LocateAnything: Fast and High-Quality Vision-Language Grounding with Parallel Box Decoding

Shihao Wang[1], Shilong Liu[2], Yuanguo Kuang[1], Xinyu Wei[1], Yangzhou Liu[3], Zhiqi Li[4], Yunze Man[4], Guo Chen[3], Andrew Tao[4], Guilin Liu[4], Jan Kautz[4], Lei Zhang[1], and Zhiding Yu[4†]

[1] The Hong Kong Polytechnic University
[2] Princeton University
[3] Nanjing University
[4] NVIDIA

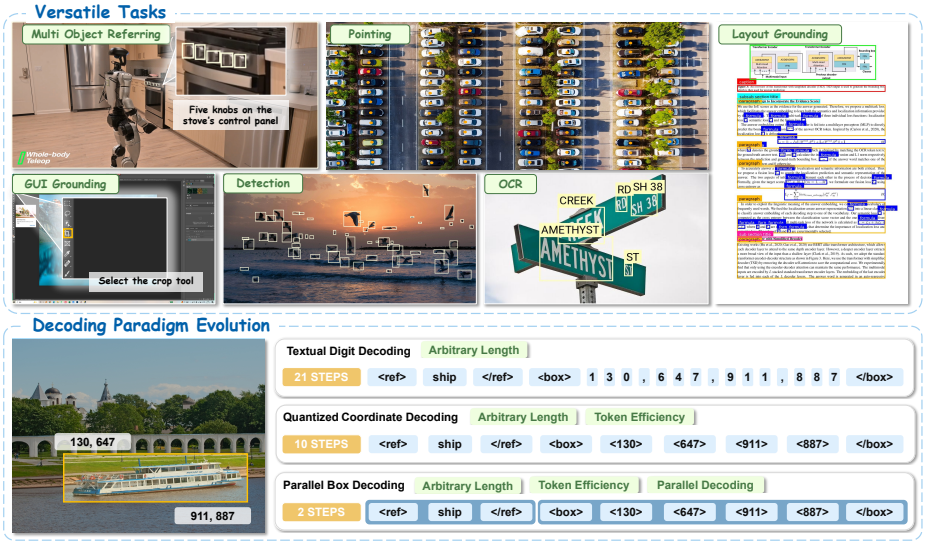

Fig. 1: Versatile tasks of LocateAnything with parallel box decoding. Top: LocateAnything supports diverse localization tasks under a unified vision-language model. **Bottom:** Textual digit decoding spells coordinates digit by digit, and quantized coordinate decoding predicts coordinate tokens sequentially. In contrast, **Parallel Box Decoding** predicts each geometric unit (*e.g.*, a bounding box) in a single forward pass.

Abstract. Vision-language models (VLMs) commonly formulate visual grounding and detection as a coordinate-token generation problem, serializing each 2D box into multiple 1D tokens that are learned and decoded largely independently. This token-by-token decoding mismatches the coupled structure of box geometry and creates a practical inference bottleneck due to strictly sequential generation. We introduce **LocateAnything**, a unified generative grounding and detection framework based on **Parallel Box Decoding (PBD)**. By decoding geometric elements such as

† Corresponding author.

bounding boxes and points as atomic units in a single step, LocateAnything preserves intra-box geometric coherence and unlocks substantial parallelism. We show that PBD improves both decoding throughput and localization accuracy. We further develop a scalable data engine and curate **LocateAnything-Data**, a large-scale dataset with more than 138 million training samples, substantially increasing data diversity for high-precision localization. Extensive evaluations show that LocateAnything advances the speed–accuracy frontier, achieving significantly higher decoding throughput while improving high-IoU localization quality across diverse benchmarks. The results highlight the complementary benefits of Parallel Box Decoding and large-scale training data in enabling efficient and precise unified visual grounding and detection.

Keywords: Visual Grounding · Detection · Vision-Language Models

1 Introduction

Vision-language models (VLMs) [4, 9, 19, 32, 41, 80, 89] are increasingly adopted as a general-purpose backbone for interactive and embodied systems [16, 58, 77, 79] due to their broader knowledge and stronger instruction-following capabilities than conventional specialized models [8, 45, 45, 53, 70, 81, 81, 100]. To act in the world, VLMs [2, 4, 23, 80, 98] must be tightly grounded in *perception* — in particular, they *localize* task-relevant entities (*e.g.*, objects [33, 78, 94, 101], UI elements [21, 46, 56, 63], regions [13, 31, 38, 69, 72, 95, 96]) from natural-language intents with high quality and low latency, which requires high vision-language grounding capabilities.

Object detection and grounding in VLMs [33, 39, 62, 65, 94, 98, 102] are often formulated as a *generative* problem. Under the next-token prediction (NTP) paradigm [11, 33, 65], a VLM can answer open-ended queries by emitting spatial coordinates as a token sequence. As illustrated in the bottom panel of Fig. 1, existing methods [33, 65, 67, 93, 102] commonly represent coordinates as either **Textual Digits** (*e.g.*, "1024" as "1", "0", "2", "4") or **Quantized Tokens** (*e.g.*, $x_1 \rightarrow y_1 \rightarrow x_2 \rightarrow y_2$). Despite their differences, these representations serialize a 2D geometric object into a 1D stream, forcing token-by-token generation at inference time. This token-level sequential decoding becomes a practical bottleneck (higher latency and lower throughput) and under-utilizes the strong structured correlation among coordinates (x_1, y_1, x_2, y_2).

Multi-Token Prediction (MTP) [54, 64, 92, 97] offers a natural approach to reducing decoding steps by predicting multiple tokens in parallel. In language modeling, MTP is usually implemented by randomly (i) choosing positions in the sequence and training the model to predict a following span in parallel (*i.e.*, next-block prediction) [7, 44, 51, 54], or (ii) masking some tokens of the sequence and training the model to reconstruct the original text, such as masked diffusion modeling [1, 42, 52, 64]. However, these formulations are largely *structure-agnostic*: they treat inputs as generic token streams and mainly capture correlations driven by co-occurrence. Inferring the missing tokens from random subsets requires the

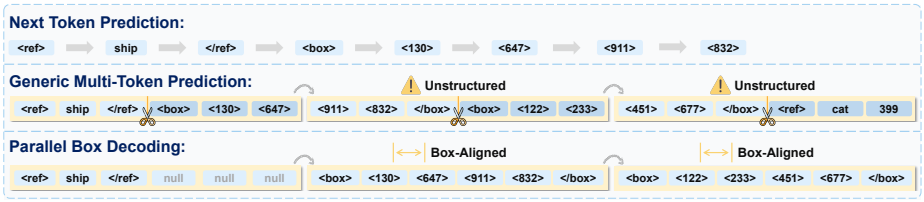

Fig. 2: Comparison of Token Decoding Methods. The NTP generates coordinate values one-by-one. The standard MTP method results in irregular distributions and non-coherent, unstructured patterns. Our proposed PBD generates a single atomic box (or point) unit in a parallel step, ensuring box-aligned and structured output.

model to represent complex and irregular conditional distributions. For tightly coupled units such as bounding boxes, this supervision does not match well the training objective because it can learn to generate token combinations across bounding-box boundaries and even object categories, as demonstrated in Fig. 2. Consequently, the model must fit many unreliable patterns, inducing spurious correlations, sacrificing structured decoding, and amplifying error propagation, which together reduce accuracy, reliability, and decoding speed.

To reconcile high-throughput decoding with reliable localization, we propose **LocateAnything**, a unified framework for VLM-based visual detection and grounding built upon **Parallel Box Decoding (PBD)**. Our key idea is to align MTP blocks with structured units: during training, LocateAnything treats each bounding box (or point) as an *atomic unit* and learns to predict the full coordinate set (x_1, y_1, x_2, y_2) in one parallel step. This *box-aligned* training target avoids arbitrary chunking of coordinate tokens. As a result, our strategy improves the localization performance of the model, while simultaneously unlocking the speed benefits of parallel decoding.

With the proposed PBD, we study various strategies for structured bounding-box decoding to balance throughput and accuracy. Our observations motivate a flexible inference design to meet different latency–robustness requirements by providing three on-demand modes. (i) **Fast Mode** (MTP) predicts full boxes in parallel for maximum throughput, which is suitable for latency- and compute-constrained settings, such as on-device robotics and embodied agents. (ii) **Slow Mode** (NTP) decodes coordinate tokens autoregressively for maximum stability, which is appropriate for high-precision labeling, final-pass dataset curation, and accuracy-oriented offline evaluation. (iii) **Hybrid Mode** uses Fast Mode by default and falls back to Slow Mode when the parallel output is unreliable, *e.g.*, due to format or consistency violations; this mode is intended for production pipelines that require both speed and accuracy. Overall, Hybrid Mode preserves most of the speed gains of parallel decoding while maintaining robust outputs.

Our main contributions are summarized as follows:

– We introduce **LocateAnything**, an early exploration of applying multi-token prediction to VLM-based detection/grounding via **Parallel Box Decoding**, performing box-aligned decoding to improve throughput and accuracy.

