More than the Sum: Panorama-Language Models for Adverse Omni-Scenes

Supplementary Material

A. Sparse Attention

A.1. Preliminary of Simplified Sparse Attention

To verify the effectiveness of sparse attention with an efficient indexing mechanism in panoramic tasks, we implement a simple yet effective sparse attention, called Simplified Sparse Attention (SSA). Similar to PSA, SSA employs a lightweight indexer to select the Top-K key tokens. However, SSA simplifies the similarity scoring by removing the position-aware gating network. Specifically, the indexer computes a score matrix $\mathbf{I} \in \mathbb{R}^{L\times L}$. The score $I_{t,s}$ between a query token t and a key token s is calculated by

$$I_{t,s} = \sum_{j=1}^{H^I} w_{t,j} \cdot \mathrm{ReLU}(\mathbf{q}_{I,(t,j)}^{\top}\mathbf{k}_{I,(s,j)}), \tag{14}$$

where H^I is the number of selector heads. For each head j, the query vector $\mathbf{q}_{I,(t,j)} \in \mathbb{R}^{d_I}$ and key vector $\mathbf{k}_{I,(s,j)} \in \mathbb{R}^{d_I}$ are obtained via linear projections of the input states $\mathbf{h}_t^{(l-1)}$ and $\mathbf{h}_s^{(l-1)}$. Differ from the gate mechanism in PSA, the weighting term $w_{t,j}$ is derived solely from the query token, i.e.,

$$\mathbf{w}_t^{(l)} = \mathbf{h}_t^{(l-1)}\mathbf{W}_w^{(l)}, \tag{15}$$

where $\mathbf{W}_w^{(l)} \in \mathbb{R}^{d\times H^I}$, and $w_{t,j}$ is the j-th scalar component of $\mathbf{w}_t^{(l)}$, representing the importance of head j for the current query t. The ReLU activation ensures sparsity in the dot-product similarity. Based on the score matrix $\mathbf{I}$, we select the indices of the Top-K relevant keys for each query t, denoted as $\mathcal{S}_t = \{s \mid I_{t,s} \in \text{Top-K}(\mathbf{I}_{t,:})\}$. The subsequent attention computation remains consistent with the sparse paradigm:

$$\mathbf{o}_t^{(l)} = \text{Attention}\left(\mathbf{q}_t^{(l)}, \mathbf{K}_{\mathcal{S}_t}^{(l)}, \mathbf{V}_{\mathcal{S}_t}^{(l)}\right). \tag{16}$$

Finally, the output of the SSA is produced by aggregating the token outputs and applying a linear projection:

$$\mathrm{SSA}(\mathbf{h}^{(l-1)}) = \mathrm{Stack}(\mathbf{o}_1^{(l)}, \ldots, \mathbf{o}_L^{(l)})\mathbf{W}_{O,\text{sparse}}^{(l)}. \tag{17}$$

A.2. Discussion on Proposed PSA

We highlight three key points of our PSA: (1) Unlike pixel-level Deformable CNNs, PSA utilizes *patch-wise* attention. This is specifically designed for high-resolution panoramas to capture long-range semantic dependencies (*e.g.*, wrap-around continuity) that pixel-based methods miss. (2) We introduce a gating mechanism (Eq. 8) as a semantic noise filter to prune vast redundant regions (*e.g.*, sky). This integration of gating and sparsity is supported by recent findings in efficient LLMs [33] for handling massive token sequences. (3) As shown in Fig. 7, the effectiveness of PSA is clear: while the baseline scatters attention noisily, PSA concentrates activation strictly along the semantic horizon (buildings, roads marked in red box), suppressing uninformative backgrounds.

Furthermore, to test the compatibility of our proposed PSA, we integrated it with the LLaVA-Next-7B and conducted experiments on the PanoVQA-mini dataset. The results are shown in Table 7, the addition of PSA enhances performance across all subsets, confirming PSA's robust plug-and-play capability.

