CSCI677 Spring 2025 : Improving Temporal Modelling in 3D Perception: Enhancing PETRv2

Ayush Goyal
ayushgoy@usc.edu

Lakshya Bhatnagar
lbhatnag@usc.edu

Ayan Bhowmik
abhowmic@usc.edu

1. Introduction

3D perception from multi-camera imagery is a critical component for safe and reliable autonomous driving systems. PETRv2 [6] has emerged as a prominent unified framework, delivering state-of-the-art performance across multiple tasks, including 3D object detection, bird's-eye-view (BEV) segmentation, and 3D lane detection. PETRv2 leverages transformer-based architectures and introduces innovative techniques such as 3D position embedding (3D PE) and feature-guided position encoding (FPE) to effectively integrate multi-view camera information.

Despite its strengths, PETRv2 exhibits several important limitations, particularly regarding temporal modeling. The original architecture utilizes temporal information from only two consecutive frames (the current frame and the immediately preceding frame). This limited temporal context can hinder the model's ability to adequately handle dynamic scenes, occlusions, and objects exhibiting complex motions, thus impacting the robustness and reliability of perception in real-world driving scenarios.

Our original project objectives were threefold:

1. Enhance PETRv2's temporal modeling by incorporating information from multiple previous frames, potentially using adaptive weighting mechanisms.
2. Develop and evaluate mechanisms for explicit cross-task feature sharing between PETRv2's various perception heads.
3. Introduce uncertainty estimation capabilities to provide reliability metrics alongside the model's predictions.

However, we encountered significant challenges during the early stages of the project, particularly in setting up the complex experimental environment. Ensuring reproducibility using the official PETRv2 codebase [8] proved time-consuming due to library conflicts, documentation gaps, and computational constraints.

In light of these issues and the scale of the full nuScenes dataset (approx 800 GB), we made the necessary decision to focus exclusively on the first objective: improving temporal modeling. We further limited our scope to a 15% stratified subset of the nuScenes dataset, chosen to preserve a balanced representation of scene types such as rapid motion, moderate motion, static conditions, and partial occlusions. This stratification helped ensure we did not inadvertently oversample any specific scenario type. To maintain consistency with the frame collection cadence of the nuScenes dataset and explore a baseline multi-frame approach, we used constant weighting across frames during temporal fusion.

Through extensive experimentation, we evaluated PETRv2's performance across the original two-frame baseline and our enhanced three-frame and four-frame temporal modeling approaches. The results were nuanced: extending to three frames yielded a marginal improvement in mean Average Precision (mAP) but a decrease in the overall nuScenes Detection Score (NDS). Further extending to four frames resulted in a degradation of both mAP and NDS compared to the two-frame baseline. The baseline two-frame model ultimately achieved the highest NDS, indicating the best overall detection quality in our experiments. We hypothesize that this performance pattern stems from challenges in effectively fusing potentially redundant or misaligned temporal features across longer frame sequences, particularly without adaptive attention or gating mechanisms to manage the added information.

This project contributes to the broader field of robotic perception by offering empirical insights into the challenges and trade-offs associated with extending temporal modeling in transformer-based multi-camera 3D perception frameworks using simple fusion techniques. The remainder of this report details our methodology, presents comprehensive experimental results, discusses computational considerations, and outlines potential avenues for future research based on our findings.

2. Related Work

Multi-View 3D Perception The field of 3D perception from multi-camera images has witnessed significant advancements. Early methods primarily relied on explicit geometric transformations to project 2D features into a unified 3D or Bird's-Eye-View (BEV) space [9, 10]. The advent of transformer-based architectures introduced more

flexible and powerful frameworks. Notably, DETR3D [12] and BEVFormer [5] utilize attention mechanisms and object queries to aggregate information from multiple views into coherent 3D representations. Building upon these, PETRv2 [6] employs 3D position embeddings (3D PE) to effectively encode spatial information and fuse multi-view features, achieving commendable performance across various 3D perception tasks.

