Locate Anything in Videos: Rethinking Efficient Generative Spatio-Temporal Video Grounding

Hanoona Rasheed[1], **Haania Siddiqui**[1], **Ming-Hsuan Yang**[2], **Fahad Shahbaz Khan**[1,3], **Salman Khan**[1,3]
[1]Mohamed bin Zayed University of Artificial Intelligence, [2]University of California, Merced, [3]Apertix

Spatio-temporal video grounding (STVG) requires models to identify when a referred event occurs and localize the target entity throughout that interval. Existing multimodal large language models typically serialize dense localization trajectories autoregressively, causing decoding latency to grow with tube length and allowing localization errors to propagate across time. We introduce **Parallel Tube Decoding (PTD)**, a generative formulation that decomposes grounding into a temporal block followed by time-conditioned spatial blocks decoded simultaneously. This removes both token-level and trajectory-level dependencies, reducing the sequential decoding depth to a fixed $1 + 1$ rounds, independent of tube length. To enable parallel spatial generation, we introduce Decoupled Block Attention, which preserves access to shared video-query context while eliminating cross-box dependencies, together with localization-aware policy optimization for temporal boundaries and spatial geometry. On VidSTG, PTD reduces Tube Completion Latency by 79× and increases spatial decoding throughput by 92× over standard autoregressive decoding, while also improving grounding accuracy. With a compact 4B backbone, our model performs favorably well on VidSTG and HC-STVG, and generalizes zero-shot to temporal grounding, grounded VideoQA, and referring video object tracking. Our results show parallel tube generation is an efficient and effective alternative to autoregressive localization in videos.

Website https://mbzuai-oryx.github.io/parallel-tube-decoding/
Code https://github.com/mbzuai-oryx/ParallelTubeDecoding

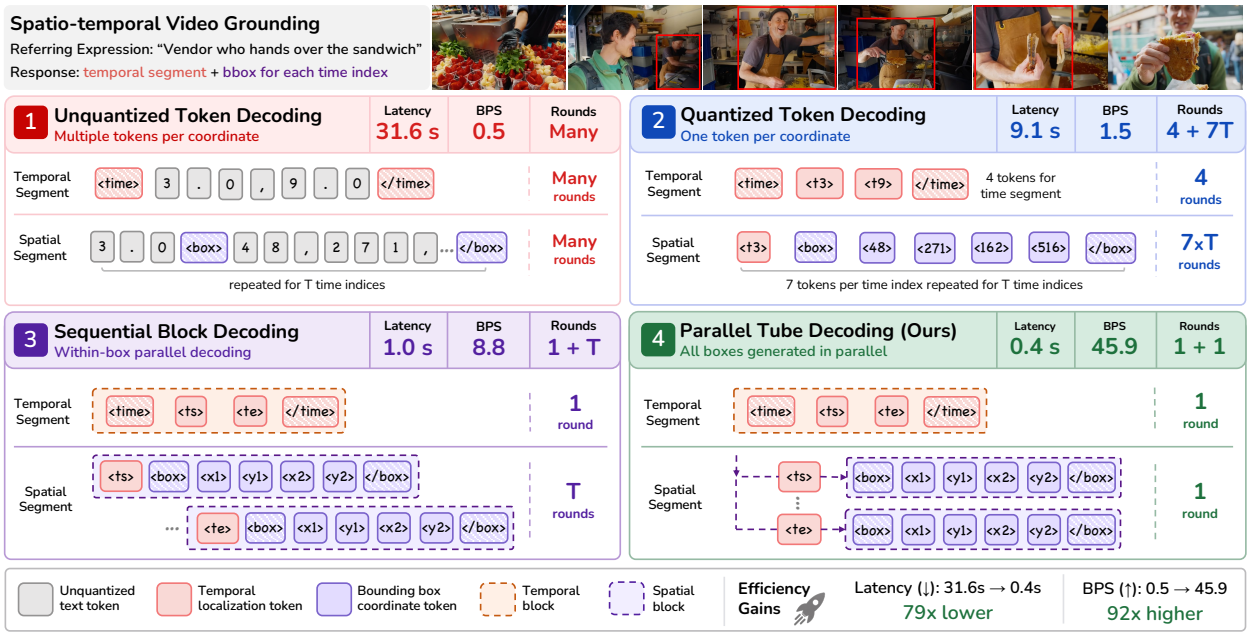

Figure 1 Autoregressive localization vs. Parallel Tube Decoding. Given a video and a referring expression, STVG predicts when the event occurs and the bounding box of the referred entity throughout that interval. We compare four decoding paradigms. Unquantized Token Decoding generates textual localization values autoregressively, while Quantized Token Decoding reduces representation overhead with discrete temporal and spatial tokens. Sequential Block Decoding further removes within-block dependencies but remains sequential across the tube. In contrast, our Parallel Tube Decoding (PTD) generates all time-conditioned spatial blocks in parallel after temporal localization, reducing the sequential decoding depth to $1 + 1$. Compared with standard Unquantized Token Decoding, PTD achieves 79× lower Tube Completion Latency and 92× higher spatial decoding throughput, measured as Boxes Per Second (BPS), while simultaneously improving spatio-temporal grounding accuracy.

1 Introduction

Real-world video understanding requires interpreting natural-language queries about entities and events unfolding across space and time. Spatio-temporal video grounding (STVG) formalizes this by identifying when a queried event occurs and localizing the referred entity throughout that interval [65, 50, 58]. *Accurate* and *efficient* spatio-temporal localization is becoming increasingly important as video-language models are deployed for long-video search [55, 46, 28], evidence-grounded question answering [9, 8, 38, 44], language-guided tracking [14, 60, 5, 39], and embodied interaction [31, 66, 12]. Since the query may identify the target through its attributes, actions, or relations rather than a category name, the model must reason over the video to find the intended target, making multimodal large language models (MLLMs) a natural foundation [3, 33, 53].

However, STVG remains challenging because temporal and spatial localization involve different sources of ambiguity. Temporally, the target may remain visible before and after the event, while the event itself may be brief or have ambiguous boundaries. Spatially, the target may undergo motion, occlusion, scale and appearance changes, or interference from nearby distractors [65, 50, 58, 22]. The output structure amplifies this. Unlike temporal grounding, which predicts a start and an end, or image grounding, which predicts a single region, STVG emits a bounding box at every grounded time step. In MLLMs, serializing this tube into an autoregressive coordinate sequence makes decoding depth and latency grow with tube length.

Because of this cost, many MLLM-based methods offload spatial localization to prediction heads, external detectors, spatial decoders, or detection-and-tracking pipelines [53, 20, 51, 63]; those that keep the tube inside the native token interface stay unified [24] but inherit the scaling cost of decoding. The cost can be attacked at the token level, as Figure 1 shows. Unquantized Token Decoding writes timestamps and coordinates as text, so a single value may be split into multiple tokens and require several dependent predictions. Quantized Token Decoding maps each value to a single token and substantially reduces this representation overhead [11, 29, 54, 6]. However, it still generates the tokens that describe a bounding box one by one. Inspired by the box-aligned multi-token formulation of [54], we adapt the block decoding to STVG, referring to this formulation as Sequential Block Decoding. Here, each temporal interval or bounding box is predicted as a single joint block. This removes dependencies within a localization unit but not across them. As a result, the tube is still decoded autoregressively box by box, so decoding depth grows with the tube length.

Removing dependencies inside a box does not remove the dependency between boxes, and this is costly in two ways for STVG. First, it sets a decoding depth that *grows linearly* with tube length, and secondly, it hurts *accuracy*: conditioning each box on an expanding history of generated coordinates lets an early error propagate along the trajectory and shifts the model's attention toward its own output and away from the visual evidence (Figure 4b and c). Neither cost is inherent to the STVG task, since the correct box at a time step is determined by the query and the visual content at that step, not by the boxes emitted before it.

To address these challenges, we introduce **Parallel Tube Decoding (PTD)**, a generative paradigm for efficient and accurate spatio-temporal video grounding. Our key idea is to decompose tube generation into a temporal block that identifies the queried event and a set of time-conditioned spatial blocks, one per temporal position, that are predicted in parallel across the interval. Tube generation therefore takes two decoding rounds, one temporal and one spatial, regardless of tube length. To enable this while preserving the model's native autoregressive capability, we jointly train complementary next-token and multi-token formulations and introduce Decoupled Block Attention, which lets every spatial block access the shared video-query context while blocking dependencies on other predicted boxes. We further add localization-aware policy optimization, improving temporal boundaries and box geometry through complementary spatio-temporal rewards.

The main contributions of this work are:

- **Decoding formulation.** We study four decoding strategies for generative STVG and introduce Parallel Tube Decoding, reducing tube generation from $4+7T$ rounds under Quantized Token Decoding and $1+T$ under Sequential Block Decoding to just two rounds, independent of the number of boxes (Sec. 3.1).
- **Attention mechanism.** We enable PTD through Decoupled Block Attention, which removes cross-box dependencies while preserving shared multimodal evidence, keeping each box grounded in the visual content at its own temporal position (Sec. 3.2).

- **Localization-aware optimization.** Policy optimization with complementary temporal and spatial rewards directly improves event boundaries and bounding-box geometry (Sec. 3.4).
- **Efficiency and generalization.** PTD cuts Tube Completion Latency by 79× and raises spatial decoding throughput by 92× over unquantized token decoding while improving localization quality (Sec. 4.2). With a 4B backbone it reaches state-of-the-art or competitive STVG performance (Sec. 4.3) and generalizes zero-shot to temporal grounding, evidence-grounded VideoQA, and referring video object tracking (Sec. 4.4).

2 Related Works

Spatio-Temporal Video Grounding. Early STVG approaches rely on task-specific graph reasoning, cross-modal transformers, and DETR-style decoders to jointly infer temporal boundaries and regress the corresponding spatial tube [65, 58, 22, 23]. Concurrently, MLLMs have demonstrated strong grounding capabilities in static images through coordinate or region generation [42, 10, 37], while video MLLMs extend this capability to temporal localization through timestamp-aware representations and event-boundary prediction [28, 46, 25]. Recent MLLM-based STVG methods build on these advances along two directions. LLaVA-ST and STVG-o1 directly serialize temporal boundaries and frame-wise boxes through the language decoder, preserving a unified generative interface [33, 24]. In contrast, many competitive approaches retain the MLLM primarily for semantic reasoning and temporal localization, while delegating dense spatial grounding to dedicated components: SpaceVLLM and Bridge-STG employ specialized spatial decoders [53, 51], DEViL couples the MLLM with open-vocabulary detectors [20], and STVG-R1 reformulates coordinate prediction as instance-ID selection over preconstructed object trajectories [63]. These formulations expose a central trade-off: direct generation retains a unified MLLM output space but produces long serialized tubes, whereas decoupled systems reduce the generation burden by shifting spatial localization to additional modules.

Efficient Localization Decoding. This architectural distinction has also shaped how recent STVG methods address inference efficiency. Bridge-STG and DEViL explicitly reduce MLLM output length and decoding cost by offloading dense spatial localization to a dedicated decoder or detector [51, 20]. For models that retain localization within the language decoder, discrete spatial and temporal tokens reduce representation overhead, but the complete tube remains restricted by token-by-token generation. Recent advances in multi-token prediction and block-based generation mitigate this limitation by predicting several tokens jointly, although they typically operate on arbitrary token spans or preserve causal dependencies across successive blocks [36, 21, 2, 40, 61]. Specifically, LocateAnything treats the entire coordinate set of a bounding box as an atomic prediction block, significantly accelerating visual grounding while preserving geometric coherence [54]. Crucially, its box blocks remain causally ordered, such that generating a longer sequence of boxes still requires additional dependent decoding rounds. Based on the above discussion, efficient native STVG generation requires removing not only token-level dependencies within each box, but also the trajectory-level dependency across the complete spatio-temporal tube.

3 Method

In this section, we present our approach to efficient spatio-temporal video grounding (STVG). We begin by formulating the task and then discuss *four* decoding strategies that progressively reduce sequential localization dependencies (Sec. 3.1). To address the remaining sequential dependency, we introduce Parallel Tube Decoding (PTD) (Sec. 3.2). Next, we define latency and throughput metrics for decoding efficiency (Sec. 3.3) and present localization-aware policy optimization for improved spatial and temporal grounding (Sec. 3.4).