Table 7. compatibility of PSA on LLaVA-Next-7B

Model	Avg(N)	Avg(O)	Avg(D)	Avg
LLaVA-Next-7B	15.36	34.14	26.25	24.59
LLaVA-Next-7B + PSA	16.05	33.57	28.53	25.63

B. PanoVQA Benchmark

B.1. Data Collection

PanoVQA-N (NuScenes) As a cornerstone for real-world autonomous driving perception, the NuScenes dataset [4] comprises 1,000 scenes collected in Boston and Singapore. It provides high-quality data from a sensor suite including 6 cameras, LiDAR, and Radar, offering a complete 360° field of view. We utilize the multi-view camera images to construct panoramic inputs, serving as the primary source for training the model's fundamental perception capabilities, such as object detection, and spatial relationships.

PanoVQA-O (BlendPASS). To evaluate the performance of existing VLMs, particularly in occlusion scenarios, we incorporate the BlendPASS dataset. Utilizing a native 360° camera, this dataset captures a seamless, full-surround Field of View (FoV). BlendPASS is characterized by its high-fidelity rendering and diverse visual scenarios, which are often underrepresented in standard real-world collections. By including this data, we significantly improve the model's robustness against varying lighting conditions and environmental contexts, ensuring consistent performance across different visual domains.

PanoVQA-D (DeepAccident). Standard datasets like NuScenes rarely contain collision data, limiting a model's ability to reason about safety-critical situations. DeepAccident [41] is a large-scale synthetic dataset explicitly de-

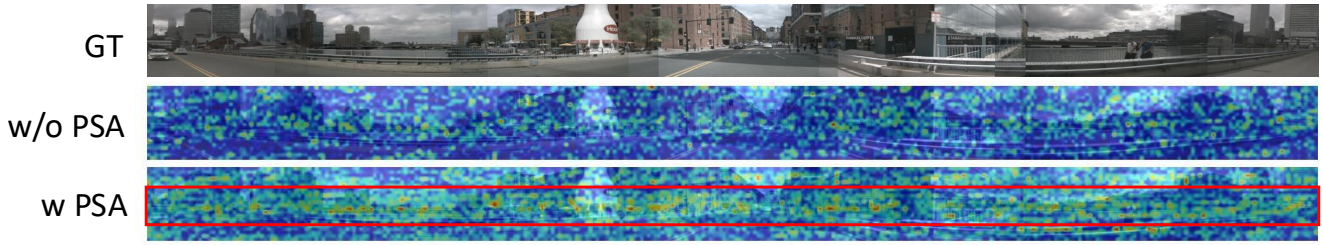

Figure 7. Attention visualization of our proposed PSA. PSA filters uninformative regions like "sky" and distant backgrounds, while concentrating on the critical navigation areas (e.g., the road and surrounding vehicles) across the 360-degree panoramic view.

signed for accident prediction and causal reasoning. It covers diverse accident types and complex interaction scenarios. We leverage this dataset to construct accident-related QA pairs, requiring the model not only to perceive the environment but also to predict potential hazards and explain the underlying causes of accidents, which is crucial for safety-aware autonomous driving.

B.2. Generating Details

Our dataset generation pipeline integrates three distinct autonomous driving sources, including NuScenes, BlendPASS, and DeepAccident, to construct a large-scale panoramic VQA benchmark. The process comprises three sequential stages: (1) geometry-based panoramic synthesis, (2) structured annotation preprocessing, and (3) automated QA generation with quality assurance.

B.2.1 Panoramic Image Synthesis

For the NuScenes and DeepAccident datasets, we synthesize panoramic images using a geometry-based stitching approach following by OneBEV [43]. We model the environment using a viewing sphere centered at the optical origin O. Each of the six camera views, denoted as $\mathcal{I}_i$ for $i \in \{1, \ldots, 6\}$, is treated as a 2D image plane tangent to this sphere at a specific point M_i, aligned with the camera's extrinsic optical axis.