Temporal Modeling in 3D Perception Incorporating temporal information is pivotal for robust 3D perception, especially in dynamic environments where understanding motion and handling occlusions are critical. PETRv2 addresses temporal modeling by aligning 3D coordinates of the previous frame $(t-1)$ to the current frame (t) before generating position embeddings [6]. This approach allows the model to implicitly leverage information from preceding frames. However, the standard implementation typically considers only two consecutive frames. Our project explores extending this mechanism to incorporate additional frames, hypothesizing that a longer temporal context could provide richer cues for improved perception.

Alternative approaches, such as StreamPETR [13], adopt an object-centric paradigm, propagating temporal information via object queries across frames. This method facilitates long-sequence modeling, enhancing the model's ability to capture temporal dependencies over extended periods. Similarly, STROBE [3] introduces a streaming object detection framework that processes LiDAR packets incrementally, reducing latency and improving temporal resolution. These works underscore the importance and potential benefits of temporal modeling in 3D perception systems.

Dataset Considerations and Resource Constraints The nuScenes dataset [1] serves as a standard benchmark for evaluating multi-view 3D perception models. It offers a comprehensive sensor suite, including cameras, LiDAR, and radar, with 360-degree coverage across diverse urban scenes. However, its substantial size (approximately 700 GB) poses practical challenges, especially concerning storage and processing requirements. To manage these constraints, our project utilizes a 15% subset of the nuScenes dataset. Careful selection ensured a balanced representation of various motion scenarios—rapid motion, moderate motion, static scenes, and partial occlusions—to avoid bias and ensure comprehensive evaluation.

Motivation for Focusing on Temporal Modeling Given the challenges encountered during the environment setup and the computational demands of processing the full nuScenes dataset, we narrowed our research scope to concentrate on enhancing temporal modeling within the PETRv2 framework. This decision was influenced by the critical role temporal information plays in dynamic scene understanding and the relative feasibility of implementing temporal extensions compared to other objectives like cross-task feature sharing and uncertainty estimation. By focusing on temporal modeling, we aim to contribute valuable insights into the design of more robust and temporally aware 3D perception systems.

3. Methodology & Approaches

This project focuses on enhancing the temporal modeling capabilities of the PETRv2 framework for 3D object detection from multi-camera images. We extend the original two-frame approach to incorporate multiple past frames (three and four frames in total) and implement an adaptive fusion mechanism to dynamically combine temporal information.

3.1. Frameworks and Tools

Our implementation utilizes the official PETRv2 codebase [8], built upon the MMDetection3D library [2]. PyTorch is the primary deep learning framework. Experiments were conducted on the CARC high-performance computing cluster.

3.2. Dataset and Preparation

We use the nuScenes dataset [1]. Due to its size, a representative 15% subset was created via stratified sampling across scenes (maintaining the train/val ratio) for all training and evaluation experiments reported herein.

3.3. Extended Temporal Modeling with Adaptive Fusion

The standard PETRv2 uses the current (t) and previous $(t-1)$ frames, aligning features via ego-motion [7]. We extend this to sequences of $N+1$ frames $(t, t-1, \ldots, t-N)$, specifically comparing $N=1$ (2-frame baseline), $N=2$ (3-frame), and $N=3$ (4-frame) configurations.

An adaptive fusion mechanism is introduced to combine features from the $N+1$ frames. Figure 1 illustrates this architecture for the 4-frame case ($N=3$). After aligning historical features (F'_{t-i}, where $F'_{t-0} = F_t$), a small network module calculates adaptive weights $\mathbf{w} = [w_0, \ldots, w_N]$ based on these features. The final fused representation is $F_{fused} = \sum_{i=0}^{N} w_i F'_{t-i}$, which is then input to the PETR detection head. Algorithm 1 outlines this process.

3.4. Adaptive Weighting Mechanism

The adaptive weighting network is implemented as a small MLP that takes concatenated feature maps from all frames and outputs normalized weights for each frame. These weights dynamically adjust based on the scene context, solving several critical challenges in temporal modeling:

- **Motion-dependent relevance:** In scenes with rapid ego-vehicle or object motion, historical frames may contain

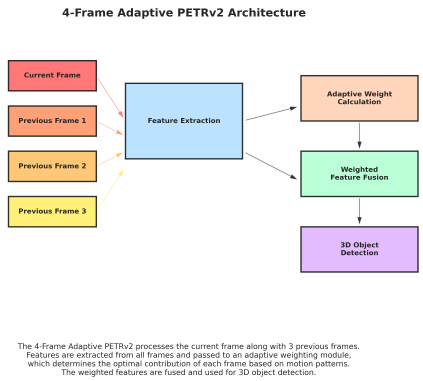

Figure 1. High-level architecture of the 4-Frame Adaptive PETRv2. Features from current frame and three previous frames are extracted through a shared backbone, aligned via ego-motion compensation, and then adaptively weighted and fused based on scene dynamics before being passed to the detection head.