Task Formulation. Given a video and a referring expression, STVG requires identifying *when* the event described by the query occurs and *where* the referred entity is located throughout that interval. This is challenging as the entity may remain visible beyond the event, while its location can change substantially over time. The model performs temporal localization by predicting the interval $[t_s, t_e]$, and spatial localization by predicting a bounding box $b_i = (x1_i, y1_i, x2_i, y2_i)$ for each time index t_i within that interval. The output is $\mathcal{Y} = ([t_s, t_e], \mathcal{B})$, where $\mathcal{B} = (t_i, b_i)_{i=s}^{e}$ denotes the spatio-temporal tube. We represent the prediction as

$$\texttt{<time>}\ \texttt{<}t_s\texttt{>}\ \texttt{<}t_e\texttt{>}\ \texttt{</time>} \quad [\texttt{<}t_i\texttt{>}\ \texttt{<box>}\ \texttt{<}x1_i\texttt{>}\ \texttt{<}y1_i\texttt{>}\ \texttt{<}x2_i\texttt{>}\ \texttt{<}y2_i\texttt{>}\ \texttt{</box>}]_{i=s}^{e}\,. \tag{1}$$

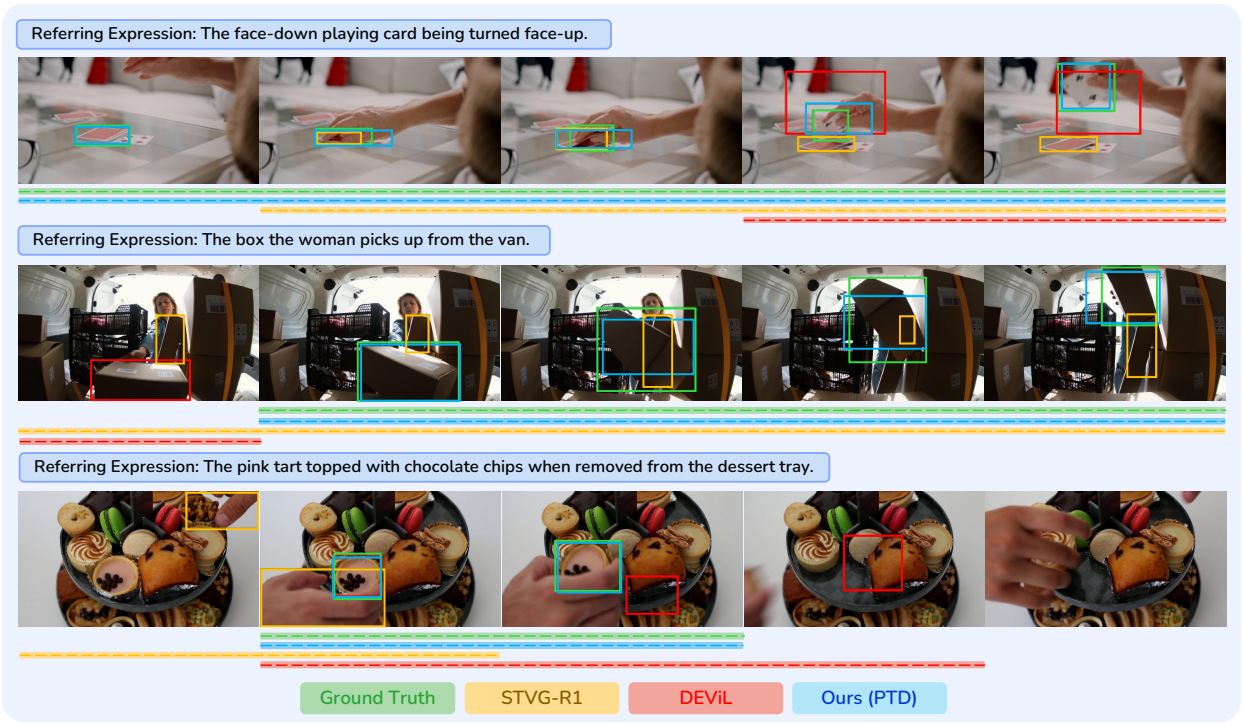

Figure 2 Qualitative comparison with prior STVG methods. We compare our model with competitive STVG methods on challenging grounding examples. Like many existing approaches, STVG-R1 [63] and DEViL [20] use the MLLM for temporal localization but construct the spatial tube using external detection or tracking modules. In contrast, PTD directly generates both the temporal interval and complete spatial tube, producing more accurate temporal boundaries and spatial localization. The examples highlight fine-grained target identification among visually similar distractors, large changes in object scale as the target approaches the camera, and event-specific temporal localization rather than simply predicting the full period in which the target is visible. Bounding boxes show spatial predictions, while the horizontal lines indicate the predicted temporal intervals.

3.1 From Autoregressive Localization to Parallel Tube Decoding

We compare four decoding strategies in terms of *sequential decoding depth*, defined as the number of dependent generation rounds required to produce the complete spatio-temporal tube.

i) Unquantized Token Decoding. Under standard autoregressive decoding, time indices and spatial coordinates are represented as text and generated through the model's language-token prediction [3, 24]. For example, a timestamp such as `3.0 seconds` or a coordinate such as `512` may be decomposed into multiple tokens depending on the tokenizer. Each localization can therefore require *several* sequential decoding steps, resulting in long generation sequences for dense spatio-temporal tubes.

ii) Quantized Token Decoding. Quantization addresses this representation overhead by mapping spatial and temporal locations to discrete tokens. Following prior quantized coordinate representations [11, 29, 54, 6, 33], normalized spatial coordinates are quantized into 1,001 bins, such that a coordinate such as 512 is represented by a single token `<512>`. For temporal localization, we similarly quantize text timestamps into discrete temporal tokens [52, 33]. Each video temporal patch is prefixed with its corresponding token $\texttt{<}t_i\texttt{>}$, and a time interval is represented as $\texttt{<time> <}t_s\texttt{> <}t_e\texttt{> </time>}$, where t_s and t_e denote the start and end positions.

This quantized representation makes the grounding output compact and *fixed in length*, while preserving standard autoregressive decoding. Following the output formulation above, the temporal interval requires *four* sequential token predictions, while each time-indexed box prediction requires *seven*: $\texttt{<}t_i\texttt{>}$, `<box>`, four coordinate tokens, and `</box>`. Thus, for a tube containing T boxes, the sequential decoding depth is $4 + 7T$. Quantization therefore improves token efficiency, but the complete tube is still generated *one token at a time*.

iii) Sequential Block Decoding. While quantization reduces token count, decoding remains fully autoregressive.

Multi-token prediction (MTP) [36, 61] reduces this sequential depth by predicting multiple tokens jointly. Following structured block decoding [54], we define each prediction block as a complete localization unit. For STVG, we define two block types: a temporal block, <time> $<t_s>$ $<t_e>$ </time>, and a time-indexed spatial block, $<t_i>$ <box> $<x1_i>$ $<y1_i>$ $<x2_i>$ $<y2_i>$ </box>. All tokens within a block are predicted jointly in a single decoding round. We use a block size of seven, padding the shorter temporal block with <null> tokens. Each target block is prompted using the final token of the preceding block followed by <mask> tokens, allowing all tokens in the block to be predicted in parallel [36, 54].

We refer to this formulation as *Sequential Block Decoding*. Although each block is decoded jointly, they are still generated sequentially along the tube. Thus, a tube with T boxes requires one round for the temporal block and T rounds for spatial localization, reducing the sequential decoding depth from $4+7T$ to $1+T$. This removes token-level dependencies within each block, but leaves a trajectory-level dependency across time. As the tube becomes longer, each additional spatial prediction still introduces another sequential decoding round.

iv) Parallel Tube Decoding. We introduce *Parallel Tube Decoding* (PTD) to remove the remaining trajectory-level dependency by predicting all spatial blocks in parallel after temporal localization. The sequential decoding depth therefore becomes $1+1$, independent of tube length T. Longer tubes increase only the parallel decoding width, rather than the number of sequential decoding rounds. Detailed formulations for all four decoding variants are provided in Appx. Sec. B.3.

3.2 Parallel Tube Decoding

Sequential block decoding removes token-level dependencies within each localization block, but still generates the spatial trajectory one time step at a time. However, this trajectory-level dependency is not inherent to STVG. Once the temporal interval is localized, spatial grounding at each time index can be performed directly from the corresponding visual evidence, without relying on boxes predicted at earlier time steps. Based on this observation, we introduce **Parallel Tube Decoding (PTD)**, which decomposes tube generation into two stages: a *temporal block* that identifies the relevant interval, followed by a set of **time-conditioned spatial blocks** that localize the referred entity throughout that interval. The key here is that all spatial blocks share the same multimodal context, while each is conditioned on its corresponding temporal token. Crucially, all spatial blocks are decoded in parallel, allowing the complete spatial trajectory to be generated in a single round.

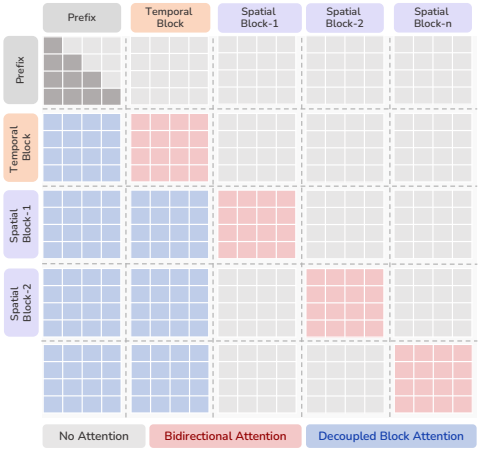

Figure 3 Illustration of Decoupled Block Attention in PTD. Each spatial block attends to the shared multimodal prefix and temporal block, uses bidirectional attention within the block, and remains isolated from other spatial blocks, enabling parallel tube generation.

Let P denote the shared multimodal prefix, consisting of the video, the referring expression, and the predicted temporal block. Each video temporal patch is associated with a temporal token $<t_i>$, which serves as the *temporal anchor* for its corresponding spatial block. Once the temporal boundaries $[t_s, t_e]$ are predicted, we directly instantiate the corresponding anchors $<t_i>_{i=s}^{e}$ from the localized interval rather than generating them autoregressively. Each anchor initializes a spatial block B_i with $<t_i>$, <mask>[5] as input, from which the model jointly predicts <box>, $<x1_i>$, $<y1_i>$, $<x2_i>$, $<y2_i>$, </box>. We therefore formulate tube prediction as

$$p(B_{s:e} \mid P) = \prod_{i=s}^{e} p(B_i \mid P, <t_i>)\,. \tag{2}$$

To train this parallel decoding formulation while retaining the model's native autoregressive behavior, we adopt a dual-formulation training strategy following [54]. Specifically, each training sample is supervised through two complementary target sequences: an NTP sequence that maintains standard causal generation and an MTP sequence that enables structured block prediction, representing the same sequence as a temporal block followed by a set of time-conditioned spatial blocks, $B_s, \ldots, B_e$. Both target sequences are concatenated after the shared video-query context and jointly optimized within a single forward pass using $\mathcal{L}_{\text{SFT}} = \mathcal{L}_{\text{NTP}} + \mathcal{L}_{\text{MTP}}$.

Our attention design combines three attention mechanisms to support parallel tube generation. i) The NTP sequence uses standard causal attention, preserving the model's autoregressive generation ability. ii) Tokens within each MTP block use bidirectional attention, allowing the structured localization unit to be predicted jointly [54, 18, 2]. iii) We introduce **Decoupled Block Attention** across time-conditioned spatial blocks. Each spatial block can attend to the shared multimodal prefix, including the predicted temporal block, but is restricted from attending to any other spatial block. Unlike sequential block decoding [54], which uses causal attention across blocks, our design eliminates this cross-block dependency and seamlessly enables the entire spatial trajectory to be decoded in a single generation round (see Appx. Figure 9 for attention masks). The NTP causal attention cannot attend to the MTP blocks, preventing information leakage during joint training.

3.3 Latency Metrics

To quantify the decoding efficiency of different decoding strategies, we consider two complementary wall-clock metrics. i) **Tube Completion Latency (TCL)** measures the time required to complete the spatio-temporal tube prediction once generation begins. Let T_{full} denote the end-to-end generation time and T_{TTFT} the time to first token. We define TCL $= T_{\text{full}} - T_{\text{TTFT}}$, thereby excluding the initial video and prompt processing cost and isolating the latency associated with tube generation. ii) **Boxes Per Second (BPS)** measures spatial decoding throughput. For a prediction containing T time-conditioned spatial boxes, we define BPS $= T/\text{TCL}$. These two metrics capture complementary aspects of decoding efficiency: TCL measures how quickly the complete spatio-temporal tube becomes available, while BPS measures the rate at which spatial predictions are produced. Importantly, they expose how each decoding strategy scales with tube length T. Token-based decoding and Sequential Block Decoding require increasingly more dependent generation rounds as the tube grows, whereas PTD significantly accelerates tube generation. We evaluate this scaling in Sec. 4.2.