The pixel mapping is computed via ray casting. For each target pixel $P_{j,k}$ in the panoramic coordinate system, we cast a 3D ray $\vec{l}_{j,k}$ outward from the origin O. The source pixel correspondence is established by calculating where this ray intersects the camera planes:

$$\mathbf{N}_{j,k}^{(i)} = \text{Intersect}(\vec{l}_{j,k}, \mathcal{I}_i), \tag{18}$$

$$\mathcal{P}(P_{j,k}) = \mathcal{I}_{i^*}(\text{Proj}(\mathbf{N}_{j,k}^{(i^*)})), \tag{19}$$

where $\mathcal{P}(P_{j,k})$ denotes the assigned pixel value (e.g., RGB color) in the final panorama $\mathcal{P}$ at spatial coordinate (j,k). $\mathbf{N}_{j,k}^{(i)}$ represents the 3D intersection point of the ray with the i-th camera plane, and $\text{Proj}(\cdot)$ projects this 3D point back into the 2D pixel coordinates of that specific camera to fetch the source value.

Since the cameras' fields of view naturally overlap, a single ray might hit multiple valid camera planes. To prevent blurry ghosting artifacts caused by pixel blending, we resolve these overlaps using a strict deterministic priority rule: $i^* = \min\{i \mid \mathbf{N}_{j,k}^{(i)} \in \text{Valid}(\mathcal{I}_i)\}$. This guarantees spatial consistency by exclusively selecting the valid camera with the highest priority (the lowest index i^*). Note that the BlendPASS dataset provides native panoramic imagery, thereby bypassing this entire stitching phase.

B.2.2 Structured Annotation Processing

To standardize object properties across heterogeneous sources, we use a quadruple to denote each object in the scene. Each object of interest is encoded as a tuple $\mathcal{Q} = (c, \theta, d, v/s)$, representing category, direction, distance, and visibility/speed, respectively.

PanoVQA-N (NuScenes Baseline). We derive spatial attributes from 3D bounding boxes. The direction θ is the angular displacement relative to the ego-vehicle's heading vector $\vec{v}_{\text{ego}}$, and visibility v is original meta informations.

PanoVQA-O (BlendPASS Occlusion). Focusing on occlusion handling, we employ the DA2 [23] for pixel-level depth estimation. For an object represented by a polygonal region $\mathcal{R}$, the distance is computed as the regional average:

$$\bar{D} = \frac{1}{|\mathcal{R}|} \sum_{(u,v)\in\mathcal{R}} D_{(u,v)}, \tag{20}$$

where $D(u,v)$ is the estimated depth at pixel (u,v). Occlusion relationships are determined by comparing the geometric intersection and y-axis ordering of object polygons.

PanoVQA-D (DeepAccident). To incorporate temporal dynamics from static imagery, we extend the quadruple format to include speed s. The tuple becomes $\mathcal{Q}_D = (c, \theta, d, s)$, where ground-truth speed data is used to compensate for the ambiguity of motion inference in single-frame inputs.