- We present a Hybrid decoding policy that detects unreliable parallel blocks and performs localized NTP re-decoding only for the problematic block, reducing worst-case failures while retaining most speed gains.
- Extensive evaluations, including layout grounding, long-tail detection, and GUI grounding, show that LocateAnything advances the **speed–accuracy frontier**, outperforming the SOTA by a large margin. It achieves up to **2.5**× higher decoding throughput while improving localization quality.

2 Related Work

Visual Detection and Grounding in VLMs. Visual grounding/detection tasks traditionally rely on task-specific heads [8, 36, 70, 71], but recent VLMs like Qwen-VL series [3, 4], InternVL [12] and Shikra [10] formulate it as an autoregressive token generation problem. This generative paradigm, however, often suffers from structural hallucinations and high latency [43]. To mitigate these issues, Rex-Omni [33] employs point-based prediction, while Patch-as-Decodable-Token (PaDT) [74] and Groma [60] utilize visual reference tokens to point directly to image patches. Complementary innovations such as Pink [88], ViP-LLaVA [6], Griffon [98], DnU [48] and PAM [49] focus on enhancing 2D referential comprehension through visual prompt engineering and multi-granularity feature scaling. LLMDet [23] boosts detection recall by data distribution tuning. To bypass serial decoding bottlenecks, WeDetect [22] treats detection as a parallel retrieval task. Advanced perception logic is further integrated via Chain-of-Thought (CoT) [67], while post-training strategies such as Vision-R1 [99], UniVG-R1 [5] and GW-VLM [35] utilize reinforcement learning to align model outputs with visual feedback [87] and reduce grounding errors [102].

Parallel Decoding via MTP and Diffusion LLMs. To mitigate autoregressive latency, parallel generation techniques such as MTP [7, 26, 73] predict multiple future tokens simultaneously, often coupled with speculative decoding to accelerate inference. Recent extensions such as Future Summary Prediction [61] capture long-term dependencies via auxiliary heads. Concurrently, Diffusion Language Models (DLMs) such as LLaDA [64], Dream [92], and DiffuCoder [28] frame sequence generation as a discrete denoising process, enabling bidirectional context modeling and non-autoregressive decoding. Hybrid semi-autoregressive paradigms, including Block Diffusion [1], SDLM [54] and Fast-dLLM v2 [83], decode fixed-size token blocks in parallel while maintaining causal dependencies to preserve KV-caching compatibility. More advanced frameworks [57, 82] unlock inter-block parallelism and adaptive block scheduling. These paradigms have been extended to the multimodal domain via DiffusionVL [97], translating autoregressive LMMs into high-performance diffusion-based models.

LocateAnything differs from existing works in two key aspects. First, instead of generating bounding boxes via slow NTP, we output the complete box in a single parallel step. Second, recent MTP paradigms group tokens into arbitrary chunks. Instead, our PBD treats the entire coordinate set as a single atomic block, resolving both the fragmentation of NTP and the arbitrary chunking of MTP, seamlessly unifying high throughput with structural coherence.

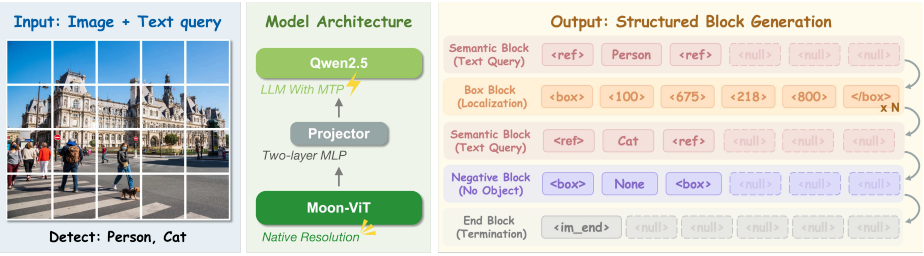

Fig. 3: Architecture and Block-Based Output Representation. LocateAnything formulates localization as generating a sequence of fixed-length, *box-aligned atomic blocks*. Four functional block types—Semantic, Box, Negative, and End blocks—are defined to jointly specify predicted entities or termination states.

3 Method

This section presents **LocateAnything**, a fast and effective framework that integrates **Parallel Box Decoding (PBD)** into VLMs for visual detection and grounding. Section 3.1 introduces the model architecture and the block-based output formulation. Section 3.2 details the joint training strategy, which aligns NTP with block-level MTP. Section 3.3 describes the on-demand inference mechanism, featuring a hybrid mode that dynamically balances decoding throughput and robustness. Finally, Section 3.4 outlines the construction of our large-scale training dataset, **LocateAnything-Data**.

3.1 Model Architecture and Formulation

Overview. As illustrated in Fig. 3, LocateAnything builds upon a native-resolution VLM pre-trained on large-scale image-text corpora. The architecture comprises a Moon-ViT [76] vision encoder and a Qwen2.5 [68] language decoder, bridged by a MLP projector. Given an input image $\mathcal{I}$, the vision encoder extracts visual tokens $Z = \text{Encoder}(\mathcal{I})$ at the native resolution, preserving the fine-grained spatial details crucial for high-precision localization. These tokens are subsequently fed into the language model, which directly converts them into a sequence of box-aligned block-level predictions.

Block-Based Output Formulation. To facilitate PBD, we abandon standard NTP coordinate generation. Instead, continuous coordinates are normalized to $[0, 1000]$, discretized into tokens [11,33], and reorganized into a sequence of blocks $\mathbf{B} = (b_1, b_2, \ldots, b_N)$. Conditioned on the visual features Z and a text query $\mathcal{E}$, the joint probability is formulated as $P(\mathbf{B} \mid \mathcal{Z}, \mathcal{E}) = \prod_{i=1}^{N} P(b_i \mid b_{<i}, Z, \mathcal{E})$.

Each block b_i acts as an atomic unit of constant length $L = 6$, accommodating a bounding box and two structural tokens (*e.g.*, `<box>` and `</box>`). To guarantee uniform tensor shapes for parallel decoding, any unoccupied positions are padded with a `<null>` token. As depicted in Fig. 3, we define four functional block types. *(1) Semantic Block*: Encodes the linguistic identity. If an expression exceeds the capacity of a single block, it is partitioned across multiple consecutive blocks. *(2) Box Block*: Uses four quantized coordinates representing the bounding boxes.

(3) Negative Block: Explicitly indicates the absence of a queried object. *(4) End Block*: Signals the termination of the generation process.

3.2 Training Design

Our method treats bounding box coordinates as an indivisible atomic unit, enforcing structured supervision and unlocking the capability for parallel generation. However, parallelizing the output directly in the training phase risks disrupting the model's inherent causal reasoning process. To resolve this issue, we introduce a dual-formulation training strategy that jointly optimizes two aligned representations: the NTP sequence to preserve the causal reasoning ability, and the block-wise MTP formulation for box-aligned predictions. To implement this, a single concatenated input sequence is constructed: $x_{\text{all}} = x_{\text{vis}} \oplus x_{\text{q}} \oplus x_{\text{ntp}} \oplus x_{\text{blk}}$, where $\oplus$ denotes sequence concatenation. The terms x_{vis} and x_{q} serve as the shared context (visual and text query inputs), x_{ntp} represents the standard NTP input sequence, and x_{blk} is the block-wise MTP input sequence. Essentially, they represent the identical ground truth in two distinct formats: a *token-level representation* and a *block-level representation*.

Specifically, inspired by [50, 54], x_{blk} is constructed by traversing x_{ntp} from left to right, splitting and padding the sequence according to our previously defined block rules. Within each block, we retain the first token to serve as the prediction context, while replacing all subsequent tokens with `[mask]` tokens. This structure prompts the model to simultaneously predict all masked tokens within the block in a single cohesive step. Notably, if the block size is set to 1, this MTP formulation naturally becomes equivalent to standard NTP.

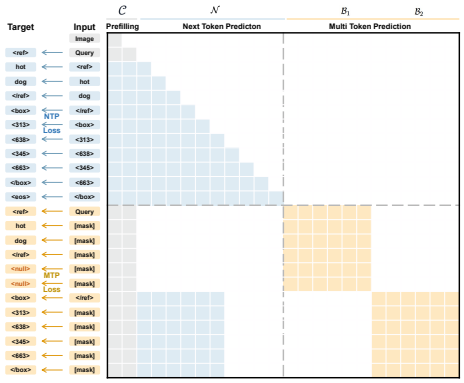

Fig. 4: Attention Mask for Joint NTP–MTP Training. The shared context and NTP stream use causal attention, the MTP blocks follow a block-causal pattern across blocks, and tokens within the same block share bidirectional attention. The two streams are isolated to prevent leakage while jointly attending to the shared context.

