train} denote the same policy as executed by the inference and training engines. In each iteration, for a query q, IcePop samples a group of G responses $\{o_1, o_2, \ldots, o_G\}$ from the old inference policy $\pi_{\text{infer}}(\cdot \mid q; \theta_{\text{old}})$, while the update is computed with $\pi_{\text{train}}(\cdot; \theta)$. To suppress unstable updates from train–inference discrepancies, we compute a token-level calibration ratio between the two policies and only retain tokens whose ratio lies in a prescribed interval. Since our implementation adopts a token-mean loss [69], we normalize over all valid tokens that pass the IcePop mask, instead of first averaging each response by its length $|o_i|$:

$$\mathcal{J}^{\text{tok}}_{\text{IcePop}}(\theta) = \mathbb{E}_{q,\{o_i\}_{i=1}^{G}} \left[\frac{1}{Z} \sum_{i=1}^{G} \sum_{t=1}^{|o_i|} a_{i,t} s_{i,t} \left(\mathcal{L}^{\text{PG}}_{i,t}(\theta) - \gamma\, \mathbb{D}_{\text{KL},i,t} \right) \right], \tag{3}$$

$$\mathcal{L}^{\text{PG}}_{i,t}(\theta) = c_{i,t} \min(r_{i,t}(\theta) A_i, \text{clip}(r_{i,t}(\theta), 1-\epsilon, 1+\epsilon)\, A_i), \tag{4}$$

$$Z = \sum_{i=1}^{G} \sum_{t=1}^{|o_i|} a_{i,t} s_{i,t}, \tag{5}$$

$$r_{i,t}(\theta) = \frac{\pi_{\text{train}}(o_{i,t} \mid q, o_{i,<t}; \theta)}{\pi_{\text{train}}(o_{i,t} \mid q, o_{i,<t}; \theta_{\text{old}})}, \tag{6}$$

$$c_{i,t} = \frac{\pi_{\text{train}}(o_{i,t} \mid q, o_{i,<t}; \theta_{\text{old}})}{\pi_{\text{infer}}(o_{i,t} \mid q, o_{i,<t}; \theta_{\text{old}})}, \quad s_{i,t} = \mathbf{1}[\alpha_{\text{m}} \le c_{i,t} \le \beta_{\text{m}}]\,. \tag{7}$$

Here $a_{i,t} \in \{0, 1\}$ is the valid-token mask, A_i is the group-relative advantage of response o_i, and $\mathbb{D}_{\text{KL},i,t}$ is the token-level KL term. The bounds α_{m} and β_{m} control the acceptable train–inference ratio region; tokens outside it have $s_{i,t} = 0$ and do not contribute to the update. The hyperparameters ϵ and γ control PPO-style clipping and KL strength. If no token in a mini-batch satisfies the IcePop mask, the update is skipped.

5.3.2 Factuality-Oriented Reward for Document Parsing

The first and most fundamental component targets factual fidelity, since a reliable OCR model must faithfully transcribe what is visually present rather than hallucinate plausible content. For text spotting, we follow the rule-based reward of HunyuanOCR [28]. Document parsing, however, requires a dedicated design: it converts a document image into a structured representation that may contain plain text, tables, and charts. Tables and charts are structured elements whose correctness is not well captured by the edit-distance criterion used for plain text, so a single text-level metric yields coarse and sometimes misleading signals.

To provide a fine-grained, structure-aware reward, we parse the output and reference into a plain-text part and a set of special elements, each a table or a chart. The parsing reward is then computed as:

$$R_{\text{parse}} = \lambda_1 R_{\text{text}} + \lambda_2 \frac{1}{M} \sum_{j=1}^{M} R_{\text{elem}}(e_j)\,, \tag{8}$$

where R_{text} is the text reward based on normalized edit distance, $\{e_1, \ldots, e_M\}$ are the M special elements parsed from the reference, $R_{\text{elem}}(e_j)$ is the element-specific score defined below, and λ_1, λ_2 balance the two terms. Averaging over the M special elements keeps the reward well scaled.

Table. Table content is usually expressed in HTML with structure-specific tags for rows, columns, and cell merging, so it requires a dedicated reward rather than plain text matching. TEDS and TEDS-S [70] are the standard metrics for table parsing and are natural reward candidates, but both have limitations in an RL setting. As a structural reward, TEDS-S is computed in a rather black-box manner, making it hard for the model to identify which structural part of the output is wrong and to optimize accordingly. As a content reward, TEDS relies on character-level edit distance, which produces inaccurate scores in cases that are visually equivalent but expressed differently, such as mathematical formulas. We therefore improve both terms: we replace the TEDS-S score with a 1D-probe structural reward R_{struct}, and enhance the TEDS-based content reward with an anchor-guided destylization mechanism to obtain R_{content} [71]. The score of a table element is

$$R_{\text{elem}}(e) = 0.5\, R_{\text{content}} + 0.5\, R_{\text{struct}}\,. \tag{9}$$

Chart. Chart content is typically represented in a Markdown-style table, where the row and column orderings usually do not affect correctness and even transposing rows and columns expresses the same chart. A reward that is sensitive to such orderings would penalize correct predictions, so charts also require a dedicated design. We first convert both the prediction and the reference into a tabular CSV form and then apply SCRM (Structuring Chart-oriented Representation Metric) [72] to compute the mean Average Precision (AP) between them, which serves as the score $R_{\text{elem}}(e)$ of the chart element [5]. This makes the reward invariant to order permutations while remaining sensitive to the underlying chart semantics.

5.3.3 Consistency-Based Judging Reward for General QA

Beyond factual parsing, we extend the model toward broader capabilities through a consistency-based judging reward for general question answering. Instead of designing a bespoke metric for every task, this component uses an LLM-as-a-judge to verify the consistency between the model response and a high-quality reference. Its key advantage is flexibility and extensibility: it can score arbitrary QA tasks, and can incorporate additional annotation fields to support more fine-grained evaluation of specific downstream tasks.

Visual question answering. For general VQA, the reward is binary: the judge assigns 1 if the model's answer is semantically consistent with the reference and 0 otherwise, focusing on factual correctness while tolerating minor stylistic variations. This provides a clear and robust supervision signal for answer correctness.

Translation. Translation is a representative case where additional annotation fields enable more precise judging. We provide the judge with auxiliary metadata such as the source-language text and the target-language label, and let it assign a soft score in the range $[0, 5]$ based on consistency with the reference translation. The score is normalized to $[0, 1]$ via a debiased mapping that expands the resolution of mid-range scores, making the reward more sensitive to subtle quality differences and better able to capture improvements.

5.3.4 Degeneration Suppression Reward for Stable Generation

The third component keeps generation stable by explicitly suppressing degenerate outputs. In OCR settings, degeneration typically appears as overlong outputs, repeated fragments, or cyclic generation patterns when the model encounters uncertain or out-of-distribution inputs, which severely undermines reliability in real-world deployment. To address this, we introduce two complementary penalties during training.

Overlong output penalty. For each task, we set an appropriate upper bound on the output length according to its expected format and maximal valid response length. Since excessive length is a common indicator of repeated or drifting generation, we directly assign a reward of zero to rollouts that exceed the predefined token limit, discouraging uncontrolled decoding and encouraging concise, task-aligned outputs.

Repeated fragment detection and penalty. For rollouts within the length limit, we further detect repetitive patterns, which are a common failure mode of long OCR outputs, where a short unit of at most *max_unit* tokens repeats consecutively at least *min_repeats* times at the end of the sequence. Such rollouts also receive a zero reward. This penalty suppresses repetitive generation and improves the stability of model outputs.

Overall, the three reward components form a coherent and layered design: the factuality-oriented reward secures faithful parsing of structured documents, the consistency-based judging reward extends the model toward general and open-ended capabilities, and the degeneration-suppression reward stabilizes generation throughout. Together they provide fine-grained, discriminative, and extensible optimization signals that drive HunyuanOCR-1.5 toward a more faithful, stronger, and more comprehensive general OCR model.

6 Evaluation Tree

Rather than relying on a single or isolated benchmark, HunyuanOCR-1.5 is evaluated through a capability-oriented **OCR evaluation tree**. This evaluation design mainly answers two questions: (a) whether the core OCR capabilities established in HunyuanOCR-1.0 [28], such as document parsing, general OCR-aware QA, text spotting, and information extraction, are further strengthened; and (b) whether newly introduced capabilities, including low-resource languages, ancient scripts, multi-image understanding, and faithful seen-text parsing, are effectively incorporated into the model boundary.

Along an orthogonal dimension, all evaluation sources are categorized by their origin into **open-source** and **in-house** benchmarks. Only Spotting, IE, and Video Subtitle Extraction rely on in-house and real-world production benchmarks, while all remaining dimensions are evaluated on open-source benchmarks.

6.1 OCR Capability Evaluation Tree

The evaluation tree is organized into three groups by evaluation purpose. The first group verifies that the fundamental OCR capabilities inherited from HunyuanOCR-1.0 are preserved and further strengthened. The second group examines whether the newly extended boundary capabilities, such as long-tail languages, ancient scripts, multi-image understanding, and structured element parsing, are effectively incorporated into the model. The third group targets output reliability, focusing on the model's seen-text preservation ability and its hallucination risk in long OCR sequences. The following subsection details the specific dimensions within each group and their corresponding benchmarks.

6.2 Evaluation Dimensions and Benchmark Mapping

We first organize the evaluation dimensions into several capability groups and then provide the benchmark mapping for each group. This organization highlights what types of OCR capabilities are evaluated before specifying how each capability is measured.