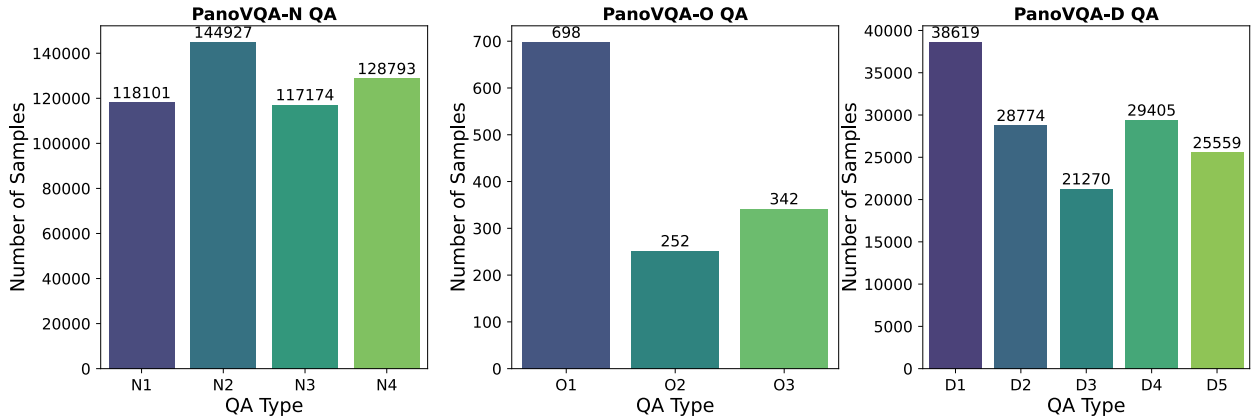

Figure 8. Distribution of QA samples in PanoVQA. The dataset features disparate scales to mimic real-world distributions: PanoVQA-N (left) offers a large-scale foundation (∼510k), PanoVQA-O (middle) introduces a challenging long-tailed setting (∼1.3k), and PanoVQA-D (right) covers safety-critical events (∼143k).

```
qwenvl/train/train_qwen.py
  --model_name_or_path "Qwen2.5-VL-7B-Instruct"
  --run_name "qwen2.5-vl-finetune"
  --deepspeed "scripts/zero3.json"
  --dataset_use "NuScenes_train%60,DeepAccident_train
      %90"
  --data_flatten True
  --tune_mm_vision True
  --tune_mm_mlp True
  --tune_mm_llm True
  --tune_mm_adaptorformer False
  --bf16 True
  --output_dir "./output/train_whole/7B(vit,mlp,llm)
      _baseline"
  --num_train_epochs 1.0
  --per_device_train_batch_size 8
  --per_device_eval_batch_size 16
  --gradient_accumulation_steps 1
  --max_pixels 50176
  --min_pixels 784
  --save_strategy "steps"
  --save_steps 100
  --save_total_limit 1
  --learning_rate 5e-6
  --weight_decay 0
  --warmup_ratio 0.03
  --max_grad_norm 1
  --lr_scheduler_type "cosine"
  --logging_steps 1
  --model_max_length 8192
  --gradient_checkpointing True
  --dataloader_num_workers 8
  --report_to "wandb"
```

Figure 9. Hyperparameters for Qwen2.5-VL Baseline.

B.2.3 Automated QA Generation and Statistics

We employ `GPT-5-mini` to generate QA pairs, incorporating structured quadruples $\mathcal{Q}$ into the template shown in Fig. 10. To optimize the trade-off between reasoning depth and efficiency, we apply adaptive reasoning effort settings: `minimal` for standard driving scenes (PanoVQA-N) and `low` for the more challenging scenarios in PanoVQA-O and PanoVQA-D.

Quality Assurance. The pipeline includes a two-stage validation: (1) *Automated Filtering* to verify semantic alignment between the generated QA and source annotations; (2) *Human Evaluation* to assess correctness and fluency.

Statistics. The final PanoVQA benchmark comprises a total of 538,635 training samples and 115,279 validation samples. The dataset features an average question length of 19.07 words and an answer length of 42.4 words, ensuring sufficient complexity for comprehensive evaluation across normal, occluded, and accident-prone scenarios.

Panoramic QA. To ensure the model processes the full 360° context (i.e., panoramic scene), we emphasize panoramic scene in three ways. (1) The system prompt explicitly include "panoramic scene" (see Figure 10) to force generate panorama-related qa pairs during generation. (2) The questions utilize keywords like "*panoramic scene*" and "*whole scene*" to trigger global reasoning. Our quadruple tuple includes a direction field that compels the model to explicitly model spatial relationships.