Attention Mask Design. The core challenge of this dual-sequence formulation is how to isolate the NTP and MTP streams while allowing both to leverage the shared context. This is achieved through a specialized attention mask (as shown in Fig. 4), which dictates information flow via three distinct behaviors:

Causal Attention for NTP. To preserve the original language capabilities of the VLM, the shared context (x_{vis} and x_{q}) and the NTP sequence (x_{ntp}) collectively employ a causal attention mask. Tokens within these segments can only attend to preceding tokens. Crucially, they are restricted from attending to x_{blk} to prevent

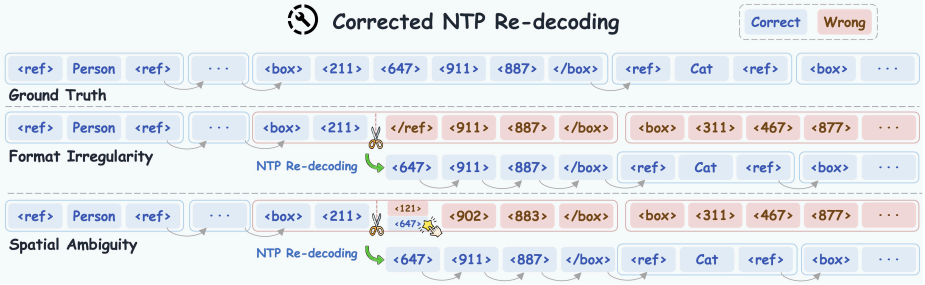

Fig. 5: Corrected NTP Re-decoding. When parallel decoding encounters *Format Irregularity* or *Spatial Ambiguity*, the model discards the erroneous block and reverts to standard NTP to ensure robust predictions.

data leakage. This strict causal formulation perfectly aligns with the standard KV Cache usage during inference.

Causal Flow Across Blocks. To align with the semi-autoregressive generation process, attention across different blocks in x_{blk} is strictly causal. Tokens in the active block can attend to the shared context and all previously committed blocks, but cannot see future blocks. This historical visibility enables the model to learn dependencies between different box predictions, effectively mitigating duplicate or missing bounding boxes.

Bidirectional Intra-Block Attention. Following the block-causal design widely adopted in recent generative modeling [1, 24, 64, 82–84], tokens within the same block share bidirectional attention. This fully-connected intra-block interaction allows the model to capture complex internal relationships (*e.g.*, geometric dependencies among a set of coordinates) and resolve all internal tokens simultaneously within a single functional unit.

Objective. Guided by this mask, we jointly minimize the cross-entropy losses for both sequences, *i.e.*, $\mathcal{L} = \mathcal{L}_{\text{ntp}} + \mathcal{L}_{\text{blk}}$.

3.3 On-Demand Inference Modes

While our proposed PBD significantly accelerates inference, parallel decoding faces an inherent exploration-exploitation dilemma in highly complex scenes, as shown in Fig. 5. The first is *Format Irregularity*, which occurs in complex scenes containing multiple instances across categories. During parallel decoding, the model may struggle at category boundaries, hesitating between continuing to predict for the current class or transitioning to a new class. This uncertainty manifests as malformed syntax within a single predicted block, erroneously mixing structural and coordinate tokens (*e.g.*, `<box><211></ref><911><887></box>`). The second is *Spatial Ambiguity*, which arises when objects are densely arranged in regular grids, such as rows or columns. The MTP approach can blur spatial boundaries and output an intermediate coordinate situated between two objects, consequently producing low IoU predictions.

Both failure patterns can be effectively resolved using an NTP fallback mechanism. The NTP prediction can achieve higher precision when handling

complex category transitions and dense spatial layouts. Therefore, during MTP inference, we continuously validate the syntactic integrity and monitor spatial confidence. Specifically, an ambiguity trigger is activated if two conditions are met simultaneously: (1) the top-1 coordinate token's probability is below 0.7, and (2) the max-min difference among the top-5 coordinate tokens exceeds 80 within the [0, 1000] normalized space. Upon detection of a format violation or high spatial ambiguity, the compromised block is discarded, and the generation reverts to the last verified prefix. NTP is then employed to autoregressively generate the tokens for the specific problematic block. Once the block is completed, the model seamlessly switches back to MTP for subsequent predictions.

Based on the above discussion, we propose three on-demand inference modes to balance throughput and spatial robustness. *(1) Slow Mode*, which generates the output token-by-token using standard NTP. *(2) Fast Mode*, which leverages MTP to predict box-aligned blocks. For each block, `<null>` padding tokens are discarded, and the remaining tokens are appended to the output; the committed tokens are stored in the key-value cache and serve as causal context for subsequent prediction steps. *(3) Hybrid Mode*, which employs MTP by default but seamlessly switches to NTP when parallel outputs become unreliable.

Inference-Time Attention Mask. During inference, the attention mask for each MTP decoding step mirrors the training-time block-causal pattern illustrated in Fig. 4. All previously committed tokens in the KV cache follow standard causal attention, while the n_{future} tokens in the current MTP block attend to each other bidirectionally, enabling parallel token prediction. Meanwhile, the current block can attend to all preceding blocks but is prevented from accessing subsequent ones. After each MTP step, the KV cache is truncated to retain only committed tokens, evicting mask tokens and the duplicated anchor to ensure the cache stays consistent with the causal prefix seen during training.

3.4 LocateAnything-Data

To train a highly capable model for general-purpose visual detection and grounding, we curate **LocateAnything-Data**, a large-scale, multi-domain dataset. The dataset construction details can be found in the **supplementary materials**.

As illustrated in Fig. 6, the dataset contains 12M unique images and 138M natural language queries. Furthermore, the dataset includes 785M annotated bounding boxes, providing massive and dense supervisory signals to guide the spatial learning of the LocateAnything model. The training corpus is categorized into six distinct tasks. (1) **General object detection** constitutes the foundation, representing 66.9% of the queries and providing the essential bounding box supervision (83.1%) to help the model achieve precise and dense coordinate alignments. (2) **Grounding user interface** elements (16.5% of queries) enable the model to support embodied agents and graphical user interface navigation tasks. (3) **Natural language referring** comprehension (7.3% of queries) enables the model to link complex linguistic intents to specific spatial regions. (4) **Text localization** (3.6% of queries) ensures that the model can perceive and tightly ground textual information within images. (5) **Document and scene layout**

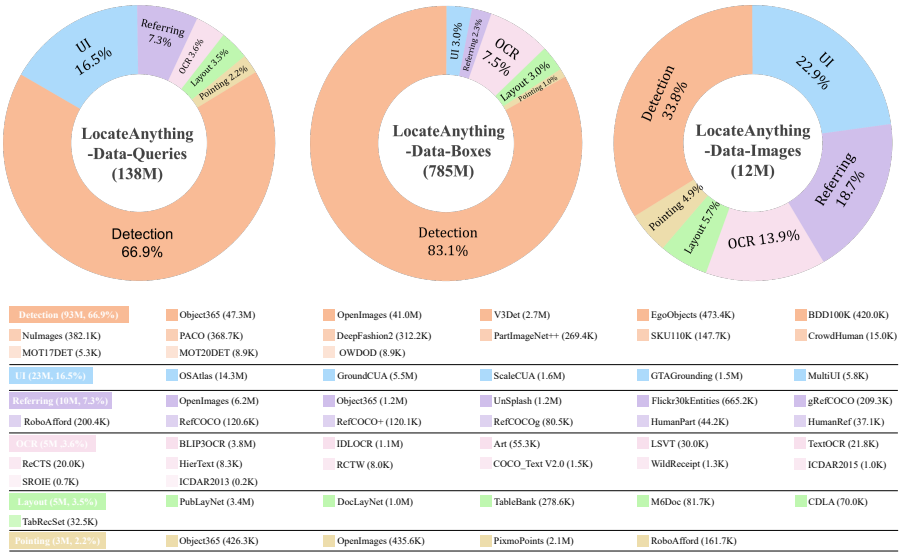

Fig. 6: Overview of the **LocateAnything-Data** dataset. The pie charts illustrate the task distribution across natural language queries, bounding boxes, and unique images. The bottom panel provides a detailed breakdown specifically for the language queries, showing the absolute count and percentage for each task category.

grounding (3.5% of queries) enriches the structural reasoning capabilities of the model. (6) **Point-based localization** tasks (2.2% of queries) further refine the spatial precision of the model for fine-grained predictions.

4 Experiments

4.1 Training Details and Evaluation Setup

Training Details. We first conduct an initial training on the base VLM with focus entirely on world-knowledge alignment, during which all detection and grounding data are excluded. We then apply a two-stage supervised fine-tuning to the base VLM to train our LocateAnything model. In Stage-1, we incorporate a massive mixture of 138M queries into the overall training data to equip the model with comprehensive grounding and detection capabilities. In Stage-2, we reduce the proportion of general training data to 20% while significantly increasing the proportion of data containing many objects per image (*e.g.*, MOT20Det [18], SKU110K [27]) to enhance the model's ability in dense detection. For model ablations, we train all models exclusively on the COCO dataset [47] to strictly isolate PBD's architectural benefits from our massive 138M data. Detailed configurations for both the base VLM and the subsequent LocateAnything model training are provided in the **supplementary materials**.

Compared Methods. We compare LocateAnything against three categories of methods. **(1) Specialized detectors**, including representative general detection

Table 1: Results on LVIS and COCO. Throughout all tables, "-" means that the information was not reported in the respective papers or the model does not support the corresponding task, **bold** and underline highlight the best and second-best, and BPS (Boxes Per Second) measures decoding throughput.