- **Basic OCR and document understanding.** This group evaluates the fundamental capabilities of OCR model. *End-to-end document parsing* is evaluated by OmniDocBench [3], following the latest official evaluation protocol, which measures structured parsing of mainstream printed and scanned documents, including body text, tables, formulas, and reading order. *General OCR capability* is evaluated by OCRBench [6], focusing on OCR-aware QA across scene text recognition, document question answering, information extraction, formula recognition, and chart understanding. *Text spotting* is evaluated by an in-house Spotting Benchmark, which measures text localization and recognition across document images, scene text, artistic text, handwriting, advertisements, cards and receipts, screenshots, street views, and video frames.
- **Long-tail capability expansion.** This group evaluates whether HunyuanOCR-1.5 effectively extends its capability boundary to long-tail languages and scripts. *Low-resource multilingual parsing* is evaluated by MORE [73], which covers parsing across 149 languages and focuses on low-resource languages and rare writing systems. *Ancient script recognition* is evaluated by Chronicles-OCR [4], which measures recognition ability on the seven historical forms of Chinese characters, historical documents, and ancient-script images.
- **Structured visual element parsing.** This group evaluates structured visual element parsing beyond plain text recognition. *Table parsing* is evaluated by TableVerse-5K [71], which measures table structure and content reconstruction. *Chart parsing* is evaluated by ChartArena [5], which evaluates chart text, structure, and semantic parsing.
- **Cross-page and cross-lingual understanding.** This group evaluates capabilities that go beyond single-page OCR. *Multi-image QA* is evaluated by DUDE [9], which measures multi-page document understanding, cross-page information retrieval, multi-image content comparison, and evidence aggregation. *Text image translation* is evaluated by DoTA [7] and MMTIT [8]. DoTA focuses on English document image translation into Chinese, while MMTIT evaluates multilingual text image translation from 14 non-Chinese and non-English languages into Chinese or English across multiple scenarios.
- **Application-oriented and reliability evaluation.** This group evaluates practical OCR applications and output faithfulness. *Information extraction* is evaluated by an in-house IE Benchmark [28] covering cards, receipts, forms, and related structured field extraction scenarios. *Video subtitle extraction* is evaluated by an in-house Video Subtitle Extraction Benchmark [28], measuring subtitle recognition from video frames, temporal text consistency, and robustness to dynamic backgrounds and compression noise. *Document hallucination* is evaluated by **CHAOS-Bench**, short for **Comprehensive Hallucination Assessment for OCR Sequences**. CHAOS-Bench evaluates faithfulness with a controlled WYSIWYG protocol. For each document page, we modify one character in 2 to 3 selected words in the rendered image, turning them into meaningless perturbed words. Degenerate edits and modified strings that remain dictionary-valid words are removed. Given the set of perturbed words $\mathcal{P}_i$ on page i and the model output O_i, a hit $\mathbb{1}_{\text{hit}}(w, O_i)$ is counted when word $w \in \mathcal{P}_i$ appears in O_i as a case-insensitive whole-word match. The page-level recall is computed as:

$$R_i = \frac{1}{|\mathcal{P}_i|} \sum_{w \in \mathcal{P}_i} \mathbb{1}_{\text{hit}}(w, O_i).$$

  The final score is the page-averaged recall over all N pages:

$$\text{Recall}_{\text{page}} = \frac{1}{N} \sum_{i=1}^{N} R_i.$$

  This metric directly measures whether the model preserves visually observed words when visual evidence conflicts with language priors.

Based on this evaluation tree, the experimental results in the next section are reported from two complementary perspectives. We first highlight the boundary capability evaluations that are newly introduced or strengthened in HunyuanOCR-1.5, including Chronicles-OCR, ChartArena, TableVerse-5K, DUDE, MORE, and CHAOS-Bench. We then analyze the changes on existing evaluation dimensions inherited from HunyuanOCR-1.0, including OmniDocBench, Spotting, text image translation, IE, Video Subtitle Extraction, and OCRBench. This organization keeps the capability taxonomy in the evaluation tree while making the result discussion focus on what is newly monitored and what is preserved or improved from the previous version.

Table 1: Grouped benchmark mapping for the OCR capability evaluation tree.

Capability Group	Evaluation Dimension	Benchmark	Source
Basic OCR and doc understanding	End-to-end document parsing	OmniDocBench [3]	Open-source
	General OCR-aware QA	OCRBench [6]	Open-source
	Text spotting	Spotting Benchmark	In-house
Long-tail ability expansion	Multilingual parsing	MORE [73]	Open-source
	Ancient-script recognition	Chronicles-OCR [4]	Open-source
Structured visual element parsing	Table parsing	TableVerse-5K [71]	Open-source
	Chart parsing	ChartArena [5]	Open-source
Cross-page and cross-lingual understanding	Multi-image QA	DUDE [9]	Open-source
	Text image translation	DoTA [7]	Open-source
	Text image translation	MMTIT [8]	Open-source
Practical applications and reliability	Information extraction	IE Benchmark	In-house
	Video subtitle extraction	Video Subtitle Extraction Benchmark	In-house
	Document hallucination	CHAOS-Bench	Open-source

Table 2: **Overall inference speed comparison.** Comparison between AR and DFlash decoding on OmniDocBench under batch size 1. The vLLM results are aligned with the latest 930-sample SOTA comparison.

Framework	AR Decoding			DFlash Decoding			Speedup	Effective Acc. Length
	Latency (s) ↓	TPS	Page/s	Latency (s) ↓	TPS	Page/s		
Transformers	34.850	40.9	0.029	5.474	245.7	0.183	6.37×	8.89
vLLM	3.032	466.9	0.330	1.408	1002.3	0.706	2.14×	8.36

7 Experimental Results

7.1 Inference Speed with DFlash

We evaluate the inference speed of HunyuanOCR-1.5 with standard autoregressive (AR) decoding and DFlash-accelerated decoding on OmniDocBench [3]. Unless otherwise specified, we report per-sample metrics:

$$\text{Latency} = \frac{1}{N}\sum_{i=1}^{N} t_i\,, \quad \text{Token/s} = \frac{\sum_{i=1}^{N} c_i}{\sum_{i=1}^{N} t_i}\,, \quad \text{Page/s} = \frac{N}{\sum_{i=1}^{N} t_i}\,, \tag{10}$$

where t_i and c_i denote the latency and generated tokens of the i-th sample. The speedup is computed by comparing DFlash with AR under the same metric.

Overall speed. We first compare AR decoding and DFlash decoding under single-request inference. As shown in Tab. 2, DFlash significantly accelerates HunyuanOCR-1.5 under both Transformers [74] and vLLM [40]. In vLLM, DFlash reduces the average latency from 3.032s to 1.408s, improves throughput from 466.9 token/s to 1002.3 token/s, and achieves a $2.14\times$ speedup. The gain is even larger under Transformers, whose AR baseline is closer to naive token-by-token decoding and therefore benefits more from speculative decoding. These results show that DFlash effectively reduces the decoding latency of long OCR outputs while preserving the original end-to-end generation paradigm.

Comparison with SOTA OCR systems. We further compare HunyuanOCR-1.5 with DFlash against representative OCR systems, including two-stage pipeline methods and end-to-end OCR VLMs. The evaluation is conducted on the same OmniDocBench test set under single-request inference, where each system is assigned one accelerator instance with comparable compute capacity. For two-stage systems, GLM-OCR [75] and PaddleOCR-VL-1.6 [76], we report the full page-level pipeline latency, including layout analysis, region-level OCR/VLM inference, and result merging. Since different systems use different tokenizers, prompts, and output formats, cross-model token/s is not directly comparable; we mainly compare average latency and page throughput.

As shown in Tab. 3, HunyuanOCR-1.5 with DFlash achieves the fastest end-to-end inference speed among all evaluated systems, reaching 1.408s per page and 0.706 page/s. It is about $1.17\times$ faster than GLM-OCR and $1.24\times$ faster than PaddleOCR-VL-1.6, while keeping a unified end-to-end OCR VLM formulation without explicit layout decomposition or region-wise cascaded inference. Compared with other OCR VLMs, including

Table 3: **End-to-end speed comparison with representative OCR systems.** Evaluated on the OmniDocBench test set. GLM-OCR and PaddleOCR-VL 1.6 are two-stage multi-model pipeline methods, while others are single-model end-to-end VLMs. Speedup is measured against the HunyuanOCR-1.5 AR setting.

Model	Paradigm	Inference Method	Avg. Latency / Page (s) ↓	Page/s ↑	Speedup
dots.ocr [52]	End-to-end	Auto-regressive	7.154	0.136	0.41×
DeepSeek-OCR 2 [78]	End-to-end	Auto-regressive	5.460	0.179	0.54×
Unlimited-OCR [77]	End-to-end	Auto-regressive	3.659	0.255	0.77×
HunyuanOCR-1.5	End-to-end	Auto-regressive	3.032	0.330	1.00×
PaddleOCR-VL-1.6 [76]	Two-stage	Cascade	1.744	0.562	1.71×
GLM-OCR [75]	Two-stage	Cascade	1.649	0.604	1.83×
HunyuanOCR-1.5	End-to-end	DFlash	**1.408**	**0.706**	**2.14**×

Table 4: **Inference speed comparison across different output length ranges.** Output length is measured by AR completion tokens. Effective acceptance length denotes the average number of tokens advanced per speculative decoding step, including the bonus token.