B.3. Data Statistics

As illustrated in Figure 8, PanoVQA spans diverse scales while maintaining internal balance. **PanoVQA-N** provides a massive foundation (>500k) with samples evenly distributed across four QA types. Similarly, **PanoVQA-D** (∼143k) maintains a relatively uniform distribution across its five accident categories. In the **PanoVQA-O** subset (∼1.3k), the data remains distributed across its types without extreme skew. This internal balance prevents the model from overfitting to specific question patterns, while the global scale disparity effectively tests robust generalization.

B.4. Human Evaluation

To assess annotation fidelity, we implemented a dual-verification protocol on the *PanoVQA-mini* subset involving five human experts. We partitioned the evaluation samples

Table 8. Comparison of multi-view (6 cameras) and panoramic (1 camera) modeling across Normal (N), Occluded (O), and Accidental (D) scenes.

Model	Input Strategy	Metrics (↑)			
		Avg(N)	Avg(O)	Avg(D)	**Avg**
Qwen2.5-VL-3B	Multi-view (6 cam)	16.82	**32.40**	**34.92**	**28.26**
	Uni-view (2×3 grid)	15.69	31.27	30.39	25.71
	Panoramic (1 pano)	**16.86**	30.44	31.25	26.25
Qwen2.5-VL-3B-SFT	Multi-view (6 cam)	26.33	39.88	**54.45**	40.22
	Uni-view (2×3 grid)	26.04	40.52	50.95	40.04
	Panoramic (1 pano)	**29.68**	**40.98**	51.08	**41.42**

equally between human reviewers and GPT-4o; the consistent scoring distributions observed across both groups attest to the dataset's reliability. Quantitatively, as detailed in Table 4, 96% of the samples were classified as "Valid" or "High Quality." Furthermore, the high degree of alignment—with a deviation of less than 0.5% between GPT-4o and human evaluations—confirms that our automated pipeline serves as a robust proxy for human judgment.

C. Implementation Details

C.1. Training Script

To ensure the reproducibility of our experiments, we present the detailed training command and hyperparameter configuration in Fig. 9. We utilize `Qwen2.5-VL-7B-Instruct` as our base foundation model. We use supervised fine-tuning and unfreezing all components, including the visual encoder, MLP-based merger, and LLM.

C.2. Evaluation Prompt

Fig. 11 illustrates the prompt utilized to score the model inference on a scale of 1∼5. After obtaining the raw responses from the evaluator, our post-processing script automatically parses the numerical ratings and aggregates them to assess model performance from multiple perspectives. Specifically, it computes normalized scores for each item based on our pre-defined categories, respectively. Finally, the system derives a comprehensive overall score, reported with a precision of two decimal places, to facilitate accurate quantitative comparisons.

D. More Experiment Analysis

D.1. Qualitative Results

To systematically evaluate the effectiveness of our panoramic approach, we compare three distinct visual input modalities for the VLM. As summarized in Table 8, the experimental settings are defined as follows:

- **Multi-view (6-cam):** The 6 surrounding camera images are input as independent sequences of visual token. We utilize text prompts to identify each view (e.g., "Front: <image1>, Front-Left: <image2>..."), relying on the LLM to mentally reconstruct the spatial topology.
- **Uni-view (2×3 grid):** The 6 images are spatially concatenated into a single 2×3 grid image before feeding into the vision encoder. This preserves the original resolution without an extremely unbalanced aspect ratio.
- **Panoramic (1-pano):** Our proposed method, where images are cylindrically projected and stitched. Note that this process involves *vertical cropping* to remove distortion and blank areas, resulting in a reduction of total pixels compared to the raw 6-camera input.

Table 9. Proprietary results on PanoVQA-mini.