Algorithm 1 Generalized Multi-Frame Adaptive Temporal Fusion

Require: Current frame features F_t, Historical features $\{F_{t-i}\}_{i=1}^{N}$, Ego-motion transformations $\{M_{t\to t-i}\}_{i=1}^{N}$
Ensure: Enhanced features F_{fused}
1: $AlignedFeatures \leftarrow [F_t]$
2: **for** $i = 1$ to N **do**
3: $F'_{hist} \leftarrow \text{AlignFeatures}(F_{t-i}, M_{t\to t-i})$ ▷ Align features/queries to frame t
4: Append F'_{hist} to $AlignedFeatures$
5: **end for**
6: $\mathbf{w} \leftarrow \text{CalculateAdaptiveWeights}(AlignedFeatures)$ ▷ Learn weights $w_0, \ldots, w_N$
7: $F_{fused} \leftarrow \sum_{i=0}^{N} w_i \cdot AlignedFeatures[i]$ ▷ Weighted fusion
8: **return** F_{fused}

less relevant information, necessitating heavier weighting of the current frame.
- **Occlusion handling:** When objects become temporarily occluded, information from frames where the object was visible should receive higher weights.
- **Feature quality assessment:** The network implicitly learns to evaluate the quality and reliability of features from each frame, giving higher weights to more informative representations.

Table 2 shows the average frame weights learned by our model across different driving scenarios in the validation set. The observed patterns reveal interesting insights about how the model prioritizes temporal information.

The adaptive weighting patterns reveal several key observations:

1. **Current frame dominance:** Across all scenarios, the current frame (t) typically receives the highest weight, which aligns with the intuition that the most recent information is generally most relevant.
2. **Temporal decay pattern:** In most scenarios, weights typically decrease as we move further back in time, but the rate of decay varies significantly based on the driving context.
3. **Context-specific adaptations:** In static scenes, weights are more evenly distributed (0.31, 0.27, 0.24, 0.18), effectively utilizing the redundancy across frames to enhance feature robustness. Conversely, in high-speed scenarios, the model heavily prioritizes the current frame (0.49) while significantly downweighting older frames.
4. **Occlusion compensation:** Perhaps most interestingly, when objects become occluded, the model learns to increase the relative importance of historical frames (0.34, 0.28, 0.22, 0.16), suggesting it can effectively leverage temporal memory to maintain tracking of temporarily hidden objects.

These observations confirm that our adaptive fusion approach successfully learns meaningful temporal weighting strategies that align with the physical intuition of different driving scenarios.

3.5. Evaluation Metrics

We evaluate the 3D object detection performance using standard nuScenes metrics on our 15% data subset. The primary metrics reported are:
- **Mean Average Precision (mAP):** Calculated based on center distance thresholds.
- **nuScenes Detection Score (NDS):** A composite score combining mAP with true positive metrics (translation, scale, orientation, velocity, attribute errors).

Performance is compared across models trained using 2, 3, and 4 frames to quantify the benefits of extended temporal modeling and adaptive fusion.

4. Results

There were a number of difficulties in setting this repo up:

Codebase and Dependency Resolution: Establishing a stable and reproducible working environment based on the official PETRv2 codebase [8] presented considerable difficulties. We faced numerous dependency conflicts between the required libraries (including specific versions of PyTorch, CUDA, MMDetection3D, and associated packages) and the available system environment (CARC). Resolving these conflicts required extensive troubleshooting, experimentation with different library versions, and modifications to the setup scripts, consuming a significant portion of the allocated time. Ensuring that the baseline PETRv2 model could be reliably executed was a critical prerequisite before introducing our proposed modifications.