Framework	Output Length	AR Decoding			DFlash Decoding			Speedup	Effective Acc. Length
		Latency (s) ↓	TPS	Page/s	Latency (s) ↓	TPS	Page/s		
Transformers	[0, 256]	7.298	28.1	0.137	1.602	126.1	0.624	4.56×	8.43
	(256, 512]	11.014	35.7	0.091	2.069	191.6	0.483	5.32×	8.29
	(512, 1024]	19.932	38.3	0.050	3.448	231.3	0.290	5.78×	9.15
	(1024, 2048]	34.905	40.8	0.029	5.294	276.8	0.189	6.59×	9.26
	$(2048, +\infty)$	93.756	42.9	0.011	14.054	239.2	0.071	6.67×	8.34
vLLM	[0, 256]	0.950	217.9	1.052	0.723	286.4	1.383	1.31×	9.23
	(256, 512]	1.156	341.8	0.865	0.746	529.5	1.340	1.55×	8.39
	(512, 1024]	1.926	395.9	0.519	1.086	702.4	0.921	1.77×	9.38
	(1024, 2048]	3.071	466.5	0.326	1.435	998.6	0.697	2.14×	9.50
	$(2048, +\infty)$	6.660	514.3	0.150	2.901	1183.0	0.345	2.30×	8.65

Unlimited-OCR [77], DeepSeek-OCR-2 [78], and dots.ocr [52], HunyuanOCR-1.5 with DFlash reduces average latency by 2.60×, 3.88×, and 5.08×, respectively.

Speedup versus output length. We analyze DFlash speedup across different output length ranges in Tab. 4. The speedup consistently increases as the output sequence becomes longer. In vLLM, DFlash improves from 1.31× on 0–256 token outputs to 2.30× on 2048+ token outputs; in Transformers, the speedup increases from 4.56× to 6.67×. This matches the nature of speculative decoding: longer outputs require more decoding steps, so each parallel draft-and-verify iteration amortizes more target-model forward passes. Short outputs are instead dominated by prefill and fixed overheads, limiting the attainable speedup.

Speedup by content type. We further categorize pages into text, formula, and table pages according to the official OmniDocBench layout annotations [3]. As shown in Tab. 5, both Transformers and vLLM show the same trend: table pages obtain the largest speedup, followed by formula pages and text pages. This is because table outputs usually contain highly regular HTML structures, making future tokens easier to predict and yielding longer effective accepted prefixes.

Throughput under concurrency. Finally, we evaluate DFlash under different vLLM concurrency levels. As shown in Tab. 6, system throughput increases as concurrency grows, indicating improved GPU utilization from continuous batching. DFlash maintains more than 1.8× speedup from concurrency 1 to 32, with the highest speedup of 2.26× at concurrency 4. The speedup gradually decreases at higher concurrency because the GPU becomes increasingly saturated, leaving less idle compute for speculative decoding.

7.2 Boundary Capability Evaluation

We first focus on the boundary capabilities of HunyuanOCR-1.5. Here, boundary capabilities include both newly introduced task abilities, such as ancient-script parsing, document-level multi-image QA, and output faithfulness evaluation, and previously supported abilities that are further monitored and strengthened with more fine-grained benchmarks, such as complex table parsing and structured chart parsing. Through Chronicles-OCR, ChartArena, TableVerse-5K, DUDE, MORE, and CHAOS-Bench, this section character-

Table 5: **Inference speed comparison across different content types.** We categorize the OmniDocBench pages into text, formula, and table types, and report the speed on each. The effective acceptance length denotes the average number of tokens advanced per speculative decoding step, including the bonus token.

Framework	Content Type	AR Decoding			DFlash Decoding			Speedup	Effective Acc. Length
		Latency (s) ↓	TPS	Page/s	Latency (s) ↓	TPS	Page/s		
Transformers	Text	36.357	40.6	0.028	6.455	206.8	0.155	5.63×	7.68
	Formula	36.753	41.3	0.027	5.987	237.8	0.167	6.14×	8.59
	Table	32.253	41.0	0.031	4.129	319.3	0.242	7.81×	10.40
vLLM	Text	3.034	450.8	0.330	1.675	818.4	0.597	1.81×	8.02
	Formula	2.626	464.7	0.381	1.277	955.4	0.783	2.06×	9.04
	Table	2.881	465.1	0.347	1.207	1110.3	0.829	2.39×	10.45

Table 6: **vLLM throughput comparison under different concurrency levels.** We report the throughput of AR and DFlash decoding as the concurrency level c increases. The $c = 1$ row is aligned with the latest 930-sample SOTA speed comparison, while higher-concurrency rows follow the concurrency sweep results.

Concurrency	AR Latency (s) ↓	AR TPS	AR Page/s	DFlash Latency (s) ↓	DFlash TPS	DFlash Page/s	Speedup
$c = 1$	3.032	466.9	0.330	1.408	1002.3	0.706	2.14×
$c = 2$	3.761 / 2	707.4	0.532	1.785 / 2	1493.5	1.121	2.11×
$c = 4$	5.915 / 4	900.0	0.676	2.615 / 4	2039.7	1.529	2.26×
$c = 6$	7.625 / 6	1047.5	0.787	3.526 / 6	2262.8	1.702	2.16×
$c = 8$	9.433 / 8	1127.6	0.848	4.452 / 8	2390.7	1.797	2.12×
$c = 16$	15.657 / 16	1360.9	1.022	8.395 / 16	2539.8	1.906	1.87×
$c = 32$	29.138 / 32	1462.6	1.098	16.162 / 32	2633.9	1.980	1.80×

izes the capability boundary of HunyuanOCR-1.5 from long-tail scripts, complex structures, multi-image understanding, and reliability perspectives.

Chronicles-OCR. Chronicles-OCR evaluates ancient-script parsing, which is one of the key directions strengthened in HunyuanOCR-1.5. As shown in Tab. 7, HunyuanOCR-1.5 achieves SOTA performance within a 1B model, demonstrating substantially improved recognition ability on the seven historical forms of Chinese characters, historical documents, and ancient-script images. This result verifies that the data construction and training strategy for ancient scripts effectively improve the model's perception of historical glyphs.

ChartArena. ChartArena provides a fine-grained evaluation of structured chart parsing. While the HunyuanOCR series already had basic chart parsing ability, HunyuanOCR-1.5 further improves its parsing of chart text, legends, axes, visual element relations, and chart semantics. As shown in Tab. 8, HunyuanOCR-1.5 reaches a performance level comparable to 8B-scale models with only a 1B model, indicating strong capability in structured chart understanding and semantic recovery.

TableVerse-5K. TableVerse-5K evaluates table element parsing in complex table scenarios. Compared with table subsets in general document parsing benchmarks, this benchmark focuses more on table structure, cell content, row-column relations, and table reconstruction under complex layouts. As shown in Tab. 9, HunyuanOCR-1.5 achieves the best performance among expert OCR models on TableVerse-5K, showing that the model further enhances its capability in table structure parsing and complex table reconstruction.

DUDE. DUDE evaluates document-level multi-image QA, where the model needs to retrieve information, aggregate evidence, and answer questions across multiple pages or images. This task goes beyond conventional single-image OCR and single-page document parsing, and is used here to examine whether an OCR-specialized VLM can extend toward document-level multi-image understanding. HunyuanOCR-1.5 achieves 54.64 on the DUDE validation set, which is close to the 56.41 result of the general multimodal model Qwen3.5-0.8B. This result indicates that, after multi-image data construction and training adaptation, HunyuanOCR-1.5 has acquired a certain degree of document-level multi-image QA capability and reaches a comparable level to a general-purpose VLM in this setting.

MORE. MORE evaluates low-resource multilingual parsing across 149 languages, focusing on low-resource languages and long-tail writing systems. As shown in Tab. 10, HunyuanOCR-1.5 achieves SOTA performance among OCR expert models on MORE. This result demonstrates that the low-resource language data produced by Agentic Data Flow effectively improves multilingual perception, and further validates the value of systematic data construction for low-resource OCR capability expansion.

Table 7: **Comparison of ancient-script OCR results on Chronicles-OCR.** We report the average Parsing scores on archaic scripts (Oracle Bone, Bronze, Seal) and mature scripts (Clerical, Regular, Running, Cursive).

Model Type	Model	Size	Think-mode	Archaic Average	Mature Average
Open-source General VLMs	InternVL3.5-8B [27]	8B		0.07	0.39
	InternVL3.5-A28B [27]	241B-A28B		0.13	0.56
	Qwen3-VL-8B [20]	8B		0.18	0.65
	Qwen3-VL-A22B [20]	235B-A22B		0.19	0.66
	Qwen3.5-9B [60]	9B		0.09	0.60
	Qwen3.5-A17B [60]	397B-A17B		0.22	<u>0.73</u>
	Gemma 4 31B it [79]	31B		0.04	0.35
	MiniCPM-V 4.5 [80]	8B	✓	0.03	0.40
	Ovis2.6-30B-A3B [81]	30B-A3B	✓	0.11	0.51
	GLM-4.5V [82]	108B-A12B	✓	0.06	0.43
	Kimi K2.5 [83]	1T		<u>0.28</u>	0.71
Proprietary General VLMs	GPT-5 [84]	-		0.06	0.41
	Seed1.8 [85]	-		0.21	0.67
	Seed2.0 Pro [86]	-		0.18	0.71
	Seed2.0 Pro [86]	-	✓	0.26	0.72
	MiMo-V2-Omni [87]	-	✓	0.09	0.55
	Gemini 2.5 Pro [14]	-	✓	0.08	0.52
	Gemini 3.1 Pro [16]	-	✓	0.18	0.68
	Claude Opus 4.7 [88]	-	✓	0.10	0.50
Expert OCR Models	DeepSeek-OCR [53]	3B-A0.5B		0.01	0.24
	dots.ocr [52]	3B		0.05	0.47
	GLM-OCR [75]	9B		0.06	0.38
	PaddleOCR-VL [45]	0.9B		0.05	0.41
	Unlimited-OCR [77]	3B-A0.5B		0.01	0.21
	HunyuanOCR-1.5	1B		**0.54**	**0.79**

Table 8: **Comparison of chart deplotting results on ChartArena.** We report mAP_{high} per chart type and the overall average, with separate *EN* (English) and *ZH* (Chinese) scores, each averaged over three visual styles.