Model	Avg(N)	Avg(O)	Avg(D)	Avg
Gemini-3-Pro	18.04	20.86	37.32	26.78
GPT-5	22.02	40.55	51.64	38.99
Claude-Sonnet-4.5	19.29	41.73	52.76	38.84

Analysis. The quantitative results in Table 8 reveal a compelling finding. While the Multi-view approach preserves the highest image fidelity and vertical Field of View (FoV), it forces the model to process fragmented visual information. In contrast, despite the inevitable resolution loss and vertical cropping inherent in the stitching process, the **Panoramic** input achieves superior performance after Supervised Fine-Tuning (SFT), reaching an overall score of **41.42** compared to **40.22** for the Multi-view baseline.

This performance gain is particularly evident in *PanoVQA-N* (29.68 vs. 26.33) and *PanoVQA-O* (40.98 vs. 39.88) scenarios. We attribute this to the **seamless spatial context** provided by the panorama. The continuous 360° view eliminates the cognitive burden of stitching disjointed images, allowing the model to better understand spatial relationships and object continuity across camera boundaries. This demonstrates that for holistic scene understanding in autonomous driving, spatial continuity is often more critical than maximizing pixel count.

D.2. More Results on PanoVQA-mini

We additionally evaluate several advanced proprietary models on PanoVQA-mini, including Gemini-3-Pro, GPT-5, and Claude-Sonnet-4.5. As shown in 9, their performance indicates that panoramic understanding remains a challenge for generalist models, likely due to the scarcity of panoramic data in their pre-training corpora.

D.3. More Examples

Reasoning in Normal Scenarios. Fig. 12 illustrates a perception task from PanoVQA-N, where the model is asked to identify the visibility and location of the closest adult. The 1-Pano model accurately locates the adult in the "front"

SYSTEM_MESSAGE = ‘‘Generate 15 QA pairs from one panoramic scene JSON file from perspectives from users. Each scene includes object annotations (category, attributes, and relative direction). Generate Question Answer pairs for each panoramic scene, which should cover all 4 QA Types. (At least 1 pair for each type) The Question Answer pairs must follow the instructions and rules below, taking the given sample as an example.’’

PROMPT = ‘‘
Rules:
1. Analyze the panoramic scene annotations, focusing on:
- Use a quadruple tuple (category, direction, distance, visibility) to describe an object (e.g., ‘a fully visible pedestrian in the back right around 9 meters’). Use vague numbers to express distances, without decimal points.
- Object attributes and spatial relationships (visibility, distance, and direction relative to the ego car).
2. Only describe clear information in the images, do not fabricate or invent in the answers.
3. Base all answers only on what is actually visible in the provided json data. Do not make assumptions or invent details.
4. All positions and absolute coordinates must be described in a directional manner. (Describe exact direction such as ‘ front left’, ‘back right’, ‘front’, etc.)
5. Visibility Encoding:
1: Low visibility (0−40%)
2: Medium visibility (40−60%)
3: High visibility (60−80%)
4: Fully visible (80−100%)
6. The question can be slightly modified to produce different answers. For multi−item answers, maintain the order relevant to the question (e.g., nearest to farthest). One question should correspond to one answer.
7. All responses should be written expressions in natural language, avoid using symbols or brackets.

Instructions:
Fully consider following levels to generate questions and multiple answers:
1. Short Level QA: QA pairs that query the basic information in the json file or single panoramic image, the answer can be completely verified by the ground truth.
2. Long Level QA: QA pairs that contain multiple objects, with attributions and their relationships in concern, the answer stems mainly from the combined ground truth feature information. The questions should be short and rough, while the answers should be detailed and comprehensive. The answer can be partially verified.

QA Types:
−Type N1− Global scene understanding (overall scene description, including objects, visibility)
−Type N2− Object and attribute identification (attributes of objects present)
−Type N3− Relationship identification (object−object spatial relationships)
−Type N4− Location description (where objects are located in relation to ego car, ego−object spatial relationships)

JSON file:
{json_data}
’’

Figure 10. Prompt used in PanoVQA generation. Using PanoVQA-N as an example.

at approximately 9 meters and correctly identifies them as “fully visible.” In contrast, the 6-Cam model hallucinates the direction as “front left” and provides extraneous speed information not present in the visual cues. This discrepancy highlights that the seamless 360° context of the panoramic input aids the model in establishing a more precise egocentric coordinate system, whereas the fragmented multi-view input may introduce spatial ambiguities across camera overlaps.