3.4 Localization-Aware Policy Optimization

PTD replaces temporally serialized tube generation with parallel time-conditioned spatial prediction. While SFT supervises this formulation through token-level cross-entropy, it does not directly optimize the geometric quality of the resulting tube. We therefore further train the model with localization-aware Group Relative Policy Optimization (GRPO) [49], using complementary rewards for temporal and spatial grounding. Recent studies have also explored GRPO for STVG through localization-specific rewards [24, 63].

i) Temporal localization reward. A central challenge in STVG is to localize the queried event within a longer temporal window where the referred entity may remain continuously visible. A second challenge arises when the event is temporally brief, making its precise boundaries difficult to localize. Consequently, the model may either fail to localize the event at the correct time or concentrate on a short, visually salient portion while missing earlier or later phases of the interaction. Further, even accurate spatial predictions cannot recover a tube whose temporal boundaries are incorrect. To explicitly optimize this, we use temporal IoU as an interval-level reward between the predicted temporal segment $[\hat{t}_s, \hat{t}_e]$ and the target segment $[t_s, t_e]$,

$$R_{\text{temp}} = \text{tIoU}\left([\hat{t}_s, \hat{t}_e], [t_s, t_e]\right). \tag{3}$$

ii) Spatial localization reward. Once the relevant event interval is identified, the next challenge is to precisely localize the referred entity within the selected frames. The predicted tube may still contain boxes that are spatially shifted, overly loose, or gradually drift toward nearby objects or background regions. These localization errors require spatial supervision that captures both region-level overlap and coordinate-level precision. We therefore combine generalized IoU [47], which measures the geometric agreement between the predicted and target regions, with an L_1 coordinate penalty that further discourages shifted or loosely fitted boxes. Since temporal extent is already optimized by the temporal localization reward, we evaluate spatial quality only over the temporal intersection $\mathcal{I} = [\hat{t}_s, \hat{t}_e] \cap [t_s, t_e]$, where the predicted and target tubes are temporally aligned.

$$R_{\text{spat}} = \frac{1}{|\mathcal{I}|} \sum_{t_i \in \mathcal{I}} \left[\text{GIoU}(\hat{b}_i, b_i) - \left\| \hat{b}_i - b_i \right\|_1 \right]. \tag{4}$$

We combine the temporal and spatial localization rewards into a single objective for policy optimization. Since

both components capture complementary aspects of the spatio-temporal tube, we assign them equal weight and define the total reward as $R = R_{\text{temp}} + R_{\text{spat}}$. This jointly encourages accurate temporal localization and precise spatial grounding within the localized interval.

4 Experiments

4.1 Experimental Setup

We build our model on Qwen3-VL-4B [3] and equip it with discrete spatial and temporal localization tokens for structured spatio-temporal grounding. Specifically, we augment the vocabulary with 1,001 spatial tokens representing normalized coordinates in $[0, 1000]$ and 100 temporal tokens representing discretized timestamps. These temporal tokens are interleaved with the video representation, providing an explicit temporal anchor for each temporal patch and enabling the time-conditioned spatial blocks used by PTD. We use a temporal patch size of 1 for finer temporal granularity. We initialize the newly introduced localization tokens from semantically aligned Qwen3-VL vocabulary embeddings. Additional details on the localization-token representation and temporal anchoring are provided in Appx. Secs. B.1 and B.2.

Training stages. We train our model in two stages. We first perform supervised fine-tuning (SFT) using the dual NTP/MTP formulation introduced in Sec. 3.2, enabling the model to support both autoregressive generation and Parallel Tube Decoding (PTD) within the same training stage. We continue training the model with localization-aware Group Relative Policy Optimization (GRPO), using the temporal reward R_{temp} and spatial reward R_{spat} defined in Sec. 3.4 to further improve localization quality. For both SFT and GRPO, we use LoRA [27] with rank 32 and optimize the LoRA parameters together with the embeddings of the newly introduced localization tokens. For the controlled comparison in Sec. 4.2, all four variants are trained for 1 SFT epoch, while the main PTD model is trained for 2 epochs with all other settings unchanged. SFT is trained with a learning rate of 2×10^{-5} using the dual NTP/MTP formulation, while GRPO uses only the PTD formulation with a learning rate of 5×10^{-6}, $\beta = 0.04$, and a sampling temperature of 0.9. Videos are sampled at 2 FPS with at most 64 frames. Additional training details are provided in Appx. Sec. C.

Training data. For SFT, we combine the training splits of VidSTG [65] and HC-STVG [50]. VidSTG provides both declarative and interrogative referring expressions, while HC-STVG-v1 and HC-STVG-v2 further introduce human-centric grounding samples involving attributes, actions, and interactions in multi-person scenes. Together, these datasets yield approximately 90K SFT samples. For GRPO, rather than uniformly reusing the complete SFT pool, we construct a more informative subset that emphasizes samples with meaningful variation in localization quality. Specifically, for each training sample, the SFT model generates eight candidate responses, which are evaluated using temporal IoU (tIoU) and video IoU (vIoU). We preferentially retain samples where the sampled responses exhibit substantial within-group variation in temporal or spatio-temporal correctness, providing a stronger relative learning signal for policy optimization [59]. In contrast, samples that are consistently solved offer limited supervision, while those for which all rollouts fail provide little discrimination among candidate responses. Based on this criterion, we construct a 16K GRPO training set.

Evaluation protocol. We evaluate our approach along *three* complementary dimensions: **(i) decoding paradigm** (Sec. 4.2), where we examine how progressively reducing sequential decoding dependencies affects grounding quality and generation efficiency; **(ii) in-domain STVG performance** (Sec. 4.3), where we compare our model with prior methods under the standard STVG evaluation setting; and **(iii) generalization beyond STVG** (Sec. 4.4), where we assess zero-shot transfer to temporal grounding, grounded VideoQA, and video object tracking. Dataset splits, prompts used, and task-specific evaluation details are provided in Appx. Sec. D.

Decoding Efficiency. To characterize the efficiency of different localization strategies, we use the metrics introduced in Sec. 3.3: Tube Completion Latency (TCL) to measure the time required to complete the predicted tube and Boxes Per Second (BPS) to quantify spatial decoding throughput. For all efficiency evaluations, we use batch size 1 and BF16 inference on a single 64-GB AMD Instinct MI210 GPU using PyTorch 2.7 and ROCm 6.3. We synchronize GPU operations and measure decode-only time from the first output token to tube completion, excluding multimodal prefill, under the same hardware and runtime.

Model Design	Declarative				Interrogative				Efficiency	
	m_{tIoU}	vIoU			m_{tIoU}	vIoU			TCL (s)	BPS
		Mean	@0.3	@0.5		Mean	@0.3	@0.5	↓	↑
Unquantized Token Decoding	42.8	27.4	39.8	27.1	40.8	22.6	32.0	20.9	31.6	0.5
Quantized Token Decoding	43.3	28.7	40.6	27.3	42.0	23.0	32.2	21.2	9.1	1.5
Sequential Block Decoding	47.2	31.2	42.8	28.7	46.3	26.0	35.5	22.5	1.0	8.8
Parallel Tube Decoding (PTD)	**47.5**	**32.9**	**46.0**	**30.5**	**46.7**	**27.2**	**37.8**	**23.3**	**0.4**	**45.9**

Table 1 We report spatio-temporal grounding performance for declarative and interrogative queries, together with decoding efficiency measured by tube completion latency (TCL) and boxes per second (BPS). Quantization reduces decoding cost, sequential block prediction further improves efficiency, and the proposed **Parallel Tube Decoding (PTD)** achieves the strongest grounding performance while substantially reducing TCL and increasing BPS.

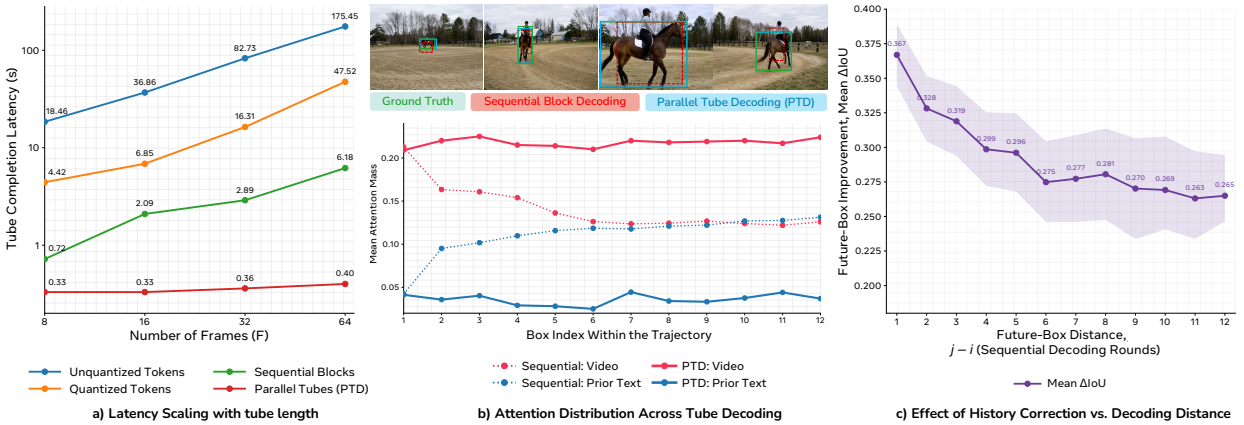

Figure 4 Analysis of decoding efficiency and trajectory-level dependency. (a) Tube completion latency as the number of grounded frames increases. PTD maintains nearly constant latency, while token-based and block decoding scale with tube length. (b) Attention distribution across tube decoding for Sequential Block Decoding (dotted) and PTD (solid). Sequential decoding progressively shifts attention from the video toward prior text. (c) History-correction analysis for Sequential Block Decoding. Replacing an erroneous box B_i with its ground-truth box improves subsequent predictions, with the effect gradually decreasing as the decoding distance $j - i$ increases.

4.2 Evaluating the Decoding Paradigm

We first evaluate how the decoding paradigm affects both grounding quality and generation efficiency. In Table 1, we compare the four decoding strategies introduced in Sec. 3.1 on VidSTG [65] and report tIoU and vIoU for grounding quality. This comparison reveals a clear progression as sequential dependencies are reduced: Unquantized Token Decoding generates textual localization values autoregressively, Quantized Token Decoding replaces each value with a discrete token, Sequential Block Decoding jointly predicts structured localization blocks while remaining sequential across time, and our Parallel Tube Decoding (PTD) removes this remaining dependency by decoding all time-conditioned spatial blocks in parallel after temporal localization. Here, the unquantized variant uses the Qwen3-VL baseline decoding scheme. We adapt existing decoding strategies for STVG to build the quantized and block decoding variants, and propose PTD on top of them. For fair comparison, all variants are trained under the same setting and differ in output representation and decoding strategy used at inference. Across the four variants, the mean predicted tube lengths are 12.3, 13.7, 14.2, and 17.7 boxes, respectively. Sequential Block Decoding serves as the natural ablation of Decoupled Block Attention. Replacing the decoupled attention mask with standard causal inter-block attention allows each spatial block to attend to preceding box blocks, thereby restoring sequential dependencies across time and preventing parallel block decoding.

The results show that progressively reducing sequential decoding dependencies improves both grounding quality and generation efficiency. Quantization provides an immediate efficiency gain over unquantized, reducing

Model	Scale	Declarative Sentences				Interrogative Sentences			
		m_{tIoU}	m_{vIoU}	vIoU@0.3	vIoU@0.5	m_{tIoU}	m_{vIoU}	vIoU@0.3	vIoU@0.5
Backbone Baselines									
Qwen2.5-VL-7B (ZS) [4]	7B	16.8	10.9	14.3	5.4	13.8	8.5	11.3	4.4
Qwen2.5-VL-7B + SFT† [24]	7B	41.6	20.3	26.1	15.4	40.9	17.1	17.6	13.9
Qwen3-VL-4B (ZS) [3]	4B	36.2	13.1	16.6	7.0	36.1	8.9	10.2	3.8
Qwen3-VL-8B (ZS) [3]	8B	37.0	13.4	16.5	7.1	35.0	9.3	11.0	3.9
Qwen3-VL-8B + SFT‡ [20]	8B	42.8	22.4	29.6	15.3	41.4	18.2	19.6	12.8
Prior MLLM Methods									
GroundingGPT [35]	7B	15.5	12.3	13.2	4.1	11.9	8.7	9.6	2.9
Gemini-2.5-Pro [15]	–	49.9	22.5	33.6	12.0	45.4	13.7	17.3	8.1
LLaVA-ST [33]	7B	45.5	24.8	36.0	22.9	43.2	20.0	28.1	17.5
SpaceVLLM-7B [53]	7B	47.7	27.4	39.1	26.2	48.5	25.4	35.9	22.2
DEViL (VideoLLaMA3-7B) [20]	7B	50.2	33.6	46.5	34.0	48.5	28.8	38.7	28.2
STVG-o1 (Qwen2.5-VL-7B) [24]	7B	52.1	33.5	48.4	32.0	50.5	27.9	39.8	26.0
Bridge-STG (Qwen3-VL-7B) [51]	7B	52.6	37.2	52.4	37.4	50.1	31.3	43.8	**31.2**
Ours (Qwen3-VL-4B)									
SFT (NTP)	4B	46.3	31.6	45.1	30.5	44.7	26.3	36.9	25.3
SFT (PTD)	4B	50.0	34.9	48.5	33.3	48.2	28.7	40.2	25.6
GRPO (PTD)	4B	**53.7**	**38.3**	**53.0**	**37.5**	**52.2**	**32.4**	**44.9**	30.3

Table 2 Comparison with prior work on **VidSTG** under declarative and interrogative settings. We report the model scale, mean temporal IoU (m_{tIoU}), mean video IoU (m_{vIoU}), and vIoU at 0.3 and 0.5 thresholds. † denotes the Qwen2.5-VL SFT baseline reported in STVG-o1 [24], while ‡ denotes the Qwen3-VL SFT baseline reported in DEViL [20]. Our results use Qwen3-VL-4B as the backbone. **SFT (NTP)** uses supervised fine-tuning with standard next-token autoregressive decoding, whereas **SFT (PTD)** uses the same supervised training objective with Parallel Tube Decoding at inference. **GRPO (PTD)** further optimizes the Qwen3-VL-4B model with GRPO while retaining Parallel Tube Decoding.