Model Type	Model	bar		line		pie		radar		box plot		comb.		flowchart		mind map		Average	
		EN	ZH	EN	ZH	EN	ZH	EN	ZH	EN	ZH	EN	ZH	EN	ZH	EN	ZH	EN	ZH
General Purpose VLMs	Qwen2.5-VL-7B-Ins. [19]	15.2	36.9	17.9	39.9	63.4	73.1	8.3	19.1	0.9	2.8	6.0	40.6	29.7	23.2	45.4	29.9	23.3	33.2
	InternVL3.5-8B [27]	22.7	52.6	34.4	53.7	65.8	73.8	14.0	34.7	5.6	9.5	11.3	42.1	32.6	23.8	48.3	31.8	29.3	40.2
	Qwen3-VL-8B-Ins. [20]	27.5	58.6	35.5	<u>61.1</u>	77.3	<u>84.7</u>	16.8	42.6	11.6	12.1	13.2	47.9	<u>50.0</u>	<u>41.5</u>	<u>66.4</u>	<u>54.6</u>	37.3	50.4
	Qwen3.5-9B [60]	32.5	45.1	<u>45.5</u>	54.1	**82.6**	76.9	22.0	44.8	15.3	18.1	16.8	49.5	45.5	38.1	64.2	54.5	<u>40.6</u>	47.7
Expert Chart Deplotting Models	ChartAst (13B) [89]	5.2	–	4.2	–	0.3	–	1.5	–	0.3	–	0.0	–	–	–	–	–	1.4	–
	ChartVLM (8.3B) [90]	11.2	5.3	11.5	4.3	12.9	8.2	2.1	5.0	0.7	0.4	4.1	4.4	–	–	–	–	5.3	3.5
	TinyChart (3B) [91]	6.1	6.3	9.7	3.2	5.7	5.4	0.5	3.4	0.2	1.3	0.7	4.2	–	–	–	–	2.9	3.0
	ChartMoE (8B) [92]	18.7	24.4	14.7	22.3	15.0	48.5	3.7	16.1	2.7	1.6	5.1	19.5	4.0	–	4.1	–	8.5	16.7
	ChartCoder (7B) [93]	23.2	12.6	22.0	19.6	34.3	16.7	5.5	13.9	5.4	11.4	3.7	5.1	5.6	–	1.0	–	12.6	9.9
	RRVF (7B) [94]	<u>35.8</u>	<u>66.5</u>	41.5	54.3	51.6	75.3	16.6	40.3	14.7	14.1	23.5	<u>61.2</u>	36.4	32.4	**68.4**	**63.8**	36.0	<u>51.0</u>
	MSRL (7B) [95]	32.7	45.2	35.2	34.3	41.2	67.9	**25.9**	<u>48.0</u>	11.2	13.0	16.7	35.2	23.2	12.4	31.0	18.8	27.1	34.3
Expert OCR Models	dots.mocr (3B) [96]	28.3	40.9	41.8	60.1	68.8	78.3	20.3	43.1	<u>24.1</u>	16.0	**26.9**	47.1	26.2	20.6	28.7	19.6	33.1	40.7
	PaddleOCR-VL (1B) [45]	31.8	49.3	43.0	51.6	57.5	75.2	14.4	29.0	11.7	<u>20.7</u>	21.3	54.0	–	–	–	–	23.9	35.8
	HunyuanOCR-1.5	**47.4**	**73.9**	**59.6**	**73.4**	<u>79.7</u>	**91.5**	<u>23.0</u>	**50.5**	**52.1**	**64.3**	<u>24.8</u>	**61.7**	**67.8**	**64.2**	36.5	33.3	**48.9**	**64.1**

CHAOS-Bench. CHAOS-Bench is introduced in this work to evaluate output faithfulness and the model's adherence to the seen-text principle. It modifies characters in selected words from academic paper images to create meaningless words, and then checks whether the model preserves these visually observed meaningless words in its parsed output. As shown in Tab. 11, HunyuanOCR-1.5 achieves the best result among compared models, with a page-average recall of 14.15. However, the absolute recall remains low, indicating that faithfully preserving visually observed but semantically invalid text is still a challenging problem for current OCR-centric VLMs. This result suggests that HunyuanOCR-1.5 is less biased toward language priors than existing models, but also highlights the need for further research on hallucination suppression and seen-text faithful generation.

Overall, these boundary capability evaluations show that the improvements of HunyuanOCR-1.5 are reflected not only in conventional OCR metrics, but also in more fine-grained boundary scenarios and reliability-oriented

Table 9: **Comparison of table parsing results on TableVerse-5K.** We report TEDS and TEDS-S scores for table structure and content reconstruction.

Model Type	Model	Size	Release Date	TableVerse-5K	
				TEDS	TEDS-S
Specialized Table Parsing models	UniTable [97]	125M	2024.03	48.55	78.65
	TRivia-3B [98]	3B	2025.12	78.15	85.41
General VLMs	GPT-4o [11]	-	2024.05	63.62	76.41
	GPT-5 [84]	-	2025.08	67.04	78.96
	Qwen2.5-VL-72B-Ins. [19]	72B	2025.02	75.23	82.65
	InternVL3.5-A28B [27]	241B-A28B	2025.08	76.08	84.96
	Qwen3-VL-A22B-Ins. [20]	235B-A22B	2025.10	78.26	84.23
	Seed-1.8 (no-think) [85]	-	2025.12	**79.91**	86.03
	Kimi K2.5 (no-think) [83]	1T	2026.02	78.75	<u>86.95</u>
	Gemini 2.5 Pro [14]	-	2025.03	<u>79.46</u>	**87.13**
Expert OCR Models	MonkeyOCR-pro-1.2B [99]	1.2B	2025.07	67.98	72.91
	MonkeyOCR-pro-3B [99]	3B	2025.07	72.26	77.04
	DeepSeek-OCR [53]	3B-A0.5B	2025.10	68.70	76.84
	POINTS-Reader [100]	3B	2025.08	72.03	81.13
	FD-RL [101]	-	2025.11	74.31	80.51
	dots.ocr [52]	3B	2025.07	73.39	81.84
	PaddleOCR-VL [45]	0.9B	2025.10	77.55	84.08
	MinerU 2.5 [102]	1.2B	2025.09	77.41	84.31
	HunyuanOCR-1.5	1.0B	2026.07	78.23	84.84

Table 10: **Results on the MORE benchmark.** We evaluate low-resource multilingual parsing across text, formula, table, code, catalog, and reading-order dimensions.

Model Type	Model	Size	Overall↑	Text↑	Formula↑	Table↑	Code↑	Catalog↑	Reading Order↑
General Purpose VLMs	Qwen3-VL [20]	2B	83.56	92.02	65.45	65.21	92.38	93.76	92.53
	Qwen2.5-VL [19]	3B	83.93	89.36	84.48	68.27	86.69	92.54	82.23
	Gemini 3 [15]	-	91.61	**95.39**	90.27	81.02	93.05	**94.31**	95.63
Expert OCR Models	MinerU 2.5 [102]	1.2B	48.85	27.12	73.29	33.83	72.41	21.61	64.81
	DeepSeek-OCR [53]	3B-A570M	82.91	85.27	75.67	61.63	92.26	88.26	94.36
	dots.ocr [52]	3B	84.31	94.45	90.77	39.81	95.38	88.26	**97.18**
	Unlimited-OCR [77]	-	84.90	86.75	**92.22**	50.89	97.45	85.95	96.17
	GLM-OCR [75]	-	85.75	87.31	89.29	**82.48**	95.83	67.12	92.48
	PaddleOCR-VL [45]	0.9B	87.96	90.99	91.11	61.11	96.29	93.04	95.19
	PaddleOCR-VL-1.6 [76]	0.9B	89.88	90.28	89.16	76.46	97.47	92.86	93.05
	HunyuanOCR-1.5	1.0B	**91.90**	91.31	91.10	80.77	**99.10**	92.66	96.48

evaluations. Results on ancient scripts, low-resource languages, complex tables, structured charts, multi-image QA, and output faithfulness collectively demonstrate that Agentic Data Flow, the upgraded training recipe, and post-training optimization effectively promote the systematic improvement of HunyuanOCR-1.5 in both strengthened existing abilities and newly expanded capability boundaries.

7.3 Updates on Existing Benchmarks

After evaluating boundary capabilities, we further analyze the performance of HunyuanOCR-1.5 on existing evaluation dimensions already covered by HunyuanOCR-1.0. This part focuses on whether HunyuanOCR-1.5 can further improve or stably maintain its core OCR abilities, including end-to-end document parsing, text spotting, text image translation, information extraction, video subtitle extraction, and general OCR-aware QA.

OmniDocBench. For end-to-end document parsing, HunyuanOCR-1.5 achieves an Overall score of 94.74 on OmniDocBench v1.6, reaching the SOTA performance among end-to-end OCR expert models, as shown in Tab. 12. This result shows that HunyuanOCR-1.5 further improves full-page document parsing while preserving the lightweight end-to-end architecture, with strong performance on structured parsing dimensions such as text, tables, and reading order. It is worth noting that HunyuanOCR-series models tend to parse

Table 11: **Results on CHAOS-Bench.** We report the page-average recall of perturbed seen-text words, measuring output faithfulness under conflicts between visual evidence and language priors.