Reasoning under Occlusion. In the complex scenario

```
messages = [
    {
        ‘‘role’’: ‘‘system’’,
        ‘‘content’’:
            ‘‘You are an intelligent evaluator designed to evaluate the correctness and similarity of generative outputs
            for question−answer pairs. ’’
            ‘‘Your task is to compare the model prediction answer with the correct answer and determine if they match
            in meaning. Here’s the scoring criteria:\n\n’’
            ’’### Scoring Criteria:\n’’
            ‘‘5 = Perfect match or Correct in meaning\n’’
            ‘‘4 = Key information correct, minor flaws\n’’
            ‘‘3 = Partially correct\n’’
            ‘‘2 = Mostly wrong answer for key query, but some relevance\n’’
            ‘‘1 = Completely wrong or nonsense sentences\n\n’’
            ‘‘Your response must ONLY be the integer score (e.g., 4). DO NOT include any text or explanation.’’
    },
    {
        ‘‘role’’: ‘‘user’’,
        ‘‘content’’:
            f‘‘Question: {question}\n’’
            f‘‘Correct Answer: {gt_answer}\n’’
            f‘‘Predicted Answer: {pred_answer}\n\n’’
            ‘‘Please provide a score from 1 to 5 based on how well the predicted answer matches the correct answer.’’
    }
]
```

Figure 11. Prompt used in evaluation.

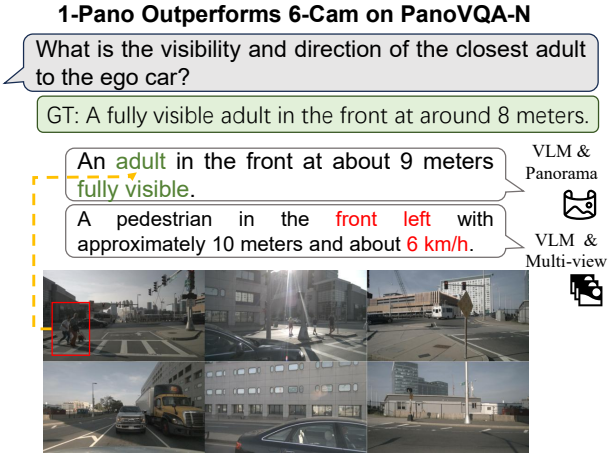

Figure 12. Qualitative comparison on **PanoVQA-N**. The panoramic model (1-Pano) correctly identifies the spatial location (“front”) and visibility of the pedestrian, whereas the multi-view model (6-Cam) hallucinates a “front left” direction due to fragmented spatial context.

shown in Fig. 13 (PanoVQA-O), the vehicle encounters a cluster of bicycles. The prompt requires a defensive driving maneuver. The 6-Cam model’s reasoning appears fragmented, potentially due to the bicycle cluster spanning across camera boundaries. Conversely, the 1-Pano model leverages the unified view to holistically assess the density of the cluster, recommending a contextually safer maneuver. This suggests that spatial continuity is critical for planning tasks where objects of interest (e.g., crowds, large vehicles) span multiple viewing angles.

Reasoning in Safety-Critical scenarios. Fig. 14 presents a rear-end collision scenario from PanoVQA-D. Interestingly, both the 6-Cam and 1-Pano models successfully identify the collision mechanics (“rear-side impact”) and accurately estimate the severity as “moderate to severe.” This result is significant as it validates that our panoramic stitching and cropping process—despite the theoretical loss of vertical field-of-view—preserves the essential fine-grained visual details required for high-level accident analysis.