TCL from 31.6s to 9.1s while preserving comparable grounding quality. Block decoding further improves both efficiency and localization performance, reaching 1.0s TCL and 8.8 BPS while consistently improving tIoU and vIoU across both query types. A key advantage of PTD is that it removes the remaining trajectory-level dependency without sacrificing grounding quality. PTD achieves the strongest performance across all grounding metrics, while further reducing TCL to 0.4s and increasing throughput to 45.9 BPS. Compared with the Qwen3-VL-4B baseline using Unquantized Token Decoding [3], PTD reduces tube completion latency by 79× and increases spatial decoding throughput by 92×, demonstrating that parallel tube generation substantially accelerates inference while simultaneously improving spatio-temporal grounding.

We further study **latency scaling with tube length**. In Figure 4, we vary the number of boxes from 8 to 64 to examine how each decoding strategy scales as the spatial trajectory becomes longer. The latency of token-based and Sequential Block Decoding increases substantially with tube length, reflecting the growing number of dependent generation rounds. In contrast, PTD exhibits nearly constant generation time, increasing only from 0.33s to 0.40s as the tube length grows by 8×, whereas block decoding rises substantially from 0.72s to 6.18s over the same range. Notably, this advantage becomes increasingly pronounced for longer tubes, where PTD avoids the latency growth incurred by sequential decoding. These results demonstrate the key scaling advantage of PTD: longer trajectories increase parallel decoding width rather than sequential decoding depth, enabling efficient tube generation with nearly tube-length-independent latency.

Cross-box dependency and error propagation. We investigate why removing trajectory-level dependencies leads to stronger spatial localization. A key observation is that the two decoding strategies exhibit different attention behavior as tube generation progresses. Figure 4b illustrates a qualitative example where block decoding progressively shifts attention from the video toward the growing localization history (dotted lines). In contrast, PTD maintains the video attention (solid lines). We directly quantify the resulting error propagation in Figure 4c, through a history-correction intervention. For a mislocalized source box B_i, we replace only B_i with its ground-truth box and autoregressively re-decode each subsequent box B_j. The x-axis $j - i$ measures the number of sequential decoding rounds between the corrected and evaluated boxes, while the y-axis reports

Model	Scale	HC-STVG v1				HC-STVG v2			
		m_{tIoU}	m_{vIoU}	vIoU@0.3	vIoU@0.5	m_{tIoU}	m_{vIoU}	vIoU@0.3	vIoU@0.5
Backbone Baselines									
Qwen2.5-VL-7B (ZS) [4]	7B	25.6	19.1	20.2	12.6	22.9	13.0	15.6	6.4
Qwen2.5-VL-7B (SFT†) [24]	7B	53.5	28.6	45.2	21.9	55.3	26.5	38.6	20.2
Qwen3-VL-4B (ZS) [3]	4B	44.6	19.5	25.4	4.9	45.2	19.4	24.7	5.5
Qwen3-VL-8B (ZS) [3]	8B	47.6	21.5	30.3	6.5	53.1	21.9	30.0	6.6
Prior MLLM Methods									
GPT-4o [41]	–	27.5	7.9	4.0	0.3	32.7	9.1	5.7	0.0
Gemini-2.5-Pro [15]	–	55.1	25.9	39.1	9.9	60.4	24.5	34.6	9.6
GroundingGPT [35]	7B	22.2	16.7	15.0	4.9	19.6	14.7	16.6	3.1
LLaVA-Video SFT [64]	7B	52.8	27.7	43.1	21.3	54.2	24.8	40.1	15.5
SpaceVLLM [53]	7B	56.9	39.3	66.6	36.9	58.0	34.0	56.9	24.7
STVG-R1 (Qwen2.5-VL-7B) [63]	7B	56.9	39.1	66.7	38.6	62.0	40.2	67.8	38.8
DEViL (VideoLLaMA3-7B) [20]	7B	59.0	43.1	70.5	44.3	61.7	42.5	67.3	42.2
Bridge-STG (Qwen3-VL-7B) [51]	7B	–	–	–	–	**64.1**	41.5	67.5	38.6
STVG-o1 (Qwen2.5-VL-7B) [24]	7B	**60.3**	44.1	73.3	43.5	63.8	41.2	68.5	39.6
Ours (Qwen3-VL-4B)									
SFT (NTP)	4B	55.2	41.0	66.5	41.7	54.4	36.5	60.3	29.7
SFT (PTD)	4B	55.7	42.7	68.6	42.5	57.0	39.2	64.7	34.0
GRPO (PTD)	4B	59.4	**45.6**	**73.8**	**46.7**	63.1	**43.6**	**71.0**	**42.2**

Table 3 Comparison with prior work on **HC-STVG v1** and **HC-STVG v2**. We report the model scale, mean temporal IoU (m_{tIoU}), mean video IoU (m_{vIoU}), and vIoU at thresholds 0.3 and 0.5. † denotes the Qwen2.5-VL SFT baseline reported in STVG-o1. Our models use Qwen3-VL-4B as the backbone. **SFT (NTP)** uses standard next-token prediction, while **SFT (PTD)** uses Parallel Tube Decoding. **GRPO (PTD)** further optimizes the PTD model with GRPO.

the resulting mean ΔIoU relative to decoding with the original history. Correcting B_i produces the largest improvement for nearby future boxes, with the effect gradually decreasing with decoding distance. Additional qualitative comparisons between Sequential Block Decoding and PTD are provided in Appx. Figure 8.

4.3 Main Results on Spatio-Temporal Video Grounding

We evaluate the spatio-temporal grounding performance of our model on VidSTG [65] and HC-STVG-v1/v2 [50], comparing it with backbone baselines and prior STVG methods.

i) VidSTG. As shown in Table 2, our model establishes the strongest overall performance on VidSTG, leading on seven of eight metrics across declarative and interrogative queries. It improves the previous best m_{tIoU} by +1.1 and +1.7 points for declarative and interrogative queries, respectively. The gains are even more pronounced when evaluating the full spatio-temporal tube, which requires both accurate temporal boundaries and consistent spatial localization throughout the event. Our model improves the previous best m_{vIoU} by +1.1 points for both query types, while remaining consistently strong under stricter tube-overlap thresholds.

ii) HC-STVG. A similar advantage extends to the more challenging human-centric HC-STVG benchmarks, as shown in Table 3. While temporal localization remains competitive with the best prior methods, the gains are more pronounced when evaluating the full spatio-temporal tube. Our model improves the previous best m_{vIoU} by +1.5 points on HC-STVG-v1 and +1.1 points on HC-STVG-v2. This advantage is also reflected under stricter overlap criteria, where it achieves the best vIoU@0.3 on both benchmarks.

The progression within our model further clarifies where these gains originate. SFT (NTP) and SFT (PTD) correspond to two inference modes of the same checkpoint, jointly trained with the NTP and MTP formulations. The former uses Quantized Token Decoding, while the latter uses Parallel Tube Decoding. As already shown in Table 1, PTD improves both grounding quality and decoding efficiency over NTP. Building on this, localization-aware policy optimization further improves grounding quality while retaining the same PTD formulation. A key advantage is that these gains are achieved with a compact Qwen3-VL-4B backbone that directly generates both the temporal interval and the complete spatial tube. In contrast, several competitive

Model	Charades-STA			
	R@0.3	R@0.5	R@0.7	mIoU
TRACE-7B [26]	–	40.3	19.4	–
TimeSuite-7B [62]	69.9	48.7	24.0	–
DEViL [20]	72.6	51.5	25.2	47.7
STVG-R1-7B [63]	73.2	52.5	–	–
Ours	**78.9**	**58.4**	**30.1**	**52.4**

Table 4 Temporal grounding on Charades-STA. We compare all models in the zero-shot setting, evaluating how accurately they localize query-relevant temporal segments at different IoU thresholds.

Model	ActivityNet Captions			
	R@0.3	R@0.5	R@0.7	mIoU
UniTime-Zero [34]	–	22.8	14.1	27.3
Momentor [43]	–	23.0	12.4	29.3
Qwen2.5-VL-7B [4]	25.5	13.4	6.1	19.1
VideoChat-TPO [57]	42.6	26.3	13.0	27.6
VideoChatR1.5-7B [56]	52.4	32.3	16.8	35.5
Ours	**63.7**	**41.4**	**23.9**	**44.6**

Table 5 Temporal grounding on ActivityNet. We compare all models in the zero-shot setting, evaluating temporal localization accuracy across multiple IoU thresholds and overall mean performance.

methods rely on additional spatial components: SpaceVLLM uses a specialized prediction head [53], DEViL an external detector [20], Bridge-STG a dedicated spatial decoder [51], and STVG-R1 a detection-and-tracking pipeline [63]. Despite using a smaller backbone and no dedicated spatial localization module, our model remains competitive with or surpasses 7B-scale models. Further, the substantial gap over zero-shot Qwen3-VL-4B/8B and existing Qwen3-VL SFT baselines shows that these improvements cannot be attributed to the backbone alone, but are instead driven by the combination of the proposed grounding formulation and localization-aware policy optimization.

4.4 Generalization Beyond STVG

In this section, we examine whether the learned grounding capabilities transfer beyond the training distribution across three settings: temporal grounding, evidence-grounded VideoQA, and referring video object tracking.

i) Zero-shot Temporal Grounding. We first examine whether the temporal localization capability learned from STVG generalizes to two unseen temporal grounding benchmarks with distinct video distributions. Charades-STA [19] primarily contains short indoor videos with densely occurring human activities, whereas ActivityNet [32] contains longer videos covering a broader range of activities. We compare our model with prior methods evaluated under the same zero-shot setting. As shown in Table 4 and Table 5, our model demonstrates strong performance on both datasets. On Charades-STA, it improves the mean performance over the strongest prior zero-shot method by +4.7 points. The advantage is more pronounced on ActivityNet, where our model surpasses the strongest zero-shot baseline by +9.1 points in m_{IoU}. Notably, the gains remain consistent under stricter overlap criteria, indicating that the model not only retrieves the relevant temporal region but also predicts more precise event boundaries.

ii) Zero-shot Grounded VideoQA. We further evaluate zero-shot generalization on ReXTime [9], where the model is required to answer each question and provide supporting evidence through temporal localization. As shown in Table 6, our model demonstrates strong performance despite receiving no task-specific supervision. A key advantage is its temporal grounding accuracy: compared with the strongest zero-shot baseline, our model improves m_{IoU} by +15.6 points, while maintaining competitive answer accuracy. Notably, it also surpasses the finetuned VideoChat-R1.5 [56] in m_{IoU} by +2.0 points. These results show that our model seamlessly supports evidence-aware video question answering without additional finetuning.

iii) Zero-shot Video Object Tracking. We further examine whether the spatial grounding capability learned from STVG generalizes to referring video object tracking, where the model must maintain consistent localization of the queried object throughout the video. As shown in Table 7, our Qwen3-VL-4B-based model demonstrates strong zero-shot performance on Ref-DAVIS [30], Ref-YT-VOS [48], and ReasonVOS [5] when paired with off-the-shelf segmentation models. For a controlled comparison, Molmo2-4B [14] similarly relies on SAM2 [45] to propagate object masks, but provides point prompts, whereas our model generates bounding boxes. Using the same SAM2 backend, our model surpasses Molmo2-4B by +8.4, +2.6, and +0.2 points on the three benchmarks, respectively, despite the latter being finetuned on Ref-DAVIS and Ref-YT-VOS. Further, using SAM3 [7] consistently improves performance across all three datasets.