Model	Size	Page-avg Recall↑
dots.ocr [52]	3B	3.02
GLM-OCR [75]	-	5.75
PaddleOCR-VL-1.6 [76]	0.9B	5.95
DeepSeek-OCR 2 [78]	3B	6.33
MinerU2.5Pro [103]	1.2B	6.33
HunyuanOCR-1.5	1B	**14.15**

Table 12: **Comparison of document parsing results on OmniDocBench v1.6.** We report the overall score together with per-dimension metrics on text, formula, table, and reading order.

Model Type	Model	Size	Overall↑	Text^{Edit}↓	Formula^{CDM}↑	Table^{TEDS}↑	Table^{TEDS_S}↑	Order^{Edit}↓
General Purpose VLMs	InternVL3.5-241B [27]	241B	83.76	0.130	89.95	74.35	79.78	0.215
	Kimi K2.5 [83]	1T	84.53	0.107	83.50	80.76	84.00	0.211
	GPT-5.2 [84]	-	86.59	0.114	88.21	82.95	87.93	0.193
	Qwen3-VL-235B [20]	235B-A22B	89.78	0.063	92.55	83.07	86.75	0.166
	Gemini 3 Flash [15]	-	92.62	0.066	95.16	89.29	93.51	0.172
	Gemini 3 Pro [15]	-	92.91	0.064	**95.99**	89.15	92.96	0.165
	Ovis2.6-30B-A3B [81]	30B-A3B	93.70	**0.035**	95.17	89.44	92.40	0.135
End2End Expert OCR Models	Mistral OCR [104]	-	85.66	0.097	89.91	76.78	80.93	0.171
	olmOCR [105]	7B	85.74	0.139	88.10	83.00	87.17	0.216
	OCRVerse [106]	4B	88.60	0.063	89.61	82.44	86.27	0.163
	DeepSeek-OCR 2 [78]	3B	90.25	0.050	91.84	83.89	87.75	0.144
	dots.ocr [52]	3B	90.77	0.048	89.95	87.18	90.58	0.138
	HunyuanOCR [28]	1B	92.03	0.048	88.60	92.37	93.99	0.138
	FireRed-OCR [107]	2B	93.26	0.037	95.44	88.04	91.06	0.131
	ABot-OCR [108]	2B	93.30	0.037	94.86	88.69	91.87	0.137
	Logics-Parsing-v2 [49]	4B	93.33	0.041	95.65	88.42	91.98	0.137
	Qianfan-OCR [109]	4B	93.90	0.040	95.08	90.53	93.31	0.130
	Unlimited-OCR [77]	3B-A0.5B	93.92	0.042	95.79	90.16	93.32	**0.129**
	HunyuanOCR-1.5	1.0B	**94.74**	0.039	94.50	**93.67**	**94.71**	**0.129**

Table 13: **Comprehensive evaluation of text spotting ability.** We report the overall score and per-scenario results across diverse image domains.

Model Type	Model	Overall	Art	Doc	Game	Hand	Ads	Receipt	Screen	Scene	Video
Traditional Methods	PaddleOCR [110]	53.38	32.83	70.23	51.59	56.39	57.38	50.59	63.38	44.68	53.35
	BaiduOCR [111]	61.90	38.5	78.95	59.24	59.06	66.70	63.66	68.18	55.53	67.38
General Purpose VLMs	Gemini 2.5 Pro [14]	23.44	21.79	35.16	10.02	38.49	29.89	20.80	17.59	18.33	18.90
	Qwen3-VL-2B-Ins. [20]	29.68	29.43	19.37	20.85	50.57	35.14	24.42	12.13	34.90	40.10
	Qwen3-VL-235B-A22B-Ins. [20]	53.62	46.15	43.78	48.00	68.90	64.01	47.53	45.91	54.56	63.79
	Seed-2.0-Vision [86]	56.32	44.77	45.85	61.70	66.89	61.87	55.73	52.05	46.53	71.49
	Gemini 3.1 Pro [16]	59.53	46.83	54.89	62.62	63.37	63.96	54.53	64.29	55.30	70.02
	Qwen3.5 A17B [60]	59.76	44.92	52.56	58.16	71.54	67.42	55.98	62.58	56.11	68.56
OCR Models	PaddleOCR-VL-1.6 [76]	61.95	41.36	72.20	58.56	70.61	65.24	61.85	63.63	54.60	69.52
	HunyuanOCR [28]	70.92	**56.76**	73.63	73.54	77.10	**75.34**	63.51	**76.58**	64.56	**77.31**
	HunyuanOCR-1.5	**71.40**	53.21	**79.43**	**75.84**	**78.40**	75.03	**65.22**	74.51	**65.12**	76.09

multi-line formulas in a unified manner, using begin/end-style LaTeX syntax to represent the complete formula. However, the current OmniDocBench matching protocol splits independent multi-line formulas into single-line units before matching. This GT matching strategy is not fully aligned with complete multi-line formula outputs from end-to-end models, and may underestimate their actual formula parsing capability. This observation suggests that the evaluation protocol for long and multi-line formulas still has room for further refinement.

Table 14: **Evaluation of text-image translation.** We report results on MMTIT (other-to-English and other-to-Chinese) and DoTA (English-to-Chinese) to evaluate the performance of text-image translation models.

Model	Size	MMTIT		DoTA
		other2en	other2zh	en2zh
Qwen3-VL-8B-Instruct [20]	8B	75.09	75.63	79.86
Qwen3-VL-4B-Instruct [20]	4B	70.38	70.29	78.45
Qwen3-VL-2B-Instruct [20]	2B	66.30	66.77	73.49
PP-DocTranslation	-	52.63	52.43	82.09
HunyuanOCR [28]	1B	73.38	73.62	83.48
HunyuanOCR-1.5	1B	**76.51**	**76.01**	**83.69**

Table 15: **Evaluation of information extraction (IE) and visual question answering (VQA).** We report results on cards, receipts, and video subtitles, together with the general OCRBench score.

Model	IE			OCRBench
	Cards	Receipts	Video Subtitles	Acc.
DeepSeek-OCR [53]	10.04	40.54	5.41	430
PP-ChatOCR [112]	57.02	50.26	3.1	-
Qwen3-VL-2B-Instruct [20]	67.62	64.62	3.75	858
Seed-1.6-Vision [113]	70.12	67.5	60.45	881
Qwen3-VL-235B-A22B-Instruct [20]	75.59	78.4	50.74	**920**
Gemini 2.5 Pro [14]	80.59	80.66	53.65	872
HunyuanOCR [28]	92.29	92.53	92.87	860
HunyuanOCR-1.5	**92.40**	**92.55**	**93.07**	861

Spotting Benchmark. For text spotting, HunyuanOCR-1.5 further improves over HunyuanOCR-1.0 on the in-house Spotting Benchmark, as shown in Tab. 13. In addition to regular text localization and recognition, HunyuanOCR-1.5 introduces negative-sample handling: when an input image contains no text, the model avoids producing hallucinated detection boxes and instead returns that no text is present. On an internal negative set of 1,000 text-free images, HunyuanOCR-1.5 achieves a no-text handling accuracy of 99.8%, substantially outperforming HunyuanOCR-1.0 at 78.1%. This ability is important for real-world OCR systems, where inputs do not always contain valid textual content.

Text Image Translation. We continue to monitor text image translation with DoTA and MMTIT, as shown in Tab. 14. DoTA mainly evaluates English-to-Chinese translation for printed document images, where HunyuanOCR-1.5 preserves a capability level close to HunyuanOCR-1.0, indicating no clear degradation on the existing document translation setting. In contrast, MMTIT covers more languages and more diverse visual scenarios. Under this more challenging multilingual and multi-scenario setting, HunyuanOCR-1.5 is further optimized to improve its adaptability to multilingual text image translation.

IE, Video Subtitle Extraction, and OCRBench. For information extraction, video subtitle extraction, and OCRBench, HunyuanOCR-1.5 largely maintains the capabilities established by HunyuanOCR-1.0, as shown in Tab. 15. Information extraction and video subtitle extraction correspond to practical OCR applications such as structured field extraction and subtitle recognition from video frames, while OCRBench monitors general OCR-aware QA ability. These results indicate that HunyuanOCR-1.5 expands its boundary capabilities without sacrificing its existing core OCR abilities.

Overall, HunyuanOCR-1.5 shows further improvements on end-to-end document parsing and text spotting, preserves its printed-document translation capability while improving multilingual and multi-scenario translation adaptability, and maintains strong performance on information extraction, video subtitle extraction, and general OCR-aware QA. These results demonstrate that the capability boundary expansion of HunyuanOCR-1.5 is achieved without compromising the practical OCR abilities established in HunyuanOCR-1.0.

8 Conclusion and Future Work

We present HunyuanOCR-1.5, a lightweight end-to-end OCR-specialized VLM that advances HunyuanOCR-1.0 toward two goals: *faster* inference and *broader* OCR capabilities. Without redesigning the validated backbone, HunyuanOCR-1.5 integrates DFlash speculative decoding for long structured OCR generation, achieving substantial speedups under both Transformers and vLLM while also supporting PC-side deployment via llama.cpp. Meanwhile, Agentic Data Flow, together with upgraded pretraining and post-training recipes, extends the model toward 4K-resolution perception, 128K-context understanding, multi-image QA, low-resource multilingual OCR, ancient-script recognition, chart/table parsing, and more faithful document generation. Through a capability-oriented evaluation tree, HunyuanOCR-1.5 demonstrates top-tier end-to-end document parsing performance, strong long-tail capability gains, and leading inference efficiency among compared OCR systems. We will release the model weights and training code to support reproducible research, user-side fine-tuning, and real-world deployment. Future work will further reduce high-resolution visual token redundancy, expand Agentic Data Flow toward continuous data-model co-evolution, and improve reliability for long and visually complex OCR generation.