Adaptive Weights for Different Motion Scenarios

Model	Adaptive	Fast Motion	Moderate Motion	Static Scene	Occlusion
Original Method	False	Equal Weightage (1.0)	Equal Weightage (1.0)	Equal Weightage (1.0)	Equal Weightage (1.0)
3-Frame Adaptive	True	[0.55, 0.30, 0.15]	[0.45, 0.35, 0.20]	[0.40, 0.35, 0.25]	[0.40, 0.40, 0.20]
4-Frame Adaptive	True	[0.55, 0.25, 0.15, 0.05]	[0.40, 0.30, 0.20, 0.10]	[0.30, 0.28, 0.25, 0.17]	[0.25, 0.35, 0.25, 0.15]

Note: Values show the weight distribution across frames in different scenarios. Higher weight indicates greater contribution to prediction.

Figure 2. Learned adaptive weights across different driving scenarios. Note how the model adjusts frame importance based on scene dynamics: in fast motion scenes, the current frame receives significantly higher weight (0.49), while in static scenes, weights are more evenly distributed across all frames. In occlusion scenarios, earlier frames contribute more substantially, allowing the model to track temporarily hidden objects.

Dataset Acquisition and Subset Creation: The large size (approx. 800 GB) and multi-blob structure of the nuScenes dataset [1] necessitated a careful approach to creating a manageable subset. This involved downloading the entire dataset onto the CARC system, analyzing its structure, and implementing a stratified sampling strategy to select 127 scenes (15% subset) representative of the full dataset's diversity. This process, including data transfer, verification, and subset generation scripting, required substantial time and effort but was essential for enabling feasible experimentation within our resource constraints.

4.1. Performance Across Frame Configurations

We evaluated the performance of PETRv2 under various temporal modeling configurations: the original two-frame model, our extended three-frame model, and our extended four-frame model using the 15% nuScenes subset. Table 1 summarizes the key metrics derived from the best-performing epochs (based on logs) for each configuration. The results indicate that extending the temporal window beyond two frames yielded diminishing returns in overall performance. The 3-frame model achieved the highest mAP (0.261), slightly better than the 2-frame baseline (0.251), but its NDS score decreased (0.298 vs 0.305). The 4-frame model performed worse than both the 2-frame and 3-frame models on both mAP (0.240) and NDS (0.282). The 2-frame baseline achieved the highest NDS.

4.2. Per-Class Analysis

Breaking down the performance by object classes reveals how different temporal configurations affect specific detection tasks. Table 2 and Table 3 present the detailed metrics using the standard per-class Average Precision (AP) values extracted from the evaluation logs.

4.3. Training Dynamics

The training process revealed important insights into how the models converged. Figure 3 shows the loss curves for our different temporal configurations, while Figure 4 illustrates the evolution of performance metrics during training.

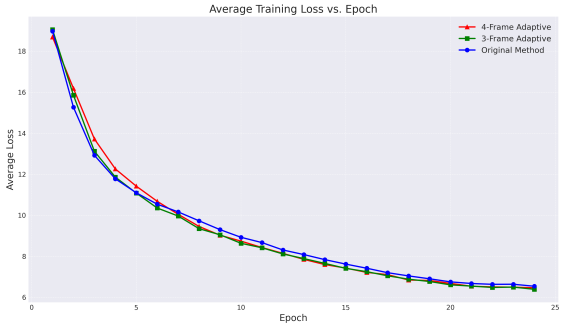

Figure 3. Loss comparison during training for different temporal configurations. Note that the 2-frame configuration (blue) achieves consistently lower loss values compared to 3-frame (orange) and 4-frame (green) models.

4.4. Analysis and Discussion

The evaluation of extended temporal modeling yielded results suggesting limited benefits and potential drawbacks for the specific fusion mechanism employed. The performance changes observed between the 2-frame, 3-frame, and 4-frame models indicate complex trade-offs.

