

# ParallelVLM: Lossless Video-LLM Acceleration with Visual Alignment Aware Parallel Speculative Decoding

## Supplementary Material

For readers who are not familiar with speculative decoding and parallel decoding, we provide more detailed explanations in the appendix. The appendix is organized as follows: experimental details are presented in Sec. A, parallel decoding in Sec. B, and theoretical analysis of the proposed method in Sec. C. Additionally, we elaborate on the origin of the unbiased nature of UV-Prune in Sec. D. Finally, we include several illustrative case studies in Sec. E.

### A. Experimental Details

**Tasks and Metrics.** Since speculative decoding [19, 45] focuses on accelerating long-paragraph decoding, we select five video understanding and description datasets VideoDetailCaption [32], VideoMME [11], MVBench [21], MVLU [53], LongVideoBench [44] with rich semantic information. We filter out videos shorter than one minute and randomly sample 50 clips from each dataset for evaluation. Unlike traditional Visual Question Answering (VQA) formats\*, we employ prompt engineering to generate finer-grained descriptions. For the lossy visual token pruning method, due to the lack of long, detailed ground truth description labels in the community, we save the autoregressive output texts after pruning and use the target model with full context to perform token-wise speculative sampling [19] to measure the acceptance rate  $\mathcal{A}$ . This approach not only saves the cost of manual annotation, but also accurately reflects the distribution shifts of the pruned models.

**Model Combinations.** Regarding the model combinations, since ParallelVLM is a training-free method, aligning the draft model with the target model poses a challenge. We therefore select model combinations with different parameter sizes from the same series to ensure that the draft and target models have a basic alignment capability. In experiments we find that models from the LLaVA-OV [20] series exhibit a higher alignment ( $M = 8.31$  for 0.5B/7B and  $M = 6.83$  for 7B/72B), while Qwen2.5-VL [3] shows a relatively lower alignment ( $M = 4.28$  for 7B/32B). Additionally, the 0.5B/7B combination within the LLaVA-OV series achieves better alignment than the 7B/72B combination, as the former has more similar model capacities and fine-tuning data scales. Although our approach is totally training-free, draft training-based methods [5, 23, 24] can be seamlessly integrated to achieve improved alignment and higher acceleration ratios in future work.

**Window Size  $\gamma$  Selection.** For simplicity of explanation in

\*Different from our finer-grained long-paragraph decoding setting, VQA evaluation suites like LLMs-Eval mainly focus on reality checking.

Table 5. Sensitivity of speed ratio  $c = T_p/T_q$  with the increase of pruning ratio  $\alpha$ . The forward time is reported in ms.

Models	Target $T_p$	Draft $T_q$		
		$\alpha = 0.0$	$\alpha = 0.5$	$\alpha = 0.9$
LLaVA-OV (0.5B & 7B)	78.3 ( $\pm 3.46$ )	46.1 ( $\pm 1.21$ )	32.5 ( $\pm 0.89$ )	25.8 ( $\pm 0.32$ )
		$c = 1.7$	$c = 2.4$	$c = 3.0$
LLaVA-OV (7B & 72B)	420 ( $\pm 8.64$ )	78.3 ( $\pm 3.46$ )	57.5 ( $\pm 2.38$ )	46.6 ( $\pm 1.52$ )
		$c = 5.3$	$c = 7.3$	$c = 9.0$
Qwen2.5-VL (7B & 32B)	203 ( $\pm 6.89$ )	63.7 ( $\pm 2.42$ )	48.2 ( $\pm 1.66$ )	39.8 ( $\pm 1.24$ )
		$c = 3.2$	$c = 4.2$	$c = 5.1$
LLaVA-OV (7B & 7B)	78.3 ( $\pm 3.46$ )	78.3 ( $\pm 3.46$ )	57.5 ( $\pm 2.38$ )	46.6 ( $\pm 1.52$ )
		$c = 1.0$	$c = 1.4$	$c = 1.7$
Qwen2.5-VL (7B & 7B)	63.7 ( $\pm 2.42$ )	63.7 ( $\pm 2.42$ )	48.2 ( $\pm 1.66$ )	39.8 ( $\pm 1.24$ )
		$c = 1.0$	$c = 1.3$	$c = 1.6$

Table 6. Peak memory usage (GB) and latency breakdown of LLaVA-OV-7B/72B (s). The output length is set to 512.