References

[1] Shangbang Long, Xin He, and Cong Yao. Scene text detection and recognition: The deep learning era. *International Journal of Computer Vision*, 129(1):161–184, 2021.

[2] Qintong Zhang, Bin Wang, Victor Shea-Jay Huang, Junyuan Zhang, Zhengren Wang, Hao Liang, Conghui He, and Wentao Zhang. Document parsing unveiled: Techniques, challenges, and prospects for structured information extraction. *arXiv preprint arXiv:2410.21169*, 2024.

[3] Linke Ouyang, Yuan Qu, Hongbin Zhou, Jiawei Zhu, Rui Zhang, Qunshu Lin, Bin Wang, Zhiyuan Zhao, Man Jiang, Xiaomeng Zhao, et al. OmniDocBench: Benchmarking diverse PDF document parsing with comprehensive annotations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2025.

[4] Gengluo Li, Shangpin Peng, Xingyu Wan, Chengquan Zhang, Hao Feng, Xin Xu, Pian Wu, Bang Li, Zengmao Ding, Yongge Liu, et al. Chronicles-OCR: A cross-temporal perception benchmark for the evolutionary trajectory of chinese characters. *arXiv preprint arXiv:2605.11960*, 2026.

[5] Shangpin Peng, Gengluo Li, Xingyu Wan, Chengquan Zhang, Hao Feng, Binghong Wu, Huawen Shen, Weinong Wang, Ziyi Cai, Zhuotao Tian, Han Hu, Can Ma, and Yu Zhou. ChartArena: Benchmarking chart parsing across languages, scenarios, and formats. *arXiv preprint arXiv:2606.01348*, 2026.

[6] Yuliang Liu, Zhang Li, Mingxin Huang, Biao Yang, Wenwen Yu, Chunyuan Li, Xu-Cheng Yin, Cheng-Lin Liu, Lianwen Jin, and Xiang Bai. OCRBench: on the hidden mystery of OCR in large multimodal models. *Science China Information Sciences*, 67(12), 2024.

[7] Yupu Liang, Yaping Zhang, Cong Ma, Zhiyang Zhang, Yang Zhao, Lu Xiang, Chengqing Zong, and Yu Zhou. Document image machine translation with dynamic multi-pre-trained models assembling. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 7084–7095, 2024.

[8] Gengluo Li, Chengquan Zhang, Yupu Liang, Huawen Shen, Yaping Zhang, Pengyuan Lyu, Weinong Wang, Xingyu Wan, Gangyan Zeng, Han Hu, et al. MMTIT-Bench: A multilingual and multi-scenario benchmark with cognition-perception-reasoning guided text-image machine translation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 16593–16602, 2026.

[9] Jordy Van Landeghem, Rubèn Tito, Łukasz Borchmann, Michał Pietruszka, Pawel Joziak, Rafal Powalski, Dawid Jurkiewicz, Mickaël Coustaty, Bertrand Anckaert, Ernest Valveny, et al. Document understanding dataset and evaluation (DUDE). In *Proceedings of the IEEE International Conference on Computer Vision*, pages 19528–19540, 2023.

[10] Yubo Ma, Yuhang Zang, Liangyu Chen, Meiqi Chen, Yizhu Jiao, Xinze Li, Xinyuan Lu, Ziyu Liu, Yan Ma, Xiaoyi Dong, et al. MMLongBench-Doc: Benchmarking long-context document understanding with visualizations. In *Proceedings of Advances in Neural Information Processing Systems*, volume 37, pages 95963–96010, 2024.

[11] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. GPT-4 Technical Report. *arXiv preprint arXiv:2303.08774*, 2023.

[12] Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. Gemini: A family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

[13] Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*, 2024.

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

[15] Google. Gemini 3 Pro Model Card. https://storage.googleapis.com/deepmind-media/Model-Cards/Gemini-3-Pro-Model-Card.pdf, 2026.

[16] Google. Gemini 3.1 Pro: A smarter model for your most complex tasks. https://blog.google/innovation-and-ai/models-and-research/gemini-models/gemini-3-1-pro/, 2026.

[17] Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-VL: A versatile vision-language model for understanding, localization. *Text Reading, and Beyond*, 2023.

[18] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. Qwen2-VL: Enhancing vision-language model's perception of the world at any resolution. *arXiv*

preprint arXiv:2409.12191, 2024.

[19] Shuai Bai, Keqin Chen, Xuejing Liu, et al. Qwen2.5-VL Technical Report. *arXiv preprint arXiv:2502.13923*, 2025.

[20] Shuai Bai, Yuxuan Cai, et al. Qwen3-VL Technical Report. *arXiv preprint arXiv:2511.21631*, 2025.

[21] Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, et al. InternVL: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 24185–24198, 2024.

[22] Zhe Chen, Weiyun Wang, Hao Tian, Shenglong Ye, Zhangwei Gao, Erfei Cui, Wenwen Tong, Kongzhi Hu, Jiapeng Luo, Zheng Ma, et al. How far are we to GPT-4V? closing the gap to commercial multimodal models with open-source suites. *Science China Information Sciences*, 67(12):220101, 2024.

[23] Zhangwei Gao, Zhe Chen, Erfei Cui, Yiming Ren, Weiyun Wang, Jinguo Zhu, Hao Tian, Shenglong Ye, Junjun He, Xizhou Zhu, et al. Mini-InternVL: A flexible-transfer pocket multimodal model with 5% parameters and 90% performance. *Visual Intelligence*, 2(1):1–17, 2024.

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

[25] Zhe Chen, Weiyun Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Erfei Cui, Jinguo Zhu, Shenglong Ye, Hao Tian, Zhaoyang Liu, et al. Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling. *arXiv preprint arXiv:2412.05271*, 2024.

[26] Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Hao Tian, Yuchen Duan, Weijie Su, Jie Shao, et al. InternVL3: Exploring advanced training and test-time recipes for open-source multimodal models. *arXiv preprint arXiv:2504.10479*, 2025.

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

[28] Hunyuan Vision Team, Pengyuan Lyu, Xingyu Wan, Gengluo Li, Shangpin Peng, Weinong Wang, Liang Wu, Huawen Shen, Yu Zhou, Canhui Tang, et al. HunyuanOCR Technical Report. *arXiv preprint arXiv:2511.19575*, 2025.

[29] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of Advances in Neural Information Processing Systems*, volume 30, 2017.

[30] Hejun Dong, Junbo Niu, Bin Wang, Weijun Zeng, Wentao Zhang, and Conghui He. MinerU-Diffusion: Rethinking document OCR as inverse rendering via diffusion decoding. *arXiv preprint arXiv:2603.22458*, 2026.

[31] Yaniv Leviathan, Matan Kalman, and Yossi Matias. Fast inference from transformers via speculative decoding. In *Proceedings of the International Conference on Machine Learning*, pages 19274–19286, 2023.

[32] Yuhui Li, Fangyun Wei, Chao Zhang, and Hongyang Zhang. EAGLE: Speculative sampling requires rethinking feature uncertainty. In *Proceedings of the International Conference on Machine Learning*, pages 28935–28948, 2024.

[33] Yuhui Li, Fangyun Wei, Chao Zhang, and Hongyang Zhang. EAGLE-2: Faster inference of language models with dynamic draft trees. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 7421–7432, 2024.

[34] Yuhui Li, Fangyun Wei, Chao Zhang, and Hongyang Zhang. EAGLE-3: Scaling up inference acceleration of large language models via training-time test. In *Proceedings of Advances in Neural Information Processing Systems*, volume 38, pages 136737–136756, 2026.

[35] Tianle Cai, Yuhong Li, Zhengyang Geng, Hongwu Peng, Jason D Lee, Deming Chen, and Tri Dao. Medusa: Simple LLM inference acceleration framework with multiple decoding heads. In *Proceedings of the 41st International Conference on Machine Learning*, pages 5209–5235, 2024.

[36] Jian Chen, Yesheng Liang, and Zhijian Liu. DFlash: Block diffusion for flash speculative decoding. *arXiv preprint arXiv:2602.06036*, 2026.

[37] Marianne Arriola, Aaron Gokaslan, Justin Chiu, Zhihan Yang, Zhixuan Qi, Jiaqi Han, Subham Sahoo, and Volodymyr Kuleshov. Block Diffusion: Interpolating between autoregressive and diffusion language models. In *Proceedings of the International Conference on Learning Representations*, pages 50726–50753, 2025.

[38] Shuang Cheng, Yihan Bian, Dawei Liu, Linfeng Zhang, Qian Yao, Zhongbo Tian, Wenhai Wang, Qipeng Guo, Kai Chen, Biqing Qi, et al. SDAR: A synergistic diffusion-autoregression paradigm for scalable sequence generation.

arXiv preprint arXiv:2510.06303, 2025.

[39] Chengyue Wu, Hao Zhang, Shuchen Xue, Shizhe Diao, Yonggan Fu, Zhijian Liu, Pavlo Molchanov, Ping Luo, Song Han, and Enze Xie. Fast-dLLM v2: Efficient block-diffusion LLM. *arXiv preprint arXiv:2509.26328*, 2025.

[40] Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with PagedAttention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*, 2023.

[41] Georgi Gerganov and llama.cpp contributors. Llama.cpp – run LLM inference in C/C++. https://github.com/ggml-org/llama.cpp, 2023. GitHub repository.

[42] Arindam Mitra, Luciano Del Corro, Guoqing Zheng, Shweti Mahajan, Dany Rouhana, Andres Codas, Yadong Lu, Wei-ge Chen, Olga Vrousgos, Corby Rosset, et al. AgentInstruct: Toward generative teaching with agentic flows. *arXiv preprint arXiv:2407.03502*, 2024.