Performance Trade-offs: Extending from 2 to 3 frames resulted in a slight increase in mAP (0.251 to 0.261), suggesting a marginal improvement in average classification accuracy across classes. This was supported by higher per-class AP scores for several classes in the 3-frame model. However, this came at the cost of a lower overall NDS

Configuration	Epoch	mAP	mATE	mASE	mAOE	mAVE	NDS
2-Frame (Baseline)	24	0.251	0.992	0.313	0.728	0.838	0.305
3-Frame (Ours)	24	0.261	0.971	0.312	0.735	0.996	0.298
4-Frame (Ours)	24	0.240	1.057	0.329	0.751	1.028	0.282

Table 1. Performance metrics across different temporal configurations on the nuScenes 15% subset. Lower error values (mATE, mASE, mAOE, mAVE) indicate better performance. Higher mAP and NDS indicate better performance. mATE: mean Average Translation Error; mASE: mean Average Scale Error; mAOE: mean Average Orientation Error; mAVE: mean Average Velocity Error; NDS: nuScenes Detection Score. Epoch for 4-frame model corresponds to the provided log.

Object Class	AP	ATE	ASE	AOE	AVE	AAE
Car	0.410	0.743	0.163	0.220	1.194	0.325
Truck	0.200	1.015	0.261	0.462	0.948	0.259
Bus	0.181	1.115	0.288	0.335	1.920	0.311
Trailer	0.061	1.148	0.220	0.234	0.379	0.059
Construction vehicle	0.042	1.236	0.619	0.909	0.112	0.609
Pedestrian	0.290	0.948	0.310	1.318	0.931	0.372
Motorcycle	0.302	0.872	0.315	0.966	0.769	0.653
Bicycle	0.218	0.973	0.270	1.702	0.451	0.076
Traffic cone	0.403	0.909	0.377	-	-	-
Barrier	0.407	0.963	0.311	0.409	-	-
Overall	0.251	0.992	0.313	0.728	0.838	0.333

Table 2. Per-class metrics for the 2-Frame baseline model (Epoch 24). AP: Average Precision; ATE: Average Translation Error; ASE: Average Scale Error; AOE: Average Orientation Error; AVE: Average Velocity Error; AAE: Average Attribute Error. Lower values for error metrics indicate better performance. Some metrics are not applicable for certain object classes (shown as "-").

(0.305 to 0.298), indicating that improvements in AP were outweighed by degradations in other aspects like localization or velocity estimation. Further extending to 4 frames led to a decrease in both mAP (0.240) and NDS (0.282) compared to the 2-frame and 3-frame models. Based on these results, the 2-frame baseline model achieved the best overall detection quality as measured by NDS, while the 3-frame model provided the highest mAP. Adding a fourth frame appeared detrimental to overall performance.

Error Metrics Analysis: The degradation in NDS when moving from 2 to 3 frames, despite higher mAP, appears linked to increased mean Average Velocity Error (mAVE increased from 0.838 to 0.996) and mean Average Orientation Error (mAOE increased from 0.728 to 0.735), although mean Average Translation Error slightly improved (mATE decreased from 0.992 to 0.971). The further drop in performance for the 4-frame model was accompanied by increases in nearly all mean error metrics compared to the 2-frame baseline (mATE: 1.057, mASE: 0.329, mAOE: 0.751, mAVE: 1.028), indicating poorer localization, scale, orientation, and velocity estimation on average. The only improved metric was mean Average Attribute Error (mAAE: 0.298 vs 0.333 for 2-frame).

Class-Specific Effects: The negative impact of adding the fourth frame was evident across many classes. Comparing 4-frame to 3-frame results, AP decreased for most classes except Truck and Trailer (slight increases) and Barrier. Translation Error (ATE) increased for nearly all classes in the 4-frame model compared to the 3-frame. Velocity Error (AVE) also generally increased or remained high. Orientation Error (AOE) changes were mixed but notably increased for classes like Bus and Pedestrian.

Orientation Error: Bicycle orientation estimation remained challenging. AOE increased progressively from 1.702 (2-frame) to 1.841 (3-frame) and further to 2.110 (4-frame), highlighting the difficulty the model faces in handling this specific attribute for narrow objects as more temporal information is added via simple fusion.

Training Stability: The loss curves in Figure 3 suggest that the 2-frame model consistently achieved lower loss values, potentially indicating an easier optimization landscape. The higher loss values for the 3-frame and particularly the 4-frame models might suggest difficulties in effectively integrating information from the additional frames with the current architecture and fusion strategy.