Model	ReXTime				
	Acc		IoU		
	Overall	@0.5	Mean	@0.3	@0.5
Finetuned					
TimeChat [46]	49.4	11.1	26.5	40.5	21.9
VTimeLLM [28]	58.2	18.3	30.0	44.1	26.6
GraphThinker [13]	71.3	30.8	41.5	57.5	40.4
VideoChat-R1.5 [56]	74.8	38.1	45.8	61.8	46.4
Zero-Shot					
VTimeLLM [28]	36.2	–	20.1	28.8	17.4
GraphThinker [13]	66.8	15.2	25.3	33.9	20.3
VideoTG-R1 [17]	75.7	25.9	32.2	41.2	31.7
Ours (ZS)	73.1	**43.0**	**47.8**	**62.1**	**49.2**

Table 6 Grounded VideoQA on ReXTime. We evaluate on ReXTime [9], which requires models to answer questions while localizing the temporal evidence supporting each prediction to measure joint reasoning and temporal localization.

Model	Video Object Tracking		
	Ref-DAVIS	Ref-YT-VOS	ReasonVOS
Finetuned			
Molmo [16] + SAM2 [45]	65.2	–	45.7
Sa2VA-Qwen3VL-4B [60]	76.0	68.1	50.0
VideoGLaMM-7B [39]	69.5	66.8	33.9
VideoMolmo-7B [1]	72.5	67.3	51.1
Molmo2-4B [14]	73.5	70.2	61.9
Zero-Shot			
Qwen3-VL-4B [3]	44.4	32.1	26.5
Qwen3-VL-8B [3]	41.0	48.3	24.9
Ours (ZS) + SAM2	81.9	72.8	62.1
Ours (ZS) + SAM3	**82.9**	**74.9**	**64.7**

Table 7 Video object tracking. We evaluate on referring video object segmentation benchmarks [30, 48, 5] using the $\mathcal{J}\&\mathcal{F}$ metric. Gray entries denote finetuned results, while ReasonVOS is evaluated zero-shot.

Reward	Declarative Sentences				Interrogative Sentences			
	m_{tIoU}	m_{vIoU}	vIoU@0.3	vIoU@0.5	m_{tIoU}	m_{vIoU}	vIoU@0.3	vIoU@0.5
SFT	50.0	34.9	48.5	33.3	48.2	28.7	40.2	25.6
Temporal	53.8	37.4	51.9	36.4	52.1	31.1	43.3	28.0
Spatial	51.8	37.2	51.6	35.9	49.9	31.3	43.2	29.2
Temporal + Spatial	**53.7**	**38.3**	**53.0**	**37.5**	**52.2**	**32.4**	**44.9**	**30.3**

Table 8 Ablation of the reward design for GRPO on **VidSTG**. We compare temporal-only, spatial-only, and the full temporal-spatial reward under declarative and interrogative settings. All variants use Qwen3-VL-4B with Parallel Tube Decoding.

4.5 Ablation Studies

Reward design. In Table 8, we examine how the temporal and spatial rewards contribute to localization-aware policy optimization. With the temporal reward, we observe that the model corrects two common failure patterns: temporally displaced predictions and incomplete event coverage, where only the most salient part of an interaction is localized. Recovering a better-aligned event interval also improves m_{vIoU}, since accurate spatial predictions can only contribute to the tube when they are generated over the correct temporal interval. Spatial reward addresses complementary localization errors in which the predicted boxes do not tightly align with the referred entity, include excessive surrounding context, or lose target consistency across frames. Combining both rewards achieves the strongest overall accuracy. We perform GRPO using outcome-level temporal and spatial rewards without introducing thinking tokens to avoid the additional decoding overhead. Qualitative results on the effects of the temporal and spatial rewards are shown in Appx. Figures 6 and 7.

5 Limitations

Our qualitative analysis in Figure 5 reveals two representative sources of error. We observe that spatial localization becomes more difficult when target identity must be recovered after occlusion or when the target is small and rapidly moving, which can lead to imprecise boxes (bottom). For temporal localization, failures are concentrated around short state transitions with visually ambiguous boundaries, causing the predicted interval to extend beyond or end before the complete event (top).

Beyond these failure cases, the current study remains bounded by the scope of existing STVG formulations

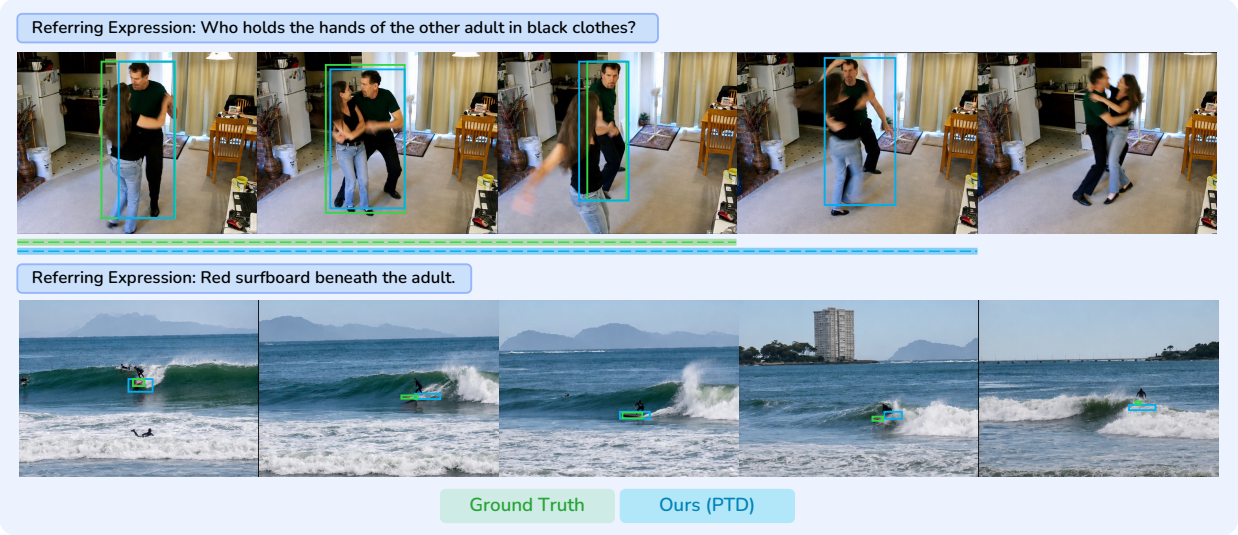

Figure 5 Representative failure cases. Temporally subtle state changes can produce ambiguous event boundaries (top), while spatial localization becomes difficult for small, rapidly moving, or occluded targets (bottom). Green and blue denote the ground-truth and PTD predictions, respectively; the horizontal lines indicate their temporal intervals.

and training resources. Standard STVG benchmarks represent each query using a single continuous temporal interval and one spatial tube, leaving multiple disjoint event occurrences, temporary out-of-view periods caused by camera motion, and multiple instances satisfying the same referring expression largely unexplored. Supporting these scenarios is important for real-world applications and motivates extending PTD toward multi-segment and multi-instance grounding. Moreover, large-scale STVG training data with dense tube annotations remains limited, with VidSTG [65] and HC-STVG-v1/v2 [50] serving as the primary datasets available for this setting. In future work, we plan to expand the scope toward more generalized grounding settings by supporting richer multi-segment and multi-instance outputs while diversifying the training data across broader video domains, entities, events, and interaction types.

6 Conclusion

We introduced Parallel Tube Decoding (PTD), a generative formulation for spatio-temporal video grounding that removes the need to decode dense spatial trajectories autoregressively. By first predicting the relevant temporal interval and then generating all time-conditioned spatial blocks in parallel, PTD reduces tube generation to just two sequential decoding rounds, independent of tube length. Decoupled Block Attention further removes dependencies between spatial predictions while preserving access to the shared multimodal evidence, and localization-aware policy optimization improves temporal boundaries and bounding-box geometry.

Our experiments show that removing these sequential dependencies improves both efficiency and localization quality. On VidSTG, PTD reduces Tube Completion Latency by 79× and increases spatial decoding throughput by 92× relative to standard autoregressive decoding, while achieving stronger grounding accuracy. Using a 4B backbone, the resulting model performs favorably well on VidSTG and HC-STVG, despite relying on no dedicated spatial localization module. The learned grounding capability also transfers zero-shot to temporal grounding, grounded VideoQA, and referring video object tracking. Overall, our results show that dense spatio-temporal localization can be generated efficiently without sequential trajectory decoding.

Appendix

A Qualitative Analysis

A.1 Effect of Temporal and Spatial Rewards

Figure 6 Qualitative results illustrating the effect of the temporal reward. Green, red, and blue denote the ground-truth, SFT, and GRPO with temporal reward predictions, respectively. In the first example, SFT localizes only the most salient portion of the event despite accurate spatial boxes, whereas the temporal reward recovers the complete interaction. In the second, the temporal reward prevents the prediction from extending beyond the queried event while the target remains visible. These examples show how the temporal reward improves boundary prediction, ensuring that the generated tube aligns with the complete temporal extent of the queried event.

To complement the quantitative reward ablation in Table 8, we qualitatively examine how the temporal and spatial rewards correct distinct localization errors. For **temporal grounding**, Figure 6 shows that the SFT model localizes only the most salient portion of the caressing interaction in the first example. Although its boxes align well with the target during the predicted interval, the incomplete temporal coverage lowers vIoU, since accurate spatial predictions can only contribute to the tube when they are generated over the correct temporal interval. In the second example, the SFT model instead extends its prediction beyond the interval in which the cat is leaning on the adult. The temporal reward corrects both failure patterns, showing how it helps distinguish the queried event from the broader period of target visibility while covering the complete interaction rather than only its most salient portion. For **spatial grounding**, Figure 7 shows that the spatial reward produces tighter boxes and maintains localization of the rider throughout the motion-heavy biking example. In the second example, it preserves target consistency despite substantial changes in object scale and interference from nearby entities, avoiding the spatial drift exhibited by the SFT model. These comparisons show how the spatial reward improves both tight geometric alignment and consistent target identification under motion, scale changes, and interference from nearby distractors.

A.2 Qualitative Analysis of Cross-Box Dependencies

We further qualitatively compare Sequential Block Decoding and PTD to examine how removing trajectory-level dependencies affects spatial localization in Figure 8. In the first example, the baby is pushed back and forth in the stroller, resulting in substantial changes in target scale. Sequential Block Decoding fails to consistently adjust its box extent as the baby moves toward and away from the camera, whereas PTD closely follows the changing target size. In the second example, partial occlusion and close proximity to another person make consistent localization of the referred woman challenging. Block decoding produces increasingly

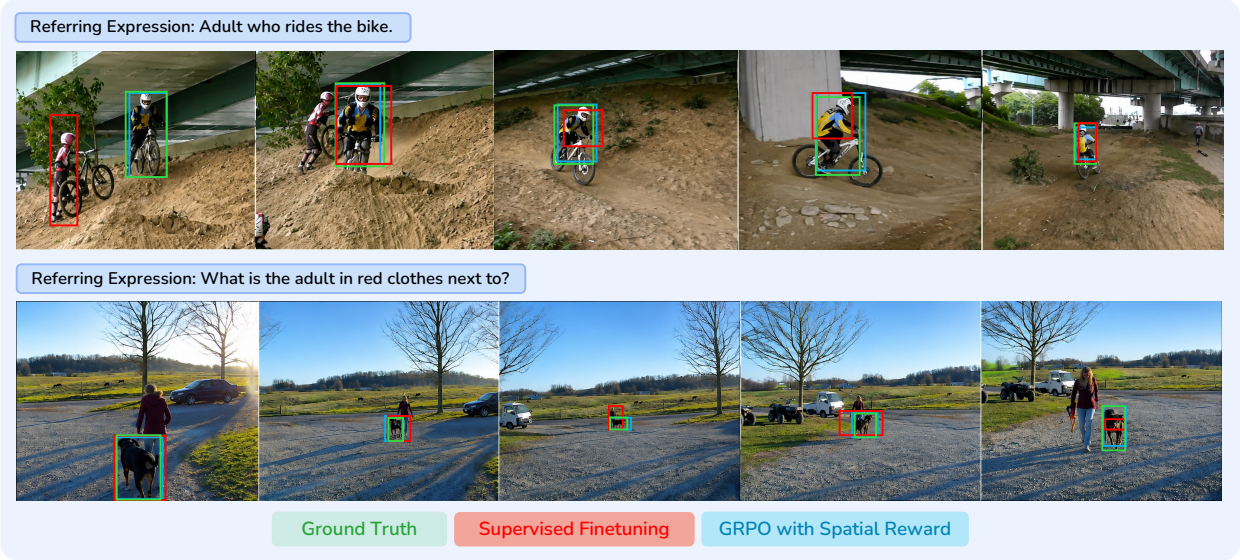

Figure 7 Qualitative results illustrating the effect of the spatial reward. Green, red, and blue denote the ground-truth, SFT, and GRPO with spatial reward predictions, respectively. In the first example, the spatial reward maintains tight localization of the rider under rapid motion. In the second, it preserves target consistency despite substantial changes in object scale and interference from nearby entities. These examples show how the spatial reward improves both bounding-box precision and consistent target localization across time.

loose boxes, while PTD remains focused on the referred woman and maintains tighter localization. These examples illustrate how conditioning on preceding boxes can cause earlier spatial estimates to influence later predictions, particularly under abrupt scale changes, occlusion, or ambiguous visual evidence. By removing cross-box dependencies, PTD grounds each time-conditioned spatial block directly in the corresponding video evidence, improving both geometric precision and target consistency across time.