Operation	Method		
	Autoregressive	SD	ParallelVLM
Peak Memory	198.85	225.19	225.34
Draft Prefill	7.92	7.92	/
Target Prefill	44.23	44.23	44.23
Draft Decode	/	58.14	/
Target Decode	228.14	54.85	67.85
Token/s	2.25	4.53	7.55
Total Latency	280.29	165.15	112.08

the main text, we directly rounded the speed ratio between the draft model and the target model  $c = T_p/T_q$  to an integer value. In practice, their acceleration ratio is not strictly an integer. In Tab. 5, we present the exact acceleration ratios for all the five combinations. Rounding the window size up or down relative to the speed ratio  $c$  are both reasonable options, the superiority between which depends on the relative magnitudes of the window size  $\gamma$  and the mean accepted length  $M$  [30]. For combinations of LLaVA-OV-7B/72B and Qwen2.5-VL-7B/32B with aggressive window sizes and relatively conservative accepted lengths, rounding down to reduce mutual waiting is a wiser choice. For Self-SD with narrow window sizes and high accepted lengths, rounding up is a more sensible option.

**Memory Consumption and Latency Breakdown.** We report the peak memory usage and latency breakdown in Tab. 6. Although performing the draft/target forward functions simultaneously, ParallelVLM incurs no more mem-

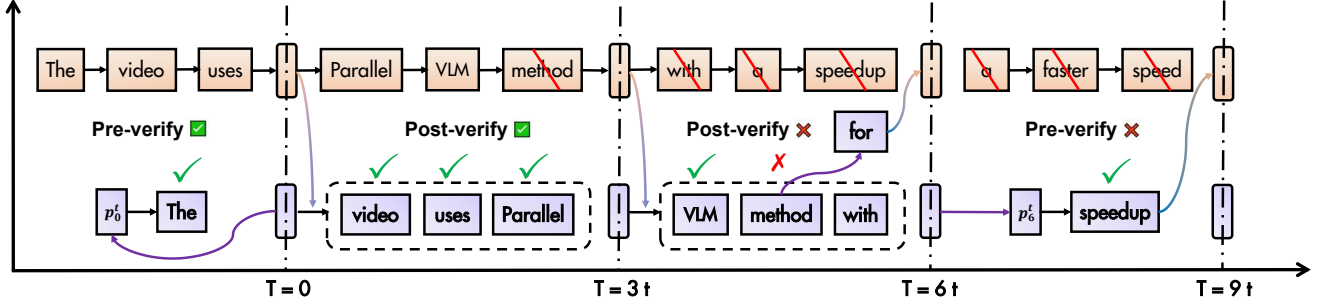


Figure 8. An illustrative example of decoding “The video uses ParallelVLM for speedup” is provided to demonstrate the verification mode switching of the parallel pipeline, as well as its advancement and rollback mechanisms. Here  $t$  denotes the one single forward time of the draft model and the window size  $\gamma$  is set to 3.

ory footprint beyond vanilla SD with draft model pruning (+26 GB than autoregressive). Furthermore, ParallelVLM directly saves the draft model’s prefilling time with parallel prefilling (−7.92 s), and significantly accelerates the decoding speed through the visual alignment aware parallelism. Our approach achieves a notable 7.55 tokens/s throughput, which is  $3.36\times$  faster than autoregressive decoding. While the throughput metric may vary on different devices, the speedup ratio will persist for methodological superiority.

**Tree-Attention Integration.** Applying a draft tree structure to speculative decoding can help improve the accepted length, but it is not feasible under a parallel architecture. The draft tree generates KV caches for multiple candidates, which constitutes a batch of additional overhead. This overhead grows exponentially with the depth  $h$  of the tree during parallelization (even sparse trees [23] exhibit considerable growth). In vanilla speculative decoding, the depth of the draft tree is generally equal to the window size  $h = \gamma$ . However, for parallel decoding, the depth of the draft tree depends on the adaptive draft length [30] and may increase  $h$  by multiple times. To address the limitation of applying the tree structure to parallel decoding, SpecBranch [38] attempts to achieve a balance with a sparser “branch prediction” structure, which we anticipate will find broader applications in the future.

## B. Parallel Pipeline Details

For clarity, we collectively denote the prefix of visual-text tokens  $V_{1:m}, X_{1:n}$  as  $\mathbf{x}$  here to describe the pipeline advancement and rollback mechanisms of parallel decoding.