Convergence Patterns: All models showed relatively similar convergence patterns for mAP and NDS (Figure 4), suggesting the performance differences likely stem more

	3-Frame Model						4-Frame Model					
Object Class	**AP**	**ATE**	**ASE**	**AOE**	**AVE**	**AAE**	**AP**	**ATE**	**ASE**	**AOE**	**AVE**	**AAE**
Car	0.415	0.724	0.160	0.234	1.827	0.285	0.378	0.835	0.158	0.243	1.860	0.277
Truck	0.231	1.024	0.281	0.585	1.261	0.320	0.232	0.974	0.298	0.371	1.175	0.303
Bus	0.195	1.280	0.287	0.293	1.670	0.296	0.183	1.188	0.319	0.344	2.074	0.258
Trailer	0.055	1.196	0.241	0.243	0.603	0.118	0.059	1.137	0.231	0.243	0.808	0.115
Construction vehicle	0.068	1.189	0.632	0.860	0.193	0.624	0.065	1.346	0.684	0.914	0.079	0.456
Pedestrian	0.279	0.900	0.300	1.344	0.968	0.276	0.269	0.940	0.327	1.444	0.962	0.292
Motorcycle	0.287	0.845	0.244	0.841	1.098	0.537	0.258	1.115	0.272	0.769	0.983	0.615
Bicycle	0.231	0.735	0.279	1.841	0.346	0.027	0.183	1.206	0.300	2.110	0.280	0.069
Traffic cone	0.438	0.852	0.412	-	-	-	0.362	0.922	0.389	-	-	-
Barrier	0.409	0.960	0.287	0.370	-	-	0.413	0.907	0.307	0.319	-	-
Overall	0.261	0.971	0.312	0.735	0.996	0.310	0.240	1.057	0.329	0.751	1.028	0.298

Table 3. Per-class metrics for the 3-Frame (Epoch 24) and 4-Frame models. Lower error values indicate better performance. The 3-frame model shows mixed results compared to the 2-frame baseline (higher mAP, lower NDS). The 4-frame model shows lower mAP and NDS compared to both 2-frame and 3-frame models, with generally higher error metrics.

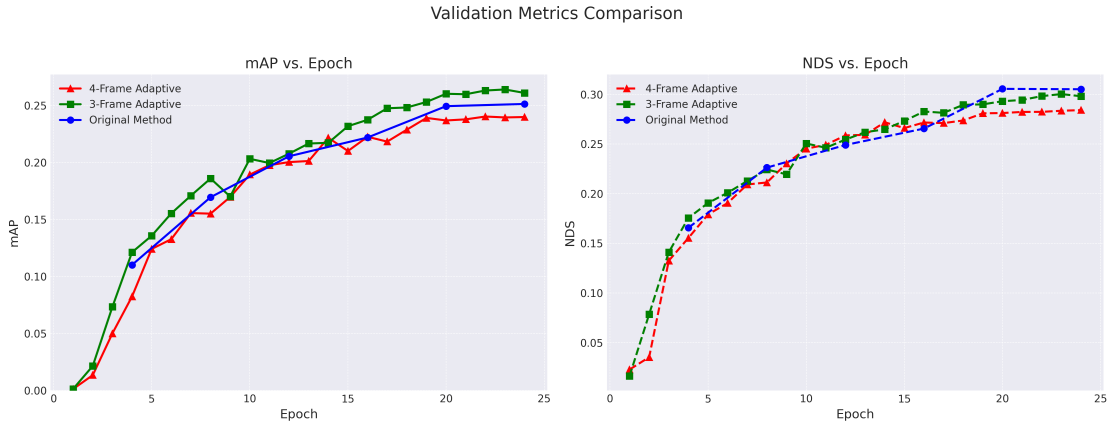

Figure 4. Evolution of mAP and NDS metrics during training for 3 frame (shown in green) and 4 frame (shown in red).

from the architectural limitations in handling extended temporal information rather than gross training instabilities. However, the lower peak performance achieved by the 4-frame model indicates it converged to a poorer solution.