Method	Throughput	Zero-Shot (LVIS)	LVIS (F1@IoU)			Zero-Shot (COCO)	COCO (F1@IoU)		
			0.5	0.95	Mean		0.5	0.95	Mean
Open-set Specialized Detectors									
Grounding DINO-Swin-T [53]	-	Yes	47.7	22.7	38.8	Yes	69.8	23.0	56.6
Closed-set Specialized Detectors									
Faster RCNN-R50 [70]	-	-	-	-	-	No	60.6	7.1	48.1
DETR-R50 [8]	-	-	-	-	-	No	65.9	13.6	48.3
Deformable-DETR-R50 [105]	-	-	-	-	-	No	69.7	17.7	54.7
DINO-R50 [100]	-	-	-	-	-	No	68.8	21.1	55.6
DINO-Swin-L [100]	-	-	-	-	-	No	**75.6**	**25.4**	**62.1**
Vision-Language Models									
DeepSeek-VL2-Small [85]	-	-	56.2	21.0	41.8	-	60.9	14.9	45.9
MiMo-VL-7B [75]	1.0	-	49.5	8.8	31.4	-	56.5	6.7	35.9
OVIS2.5-2B [59]	1.3	-	54.4	15.8	37.4	-	56.2	10.3	38.7
Qwen3-VL-4B [3]	1.1	-	59.8	20.0	43.5	-	63.0	14.2	46.1
Qwen3-VL-8B [3]	1.0	-	61.5	20.2	44.8	-	62.8	14.0	45.7
SEED1.5-VL [29]	-	Yes	**65.6**	19.5	46.7	Yes	71.3	14.3	51.4
Rex-Omni-3B [33]	5.0	Yes	64.3	20.7	46.9	Yes	72.0	15.9	52.9
LocateAnything-3B	**12.7**	Yes	62.3	**31.1**	**50.7**	Yes	70.1	19.3	54.7

models such as DETR [8] and Deformable-DETR [105], *etc.*, open-set detectors such as Grounding DINO [53], leading document layout analysis model DocLayout-YOLO [103], and text detection model PaddleOCRv5 [17]. **(2) General-purpose VLMs with grounding capabilities**, including Qwen3-VL [3], DeepSeek-VL2 [85], OVIS2.5 [59], MiMo-VL [75], and SEED1.5-VL [29], *etc.* These models adopt textual coordinate representations with standard next-token prediction, providing a direct comparison to our parallel box decoding paradigm. **(3) VLM-based detection and grounding specialists**, including Rex-Omni [33], which is the most related work to ours targeting unified detection and grounding in a VLM framework. For GUI grounding, we also include several domain-specific expert models [25, 55, 56, 86, 90, 91, 104].

Evaluation Setup. Following the evaluation framework established in Rex-Omni [33], we conduct a comprehensive assessment across multiple visual perception tasks. Object Detection is evaluated on COCO for common objects, LVIS [30] for long-tailed distributions, and VisDrone [20] and Dense200 [33] for dense and tiny object scenarios. Language-aware Grounding tasks include Referring Expression Comprehension (REC) on RefCOCOg [37] and HumanRef [34]. Interactive tasks are evaluated through GUI Grounding on ScreenSpot-Pro [40]. Additionally, Layout Grounding on DocLayNet [66] and M6Doc [14], along with OCR (text detection and recognition) on TotalText [15], are reported together under scene text and document understanding tasks.

The metric for each task is summarized as follows. *(1) Box-based outputs*: For detection, layout, and OCR tasks, a prediction is considered correct (*i.e.*, a true positive) if its Intersection over Union (IoU) with the ground truth exceeds a

Table 2: Results on dense object detection benchmark Dense200 and VisDrone.

Method	Score Thresh.	Dense200			VisDrone		
		F1@IoU 0.5	F1@IoU 0.95	F1@IoU Mean	F1@IoU 0.5	F1@IoU 0.95	F1@IoU Mean
Open-set Specialized Detectors							
Grounding DINO-Swin-T [53]	0.25	36.9	**19.7**	33.1	55.2	**3.9**	38.5
Vision-Language Models							
DeepSeek-VL2-Small [85]	-	16.0	3.9	12.7	35.8	1.7	23.3
OVIS2.5-2B [59]	-	17.9	0.0	6.7	21.0	0.1	9.2
MiMo-VL-7B [75]	-	29.7	0.4	15.9	27.7	0.3	14.3
Qwen3-VL-4B [3]	-	17.5	2.4	12.5	42.3	1.4	26.0
Qwen3-VL-8B [3]	-	13.5	1.7	9.6	42.8	1.4	25.8
SEED1.5-VL [29]	-	76.9	5.3	53.2	55.9	0.6	27.4
Rex-Omni-SFT-3B [33]	-	60.2	10.6	46.4	55.6	1.9	32.4
Rex-Omni-3B [33]	-	**78.4**	10.3	58.3	61.6	1.5	35.8
LocateAnything-3B	-	74.0	18.5	**58.7**	**63.0**	3.2	**39.9**

Table 3: Results for the GUI Grounding task. The * denotes our reproduced results.

Method	ScreenSpot-Pro												Avg
	Dev.		Creative		CAD		Sci.		Office		OS		
	Text	Icon	Text	Icon	Text	Icon	Text	Icon	Text	Icon	Text	Icon	
InfiGUI-R1-3B [55]	51.3	12.4	44.9	7.0	33.0	14.1	58.3	20.0	65.5	28.3	43.9	12.4	35.7
JEDI-3B [86]	61.0	13.8	53.5	8.4	27.4	9.4	54.2	18.2	64.4	32.1	38.3	9.0	36.1
Rex-Omni-3B [33]	61.7	9.7	52.5	12.6	22.3	9.4	59.0	26.4	63.3	28.3	24.1	15.7	36.8
ScaleCUA-3B [56]	57.8	18.6	38.8	42.9	16.8	32.0	54.3	28.1	47.9	64.6	35.5	**52.0**	40.8
GTA1-7B [90]	62.6	18.2	53.3	17.2	**66.9**	20.7	76.4	31.8	82.5	50.9	48.6	25.9	50.1
Qwen3-VL-30B-A3B* [3]	76.0	24.8	**69.2**	20.3	51.8	15.6	76.4	27.3	80.8	37.7	**75.7**	38.2	53.7
GUI-Owl-7B [91]	76.6	31.0	59.6	27.3	64.5	21.9	79.1	37.3	77.4	39.6	59.8	33.7	54.9
MAI-UI-2B [104]	76.6	32.4	**69.2**	21.7	61.4	23.4	81.2	34.5	**85.9**	39.6	68.2	41.6	57.4
UI-Venus-1.5-2B [25]	70.1	43.4	63.6	28.7	54.3	32.8	76.4	38.2	81.9	47.2	73.8	51.7	57.7
GUI-Owl-32B [91]	**84.4**	39.3	65.2	18.2	62.4	28.1	**82.6**	39.1	81.4	39.6	70.1	36.0	58.0
LocateAnything-3B	70.8	**50.3**	60.1	**46.9**	57.9	**40.6**	69.4	**58.2**	77.2	**69.8**	65.4	43.8	**60.3**

certain threshold. The F1-score is reported at $IoU = 0.5$, $IoU = 0.95$, and as a mean over thresholds ($mIoU$). *(2) Point-based outputs*: For pointing tasks, a prediction is considered correct if the predicted point falls within the ground-truth segmentation mask or bounding box. We similarly report the F1-score for these point-based outputs based on this correctness criterion.

4.2 Main Results

In this section, we report the accuracy metrics and the throughput (measured in boxes per second, BPS on a single NVIDIA H100 GPU with a batch size of 1) of LocateAnything under the default *Hybrid Mode*. The results of *Fast* and *Slow Mode* are provided in the **supplementary materials**.

High-Quality Multi-Object Detection. Our model exhibits robust generalization in both common and complex dense object detection scenarios. On general detection benchmarks reported in Tab. 1, LocateAnything improves the mean F1 by +3.8% on LVIS and +1.8% on COCO compared to Rex-Omni, despite sharing an identical model size. Crucially, the model effectively learns the generalized spatial distribution, transferring its detection capabilities to unseen, heavily packed object types. This is evidenced by its performance on the dense detection benchmarks in Tab. 2, where it achieves 39.9 mean F1 on VisDrone, substantially outperforming Rex-Omni which scores 35.8. Similarly, it reaches

Table 4: Performance comparison on document layout grounding and OCR tasks.

Method	Score Thresh.	DocLayNet			M6Doc			TotalText		
		F1@IoU 0.5	F1@IoU 0.95	F1@IoU Mean	F1@IoU 0.5	F1@IoU 0.95	F1@IoU Mean	F1@IoU 0.5	F1@IoU 0.95	F1@IoU Mean
Specialized Detectors										
DocLayout-YOLO [103]	0.3	**91.2**	**52.1**	**81.1**	-	-	-	-	-	-
PaddleOCRv5 [17]	-	-	-	-	-	-	-	40.2	0.7	25.7
Vision-Language Models										
SEED1.5-VL [29]	-	54.9	4.3	28.7	48.0	3.4	28.0	35.0	0.3	19.5
Qwen3-VL-4B [3]	-	60.8	8.2	37.2	30.6	4.9	19.0	55.4	3.6	36.1
Qwen3-VL-8B [3]	-	54.7	6.7	34.1	37.2	4.9	22.7	**59.4**	2.7	37.3
Rex-Omni-3B [33]	-	89.5	28.4	70.7	76.3	18.7	55.6	56.6	3.9	40.6
LocateAnything-3B	-	91.1	35.8	76.8	**90.6**	**25.8**	**70.1**	58.9	**5.1**	**43.3**

Table 5: Evaluation results on referring expression comprehension benchmarks.