B Additional Implementation Details

B.1 Localization Token Representation

We build Parallel Tube Decoding (PTD) on the standard autoregressive generation framework of Qwen3-VL [3]. To support structured spatio-temporal grounding, we augment the model vocabulary with discrete spatial and temporal localization tokens, where each temporal token is aligned with its corresponding visual patch. The same localization representation is shared by the NTP and MTP formulations, such that both represent the same spatio-temporal output while differing only in the decoding formulation used to generate it.

Localization vocabulary. We represent each spatial coordinate using one of 1,001 discrete tokens corresponding to normalized values in $[0, 1000]$. Following prior quantized coordinate representations, each coordinate is first clipped to the frame boundary, normalized by the corresponding frame width or height, and quantized to the nearest spatial token. During evaluation, the predicted spatial tokens are mapped back to the original frame dimensions. For temporal localization, we similarly introduce 100 discrete temporal tokens, denoted by $\texttt{<}t_i\texttt{>}$, where each token corresponds to a position on the sampled temporal grid rather than an absolute timestamp or a normalized position within the complete video. Each predicted temporal token is mapped back to the original video timeline using the source-frame index associated with its sampled position.

To indicate the start and end of temporal and spatial localization outputs, we use dedicated delimiter tokens. For spatial localization, we directly reuse the existing Qwen3-VL tokens `<box_start>` and `<box_end>`, together with their pretrained embeddings, and denote them as `<box>` and `</box>` throughout the paper for readability. For temporal localization, we introduce `<time_start>` and `<time_end>`, denoted as `<time>` and `</time>`. For the MTP formulation, we also introduce two additional tokens, `<mask>` and `<null>`.

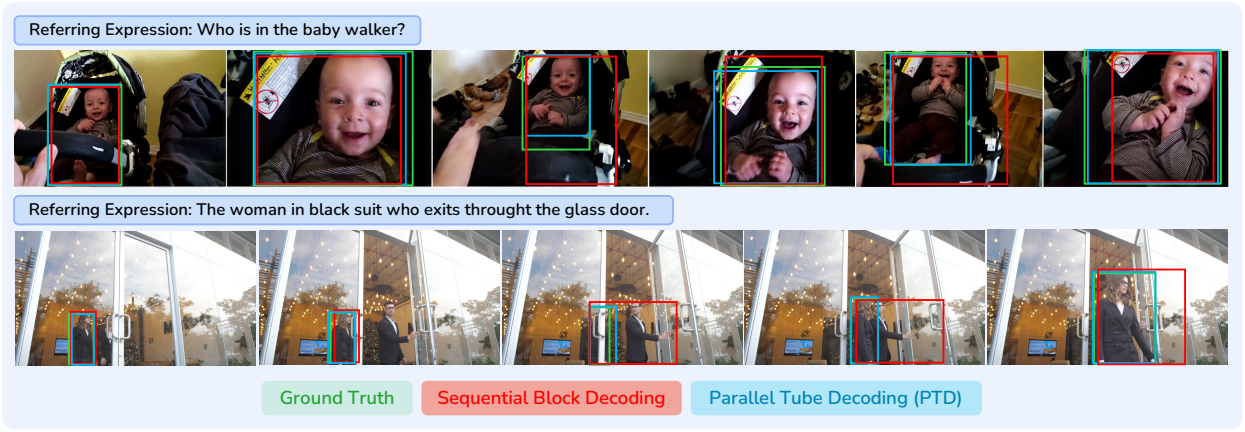

Figure 8 Qualitative comparison of Sequential Block Decoding and PTD. Green, red, and blue denote the ground-truth, Sequential Block Decoding, and Parallel Tube Decoding (PTD) predictions, respectively. The first example contains abrupt scale changes as the stroller moves back and forth, while the second contains partial occlusion as the woman exits through the glass door. Sequential Block Decoding propagates localization errors across subsequent boxes, whereas PTD maintains tight target localization by removing cross-box dependencies.

Embedding initialization. To facilitate the adaptation of newly introduced tokens during training, we initialize each token from a semantically aligned text representation in the pretrained Qwen3-VL vocabulary. When the corresponding representation is tokenized into multiple vocabulary tokens, we initialize the new embedding using the average of their pretrained embeddings (e.g., if '25' is tokenized as '2' and '5', the embedding of <25> is initialized from their mean). Specifically, spatial tokens are initialized from their corresponding numeral strings, temporal tokens from the phrase 'time segment i', and the temporal delimiters and auxiliary tokens from their corresponding text representations.

B.2 Temporal anchoring of visual tokens

Qwen3-VL interleaves text timestamps (e.g., '<0.2 seconds>') with the video representation, providing each temporal visual patch with an explicit temporal reference. We follow the same strategy, but replace textual timestamps with the discrete temporal tokens $<t_i>$, introduced above. Specifically, for N temporal visual patches, the video representation is represented as: $[<t_1>\mathcal{V}_1][<t_2>\mathcal{V}_2]\cdots[<t_N>\mathcal{V}_N]$ where $\mathcal{V}_i$ denotes the visual tokens associated with the i-th temporal patch and $<t_i>$ provides its corresponding temporal anchor. Qwen3-VL uses a temporal patch size of 2 by default, where two consecutive sampled frames are merged into a single temporal visual patch. We instead use a temporal patch size of 1 to preserve finer temporal granularity, such that each sampled frame is associated with a unique temporal token. This choice is not required by PTD: the same formulation can be applied with a temporal patch size of 2 or higher, as long as each time-conditioned spatial block is associated with the corresponding temporal token.

B.3 Construction of the Four Decoding Variants

We construct four decoding variants that progressively reduce the sequential dependencies involved in generating the spatio-temporal tube. Unquantized Token Decoding follows the standard Qwen3-VL generation formulation [3], where temporal and spatial localization values are represented as text and generated autoregressively. The remaining variants use the discrete localization representation introduced in Appx. Sec. B.1. Quantized Token Decoding uses the NTP formulation and generates each localization token sequentially. Sequential Block Decoding and PTD instead use multi-token prediction, but differ in how the spatial blocks are constructed and whether dependencies are retained across time.

For clarity, we illustrate how each decoding variant represents the same spatio-temporal target. Consider a tube spanning $[<t_3>, <t_5>]$ with boxes $b_3 = (125, 210, 420, 780)$, $b_4 = (140, 205, 438, 792)$, and $b_5 = (166, 198, 460, 805)$. We assume that $<t_3>$, $<t_4>$, and $<t_5>$ correspond to 1.5, 2.0, and 2.5 seconds, respectively. The four variants describe the same tube, but differ in how the output is represented, factorized, and decoded.

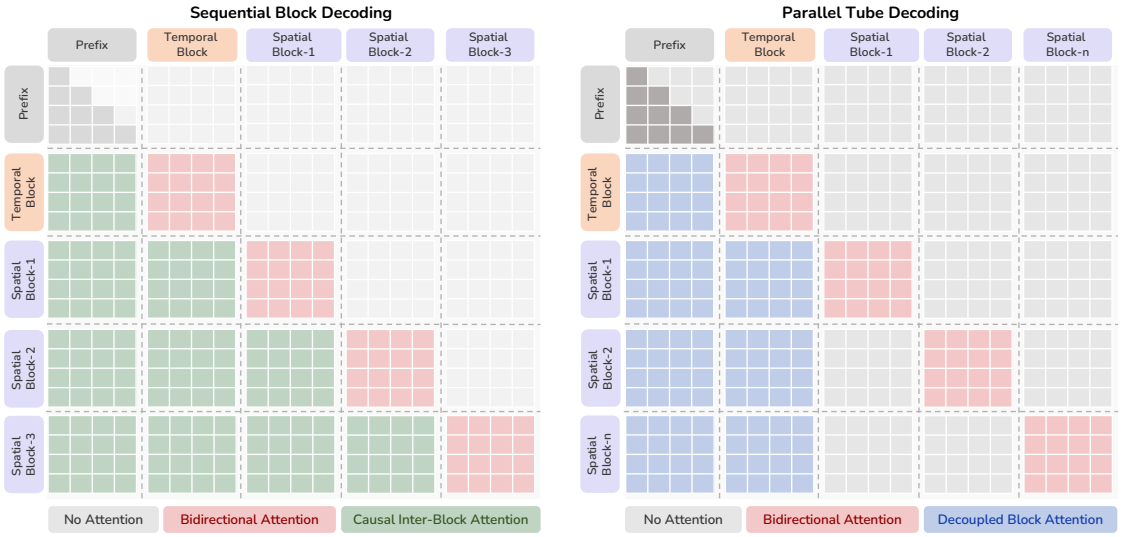

Figure 9 MTP attention masks for Sequential Block Decoding and Parallel Tube Decoding (PTD). Each row represents a query block, while each column represents a key block available as attention context. Green masks form the lower-triangular pattern used for causal attention across blocks in Sequential Block Decoding. Red denotes bidirectional attention among tokens within the same block. Blue masks denote Decoupled Block Attention in PTD, where each spatial block accesses the shared multimodal prefix and temporal block while all other spatial blocks remain masked. The absence of connections across spatial blocks allows all time-conditioned spatial blocks to be decoded in parallel.

i) Unquantized Token Decoding. Unquantized decoding follows the standard next-token generation formulation of Qwen3-VL, while representing temporal and spatial localization values as text values. For the example above, the temporal interval followed by the corresponding time-indexed spatial records is represented as

$$\texttt{<time\_start>}, \texttt{1.5 seconds}, \texttt{2.5 seconds}, \texttt{<time\_end>},$$
$$\texttt{1.5 seconds}, \texttt{<box\_start>}, 125, 210, 420, 780, \texttt{<box\_end>},$$
$$\texttt{2.0 seconds}, \texttt{<box\_start>}, 140, 205, 438, 792, \texttt{<box\_end>},$$
$$\texttt{2.5 seconds}, \texttt{<box\_start>}, 166, 198, 460, 805, \texttt{<box\_end>}.$$

The complete output is generated one token at a time using the original tokenizer. Since a timestamp or coordinate may be decomposed into multiple text tokens, the sequential decoding depth depends on both the tokenization of the localization values and the tube length. This variant therefore includes both representation overhead and token-level autoregressive dependencies. We train this variant using only the NTP formulation with standard causal attention.

ii) Quantized Token Decoding. Quantization replaces each temporal position and spatial coordinate with a dedicated discrete token while retaining standard next-token prediction. The temporal interval and each corresponding time-indexed spatial record are represented as

$$\texttt{<time\_start>}, \texttt{<}t_3\texttt{>}, \texttt{<}t_5\texttt{>}, \texttt{<time\_end>},$$
$$\texttt{<}t_3\texttt{>}, \texttt{<box\_start>}, \texttt{<125>}, \texttt{<210>}, \texttt{<420>}, \texttt{<780>}, \texttt{<box\_end>},$$
$$\texttt{<}t_4\texttt{>}, \texttt{<box\_start>}, \texttt{<140>}, \texttt{<205>}, \texttt{<438>}, \texttt{<792>}, \texttt{<box\_end>},$$
$$\texttt{<}t_5\texttt{>}, \texttt{<box\_start>}, \texttt{<166>}, \texttt{<198>}, \texttt{<460>}, \texttt{<805>}, \texttt{<box\_end>}.$$

The complete tube contains one such spatial record for each grounded temporal position. Since every localization value is represented by a single token, a tube containing T boxes requires a sequential decoding depth of $4 + 7T$, i.e., 25 rounds for the example. This variant is also trained only on the NTP formulation using standard causal attention. All tokens are therefore generated autoregressively, one at a time.

iii) Sequential Block Decoding. Block decoding represents the same quantized target as a sequence of structured localization blocks, where all tokens within each block are predicted jointly. For the example above, the

temporal block and time-indexed spatial blocks are represented as

$$[\texttt{<time\_start>}, \texttt{<}t_3\texttt{>}, \texttt{<}t_5\texttt{>}, \texttt{<time\_end>}, \texttt{<null>}^3],$$
$$[\texttt{<}t_3\texttt{>}, \texttt{<box\_start>}, \texttt{<125>}, \texttt{<210>}, \texttt{<420>}, \texttt{<780>}, \texttt{<box\_end>}],$$
$$[\texttt{<}t_4\texttt{>}, \texttt{<box\_start>}, \texttt{<140>}, \texttt{<205>}, \texttt{<438>}, \texttt{<792>}, \texttt{<box\_end>}],$$
$$[\texttt{<}t_5\texttt{>}, \texttt{<box\_start>}, \texttt{<166>}, \texttt{<198>}, \texttt{<460>}, \texttt{<805>}, \texttt{<box\_end>}].$$