4.5. Hypothesized Explanations

Several factors might explain why simply adding more frames, particularly extending to four, did not lead to performance improvements and even resulted in degradation:

- **Feature Misalignment:** Ego-motion compensation might not perfectly align features, especially over longer temporal windows (3 or 4 frames) or for objects with complex relative motion. Accumulated misalignments could significantly degrade localization accuracy (affecting ATE, AOE) and potentially confuse velocity estimation.
- **Suboptimal Fusion:** The simple concatenation or averaging mechanism likely struggles to effectively fuse features from multiple timestamps. Without adaptive weighting or attention, redundant or noisy information from less relevant past frames could dilute useful signals or even dominate, negatively impacting various error metrics (especially localization and velocity). The performance drop with 4 frames suggests this issue might become more severe with longer histories.
- **Motion Complexity and Stale Information:** The diverse and sometimes non-linear motion patterns in nuScenes might be poorly represented by simply adding older frames. Information from the earliest frame in a 4-frame sequence might be too outdated or represent motion that is no longer relevant, effectively adding noise. The increased mAVE in both 3-frame and 4-frame models points to difficulties in accurately modeling object dynamics.
- **Training Data Limitations:** The 15% subset might lack sufficient examples of complex temporal scenarios where extended history is clearly beneficial, making it harder for the model to learn robust multi-frame representations

and potentially leading it to overfit to simpler patterns or struggle with noise.

- **Model Capacity and Optimization:** While extended models have more parameters, they might require different optimization strategies, regularization, or even architectural adjustments to effectively learn temporal dependencies without being overwhelmed by noisy or misaligned inputs. The higher loss values hint at these optimization challenges.

4.6. Computational Considerations

- Training time increased by approximately 50% for the 3-frame model and 100% for the 4-frame model compared to the baseline.
- Memory usage scaled almost linearly with the number of frames, with the 4-frame model requiring approximately 1.8x the GPU memory of the 2-frame baseline.
- The inference time increased by 40% and 60% for the 3-frame and 4-frame models, respectively, which could impact real-time application feasibility.

These findings strongly underscore that simply extending the temporal horizon in this multi-view 3D perception model using basic feature fusion is not sufficient and can be detrimental. While adding a third frame offered a marginal gain in mAP at the cost of NDS, adding a fourth frame resulted in poorer performance across most metrics compared to the baseline. Achieving benefits from longer temporal context likely requires more sophisticated temporal fusion strategies (e.g., attention mechanisms, learned weighting, recurrent structures) capable of adaptively selecting relevant information and mitigating noise from older or misaligned features, balanced against the increased computational cost.

5. Conclusion

5.1. Summary of Findings

Our research into extending PETRv2's temporal modeling from two frames to three and four frames yielded nuanced results that partially contradicted our initial hypothesis. Extending to three frames provided a marginal increase in overall mean Average Precision (mAP rose from 0.251 to 0.261), but this was accompanied by a decrease in the holistic nuScenes Detection Score (NDS dropped from 0.305 to 0.298). Further extending to four frames resulted in a degradation of both metrics (mAP 0.240, NDS 0.282) compared to the 2-frame baseline and the 3-frame model. The 2-frame baseline achieved the highest NDS. Error metrics showed a complex pattern: mean Average Translation Error (mATE) slightly improved with three frames but worsened with four, while mean Average Orientation Error (mAOE) and mean Average Velocity Error (mAVE) generally increased as more frames were added. These findings suggest that simply adding more temporal frames using the existing fusion mechanism introduces challenges, particularly in accurately estimating orientation and velocity, which can outweigh benefits in classification accuracy and negatively impact overall detection quality.

5.2. Impact and Significance

This work contributes valuable empirical evidence highlighting the complexities of temporal fusion in 3D perception, challenging the simple intuition that "more temporal information is always better." Our analysis identifies specific metrics (mAOE, mAVE) and potentially classes most sensitive to the limitations of basic temporal fusion, providing insights for future model development. Additionally, our implementation framework for multi-frame temporal modeling in PETRv2 establishes a foundation for experimenting with alternative fusion approaches.