Conclusion. The qualitative results corroborate our quantitative findings: while multi-view inputs preserve raw pixel fidelity, the **1-Pano representation offers superior spatial**

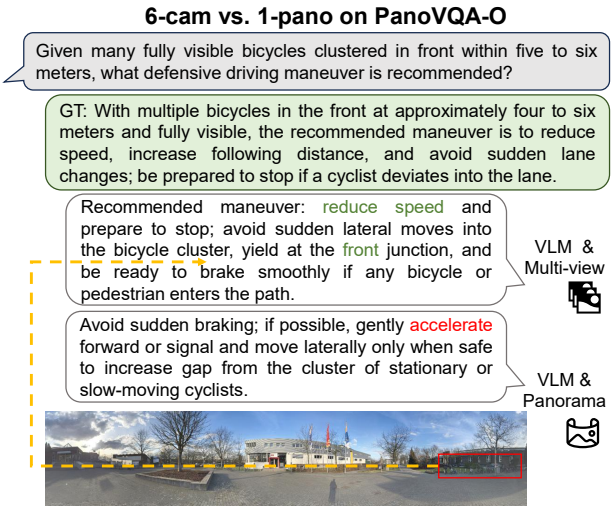

Figure 13. Qualitative comparison on **PanoVQA-O**. Facing a cluster of bicycles, the 1-Pano model benefits from the unified view to propose a coherent defensive maneuver, demonstrating the importance of seamless context for planning.

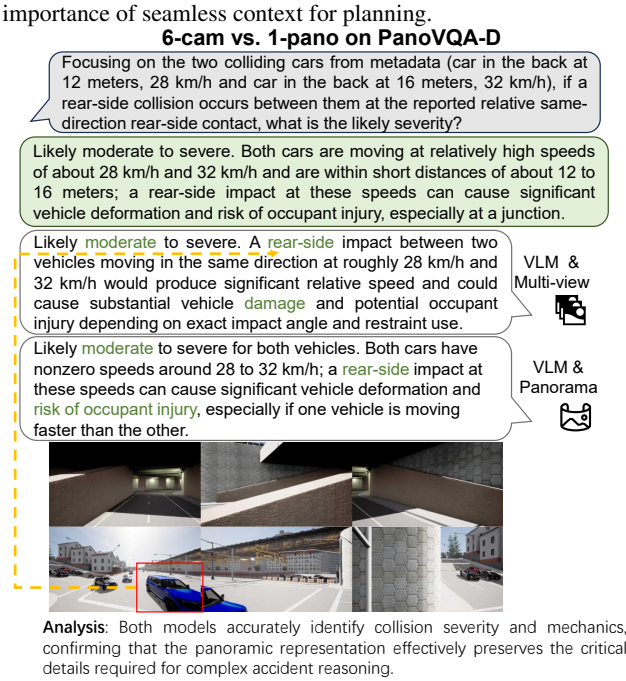

Figure 14. Qualitative comparison on **PanoVQA-D**. Both models accurately predict the severity and type of collision. This confirms that the panoramic representation retains critical visual details necessary for complex accident reasoning.

coherence. This coherence proves advantageous in tasks requiring precise localization and holistic scene understanding, without compromising performance on semantic reasoning tasks like accident analysis.

E. Limitation and Future Work

Despite the robust spatial reasoning demonstrated by our panoramic framework, the current geometric stitching process introduces inevitable limitations. To achieve a seamless projection, the method necessitates vertical cropping and induces boundary distortions, resulting in pixel-level information loss. This reduction in fidelity explains instances where raw multi-view inputs outperform the panoramic representation, especially in zero-shot settings or scenarios demanding high-resolution visual detail.

Future research will focus on enhancing visual fidelity and expanding into the temporal domain. We plan to explore more advanced image stitching techniques to reduce information loss and preserve better visual details compared to current fixed projection methods. Furthermore, acknowledging the dynamic nature of autonomous driving, we aim to extend our framework to process video inputs, enabling the model to understand movement and unfolding events rather than just static scenes.