Method	Score Thresh.	HumanRef			RefCOCOg val			RefCOCOg test		
		F1@IoU 0.5	F1@IoU 0.95	F1@IoU Mean	F1@IoU 0.5	F1@IoU 0.95	F1@IoU Mean	F1@IoU 0.5	F1@IoU 0.95	F1@IoU Mean
Open-set Specialized Detector										
Grounding DINO-Swin-T [53]	0.25	28.0	16.5	25.2	52.9	20.9	45.9	53.8	22.9	46.8
Vision-Language Model										
DeepSeek-VL2-Tiny [85]	-	39.1	16.9	31.4	67.4	16.1	50.5	69.3	16.9	52.1
OVIS2.5-2B [59]	-	70.6	12.3	50.0	87.4	29.3	73.4	87.6	30.5	73.8
MiMo-VL-7B [75]	-	77.6	26.4	63.4	84.9	14.4	65.3	84.6	14.9	65.5
DeepSeek-VL2-Small [85]	-	72.0	46.5	64.7	**92.4**	**45.6**	**81.4**	**91.8**	**47.0**	**81.6**
Qwen3-VL-4B [3]	-	77.7	54.9	71.1	88.0	34.0	74.7	87.6	33.9	74.6
Qwen3-VL-8B [3]	-	78.6	55.7	72.0	88.6	33.4	74.9	88.6	33.8	75.2
SEED1.5-VL [29]	-	**88.2**	60.0	**81.6**	84.7	30.9	71.9	85.2	32.1	73.2
Rex-Omni-3B [33]	-	85.4	65.4	79.9	86.6	35.3	73.6	86.8	36.6	74.3
LocateAnything-3B	-	82.9	**68.8**	78.7	88.6	41.5	76.7	88.8	43.4	77.6

a competitive 58.7 mean F1 on Dense200, demonstrating superior boundary delineation and instance separation in heavily overlapping environments.

Precise Open-World Localization Ability. LocateAnything demonstrates exceptional fine-grained localization capabilities across diverse open-world benchmarks, including user interface grounding, document layout parsing, and referring expression comprehension. As shown in Tab. 3, on the ScreenSpot-Pro [40], it achieves a SOTA mean F1 of 60.3, surpassing generalist VLMs like Qwen3-VL-30B-A3B and specialized models tailored for UI tasks such as GUI-Owl-32B. Furthermore, in document understanding tasks detailed in Tab. 4, LocateAnything establishes a new standard by reaching 76.8 and 70.1 mean F1 on DocLayNet and M6Doc, respectively, outperforming Rex-Omni by substantial margins. This precise spatial reasoning extends to complex referring tasks, as shown in Tab. 5, where the model seamlessly aligns nuanced human intents with visual regions, achieving 78.7 mean F1 on the HumanRef benchmark and remaining highly competitive on RefCOCOg against top-tier models.

Superior Decoding Speed. A key advantage of our model is its drastically reduced decoding steps. As shown in Tab. 1, our model achieves 12.7 BPS under the default hybrid mode, over 10× faster than textual-based Qwen3-VL (1.1 BPS) and 2.5× faster than quantized-based Rex-Omni (5.0 BPS).

4.3 Ablation Study

We conduct ablation studies on the COCO dataset to validate our core designs. The results are shown in Tab. 6 and Fig. 7.

Table 6: Ablation Studies on the COCO dataset. We decouple the analysis into three aspects: (a) coordinate representation, (b) block-based MTP Formulation, and (c) effectiveness of our on-demand decoding modes and loss design. Throughput is measured in boxes per second. For brevity, we report the Average metric across IoU thresholds for Recall (R), Precision (P), and F1 Score. "B" indicates block size in MTP.

(a) Coordinate Representations

Method	Throughput	R	P	F1
Textual	1.3	45.7	52.3	49.1
Quantized	3.9	48.2	52.2	50.1
PBD (Slow)	3.9	**49.4**	**55.2**	**52.1**
PBD (Fast)	**16.9**	45.6	54.6	49.6
PBD (Hybrid)	13.2	48.7	54.8	51.6

(b) MTP Formulations

Method	Throughput	R	P	F1
SDLM-B4 [54]	5.2	**45.4**	48.1	46.5
SDLM-B6 [54]	5.5	45.1	47.5	46.1
SDLM-B8 [54]	6.7	44.7	47.2	45.8
Block Diff-B6 [1]	4.7	45.1	44.3	44.8
PBD (Fast)	**16.9**	45.2	**54.6**	**49.6**

(c) Decoding Modes & Losses

$\mathcal{L}_{\text{ntp}}$	$\mathcal{L}_{\text{blk}}$	Mode	Throughput	R	P	F1
✓		Slow	3.9	48.2	52.2	50.1
	✓	Fast	16.7	45.6	49.0	47.2
✓	✓	Slow	3.9	**49.4**	**55.2**	**52.1**
✓	✓	Fast	**16.9**	45.6	54.6	49.6
✓	✓	Hybrid	13.2	48.7	54.8	51.6

Coordinate Representation. As Tab. 6(a) shows, under the NTP paradigm, Textual and Quantized representations yield sub-optimal performance (49.1 and 50.1 mean F1, respectively) due to forced token-by-token generation. Our PBD (Slow Mode) achieves the highest F1-score of 52.1, proving that a box-aligned formulation provides stronger supervision for spatial reasoning than 1D serialization, without sacrificing throughput.

MTP Formulation. Tab. 6(b) compares our box-aligned MTP against existing structure-agnostic MTP formulations. Methods like SDLM and Block Diffusion force the model to learn spurious, unaligned cross-boundary patterns, suffering from lower accuracy and limited acceleration (*e.g.*, SDLM-B6 achieves 46.1 F1-score at 5.5 BPS). Furthermore, structure-agnostic methods (*e.g.*, SDLM-B4, B6, B8) exhibit a strict speed-accuracy trade-off, where increasing the block size yields only marginal throughput gains while consistently degrading the F1-score. In contrast, our PBD strictly aligns MTP blocks with structured bounding box units, dramatically outpacing existing methods in throughput (16.9 BPS) while improving the mean F1 to 49.6.

Decoding Mode. Tab. 6(c) ablates the impact of our dual-formulation training ($\mathcal{L}_{\text{ntp}}$ and $\mathcal{L}_{\text{blk}}$). Training with isolated losses limits the model's potential; joint training successfully pushes the Slow Mode upper bound from 50.1 to 52.1 F1-score. During inference, Fast Mode (MTP) maximizes throughput (16.9 BPS) but induces accuracy drops in complex scenes. Hybrid Mode seamlessly resolves this trade-off, preserving most speed gains (13.2 BPS) while achieving robust, high-precision localization (51.6 F1-score).

Box Output Order. We investigate four spatial sorting strategies in Fig. 7 (left): *Top-Left Distance* (sorting by the x-coordinate of the top-left corner, then by the y-coordinate), *Center Distance* (the distance of the bounding box center point to the origin), *Area* (sorted from largest to smallest), and *Random* (shuffled randomly). Results show Top-Left Distance yields the highest F1-score. We take this setting as default in dataset construction.

Throughput. We compare the generation time and throughput against traditional NTP methods in Fig. 7 (right). As target boxes increase from 20 to 300, NTP methods suffer from a severe latency bottleneck. In contrast, the Parallel method exhibits little increase in generation time, increasing throughput from

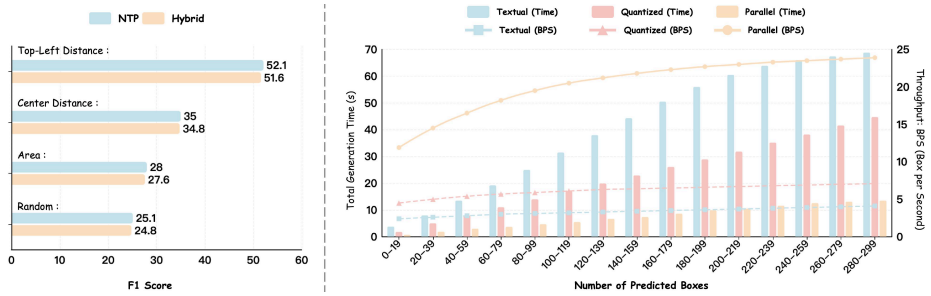

Fig. 7: Ablation Study on Box Ordering and Decoding Speed. Left: Effect of different bounding box sorting strategies on the F1-score for NTP and hybrid decoding modes. **Right:** Comparison of Generation Time (bars) and Throughput (lines) across varying numbers of predicted boxes for Textual, Quantized, and Parallel box decoding.