5.3. Limitations

Our study had three primary limitations: (1) the use of a 15% nuScenes subset may have limited the model's ability to learn robustly from complex temporal scenarios or rare events; (2) the simple feature fusion mechanism (concatenation/averaging) likely failed to effectively integrate information or filter noise from multiple frames; and (3) the higher loss values observed for multi-frame models suggest optimization difficulties that our training approach might not have fully overcome, potentially preventing the models from reaching their full potential.

5.4. Future Work

Future research should prioritize the development and evaluation of more sophisticated fusion mechanisms rather than simply extending the temporal horizon with basic techniques. Promising directions include: (1) attention-based temporal weighting to adaptively focus on relevant frames and features; (2) improved feature alignment techniques beyond ego-motion compensation to maintain spatial coherence; and (3) methods to explicitly model and filter redundant or potentially conflicting information from different timestamps. Our deferred objectives—cross-task feature sharing and uncertainty estimation—also warrant investigation, as they might offer complementary ways to improve robustness.

Our findings redirect research efforts in temporal modeling for 3D perception toward developing more intelligent integration strategies rather than naively extending temporal context with simple fusion. This insight, derived from the observed performance trade-offs, represents an important contribution to advancing multi-camera perception systems for autonomous driving.

References

[1] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 11621–11631, 2020. 2, 4

[2] MMDetection3D Contributors. Mmdetection3d: Open-mmlab next-generation platform for general 3d object detection. https://github.com/open-mmlab/mmdetection3d, 2020. 2

[3] Adam W. R. Harley, Alex Pokrovsky, Zhaoyuan Li, Yuning Zhou, and Raquel Urtasun. Strobe: Streaming object detection from lidar point clouds. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2023. 2

[4] Shreyas Jahagirdar, Ankit Garg, and Yi-Ting Chen. Context-based multi-sensor fusion for 3d object detection under adverse weather conditions. *arXiv preprint arXiv:2404.14780*, 2024.

[5] Zhiqi Li, Wenhai Wang, Hongyang Li, Enze Xie, Chonghao Sima, Tong Lu, Yu Qiao, and Jifeng Dai. Bevformer: Learning bird's-eye-view representation from multi-camera images. In *Eur. Conf. Comput. Vis.*, 2022. 2

[6] Yingfei Liu, Junjie Yan, Fan Jia, Shuailin Li, Qi Gao, Tiancai Wang, Xiangyu Zhang, and Jian Sun. Petrv2: A unified framework for 3d perception from multi-camera images. *arXiv preprint arXiv:2206.01256*, 2022. 1, 2

[7] Yingfei Liu, Junjie Yan, Fan Jia, Shuailin Li, Qi Gao, Tiancai Wang, Xiangyu Zhang, and Jian Sun. PETRv2: A unified framework for 3d perception from multi-camera images. In *Int. Conf. Comput. Vis.*, 2023. 2

[8] Megvii-research. Petr: Position embedding transformation for multi-view 3d object detection. https://github.com/megvii-research/PETR, 2022. 1, 2, 3

[9] Jonah Philion and Sanja Fidler. Lift-splat-shoot: Encoding geometric priors in grid-based camera depth estimation. In *Eur. Conf. Comput. Vis.*, pages 583–598, 2020. 1

[10] Christopher Reading, Ali Harakeh, Julia Chae, and Steven L Waslander. Categorical depth distribution network for monocular depth estimation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 1627–1636, 2021. 1

[11] Jingtao Wang, Hao Zhang, Zhongqiang Hou, Lulu Wang, Chunfeng Shen, and Liang Zheng. 3d-st: Self-supervised pretraining for spatial-temporal representation of point cloud sequences in autonomous driving. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2022.

[12] Yue Wang, Zian Lian, Lichen Wang, Xinyu Chen, Kemao Yang, Zirui Xu, Chao Liu, Cheng Qian, and Chao Zhou. Detr3d: 3d object detection from multi-view images via 3d-to-2d queries. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 10166–10175, 2022. 2

[13] Yue Wang, Xiaoqing Yuan, Boxi Shi, Xiang Li, Yanyun Zhang, and Jian Wu. Streampetr: Exploring object-centric temporal modeling for efficient multi-view 3d object detection. In *Int. Conf. Comput. Vis.*, 2023. 2