We use a block size of seven because each **time-indexed** spatial block contains one temporal anchor, two box delimiters, and four coordinate tokens. The shorter temporal block is padded with three <null> tokens to match the same block width. The temporal block is prompted using $[q_P, \texttt{<mask>}^6]$, where q_P denotes the final token of the shared multimodal prefix. Each spatial block is then prompted using the final token of the preceding block followed by 6 <mask> tokens. We train this variant using the joint NTP and MTP formulations. The NTP sequence uses standard causal attention, while the MTP sequence uses bidirectional attention within each block to jointly predict all seven tokens. Causal attention is retained across blocks, allowing each spatial block to attend to previously predicted boxes. The blocks therefore remain sequential across the tube, resulting in a decoding depth of $1 + T$, or 4 rounds for the example above.

iv) Parallel Tube Decoding. PTD retains the same quantized representation and temporal-first generation order, while removing the remaining trajectory-level dependency across spatial blocks. The temporal interval is first predicted using the same temporal block as block decoding. Once [<t_3>, <t_5>] is localized, the temporal anchors <t_3>, <t_4>, and <t_5> are instantiated from the predicted interval rather than generating them autoregressively. The corresponding **time-conditioned** spatial inputs and predicted box suffixes are

$$[\texttt{<}t_3\texttt{>}, \texttt{<mask>}^5] \rightarrow [\texttt{<box\_start>}, \texttt{<125>}, \texttt{<210>}, \texttt{<420>}, \texttt{<780>}, \texttt{<box\_end>}],$$
$$[\texttt{<}t_4\texttt{>}, \texttt{<mask>}^5] \rightarrow [\texttt{<box\_start>}, \texttt{<140>}, \texttt{<205>}, \texttt{<438>}, \texttt{<792>}, \texttt{<box\_end>}],$$
$$[\texttt{<}t_5\texttt{>}, \texttt{<mask>}^5] \rightarrow [\texttt{<box\_start>}, \texttt{<166>}, \texttt{<198>}, \texttt{<460>}, \texttt{<805>}, \texttt{<box\_end>}].$$

PTD uses a spatial block size of six because the temporal anchor is provided as input rather than predicted as part of the spatial target. We train PTD using the joint NTP and MTP formulations: NTP retains standard causal attention, while MTP uses bidirectional attention within each block together with Decoupled Block Attention. As illustrated in Figure 9, Sequential Block Decoding allows each spatial block to attend to previously predicted spatial blocks, whereas PTD restricts every spatial block to the shared multimodal prefix, the temporal block, and its own tokens. This removes cross-box dependencies and enables all spatial blocks to be decoded in parallel, giving a sequential decoding depth of 2, independent of tube length.

C Optimization Details

We optimize the model using LoRA while keeping the vision encoder, multimodal merger, base language-model parameters, and existing vocabulary embeddings frozen. Only the LoRA parameters and newly introduced localization-token embeddings are updated. Training is conducted across two compute nodes using 64-GB AMD Instinct MI210 GPUs with PyTorch 2.7 and ROCm 6.3. We use a per-device batch size of 1. Training the main PTD model for 2 SFT epochs takes approximately 3 days, while the subsequent GRPO stage takes approximately 2.5 days. The remaining optimization settings are summarized in Table 9. For the controlled comparison in Sec. 4.2, each of the four decoding variants is trained for 1 SFT epoch under the same setting.

D Evaluation Protocols

Across all evaluations, we sample videos at 2 FPS using at most 64 frames. For STVG, we evaluate on the standard test splits of VidSTG [65], HC-STVG-v1, and HC-STVG-v2 [50], which contain 10,303, 1,160, and 1,901 queries, respectively. We use the following prompt:

```
<video> Given the query: {QUERY} Localize the described object throughout the video. Use
object reference tokens, time tokens, and box tokens. Return the object reference, event time
segment, and per-time bbox coordinates.
```

Stage	Setting	Value / Notes
Video Setup	Sampling rate	2 FPS
	Maximum frames	64
	Temporal patch size	1
SFT	Training samples	∼90K
	Epochs	2
	Learning rate	2×10^{-5}
	Scheduler / warmup	Cosine / 0.03
	Global batch size	64
	LoRA rank / α	32/64
	LoRA dropout	0.05
	Training formulation	Dual NTP/MTP
GRPO	Training samples	16K
	Learning rate	5×10^{-6}
	Rollouts per prompt	8
	Maximum completion length	1024
	KL coefficient β	0.04
	Temperature / top-p / top-k	0.9/1.0/50
	LoRA rank / α	32/64
	Training formulation	PTD
	Rewards	$R_{\text{temp}} + R_{\text{spat}}$

Table 9 Optimization settings for the proposed PTD model. We report the video-processing, supervised fine-tuning, and localization-aware policy optimization settings used for the PTD model in the main STVG benchmarking.

For zero-shot temporal grounding on Charades-STA [19] and ActivityNet [32], we use the following prompt:

```
<video> Given the query: {QUERY} Localize when the described event occurs in the video. Use object reference tokens and time tokens. Return the event reference followed by the event time segment.
```

For grounded VideoQA on ReXTime [9], we use a two-stage evaluation procedure. The model first selects an answer using the following prompt:

```
<video> Given the question: {QUESTION} Options: A. {A} B. {B} C. {C} D. {D} Answer with only the letter of the correct option.
```

The selected answer is then used as the query to localize its supporting temporal interval. For referring video object tracking on Ref-DAVIS [30], Ref-YT-VOS [48], and ReasonVOS [5], the grounding stage is identical for the SAM2 [45] and SAM3 [7] segmentation backends. We use the following prompt:

```
<video> Given the query: {EXPRESSION} Localize the described object throughout the video. Use object reference tokens, time tokens, and box tokens. Return the object reference, event time segment, and per-time bbox coordinates.
```

E Additional Qualitative Results

We also present additional qualitative results in Figure 10, covering diverse grounding challenges. PTD distinguishes the queried target among visually similar instances while remaining robust to motion and partial occlusion (row 1). It maintains target consistency despite interference from nearby entities and correctly stops localization when the target leaves the scene (row 2 and 5). PTD also accurately localizes small objects as they are progressively revealed (row 3) and preserves target identity among multiple similar instances with overlapping trajectories (row 4). Overall, these results demonstrate consistent localization under frame-specific changes in position, scale, and visibility.

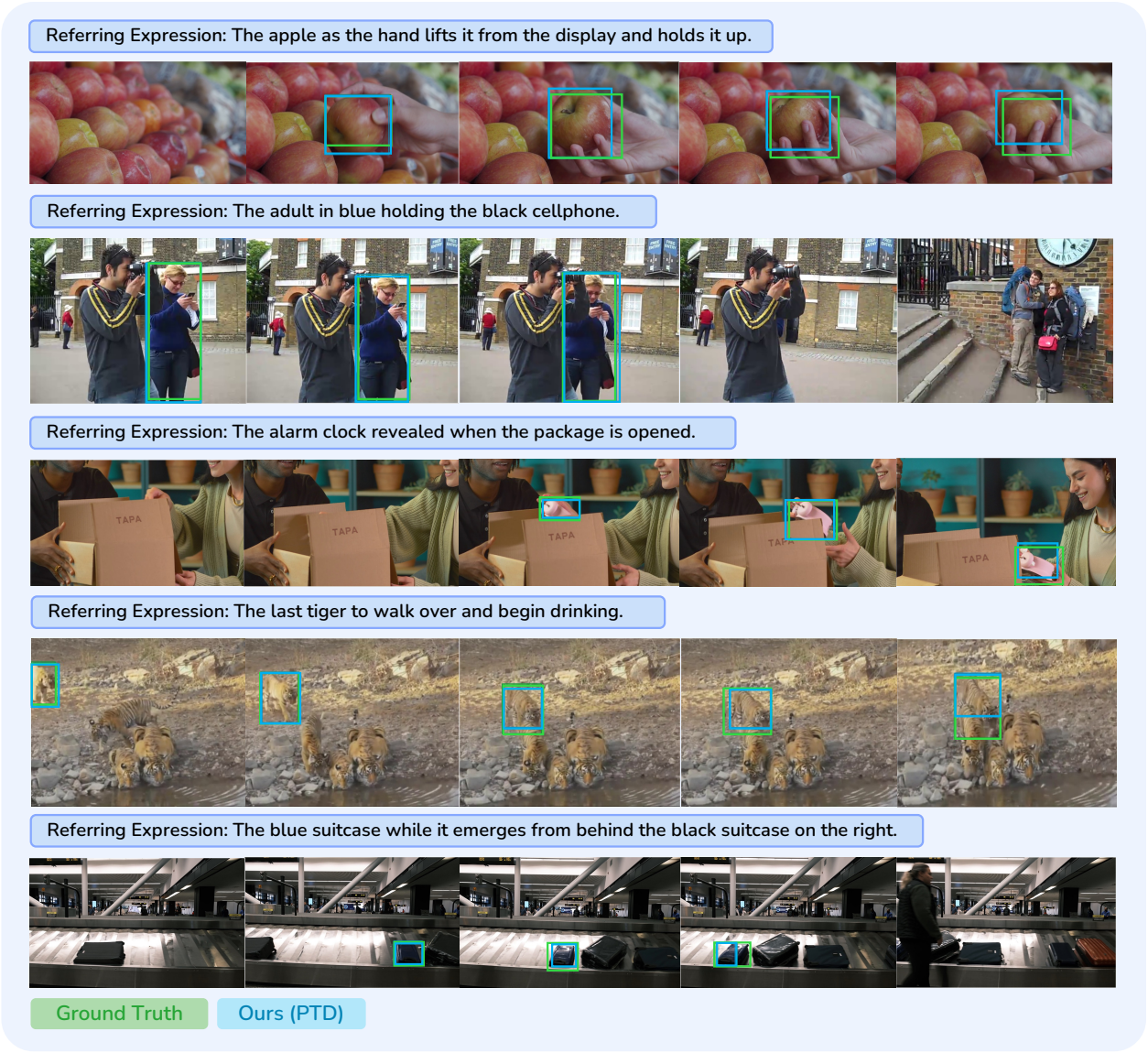

Figure 10 Additional qualitative results of PTD. PTD accurately localizes the queried entity under visually similar distractors, target motion and scale changes, partial occlusion, progressive object reveal, and multi-instance interactions.

References

[1] Ghazi Shazan Ahmad, Ahmed Heakl, Hanan Gani, Abdelrahman Shaker, Zhiqiang Shen, Fahad Shahbaz Khan, and Salman Khan. VideoMolmo: Spatio-temporal grounding meets pointing. *arXiv preprint arXiv:2506.05336*, 2025.

[2] 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 *International Conference on Learning Representations*, 2025.

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

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

[5] Zechen Bai, Tong He, Haiyang Mei, Pichao Wang, Ziteng Gao, Joya Chen, Lei Liu, Zheng Zhang, and Mike Z Shou. One token to seg them all: Language instructed reasoning segmentation in videos. *Advances in Neural Information Processing Systems*, 2024.

[6] Lucas Beyer, Andreas Steiner, André Susano Pinto, Alexander Kolesnikov, Xiao Wang, Daniel Salz, Maxim Neumann, Ibrahim Alabdulmohsin, Michael Tschannen, Emanuele Bugliarello, et al. PaliGemma: A versatile 3B VLM for transfer. *arXiv preprint arXiv:2407.07726*, 2024.

[7] Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris Coll-Vinent, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, et al. SAM 3: Segment anything with concepts. In *International Conference on Learning Representations*, 2026.

[8] Guo Chen, Yicheng Liu, Yifei Huang, Baoqi Pei, Jilan Xu, Yuping He, Tong Lu, Yali Wang, and Limin Wang. Cg-bench: Clue-grounded question answering benchmark for long video understanding. In *International Conference on Learning Representations*, 2025.

[9] Jr-Jen Chen, Yu-Chien Liao, Hsi-Che Lin, Yu-Chu Yu, Yen-Chun Chen, and Yu-Chiang Frank Wang. ReXTime: A benchmark suite for reasoning-across-time in videos. *Advances in Neural Information Processing Systems*, 37: 28662–28673, 2024.

[10] Keqin Chen, Zhao Zhang, Weili Zeng, Richong Zhang, Feng Zhu, and Rui Zhao. Shikra: Unleashing multimodal LLM's referential dialogue magic. *arXiv preprint arXiv:2306.15195*, 2023.

[11] Ting Chen, Saurabh Saxena, Lala Li, David J Fleet, and Geoffrey Hinton. Pix2seq: A language modeling framework for object detection. In *International Conference on Learning Representations*, 2022.

[12] Xinyi Chen, Yilun Chen, Yanwei Fu, Ning Gao, Jiaya Jia, Weiyang Jin, Hao Li, Yao Mu, Jiangmiao Pang, Yu Qiao, et al. InternVLA-M1: A spatially guided vision-language-action framework for generalist robot policy. *arXiv preprint arXiv:2510.13778*, 2025.