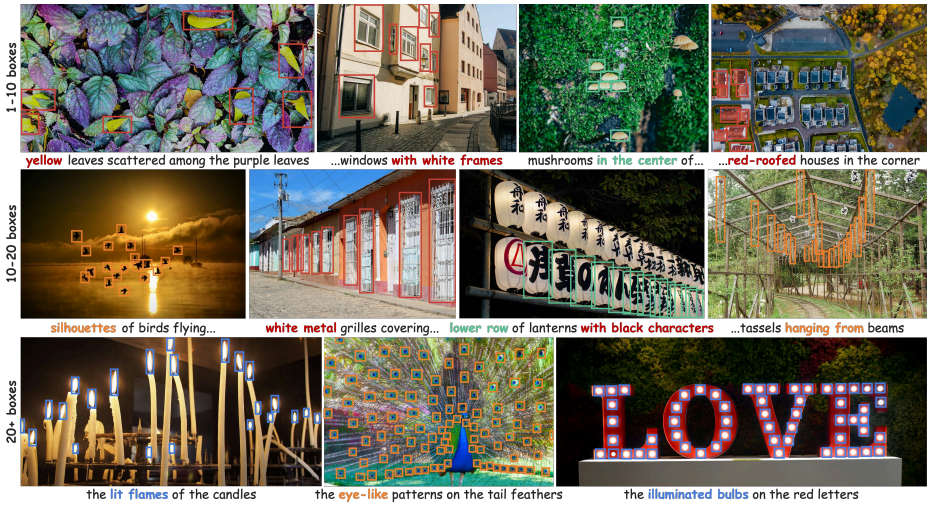

Fig. 8: Qualitative results. Each row shows test cases with varying numbers of target objects and diverse box scales. Different colors indicate different query categories, including attribute, part, reasoning, and spatial queries. Our model consistently localizes targets across diverse scene domains, arbitrary image resolutions, free-form textual queries, and an arbitrary number of objects, demonstrating strong robustness.

12 BPS to nearly 25 BPS in dense scenes. These findings confirm that PBD effectively breaks the decoding bottleneck, achieving a 2× to 6× speedup.

4.4 Qualitative Results

Fig. 8 visualizes representative grounding results of our model. Visual comparisons with other methods are provided in the **supplementary materials**. We observe three consistent behaviors. **(i) Compositional grounding:** our model handles attribute/part/spatial/reasoning-style queries well with consistent spatial alignment, supported by the diversity and coverage of our training data. **(ii) Robustness to large instance counts:** as targets grow from sparse to

crowded settings, the predicted boxes remain structured and accurate, reflecting the precision of our box-level decoding. This robustness is further strengthened by our Stage-2 training that emphasizes many-object images, improving dense localization in practice. Moreover, our Hybrid Mode maintains most of the parallel decoding speed while improving output stability in multi-instance generation. **(iii) Reliable localization in clutter:** boxes stay compact and well-separated under occlusion, repetitive textures, and grid-like dense layouts. Our hybrid inference mode further stabilizes these hard cases by detecting unreliable parallel blocks and falling back to NTP re-decoding when needed.

5 Conclusion

We presented LocateAnything, a unified framework that reformulates visual grounding and detection in VLMs via *Parallel Box Decoding*. By elevating geometric elements to atomic units rather than 1D streams, LocateAnything aligned the training supervision with the inherently coupled nature of spatial coordinates. With massive 138M text-image training queries and a flexible on-demand inference mechanism, LocateAnything not only delivered SOTA accuracy across diverse tasks, but also achieved up to a 2.5× speedup over competitive methods. Our method provided a practical and scalable route for real-time visual perception, opening the door to deploying general-purpose VLMs in latency-sensitive embodied robotics and interactive agents.

Limitation. Currently, our model is primarily trained with supervised fine-tuning. Reinforcement learning is an important next step to further optimize the block-level decoding policy, reduce fallback frequency, and encourage effective exploration in hard dense/long-tail cases, which could improve both robustness and worst-case decoding speed. We leave it for future work.

Acknowledgements

The authors would like to thank the valuable discussions and input from Qing Jiang, Amala Sanjay Deshmukh, Karan Sapra, Mingjie Liu, Yi Dong, Pavlo Molchanov, Yonggan Fu, Collin McCarthy, Mike Ranzinger, Greg Heinrich, Wonmin Byeon, Yexuan Li, Chi-Pin Huang, Fu-En Yang, Frank Wang, Jin Huang, Le An, Jaehun Jung, Shaokun Zhang, Hao Zhang, Johan Bjoerck, Jim Fan, Patrick Langechuan Liu, Sifei Liu, Xiaolong Li, Paris Zhang, Yilin Zhao, Subhashree Radhakrishnan, Shiyi Lan, Jose Alvarez, Sanja Fidler, Yan Wang, Xiaodong Yang, Yin Cui, Tsung-Yi Lin, Padmavathy Subramanian and more. We would also like to thank the NVIDIA infra, legal and data teams, including Xinyou Ma, Katherine Cheung, Timo Roman, and Yao Xu for their prompt and helpful support. Finally, the authors would like to additionally acknowledge the following teams, including Nemotron-Diffusion, Nemotron VLM, Cosmos, GR00T, Alpamayo, Gigas and Metropolis, for the engagement and downstream applications.

References

1. Arriola, M., Gokaslan, A., Chiu, J.T., Yang, Z., Qi, Z., et al.: Block diffusion: Interpolating between autoregressive and diffusion language models. In: ICLR (2025)
2. Azzolini, A., Bai, J., Brandon, H., Cao, J., Chattopadhyay, P., Chen, H., Chu, J., Cui, Y., Diamond, J., Ding, Y., et al.: Cosmos-reason1: From physical common sense to embodied reasoning. arXiv preprint arXiv:2503.15558 (2025)
3. Bai, S., Cai, Y., Chen, R., Chen, K., Chen, X., et al.: Qwen3-VL technical report. arXiv preprint arXiv:2511.21631 (2025)
4. Bai, S., Chen, K., Liu, X., Wang, J., Ge, W., et al.: Qwen2.5-VL technical report. arXiv preprint arXiv:2502.13923 (2025)
5. Bai, S., Li, M., Liu, Y., Tang, J., Zhang, H., Sun, L., Chu, X., Tang, Y.: Univg-r1: Reasoning guided universal visual grounding with reinforcement learning. arXiv preprint arXiv:2505.14231 (2025)
6. Cai, M., Liu, H., Feng, S., Lee, Y.J., et al.: ViP-LLaVA: Making large multimodal models understand arbitrary visual prompts. In: CVPR (2024)
7. Cai, T., Li, Y., Geng, Z., Peng, H., Lee, J.D., et al.: Medusa: Simple LLM inference acceleration framework with multiple decoding heads. arXiv preprint arXiv:2401.10774 (2024)
8. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., et al.: End-to-end object detection with transformers. In: ECCV. pp. 213–229 (2020)
9. Chen, G., Li, Z., Wang, S., Jiang, J., Liu, Y., Lu, L., Huang, D.A., Byeon, W., Le, M., Ehrlich, M., et al.: EAGLE 2.5: Boosting long-context post-training for frontier vision-language models. NeurIPS **38**, 91077–91100 (2026)
10. Chen, K., Zhang, Z., Zeng, W., Zhang, R., Zhu, F.P., et al.: Shikra: Unleashing multimodal LLM's referential dialogue magic. arXiv preprint arXiv:2306.15195 (2023)
11. Chen, T., Saxena, S., Li, L., Fleet, D.J., Hinton, G.: Pix2seq: A language modeling framework for object detection. In: ICLR (2022)
12. Chen, Z., Wu, J., Wang, W., Su, W., Chen, G., et al.: InternVL: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In: CVPR. pp. 24185–24198 (2024)
13. Cheng, A.C., Yin, H., Fu, Y., Guo, Q., Yang, R., et al.: SpatialRGPT: Grounded spatial reasoning in vision-language models. NeurIPS **37**, 135062–135093 (2024)
14. Cheng, H., Zhang, P., Wu, S., Zhang, J., Zhu, Q., et al.: M6Doc: A large-scale multi-format, multi-type, multi-layout, multi-language, multi-annotation category dataset for modern document layout analysis. In: CVPR. pp. 15138–15147 (2023)
15. Ch'Ng, C.K., Chan, C.S.: Total-Text: A comprehensive dataset for scene text detection and recognition. In: ICDAR. vol. 1, pp. 935–942. IEEE (2017)
16. Chou, Y.C., Wang, X., Li, Y., Wang, J., Liu, H., Xie, C., Yuille, A., Xiao, J.: Captain safari: A world engine with pose-aligned 3D memory. In: CVPR. pp. 25347–25357 (2026)
17. Cui, C., Sun, T., Lin, M., Gao, T., Zhang, Y., et al.: PaddleOCR 3.0 technical report. arXiv preprint arXiv:2507.05595 (2025)
18. Dendorfer, P., Ošep, A., Milan, A., Schindler, K., Cremers, D., et al.: MOTChallenge: A benchmark for single-camera multiple target tracking. arXiv preprint arXiv:2010.07548 (2020)
19. Deshmukh, A.S., Chumachenko, K., Rintamaki, T., Le, M., Poon, T., et al.: NVIDIA Nemotron Nano v2 VL. arXiv preprint arXiv:2511.03929 (2025)