As illustrated in Algorithm. 1, Parallel SD [30] sets two verification modes for the target model: (i) Pre-verify, which verifies the first draft token in advance, and (ii) Post-verify, which performs batch verification meanwhile generating new draft tokens. We elaborate on more details in the following with the illustrative example in Fig. 8 of decoding the sentence “The video uses ParallelVLM for speedup” with the window size  $\gamma = 3$ .

---

### Algorithm 1 Parallel Decoding Pipeline.

---

**Require:** draft model  $M_q$ , target model  $M_p$ , input prefix  $\mathbf{x}$ , output length  $K$ , window size  $\gamma$ .

- 1: mode  $\leftarrow$  “Pre-verify”
- 2: **while**  $\text{len}(\mathbf{x}) < K$  **do**
- 3:   **if** mode = “Pre-verify” **then**
- 4:      $(\mathbf{x}, \text{mode}) \leftarrow \text{Pre-verify}(M_q, M_p, \mathbf{x}, \gamma)$
- 5:   **else**
- 6:      $(\mathbf{x}, \text{mode}) \leftarrow \text{Post-verify}(M_q, M_p, \mathbf{x}, \gamma)$
- 7:   **end if**
- 8: **end while**

---

At  $T = 0$ , both the draft model and the target model complete pre-filling, they output a bonus token and execute the initial Pre-verify. The Pre-verify verifies the first token “The” before the verification stage. Upon successful matching (accept “The”), the mode switches to Post-verify. From  $T = 0 \sim 3t$ , a window size of draft tokens “video uses Parallel” is waiting to be verified. The Post-verify checks them with batch parallelism, accepts them all, and maintains the Post-verify mode. From  $T = 3t \sim 6t$ , another window size of draft tokens “VLM method with” is waiting to be verified. However, “method” is rejected by the target model, so a new token “for” is resampled from the target’s distribution. The KV cache of the draft model rollbacks to “VLM” and starts from “for” to draft new tokens, meanwhile the mode is reset back to Pre-verify. From  $T = 6t \sim 9t$ , since there are no draft tokens left, the target model verifies the first draft token “a”. Unfortunately, the target model rejects it and thinks that the resampled “speedup” is a better choice. The KV cache of the draft model thus rollbacks to “for” and restarts from “speedup” to continue its drafting. The whole pipeline from  $T = 0 \sim 9t$  thus outputs “The video uses ParallelVLM for speedup” within only three target model verification times. The maximum adaptive draft length [30] here is 5, as “The video uses ParallelVLM” is continuous without interruption of the first two draft stages.

## C. Theoretical Analysis

Let  $T_q = t$  be the single forward time of the draft model and  $T_p = c \cdot t$  be the verification time of the target model. For a predefined hyperparameter  $\gamma$  to control window size, we derive the ideal and practical speedup ratio by calculating the per-token time, respectively.

**Theorem 2** (Vanilla SD (Ideal)). *Under full acceptance of the  $\gamma$  draft tokens, the per-token time of vanilla SD is:*

$$T_{SD} = \frac{\gamma \cdot T_q + T_p}{\gamma + 1} = \frac{\gamma + c}{\gamma + 1} \cdot t. \quad (8)$$

Thus, the ideal speedup ratio is  $\mathcal{V}_{SD} = \frac{T_p}{T_{SD}} = c \cdot \frac{\gamma + 1}{\gamma + c}$ .

**Theorem 3** (Parallel SD (Ideal)). *Under full acceptance of the  $\gamma$  draft tokens, the per-token time of parallel SD is:*

$$T_{PSD} = \frac{\max(\gamma t, ct)}{\gamma} = \begin{cases} t, & \gamma \geq c \\ \frac{c}{\gamma} t, & \gamma < c \end{cases} \quad (9)$$

When  $\gamma = c$ , the ideal speedup ratio is  $\mathcal{V}_{PSD} = \frac{T_p}{T_{PSD}} = c$ , representing  $c \times$  speedup versus autoregressive decoding and  $\mathcal{V}_{PSD}/\mathcal{V}_{SD} = \frac{\gamma + c}{\gamma + 1} = \frac{2c}{c + 1} \rightarrow 2 \times (c \gg 1)$  speedup compared to vanilla SD. Note that  $\mathcal{V}_{PSD}$  can enjoy a higher gain of  $c^* > c$  with our draft model pruning setting.