[13] Zixu Cheng, Da Li, Jian Hu, Yuhang Zang, Ziquan Liu, Shaogang Gong, and Wei Li. GraphThinker: Reinforcing temporally grounded video reasoning with event graph thinking. *arXiv preprint arXiv:2602.17555*, 2026.

[14] Christopher Clark, Jieyu Zhang, Zixian Ma, Jae Sung Park, Mohammadreza Salehi, Rohun Tripathi, Sangho Lee, Zhongzheng Ren, Chris Dongjoo Kim, Yinuo Yang, et al. Molmo2: Open weights and data for vision-language models with video understanding and grounding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026.

[15] 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.

[16] Matt Deitke, Christopher Clark, Sangho Lee, Rohun Tripathi, Yue Yang, Jae Sung Park, Mohammadreza Salehi, Niklas Muennighoff, Kyle Lo, Luca Soldaini, et al. Molmo and PixMo: Open weights and open data for state-of-the-art vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

[17] Lu Dong, Haiyu Zhang, Han Lin, Ziang Yan, Xiangyu Zeng, Hongjie Zhang, Yifei Huang, Yi Wang, Zhen-Hua Ling, Limin Wang, et al. VideoTG-R1: Boosting video temporal grounding via curriculum reinforcement learning on reflected boundary annotations. In *Proceedings of the International Conference on Multimedia Retrieval*, 2026.

[18] Yonggan Fu, Lexington Whalen, Zhifan Ye, Xin Dong, Shizhe Diao, Jingyu Liu, Chengyue Wu, Hao Zhang, Enze Xie, Song Han, et al. Efficient-DLM: From autoregressive to diffusion language models, and beyond in speed. *arXiv preprint arXiv:2512.14067*, 2025.

[19] Jiyang Gao, Chen Sun, Zhenheng Yang, and Ram Nevatia. TALL: Temporal activity localization via language query. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2017.

[20] Shida Gao, Feng Xue, Xiangfeng Wang, Anlong Ming, Zhaowen Lin, Haiyang Zhang, Teng Long, Nicu Sebe, Yihua Shao, Haozhe Wang, et al. Detector-empowered video large language model for efficient spatio-temporal grounding. *arXiv preprint arXiv:2512.06673*, 2025.

[21] Fabian Gloeckle, Badr Youbi Idrissi, Baptiste Rozière, David Lopez-Paz, and Gabriel Synnaeve. Better & faster large language models via multi-token prediction. In *International Conference on Machine Learning*, 2024.

[22] Xin Gu, Heng Fan, Yan Huang, Tiejian Luo, and Libo Zhang. Context-guided spatio-temporal video grounding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

[23] Xin Gu, Yaojie Shen, Chenxi Luo, Tiejian Luo, Yan Huang, Yuewei Lin, Heng Fan, and Libo Zhang. Knowing your target: Target-aware transformer makes better spatio-temporal video grounding. In *International Conference on Learning Representations*, 2025.

[24] Xin Gu, Haoji Zhang, Qihang Fan, Jingxuan Niu, Zhipeng Zhang, Libo Zhang, Guang Chen, Fan Chen, Longyin Wen, and Sijie Zhu. Thinking with bounding boxes: Enhancing spatio-temporal video grounding via reinforcement fine-tuning. *arXiv preprint arXiv:2511.21375*, 2025.

[25] Yongxin Guo, Jingyu Liu, Mingda Li, Dingxin Cheng, Xiaoying Tang, Dianbo Sui, Qingbin Liu, Xi Chen, and Kevin Zhao. VTG-LLM: Integrating timestamp knowledge into video LLM for enhanced video temporal grounding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2025.

[26] Yongxin Guo, Jingyu Liu, Mingda Li, Qingbin Liu, Xi Chen, and Xiaoying Tang. TRACE: Temporal grounding video LLM via causal event modeling. In *International Conference on Learning Representations*, 2025.

[27] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

[28] Bin Huang, Xin Wang, Hong Chen, Zihan Song, and Wenwu Zhu. VTimeLLM: Empower LLM to grasp video moments. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

[29] Qing Jiang, Junan Huo, Xingyu Chen, Yuda Xiong, Zhaoyang Zeng, Yihao Chen, Tianhe Ren, Junzhi Yu, and Lei Zhang. Detect anything via next point prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026.

[30] Anna Khoreva, Anna Rohrbach, and Bernt Schiele. Video object segmentation with language referring expressions. In *Asian Conference on Computer Vision*, 2018.

[31] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan P. Foster, Pannag R. Sanketi, Quan Vuong, et al. OpenVLA: An open-source vision-language-action model. In *Proceedings of the Conference on Robot Learning*, 2025.

[32] Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. Dense-captioning events in videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2017.

[33] Hongyu Li, Jinyu Chen, Ziyu Wei, Shaofei Huang, Tianrui Hui, Jialin Gao, Xiaoming Wei, and Si Liu. LLaVA-ST: A multimodal large language model for fine-grained spatial-temporal understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

[34] Zeqian Li, Shangzhe Di, Zhonghua Zhai, Weilin Huang, Yanfeng Wang, and Weidi Xie. Universal video temporal grounding with generative multi-modal large language models. *Advances in Neural Information Processing Systems*, 2025.

[35] Zhaowei Li, Qi Xu, Dong Zhang, Hang Song, Yiqing Cai, Qi Qi, Ran Zhou, Junting Pan, Zefeng Li, Vu Tu, et al. GroundingGPT: Language enhanced multi-modal grounding model. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 2024.

[36] Yangzhou Liu, Yue Cao, Hao Li, Gen Luo, Zhe Chen, Weiyun Wang, Xiaobo Liang, Biqing Qi, Lijun Wu, Changyao Tian, Yanting Zhang, Yuqiang Li, Tong Lu, Yu Qiao, Jifeng Dai, and Wenhai Wang. Sequential diffusion language models. *arXiv preprint arXiv:2509.24007*, 2025.

[37] Chuofan Ma, Yi Jiang, Jiannan Wu, Zehuan Yuan, and Xiaojuan Qi. Groma: Localized visual tokenization for grounding multimodal large language models. In *The European Conference on Computer Vision*, 2024.

[38] Muhammad Maaz, Hanoona Rasheed, Fahad Shahbaz Khan, and Salman Khan. Video-R2: Reinforcing consistent and grounded reasoning in multimodal language models. *arXiv preprint arXiv:2511.23478*, 2025.

[39] Shehan Munasinghe, Hanan Gani, Wenqi Zhu, Jiale Cao, Eric Xing, Fahad Shahbaz Khan, and Salman Khan. VideoGLaMM: A large multimodal model for pixel-level visual grounding in videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

[40] Shen Nie, Fengqi Zhu, Zebin You, Xiaolu Zhang, Jingyang Ou, Jun Hu, Jun Zhou, Yankai Lin, Ji-Rong Wen, and Chongxuan Li. Large language diffusion models. *Advances in Neural Information Processing Systems*, 2025.

[41] OpenAI. GPT-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.

[42] Zhiliang Peng, Wenhui Wang, Li Dong, Yaru Hao, Shaohan Huang, Shuming Ma, and Furu Wei. Kosmos-2: Grounding multimodal large language models to the world. In *International Conference on Learning Representations*, 2024.

[43] Long Qian, Juncheng Li, Yu Wu, Yaobo Ye, Hao Fei, Tat-Seng Chua, Yueting Zhuang, and Siliang Tang. Momentor: Advancing video large language model with fine-grained temporal reasoning. In *International Conference on Machine Learning*, 2024.

[44] Hanoona Rasheed, Mohammed Zumri, Muhammad Maaz, Ming-Hsuan Yang, Fahad Shahbaz Khan, and Salman Khan. Video-CoM: Interactive video reasoning via chain of manipulations. *arXiv preprint arXiv:2511.23477*, 2025.

[45] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. SAM 2: Segment anything in images and videos. In *International Conference on Learning Representations*, 2025.

[46] Shuhuai Ren, Linli Yao, Shicheng Li, Xu Sun, and Lu Hou. TimeChat: A time-sensitive multimodal large language model for long video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

[47] Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. Generalized intersection over union: A metric and a loss for bounding box regression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

[48] Seonguk Seo, Joon-Young Lee, and Bohyung Han. URVOS: Unified referring video object segmentation network with a large-scale benchmark. In *The European Conference on Computer Vision*, 2020.

[49] 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.

[50] Zongheng Tang, Yue Liao, Si Liu, Guanbin Li, Xiaojie Jin, Hongxu Jiang, Qian Yu, and Dong Xu. Human-centric spatio-temporal video grounding with visual transformers. *IEEE Transactions on Circuits and Systems for Video Technology*, 2021.

[51] Xuezhen Tu, Jingyu Wu, Fangyu Kang, Qingpeng Nong, Kaijin Zhang, Chaoyue Niu, and Fan Wu. Bridging time and space: Decoupled spatio-temporal alignment for video grounding. *arXiv preprint arXiv:2604.08014*, 2026.

[52] Haibo Wang, Zhiyang Xu, Yu Cheng, Shizhe Diao, Yufan Zhou, Yixin Cao, Qifan Wang, Weifeng Ge, and Lifu Huang. Grounded-VideoLLM: Sharpening fine-grained temporal grounding in video large language models. In *Findings of the Association for Computational Linguistics: EMNLP*, 2025.

[53] Jiankang Wang, Zhihan Zhang, Zhihang Liu, Yang Li, Jiannan Ge, Hongtao Xie, and Yongdong Zhang. SpaceVLLM: Endowing multimodal large language model with spatio-temporal video grounding capability. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2026.

[54] Shihao Wang, Shilong Liu, Yuanguo Kuang, Xinyu Wei, Yangzhou Liu, Zhiqi Li, Yunze Man, Guo Chen, Andrew Tao, Guilin Liu, et al. LocateAnything: Fast and high-quality vision-language grounding with parallel box decoding. *arXiv preprint arXiv:2605.27365*, 2026.

[55] Xiaohan Wang, Yuhui Zhang, Orr Zohar, and Serena Yeung-Levy. VideoAgent: Long-form video understanding with large language model as agent. In *The European Conference on Computer Vision*, 2024.

[56] Ziang Yan, Yinan He, Xinhao Li, Zhengrong Yue, Xiangyu Zeng, Yali Wang, Yu Qiao, Limin Wang, and Yi Wang. VideoChat-R1.5: Visual test-time scaling to reinforce multimodal reasoning by iterative perception. *Advances in Neural Information Processing Systems*, 2025.

[57] Ziang Yan, Zhilin Li, Yinan He, Chenting Wang, Kunchang Li, Xinhao Li, Xiangyu Zeng, Zilei Wang, Yali Wang, Yu Qiao, et al. Task preference optimization: Improving multimodal large language models with vision task alignment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

[58] Antoine Yang, Antoine Miech, Josef Sivic, Ivan Laptev, and Cordelia Schmid. TubeDETR: Spatio-temporal video grounding with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[59] 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. *Advances in Neural Information Processing Systems*, 2025.

[60] Haobo Yuan, Xiangtai Li, Tao Zhang, Yueyi Sun, Zilong Huang, Shilin Xu, Shunping Ji, Yunhai Tong, Lu Qi, Jiashi Feng, et al. Sa2VA: Marrying SAM2 with MLLM for dense grounded understanding of images and videos. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2026.

[61] Lunbin Zeng, Jingfeng Yao, Bencheng Liao, Hongyuan Tao, Wenyu Liu, and Xinggang Wang. DiffusionVL: Translating any autoregressive models into diffusion vision language models. *arXiv preprint arXiv:2512.15713*, 2025.

[62] Xiangyu Zeng, Kunchang Li, Chenting Wang, Xinhao Li, Tianxiang Jiang, Ziang Yan, Songze Li, Yansong Shi, Zhengrong Yue, Yi Wang, et al. TimeSuite: Improving MLLMs for long video understanding via grounded tuning. In *International Conference on Learning Representations*, 2025.

[63] Xiaowen Zhang, Zhi Gao, Licheng Jiao, Lingling Li, and Qing Li. STVG-R1: Incentivizing instance-level reasoning and grounding in videos via reinforcement learning. *arXiv preprint arXiv:2602.11730*, 2026.

[64] Yuanhan Zhang, Jinming Wu, Wei Li, Bo Li, Zejun Ma, Ziwei Liu, and Chunyuan Li. LLaVA-Video: Video instruction tuning with synthetic data. *arXiv preprint arXiv:2410.02713*, 2024.

[65] Zhu Zhang, Zhou Zhao, Yang Zhao, Qi Wang, Huasheng Liu, and Lianli Gao. Where does it exist: Spatio-temporal video grounding for multi-form sentences. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[66] Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, et al. RT-2: Vision-language-action models transfer web knowledge to robotic control. In *Proceedings of the Conference on Robot Learning*, 2023.