20. Du, D., Zhu, P., Wen, L., Bian, X., Lin, H., et al.: VisDrone-DET2019: The vision meets drone object detection in image challenge results. In: ICCVW. pp. 0–0 (2019)
21. Feizi, A., Nayak, S., Jian, X., Lin, K.Q., Li, K., et al.: Grounding computer use agents on human demonstrations. arXiv preprint arXiv:2511.07332 (2025)
22. Fu, S., Su, Y., Rao, F., Lyu, J.X., Xie, et al.: WeDetect: Fast open-vocabulary object detection as retrieval. arXiv preprint arXiv:2512.12309 (2025)
23. Fu, S., Yang, Q., Mo, Q., Yan, J., Wei, X., et al.: LLMDet: Learning strong open-vocabulary object detectors under the supervision of large language models. arXiv preprint arXiv:2501.18954 (2025)
24. Fu, Y., Whalen, L., Ye, Z., Dong, X., Diao, S., et al.: Efficient-dLM: From autoregressive to diffusion language models, and beyond in speed. arXiv preprint arXiv:2512.14067 (2025)
25. Gao, C., Gu, Z., Liu, Y., Qiu, X., Shen, S., et al.: UI-Venus-1.5 technical report. arXiv preprint arXiv:2602.09082 (2026)
26. Gloeckle, F., Idrissi, B.Y., Roziere, B., Lopez-Paz, D., Synnaeve, G.: Better & faster large language models via multi-token prediction. In: ICML (2024)
27. Goldman, E., Herzig, R., Eisenschtat, A., Goldberger, J., Hassner, T.: Precise detection in densely packed scenes. In: CVPR. pp. 5227–5236 (2019)
28. Gong, S., Ye, J., Xie, Z., Lin, Z., Gao, J., et al.: DiffuCoder: Diffusion-based code generation with large language models. arXiv preprint arXiv:2501.01142 (2025)
29. Guo, D., Wu, F., Zhu, F., Leng, F., Shi, G., et al.: SEED1.5-VL technical report. arXiv preprint arXiv:2505.07062 (2025)
30. Gupta, A., Dollar, P., Girshick, R.: LVIS: A dataset for large vocabulary instance segmentation. In: CVPR. pp. 5356–5364 (2019)
31. Heinrich, G., Ranzinger, M., Yin, H., Lu, Y., Kautz, J., Tao, A., Catanzaro, B., Molchanov, P.: Radiov2.5: Improved baselines for agglomerative vision foundation models. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 22487–22497 (2025)
32. Huang, A., Yao, C., Han, C., Wan, F., Guo, H., et al.: Step3-VL-10B technical report. arXiv preprint arXiv:2601.09668 (2026)
33. Jiang, Q., Huo, J., Chen, X., Xiong, Y., Zeng, Z., et al.: Detect anything via next point prediction. arXiv preprint arXiv:2510.12798 (2025)
34. Jiang, Q., Wu, L., Zeng, Z., Ren, T., Xiong, Y., et al.: Referring to any person. In: CVPR. pp. 21667–21678 (2025)
35. Jiang, Q., et al.: A training-free guess what vision language model from snippets to open-vocabulary object detection. arXiv preprint arXiv:2601.11910 (2026)
36. Jiang, X., Li, S., Liu, Y., Wang, S., Jia, F., et al.: Far3D: Expanding the horizon for surround-view 3D object detection. In: AAAI. pp. 2561–2569 (2024)
37. Kazemzadeh, S., Ordonez, V., Matten, M., Berg, T.: ReferItGame: Referring to objects in photographs of natural scenes. In: Moschitti, A., Pang, B., Daelemans, W. (eds.) Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 787–798. Association for Computational Linguistics, Doha, Qatar (Oct 2014). https://doi.org/10.3115/v1/D14-1086, https://aclanthology.org/D14-1086/
38. Lai, X., Tian, Z., Chen, Y., Li, Y., Yuan, Y., et al.: LISA: Reasoning segmentation via large language model. In: CVPR. pp. 9579–9589 (2024)
39. Li, J., Xie, C., Ao, J., Leng, D., Yin, Y.: LMM-Det: Make large multimodal models excel in object detection. In: ICCV (2025)
40. Li, K., Meng, Z., Lin, H., Luo, Z., Tian, Y., et al.: ScreenSpot-Pro: GUI grounding for professional high-resolution computer use. In: ACM MM. pp. 8778–8786 (2025)

41. Li, W., Yuan, Y., Liu, J., Tang, D., Wang, S., Qin, J., Zhu, J., Zhang, L.: Token-Packer: Efficient visual projector for multimodal LLM. IJCV **133**(10), 6794–6812 (2025)
42. Li, X.L., Thickstun, J., Gulrajani, I., Liang, P.S., Hashimoto, T.B.: Diffusion-LM improves controllable text generation. In: NeurIPS (2022)
43. Li, Y., Du, Y., Zhou, K., Wang, J., Zhao, W.X., et al.: Evaluating object hallucination in large vision-language models. arXiv preprint arXiv:2305.10355 (2023)
44. Li, Z., Chen, G., Liu, S., Wang, S., VS, V., et al.: EAGLE 2: Building post-training data strategies from scratch for frontier vision-language models. arXiv preprint arXiv:2501.14818 (2025)
45. Li, Z., Wang, W., Xie, E., Yu, Z., Anandkumar, A., Alvarez, J.M., Luo, P., Lu, T.: Panoptic segformer: Delving deeper into panoptic segmentation with transformers. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1270–1279 (2022)
46. Lin, K.Q., Li, L., Gao, D., Yang, Z., Bai, Z., et al.: ShowUI: One vision-language-action model for generalist GUI agent. In: NeurIPS Workshop (2024)
47. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., et al.: Microsoft COCO: Common objects in context. In: ECCV. pp. 740–755. Springer (2014)
48. Lin, W., Wei, X., An, R., Gao, P., Zou, B., Luo, Y., Huang, S., Zhang, S., Li, H.: Draw-and-understand: Leveraging visual prompts to enable mllms to comprehend what you want. arXiv preprint arXiv:2403.20271 (2024)
49. Lin, W., Wei, X., An, R., Ren, T., Chen, T., Zhang, R., Guo, Z., Zhang, W., Zhang, L., Li, H.: Perceive anything: Recognize, explain, caption, and segment anything in images and videos. arXiv preprint arXiv:2506.05302 (2025)
50. Liu, A., He, M., Zeng, S., Zhang, S., Zhang, L., et al.: WeDLM: Reconciling diffusion language models with standard causal attention for fast inference. arXiv preprint arXiv:2512.22737 (2025)
51. Liu, A., Feng, B., Xue, B., Wang, B., Wu, B., et al.: DeepSeek-V3 technical report. arXiv preprint arXiv:2412.19437 (2024)
52. Liu, J., Dong, X., Ye, Z., Mehta, R., Fu, Y., et al.: TiDAR: Think in diffusion, talk in autoregression. arXiv preprint arXiv:2511.08923 (2025)
53. Liu, S., Zeng, Z., Ren, T., Li, F., Zhang, H., et al.: Grounding DINO: Marrying DINO with grounded pre-training for open-set object detection. arXiv preprint arXiv:2303.05499 (2023)
54. Liu, Y., Cao, Y., Li, H., Luo, G., Chen, Z., et al.: Sequential diffusion language models. arXiv preprint arXiv:2509.24007 (2025)
55. Liu, Y., Li, P., Xie, C., Hu, X., Han, X., et al.: InfiGUI-R1: Advancing multimodal GUI agents from reactive actors to deliberative reasoners. arXiv preprint arXiv:2504.14239 (2025)
56. Liu, Z., Xie, J., Ding, Z., Li, Z., Yang, B., et al.: ScaleCUA: Scaling open-source computer use agents with cross-platform data. arXiv preprint arXiv:2509.15221 (2025)
57. Lu, G., et al.: AdaBlock-dLLM: Semantic-aware diffusion LLM inference via adaptive block size. arXiv preprint arXiv:2509.26432 (2025)
58. Lu, L., Chen, G., Wei, Z., Li, Z., Liu, Y., Lu, T.: AV-Reasoner: Improving and benchmarking clue-grounded audio-visual counting for MLLMs. In: CVPR. pp. 33477–33487 (2026)
59. Lu, S., Li, Y., Xia, Y., Hu, Y., Zhao, S., et al.: Ovis2.5 technical report. arXiv preprint arXiv:2508.11737 (2025)