**Theorem 4** (Vanilla SD (Practical)). *Under partial acceptance rate  $\tau$  ( $0 < \tau < 1$ ) of  $\gamma$  draft tokens, the per-token time of vanilla SD is:*

$$T_{SD} = \frac{\gamma \cdot T_q + T_p}{\tau \cdot \gamma + 1} = \frac{\gamma + c}{\tau \cdot \gamma + 1} \cdot t. \quad (10)$$

Thus, the ideal speedup ratio is  $\mathcal{V}_{SD} = \frac{T_p}{T_{SD}} = c \cdot \frac{\tau \cdot \gamma + 1}{\gamma + c}$ , which increases monotonically with the acceptance rate  $\tau$ .

**Theorem 5** (Parallel SD (Practical)). *Under partial acceptance rate  $\tau$  ( $0 < \tau < 1$ ) of  $\gamma$  draft tokens, the per-token time of parallel SD is:*

$$T_{PSD} = \frac{\max(\gamma t, ct)}{\tau \cdot \gamma} = \begin{cases} \frac{1}{\tau} t, & \gamma \geq c \\ \frac{c}{\tau \cdot \gamma} t, & \gamma < c \end{cases} \quad (11)$$

When  $\gamma = c$ , the practical speedup ratio is  $\mathcal{V}_{PSD} = \frac{T_p}{T_{PSD}} = \tau \cdot c$ , which also increases monotonically with the acceptance rate  $\tau$ . Recall that Theorem. 1 gives  $\mathcal{V}_{iPSD} = \hat{\tau}(\mathcal{P}, \alpha) \cdot c^*(\alpha)$ . It represents an expanded speed ratio  $c^*(\alpha) > c$  with a robust acceptance rate  $\hat{\tau}(\mathcal{P}, \alpha)$ , where  $\mathcal{P}$  is the visual alignment aware pruning and  $\alpha$  is the pruning ratio.  $\mathcal{V}_{iPSD}$  strikes a balance between  $c^*(\alpha)$  and  $\hat{\tau}(\mathcal{P}, \alpha)$  that exceeds  $\mathcal{V}_{PSD}$  by up to  $0.3 \sim 0.4 \times$  (see Fig. 6 (a)).

Although Theorem. 2 - 5 provide general illustrations, the theoretical speedup ratio of parallel decoding is actually more complex in practice. As described in Sec. B, the

above theory holds only in Post-verify mode; however, if pre-verify (resulting from rollbacks) is taken into account, analysis must be conducted based on the distribution of accepted lengths. Following SpecBranch [38], we assume the random variable of the accepted lengths  $X$  follows a truncated geometric distribution  $X \sim TruncGeo(\tau, \gamma)$ .

**Lemma 1** (Expected Draft Accepted Length). For truncated geometric distribution  $X \sim TruncGeo(\tau, \gamma)$ ,

$$E[X] = \frac{\tau(1 - \tau^\gamma)}{1 - \tau}. \quad (12)$$

*Proof.* Expand the expected value of random variable  $X$ ,

$$\begin{aligned} E[X] &= \sum_{k=0}^{\gamma} k \cdot \mathbb{P}(X = k) \\ &= \sum_{k=0}^{\gamma-1} k \cdot (1 - \tau)\tau^k + \gamma \cdot \tau^\gamma. \end{aligned}$$

Let  $S = \sum_{k=0}^{\gamma-1} \tau^k = \frac{1 - \tau^\gamma}{1 - \tau}$ . Take differentiation regarding  $\tau$ , we have,

$$\frac{dS}{d\tau} = \sum_{k=0}^{\gamma-1} k\tau^{k-1} = \frac{1 - \gamma\tau^{\gamma-1} + (\gamma - 1)\tau^\gamma}{(1 - \tau)^2}$$

$$E[X] = (1 - \tau)\tau \cdot \frac{dS}{d\tau} + \gamma \cdot \tau^\gamma = \frac{\tau(1 - \tau^\gamma)}{1 - \tau} \quad \square$$

**Theorem 6** (Parallel SD (with Rollback)). *The per-token time of parallel SD under rollback is,*

$$T_{PSD_r} = \frac{2 \cdot \max(\gamma t, ct)}{(1 + \tau\gamma) \cdot \frac{\tau(1 - \tau^\gamma)}{1 - \tau}}. \quad (13)$$

*Proof.* Define the acceptance vector  $\omega = (\omega_1, \dots, \omega_\gamma) \in \{0, 1\}^\gamma$ , where  $\omega_i = 1$  if and only if token  $i$  is accepted. The accepted token count is,

$$X = \sum_{i=1}^{\gamma} \omega_i, \quad \mathbb{P}(\omega_i = 1) = \tau \text{ (i.i.d.)}.$$