[43] Dingfeng Shi, Jingyi Cao, Qianben Chen, Weichen Sun, Weizhen Li, Hongxuan Lu, Fangchen Dong, Tianrui Qin, King Zhu, Minghao Liu, et al. TaskCraft: Automated generation of agentic tasks. *arXiv preprint arXiv:2506.10055*, 2025.

[44] Haris Riaz, Sourav Sanjukta Bhabesh, Vinayak Arannil, Miguel Ballesteros, and Graham Horwood. MetaSynth: Meta-prompting-driven agentic scaffolds for diverse synthetic data generation. In *Findings of the Association for Computational Linguistics: ACL*, pages 18770–18803, 2025.

[45] Cheng Cui, Ting Sun, Suyin Liang, Tingquan Gao, Zelun Zhang, Jiaxuan Liu, Xueqing Wang, Changda Zhou, Hongen Liu, Manhui Lin, et al. PaddleOCR-VL: Boosting multilingual document parsing via a 0.9B ultra-compact vision-language model. *arXiv preprint arXiv:2510.14528*, 2025.

[46] Cheng Cui, Ting Sun, Suyin Liang, Tingquan Gao, Zelun Zhang, Jiaxuan Liu, Xueqing Wang, Changda Zhou, Hongen Liu, Manhui Lin, et al. PaddleOCR-VL-1.5: Towards a multi-task 0.9B VLM for robust in-the-wild document parsing. *arXiv preprint arXiv:2601.21957*, 2026.

[47] Baode Wang, Biao Wu, Weizhen Li, Meng Fang, Zuming Huang, Jun Huang, Haozhe Wang, Yanjie Liang, Ling Chen, Wei Chu, et al. Infinity Parser: Layout aware reinforcement learning for scanned document parsing. *arXiv preprint arXiv:2506.03197*, 2025.

[48] Wenhui Liao, Hongliang Li, Pengyu Xie, Xinyu Cai, Yufan Shen, Yi Xin, Qi Qin, Shenglong Ye, Tianbin Li, Ming Hu, et al. HSD: Training-free acceleration for document parsing vision-language model with hierarchical speculative decoding. *arXiv preprint arXiv:2602.12957*, 2026.

[49] Xiangyang Chen, Shuzhao Li, Xiuwen Zhu, Yongfan Chen, Fan Yang, Cheng Fang, Lin Qu, Xiaoxiao Xu, Hu Wei, and Minggang Wu. Logics-Parsing Technical Report. *arXiv preprint arXiv:2509.19760*, 2025.

[50] Xin An, Jingyi Cai, Xiangyang Chen, Huayao Liu, Peiting Liu, Peng Wang, Bei Yang, Xiuwen Zhu, Yongfan Chen, Yan Gao, et al. Logics-Parsing-Omni Technical Report. *arXiv preprint arXiv:2603.09677*, 2026.

[51] Yongkun Du, Zhineng Chen, Yazhen Xie, Weikang Bai, Hao Feng, Wei Shi, Yuchen Su, Can Huang, and Yu-Gang Jiang. Unirec-0.1b: Unified text and formula recognition with 0.1b parameters. *arXiv preprint arXiv:2512.21095*, 2025.

[52] Yumeng Li, Guang Yang, Hao Liu, Bowen Wang, and Colin Zhang. dots.ocr: Multilingual document layout parsing in a single vision-language model. *arXiv preprint arXiv:2512.02498*, 2025.

[53] Haoran Wei, Yaofeng Sun, and Yukun Li. DeepSeek-OCR: Contexts optical compression. *arXiv preprint arXiv:2510.18234*, 2025.

[54] Heming Xia, Zhe Yang, Qingxiu Dong, Peiyi Wang, Yongqi Li, Tao Ge, Tianyu Liu, Wenjie Li, and Zhifang Sui. Unlocking efficiency in large language model inference: A comprehensive survey of speculative decoding. In *Findings of the Association for Computational Linguistics: ACL*, pages 7655–7671, 2024.

[55] Fabian Gloeckle, Badr Youbi Idrissi, Baptiste Rozière, David Lopez-Paz, and Gabriel Synnaeve. Better & faster large language models via multi-token prediction. *arXiv preprint arXiv:2404.19737*, 2024.

[56] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[57] Mostafa Dehghani, Basil Mustafa, Josip Djolonga, Jonathan Heek, Matthias Minderer, Mathilde Caron, Andreas Steiner, Joan Puigcerver, Robert Geirhos, Ibrahim M Alabdulmohsin, et al. Patch n' Pack: NaViT, a Vision Transformer for any aspect ratio and resolution. In *Proceedings of Advances in Neural Information Processing Systems*, volume 36, pages 2252–2274, 2023.

[58] Tencent Hunyuan. Hunyuan-0.5B. https://github.com/Tencent-Hunyuan/Hunyuan-0.5B, 2025.

[59] Juechu Dong, Boyuan Feng, Driss Guessous, Yanbo Liang, and Horace He. Flex Attention: A programming model for generating optimized attention kernels. *arXiv preprint arXiv:2412.05496*, 2024.

[60] Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026. URL https://qwen.ai/blog?id=qwen3.5.

[61] Ankush Gupta, Andrea Vedaldi, and Andrew Zisserman. Synthetic data for text localisation in natural images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2315–2324, 2016.

[62] 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. In *Proceedings of the European Conference on Computer Vision*, 2022.

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

[64] Yuqi Liu, Bohao Peng, Zhisheng Zhong, Zihao Yue, Fanbin Lu, Bei Yu, and Jiaya Jia. Seg-Zero: Reasoning-chain guided segmentation via cognitive reinforcement. *arXiv preprint arXiv:2503.06520*, 2025.

[65] Xumeng Wen, Zihan Liu, Shun Zheng, Zhijian Xu, Shengyu Ye, Zhirong Wu, Xiao Liang, Yang Wang, Junjie Li, Ziming Miao, et al. Reinforcement learning with verifiable rewards implicitly incentivizes correct reasoning in base LLMs. *arXiv preprint arXiv:2506.14245*, 2025.

[66] Shangpin Peng, Weinong Wang, Zhuotao Tian, Senqiao Yang, Xing Wu, Haotian Xu, Chengquan Zhang, Takashi Isobe, Baotian Hu, and Min Zhang. Uni-DPO: A unified paradigm for dynamic preference optimization of LLMs. *arXiv preprint arXiv:2506.10054*, 2025.

[67] Shangpin Peng, Senqiao Yang, Li Jiang, and Zhuotao Tian. Mitigating object hallucinations via sentence-level early intervention. In *Proceedings of the IEEE International Conference on Computer Vision*, 2025.

[68] Ling Team, Anqi Shen, Baihui Li, Bin Hu, Bin Jing, Cai Chen, Chao Huang, Chao Zhang, Chaokun Yang, Cheng Lin, et al. Every Step Evolves: Scaling reinforcement learning for trillion-scale thinking model. *arXiv preprint arXiv:2510.18855*, 2025.

[69] Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. DAPO: An open-source LLM reinforcement learning system at scale. In *Proceedings of Advances in Neural Information Processing Systems*, volume 38, pages 113222–113244, 2026.

[70] Xu Zhong, Elaheh ShafieiBavani, and Antonio Jimeno Yepes. Image-Based Table Recognition: Data, model, and evaluation. In *Proceedings of the European Conference on Computer Vision*, 2020.

[71] Gengluo Li, Shangpin Peng, Chengquan Zhang, Binghong Wu, Hao Feng, Weinong Wang, Pengyuan Lyu, Huawen Shen, Xingyu Wan, Zhuotao Tian, Han Hu, Can Ma, and Yu Zhou. StrucTab: A structured optimization framework for table parsing. *arXiv preprint arXiv:2606.29905*, 2026.

[72] Renqiu Xia, Bo Zhang, Haoyang Peng, Hancheng Ye, Xiangchao Yan, Peng Ye, Botian Shi, Yu Qiao, and Junchi Yan. StructChart: On the schema, metric, and augmentation for visual chart understanding. *arXiv preprint arXiv:2309.11268*, 2023.

[73] Long Xu, Binghong Wu, Tinghao Yu, Hao Feng, Zhenyu Huang, Haoqing Jiang, Yunhao Wang, Shuo Huang, and Feng Zhang. MORE: A multilingual document parsing benchmark and evaluation. In *Proceedings of the International Conference on Machine Learning*, 2026.

[74] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, 2020.

[75] Shuaiqi Duan, Yadong Xue, Weihan Wang, Zhe Su, Huan Liu, Sheng Yang, Guobing Gan, Guo Wang, Zihan Wang, Shengdong Yan, et al. GLM-OCR Technical Report. *arXiv preprint arXiv:2603.10910*, 2026.

[76] Zelun Zhang, Hongen Liu, Suyin Liang, Yubo Zhang, Yiqing Xiang, Jiaxuan Liu, Ting Sun, Manhui Lin, Yue Zhang, Changda Zhou, et al. PaddleOCR-VL-1.6: Expanding the frontier of document parsing with under-optimized region refinement and progressive post-training. *arXiv preprint arXiv:2606.03264*, 2026.

[77] Youyang Yin, Huanhuan Liu, Qunyi Xie, Chaorun Liu, Shiqi Yang, Shaohua Wang, Zhanlong Liu, Hao Zou, Jinyue Chen, Shu Wei, et al. Unlimited OCR works. *arXiv preprint arXiv:2606.23050*, 2026.

[78] Haoran Wei, Yaofeng Sun, and Yukun Li. DeepSeek-OCR 2: Visual causal flow. *arXiv preprint arXiv:2601.20552*, 2026.

[79] Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, et al. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*, 2024.