60. Ma, C., Jiang, Y., Wu, J., Yuan, Z., Qi, X.: Groma: Localized visual tokenization for grounding multimodal large language models. arXiv preprint arXiv:2404.13013 (2024)
61. Mahajan, D., Goyal, S., Idrissi, B.Y., Pezeshki, M., Mitliagkas, I., et al.: Beyond multi-token prediction: Pretraining LLMs with future summaries. arXiv preprint arXiv:2510.14751 (2025)
62. Man, Y., Wang, S., Zhang, G., Bjorck, J., Li, Z., et al.: Locateanything3d: Vision-language 3d detection with chain-of-sight. arXiv preprint arXiv:2511.20648 (2025)
63. Nayak, S., Jian, X., Lin, K.Q., Rodriguez, J.A., Kalsi, M., et al.: UI-Vision: A desktop-centric GUI benchmark for visual perception and interaction. arXiv preprint arXiv:2503.15661 (2025)
64. Nie, S., Zhu, F., You, Z., Zhang, X., Ou, J., et al.: Large language diffusion models. arXiv preprint arXiv:2502.09992 (2025)
65. Peng, Z., Wang, W., Dong, L., Hao, Y., Huang, S., et al.: Kosmos-2: Grounding multimodal large language models to the world. arXiv preprint arXiv:2306.14824 (2023)
66. Pfitzmann, B., Auer, C., Dolfi, M., Nassar, A.S., Staar, P.: DocLayNet: A large human-annotated dataset for document-layout segmentation. In: KDD. pp. 3743–3751 (2022)
67. Qi, Y., Zhang, Y., Gong, C., Tan, X., Zhang, W., et al.: CoT4Det: A chain-of-thought framework for perception-oriented vision-language tasks. arXiv preprint arXiv:2512.06663 (2025)
68. Qwen Team: Qwen2.5: A party of foundation models (September 2024), https://qwenlm.github.io/blog/qwen2.5/, accessed: 2026-06-27
69. Ranzinger, M., Heinrich, G., Kautz, J., Molchanov, P.: Am-radio: Agglomerative vision foundation model reduce all domains into one. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 12490–12500 (2024)
70. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: Towards real-time object detection with region proposal networks. IEEE TPAMI **39**(6), 1137–1149 (2016)
71. Ren, T., Jiang, Q., Liu, S., Zeng, Z., Liu, W., et al.: Grounding DINO 1.5: Advance the "edge" of open-set object detection. arXiv preprint arXiv:2405.10300 (2024)
72. Ren, Z., Huang, Z., Wei, Y., Zhao, Y., Fu, D., et al.: PixelLM: Pixel reasoning with large multimodal model. In: CVPR. pp. 26374–26383 (2024)
73. Samragh, M., Kundu, A., Harrison, D., Nishu, K., Naik, D., et al.: Your LLM knows the future: Uncovering its multi-token prediction potential. arXiv preprint arXiv:2507.11851 (2025)
74. Su, Y., Zhang, H., Li, S., Liu, N., Liao, J., et al.: Patch-as-decodable-token: Towards unified multi-modal vision tasks in MLLMs. In: ICLR (2026)
75. Team, C., Yue, Z., Lin, Z., Song, Y., Wang, W., et al.: MiMo-VL technical report. arXiv preprint arXiv:2506.03569 (2025)
76. Team, K., Du, A., Yin, B., Xing, B., Qu, B., et al.: Kimi-VL technical report. arXiv preprint arXiv:2504.07491 (2025)
77. Wang, S., Chen, G., Huang, D.a., Li, Z., Li, M., Liu, G., Kautz, J., Alvarez, J.M., Zhang, L., Yu, Z.: VideoITG: Multimodal video understanding with instructed temporal grounding. In: CVPR. pp. 24640–24650 (2026)
78. Wang, S., Liu, Y., Wang, T., Li, Y., Zhang, X.: Exploring object-centric temporal modeling for efficient multi-view 3d object detection. In: ICCV. pp. 3621–3631 (2023)

79. Wang, S., Yu, Z., Jiang, X., Lan, S., Shi, M., Chang, N., Kautz, J., Li, Y., Alvarez, J.M.: OmniDrive: A holistic vision-language dataset for autonomous driving with counterfactual reasoning. In: CVPR. pp. 22442–22452 (2025)
80. Wang, W., Gao, Z., Gu, L., Pu, H., Cui, L., et al.: InternVL 3.5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. arXiv preprint arXiv:2508.18265 (2025)
81. Wang, W., Dai, J., Chen, Z., Huang, Z., Li, Z., Zhu, X., Hu, X., Lu, T., Lu, L., Li, H., Wang, X., Qiao, Y.: Internimage: Exploring large-scale vision foundation models with deformable convolutions. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 14408–14419 (2023)
82. Wang, X., et al.: Diffusion LLMs can do faster-than-AR inference via discrete diffusion forcing. arXiv preprint arXiv:2508.09192 (2025)
83. Wu, C., Zhang, H., Xue, S., Diao, S., Fu, Y., et al.: Fast-dLLM v2: Efficient block-diffusion LLM. arXiv preprint arXiv:2509.26328 (2025)
84. Wu, C., Zhang, H., Xue, S., Liu, Z., Diao, S., et al.: Fast-dLLM: Training-free acceleration of diffusion LLM by enabling KV cache and parallel decoding. arXiv preprint arXiv:2505.22618 (2025)
85. Wu, Z., Chen, X., Pan, Z., Liu, X., Liu, W., et al.: DeepSeek-VL2: Mixture-of-experts vision-language models for advanced multimodal understanding. arXiv preprint arXiv:2412.10302 (2024)
86. Xie, T., Deng, J., Li, X., Yang, J., Wu, H., et al.: Scaling computer-use grounding via user interface decomposition and synthesis. arXiv preprint arXiv:2505.13227 (2025)
87. Xiong, T., Ge, Y., Li, M., Zhang, Z., Kulkarni, P., Wang, K., He, Q., Zhu, Z., Liu, C., Chen, R., et al.: Multi-Crit: Benchmarking multimodal judges on pluralistic criteria-following. In: CVPR. pp. 8641–8652 (2026)
88. Xuan, S., Guo, Q., Yang, M., Zhang, S.: Pink: Unveiling the power of referential comprehension for multi-modal LLMs. In: CVPR (2024)
89. Yang, B., Wen, B., Ding, B., Liu, C., Chu, C., et al.: Kwai Keye-VL 1.5 technical report. arXiv preprint arXiv:2509.01563 (2025)
90. Yang, Y., Li, D., Dai, Y., Yang, Y., Luo, Z., et al.: GTA1: GUI test-time scaling agent. arXiv preprint arXiv:2507.05791 (2025)
91. Ye, J., Zhang, X., Xu, H., Liu, H., Wang, J., et al.: Mobile-Agent-v3: Fundamental agents for GUI automation. arXiv preprint arXiv:2508.15144 (2025)
92. Ye, J., Xie, Z., Lin, Z., Gao, J., Wu, Z., et al.: Dream 7B: Diffusion large language models. arXiv preprint arXiv:2508.15487 (2025)
93. You, H., Zhang, H., Gan, Z., Du, X., Zhang, B., et al.: Ferret: Refer and ground anything anywhere at any granularity. In: ICLR (2024)
94. Yu, E., Lin, K., Zhao, L., Yin, J., Wei, Y., et al.: Perception-R1: Pioneering perception policy with reinforcement learning. arXiv preprint arXiv:2504.07954 (2025)
95. Yuan, Y., Li, W., Liu, J., Tang, D., Luo, X., Qin, C., Zhang, L., Zhu, J.: Osprey: Pixel understanding with visual instruction tuning. In: CVPR. pp. 28202–28211 (2024)
96. Yuan, Y., Zhang, W., Li, X., Wang, S., Li, K., et al.: PixelRefer: A unified framework for spatio-temporal object referring with arbitrary granularity. arXiv preprint arXiv:2510.23603 (2025)
97. Zeng, L., Yao, J., Liao, B., Tao, H., Liu, W., et al.: DiffusionVL: Translating any autoregressive models into diffusion vision language models. arXiv preprint arXiv:2512.15713 (2025)

98. Zhan, Y., Zhu, Y., Chen, Z., Yang, F., Tang, M., et al.: Griffon: Spelling out all object locations at any granularity with large language models. In: ECCV (2024)
99. Zhan, Y., Zhu, Y., Zheng, S., Zhao, H., Yang, F., et al.: Vision-R1: Evolving human-free alignment in large vision-language models via vision-guided reinforcement learning. arXiv preprint arXiv:2503.18013 (2025)
100. Zhang, H., Li, F., Liu, S., Zhang, L., Su, H., et al.: DINO: DETR with improved denoising anchor boxes for end-to-end object detection. arXiv preprint arXiv:2203.03605 (2022)
101. Zhang, H., Li, H., Li, F., Ren, T., Zou, X., et al.: LLaVA-Grounding: Grounded visual chat with large multimodal models. In: ECCV. pp. 19–35. Springer (2024)
102. Zhang, H., You, H., Dufter, P., Zhang, B., Chen, C., et al.: Ferret-v2: An improved baseline for referring and grounding with large language models. arXiv preprint arXiv:2404.07973 (2024)
103. Zhao, Z., Kang, H., Wang, B., He, C.: DocLayout-YOLO: Enhancing document layout analysis through diverse synthetic data and global-to-local adaptive perception. arXiv preprint arXiv:2410.12628 (2024)
104. Zhou, H., Zhang, X., Tong, P., Zhang, J., Chen, L., et al.: MAI-UI technical report: Real-world centric foundation GUI agents. arXiv preprint arXiv:2512.22047 (2025)
105. Zhu, X., Su, W., Lu, L., Li, B., Wang, X., et al.: Deformable DETR: Deformable transformers for end-to-end object detection. arXiv preprint arXiv:2010.04159 (2021)