To compute the total number of tokens with retry, define two round of: 1)  $\gamma$  tokens (accepted if  $\omega = 1$ ); 2) Retry if Round 1 fails, which yields  $E[X]$  tokens. Thus, the total expectation is:

$$\begin{aligned} E_{total} &= \tau^\gamma(\gamma + E[X]) + (1 - \tau^\gamma) \frac{(E[X] - \gamma\tau^\gamma)}{1 - \tau^\gamma} \\ &= (1 + \tau^\gamma) \cdot E[X] \end{aligned}$$

This implies that Parallel SD (with Rollback) achieves an acceleration of  $(1 + \tau^\gamma) \times$  compared to vanilla SD (with

Rollback). As  $\tau \rightarrow 1$ , the acceleration ratio reaches  $2\times$ , matching the ideal case in Theorem. 3. Thus, we calculate the per-token time with the two rounds:

$$T_{PSD_r} = \frac{T_{total}}{E_{total}} = \frac{2 \cdot \max(\gamma t, ct)}{(1 + \tau^\gamma) \cdot \frac{\tau(1-\tau^\gamma)}{1-\tau}}. \quad \square$$

When  $\gamma = c$ , the speedup ratio of Parallel SD with rollback is  $\mathcal{V}_{PSD_r} = \frac{T_p}{T_{PSD_r}} = \frac{1}{2} \sum_{k=1}^{2c} \tau^k$ . It increases monotonically with the acceptance rate  $\tau$  in polynomial order, so maintaining robust acceptance rates is even more crucial for parallel decoding. ParallelVLM increases  $c$  by pruning and keeps an unbiased alignment for higher  $\tau$ , achieving considerable speedup. As shown in Tab. 7, under the extreme pruning ratio  $\alpha \rightarrow 1$  where  $c^*(\alpha)$  keeps almost the same, we observe the rapid decrease of speedup ratio with lower acceptance rates, confirming the high-sensitivity effect of parallel decoding performance on acceptance rates and the necessity of the visual alignment aware pruning.

Table 7. Experiments with LLaVA-OV-7B/72B on VideoDetail-Caption under extremely higher pruning ratio ( $\alpha \rightarrow 1$ ).

Pruning ratio $\alpha$	0.90	0.95	0.99	1.00
Accepted Lengths $M$	6.82	6.53	5.74	4.53
Acceptance rates $\tau$	0.41	0.37	0.33	0.27
Speedup	$3.43\times$	$3.28\times$	$2.90\times$	$2.53\times$

## D. Why is UV-Prune Essential?

In recent years, numerous studies have focused on pruning VLMs [6, 40, 47, 51] to achieve inference acceleration at the cost of a certain degree of generation quality. In contrast, our work explores the transfer of video-based understanding from more powerful target models to draft models, making it orthogonal to prior research. We retain the full context of the target model to ensure theoretically lossless generation, and perform pruning on the draft model to align with the target model – marking the first application in parallel decoding framework to allow window size expansion.

Feather [9] has discussed the issue of positional bias in the image domain. They observe that the shallow layers of LLMs serve as “active” regions for visual token comprehension and attribute the attention bias at the vision-text boundaries to the more similar RoPE positional encodings. By computing attention scores without positional encodings as the criterion, they achieve better performance gains.

However, directly applying Feather’s method is no longer feasible for the target model to guide the draft model pruning. The target and draft models differ greatly in embedding dimensions and network depth, leading to architectural mismatch when performing intra-LLM pruning directly on the draft model. In addition, Feather [9] requires

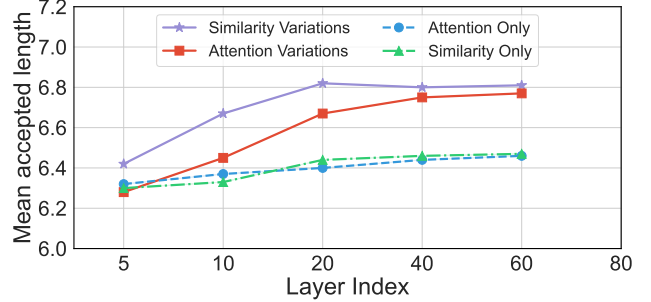


Figure 9. Experiments with LLaVA-OV-7B/72B on VideoDetail-Caption with different draft model pruning strategies.

re-computation of the target model’s attention scores without RoPE, which is not only time-consuming but also conflicts with FlashAttention [7].