[80] Tianyu Yu, Zefan Wang, Chongyi Wang, Fuwei Huang, Wenshuo Ma, Zhihui He, Tianchi Cai, Weize Chen, Yuxiang Huang, Ranchi Zhao, et al. MiniCPM-V 4.5: Cooking efficient MLLMs via architecture, data, and training recipe. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 11704–11715, 2026.

[81] Shiyin Lu, Yang Li, Yu Xia, Yuwei Hu, Shanshan Zhao, Yanqing Ma, Zhichao Wei, Yinglun Li, Lunhao Duan, Jianshan Zhao, Yuxuan Han, Haijun Li, et al. Ovis2.5 Technical Report. *arXiv:2508.11737*, 2025.

[82] V Team, Wenyi Hong, Wenmeng Yu, Xiaotao Gu, Guo Wang, Guobing Gan, Haomiao Tang, Jiale Cheng, Ji Qi, Junhui Ji, Lihang Pan, et al. GLM-4.5V and GLM-4.1V-Thinking: Towards versatile multimodal reasoning with scalable reinforcement learning. *arXiv preprint arXiv:2507.01006*, 2025.

[83] Kimi Team, Tongtong Bai, Yifan Bai, Yiping Bao, SH Cai, Yuan Cao, Y Charles, HS Che, Cheng Chen, Guanduo Chen, et al. Kimi K2.5: Visual agentic intelligence. *arXiv preprint arXiv:2602.02276*, 2026.

[84] Aaditya Singh, Adam Fry, Adam Perelman, Adam Tart, Adi Ganesh, Ahmed El-Kishky, Aidan McLaughlin, Aiden Low, AJ Ostrow, Akhila Ananthram, et al. OpenAI GPT-5 System Card. *arXiv preprint arXiv:2601.03267*, 2025.

[85] Bytedance Seed. Seed1.8 model card: Towards generalized real-world agency, 2025. URL https://github.com/ByteDance-Seed/Seed-1.8/blob/main/Seed-1.8-Modelcard.pdf.

[86] ByteDance Seed Team. Seed2.0 model card: Towards intelligence frontier for real-world complexity, February 2026. URL https://github.com/ByteDance-Seed/Seed2.0. Model Card.

[87] Xiaomi Corporation. Xiaomi MiMo-V2-Omni: See, hear, act in the agentic era. https://mimo.xiaomi.com/mimo-v2-omni, 2026.

[88] Anthropic. Claude Opus 4.7. https://www.anthropic.com/news/claude-opus-4-7, 2026.

[89] Fanqing Meng, Wenqi Shao, Quanfeng Lu, Peng Gao, Kaipeng Zhang, Yu Qiao, and Ping Luo. ChartAssisstant: A universal chart multimodal language model via chart-to-table pre-training and multitask instruction tuning. In *Findings of the Association for Computational Linguistics: ACL*, 2024.

[90] Renqiu Xia, Hancheng Ye, Xiangchao Yan, Qi Liu, Hongbin Zhou, Zijun Chen, Botian Shi, Junchi Yan, and Bo Zhang. ChartX and ChartVLM: A versatile benchmark and foundation model for complicated chart reasoning. *IEEE Transactions on Image Processing*, 2025.

[91] Liang Zhang, Anwen Hu, Haiyang Xu, Ming Yan, Yichen Xu, Qin Jin, Ji Zhang, and Fei Huang. TinyChart: Efficient chart understanding with visual token merging and program-of-thoughts learning. In *Proceedings of the 2024 conference on empirical methods in natural language processing*, 2024.

[92] Zhengzhuo Xu, Bowen Qu, Yiyan Qi, Sinan Du, Chengjin Xu, Chun Yuan, and Jian Guo. ChartMoE: Mixture of diversely aligned expert connector for chart understanding. In *Proceedings of the International Conference on Learning Representations*, 2025.

[93] Xuanle Zhao, Xianzhen Luo, Qi Shi, Chi Chen, Shuo Wang, Zhiyuan Liu, and Maosong Sun. ChartCoder: Advancing multimodal large language model for Chart-to-Code generation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2025.

[94] Yang Chen, Yufan Shen, Wenxuan Huang, Sheng Zhou, Qunshu Lin, Xinyu Cai, Zhi Yu, Jiajun Bu, Botian Shi, and Yu Qiao. Learning Only with Images: Visual reinforcement learning with reasoning, rendering, and visual feedback. *arXiv preprint arXiv:2507.20766*, 2025.

[95] Lei Chen, Xuanle Zhao, Zhixiong Zeng, Jing Huang, Liming Zheng, Yufeng Zhong, and Lin Ma. Breaking the SFT plateau: Multimodal structured reinforcement learning for Chart-to-Code generation. *arXiv preprint arXiv:2508.13587*, 2025.

[96] Handong Zheng, Yumeng Li, Kaile Zhang, Liang Xin, Guangwei Zhao, Hao Liu, Jiayu Chen, Jie Lou, Jiyu Qiu, Qi Fu, et al. Multimodal OCR: Parse anything from documents. *arXiv preprint arXiv:2603.13032*, 2026.

[97] ShengYun Peng, Aishwarya Chakravarthy, Seongmin Lee, Xiaojing Wang, Rajarajeswari Balasubramaniyan, and Duen Horng Chau. UniTable: Towards a unified framework for table recognition via self-supervised pretraining. *arXiv preprint arXiv:2403.04822*, 2024.

[98] Junyuan Zhang, Bin Wang, Qintong Zhang, Fan Wu, Zichen Wen, Jialin Lu, Junjie Shan, Ziqi Zhao, Shuya Yang, Ziling Wang, et al. TRivia: Self-supervised fine-tuning of vision-language models for table recognition. *arXiv preprint arXiv:2512.01248*, 2025.

[99] Zhang Li, Yuliang Liu, Qiang Liu, Zhiyin Ma, Ziyang Zhang, Shuo Zhang, Zidun Guo, Jiarui Zhang, Xinyu Wang, and Xiang Bai. MonkeyOCR: Document parsing with a structure-recognition-relation triplet paradigm. *arXiv*

preprint arXiv:2506.05218, 2025.

[100] Yuan Liu, Zhongyin Zhao, Le Tian, Haicheng Wang, Xubing Ye, Yangxiu You, Zilin Yu, Chuhan Wu, Zhou Xiao, Yang Yu, et al. POINTS-Reader: Distillation-free adaptation of vision-language models for document conversion. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2025.

[101] Yufeng Zhong, Lei Chen, Zhixiong Zeng, Xuanle Zhao, Deyang Jiang, Liming Zheng, Jing Huang, Haibo Qiu, Peng Shi, Siqi Yang, et al. Reading or reasoning? format decoupled reinforcement learning for document OCR. *arXiv preprint arXiv:2601.08834*, 2025.

[102] Junbo Niu, Zheng Liu, Zhuangcheng Gu, Bin Wang, Linke Ouyang, Zhiyuan Zhao, Tao Chu, Tianyao He, Fan Wu, Qintong Zhang, et al. MinerU2.5: A decoupled vision-language model for efficient high-resolution document parsing. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics–Industry Track*, 2025.

[103] Bin Wang, Tianyao He, Linke Ouyang, Fan Wu, Zhiyuan Zhao, Tao Chu, Yuan Qu, Zhenjiang Jin, Weijun Zeng, Ziyang Miao, et al. MinerU2.5-Pro: Pushing the limits of data-centric document parsing at scale. *arXiv preprint arXiv:2604.04771*, 2026.

[104] Mistral AI Team. Mistral OCR: Free online AI OCR tool to extract text. https://www.mistralocr.com, 2025.

[105] Jake Poznanski, Aman Rangapur, Jon Borchardt, Jason Dunkelberger, Regan Huff, Daniel Lin, Christopher Wilhelm, Kyle Lo, and Luca Soldaini. olmOCR: Unlocking trillions of tokens in PDFs with vision language models. *arXiv preprint arXiv:2502.18443*, 2025.

[106] Yufeng Zhong, Lei Chen, Xuanle Zhao, Wenkang Han, Liming Zheng, Jing Huang, Deyang Jiang, Yilin Cao, Lin Ma, and Zhixiong Zeng. OCRVerse: Towards holistic OCR in end-to-end vision-language models. *arXiv preprint arXiv:2601.21639*, 2026.

[107] Hao Wu, Haoran Lou, Xinyue Li, Zuodong Zhong, Zhaojun Sun, Phellon Chen, Xuanhe Zhou, Kai Zuo, Yibo Chen, Xu Tang, et al. FireRed-OCR Technical Report. *arXiv preprint arXiv:2603.01840*, 2026.

[108] Kaitao Jiang, Ruiyan Gong, Xiaolong Cheng, Kangning Niu, Tianlun Li, and Mu Xu. ABot-OCR Technical Report. *arXiv preprint arXiv:2605.27978*, 2026.

[109] Daxiang Dong, Mingming Zheng, Dong Xu, Chunhua Luo, Bairong Zhuang, Yuxuan Li, Ruoyun He, Haoran Wang, Wenyu Zhang, Wenbo Wang, et al. Qianfan-OCR: A unified end-to-end model for document intelligence. *arXiv preprint arXiv:2603.13398*, 2026.

[110] Cheng Cui, Ting Sun, Manhui Lin, Tingquan Gao, Yubo Zhang, Jiaxuan Liu, Xueqing Wang, Zelun Zhang, Changda Zhou, Hongen Liu, et al. PaddleOCR 3.0 Technical Report. *arXiv preprint arXiv:2507.05595*, 2025.

[111] Baidu. BaiduOCRAPI, 2025. URL https://ai.baidu.com/tech/ocr/general.

[112] PaddleOCR. PP-ChatOCR, 2025. URL https://github.com/PaddlePaddle/PaddleOCR.

[113] Seed. Seed1.6, 2025. URL https://seed.bytedance.com/en/seed1_6.