To avoid introducing additional computations and ensure compatibility with FlashAttention, we adopt token representations from the shallow layers of the target model as the pruning anchors. Since attention itself is a similarity weight, using cross-modal (vision-text) similarity is a well-motivated approach. To further eliminate the bias induced by positional encodings, we calculate the similarity variations of visual tokens at each position layer by layer to “offset” the interference of positional encodings. In this way, UV-Prune accurately identifies the few key video tokens prioritized by the target model among a large number of tokens, and effectively transfers this focus to the draft model, achieving salient yet robust image-text relevance.

As shown in Fig. 9, both variation and similarity enhance the effectiveness of video token selection. The variation mechanism of UV-Prune mitigates “positional bias”, leading to significant improvements for both attention and similarity criterion. We also observe that similarity-variation converges at  $L = 20$  (and it generalizes well for different model combinations). Thus, we predefine  $L = 20$  as the number of layers for the target model to guide the draft model’s pruning. The propagation of token representations in such shallow layers consumes negligible transmission time ( $\sim 0.01$  s) and ensures that the subsequent pre-filling of the draft model is fully hidden within the target model’s pre-filling span. Most importantly, our alignment-aware pruning method meets the requirement for robust acceptance rates specified in Theorem 6.

## E. Case Study

We present two generation paragraphs in Fig. 10 (LLaVA-OneVision-7B/72B) and Fig. 11 (Qwen2.5-VL-7B/32B) to intuitively demonstrate the effects of unbiased pruning guided by the target model in long-sequence video tokens, as well as the lossless generation performance of ParallelVLM compared to lossy visual token pruning schemes. We observe that ParallelVLM maintains finer-grained details and accurately reflects the realities in the videos.

### LLaVA-OV-7B-72B



Please provide a detailed description of the video, focusing on the main subjects, their actions and the background scenes.



**FastV:** The video opens with a news studio scene where two anchors are seated at a desk with a backdrop displaying the news channel's logo and the words 'Health News.' The scene transitions to a live report on snowy conditions, with a reporter standing outdoors in front of a snow-covered street. The reporter is dressed in a black jacket and is speaking to the camera. The scene then shows a woman in a blue shirt and jeans shoveling snow from a driveway. The video continues with the reporter interviewing a woman who is also shoveling snow. The woman is wearing a black coat and a scarf, and she is holding a yellow shovel. The background shows parked cars ...

**ParallelVLM:** The video begins with a news segment titled 'Health News' featuring two anchors in a studio. The background includes a large screen displaying the 'Health News' logo and a heart symbol. The scene transitions to a live report from a snowy outdoor location, where a reporter in a black jacket with a white scarf is seen speaking to the camera. The reporter is standing in front of a snow-covered street with parked cars and a building in the background. The text 'NEWS8 Shoveling Safety' and 'February Freeze' appears on the screen, indicating the topic of the report. A woman is seen shoveling snow with a yellow shovel, clearing a path ...

Figure 10. Case study for LLaVA-OV-7B/72B.

### Qwen2.5-VL-7B-32B



Please provide a detailed description of the video, focusing on the main subjects, their actions and the background scenes.



**FastV:** The video features a lively outdoor dance class, likely part of a public event or festival, given the large crowd and festive atmosphere. Here's a detailed breakdown of the video: ### \*\*Main Subjects:\*\* 1. \*\*Instructor:\*\* - The central figure is a male instructor wearing a white T-shirt with a green logo and black athletic pants. He is leading the dance class, demonstrating energetic dance moves... 2. \*\*Participants:\*\* - A diverse group of people, men and women, are following the instructor's lead. They are dressed in casual, comfortable clothing suitable for exercise, such as tank tops, leggings, and athletic shoes ...

**ParallelVLM:** The video is vibrant and dynamic that combines elements of dance, fitness, and visual artistry. ### \*\*Segment 1: Neon Light Show\*\* The video begins with a mesmerizing display of neon lights ... ### \*\*Segment 2: Outdoor Dance Class\*\* The scene shifts to an outdoor dance class taking place in a large, open area, likely a public square or park. \*\*Instructor:\*\* A male instructor, wearing a white t-shirt and black pants, leads the class. He demonstrates dance moves and exercises, guiding the participants through synchronized routines. \*\*Participants:\*\* A large group of people, mostly women, are participating in ...

Figure 11. Case study for Qwen2.5-VL-7B/32B.