
DrivingRecon: Large 4D Gaussian Reconstruction Model For Autonomous Driving

Anonymous Author(s)

Affiliation

Address

email

1 Prune and Dilate Block

Below is the PyTorch pseudo-code for the Prune and Dilate Block (PD-Block) presented 1. The pseudo-code outlines the key steps of the PD-Block, including feature concatenation, region partitioning, center proposal, similarity computation, mask generation, and feature aggregation.

The Prune and Dilate Block (PD-Block) begins by computing a value feature and a range view feature from the input feature map. These features are reshaped to accommodate multiple attention heads. If folding is enabled (i.e., $\text{fold_w} > 1$ and $\text{fold_h} > 1$), the feature maps are partitioned into smaller regions to reduce computational overhead.

Next, the block proposes a set of center points evenly distributed in space and computes their corresponding features by averaging the nearest points. A pair-wise cosine similarity matrix between the region features and the center points is calculated and passed through a sigmoid activation after scaling and shifting. A mask is generated based on a threshold to retain significant similarities, ensuring that the most similar points to each center are preserved.

The features are then aggregated by combining the long-term and local features weighted by the mask. Depending on the configuration, the aggregated features can either be returned as center features or dispatched back to each point in the cluster. If regions were previously split, they are merged back into the full feature map. Finally, the output is reshaped to restore the multi-head configuration and projected to produce the final feature map.

2 Visualization

Reconstructions, Depth Maps, and Segmentation Maps. To demonstrate the effectiveness of our algorithm, we randomly selected several examples of scene reconstructions, depth predictions, and segmentation results, as illustrated in Figures 1 and 2. These images reveal that our model consistently achieves high-quality reconstructions across diverse environments, including urban and suburban settings, as well as varying lighting conditions such as day and night. Notably, our method accurately distinguishes between static and moving objects, underscoring its robustness and precision in complex scenes.

Additional Cases of Novel View Synthesis. Novel view synthesis is a fundamental capability in scene reconstruction, playing a crucial role in enhancing the generalization performance of downstream tasks. To further validate the effectiveness of our approach, we present additional examples of novel view renderings in Figures 3 and 4. The high quality of these synthesized views demonstrates the efficacy of our method in generating realistic and coherent scene perspectives from new viewpoints.

Algorithm 1 Prune and Dilate Block (PD-Block)

Require: Input feature map $x \in \mathbb{R}^{B \times C \times W \times H}$
Ensure: Output feature map $\text{out} \in \mathbb{R}^{B \times C' \times W \times H}$

- 1: Compute value features: $\text{value} \leftarrow \text{self.v}(x)$
- 2: Compute range view features: $x \leftarrow \text{self.f}(x)$
- 3: Rearrange features for multi-head processing:
- 4: $x \leftarrow \text{rearrange}(x, \text{"b (e c) w h"} \rightarrow \text{"(b e) c w h"}, e = \text{heads})$
- 5: $\text{value} \leftarrow \text{rearrange}(\text{value}, \text{"b (e c) w h"} \rightarrow \text{"(b e) c w h"}, e = \text{heads})$
- 6: **if** $\text{fold_w} > 1$ **and** $\text{fold_h} > 1$ **then**
- 7: Get current shape: $(b_0, c_0, w_0, h_0) \leftarrow x.\text{shape}$
- 8: Assert feature map is foldable:
- 9: **assert** $w_0 \bmod \text{fold_w} = 0$ **and** $h_0 \bmod \text{fold_h} = 0$
- 10: Fold feature maps:
- 11: $x \leftarrow \text{rearrange}(x, \text{"b c (f1 w) (f2 h)"} \rightarrow \text{"(b f1 f2) c w h"},$
- 12: $f1 = \text{fold_w}, f2 = \text{fold_h})$
- 13: $\text{value} \leftarrow \text{rearrange}(\text{value}, \text{"b c (f1 w) (f2 h)"} \rightarrow \text{"(b f1 f2) c w h"},$
- 14: $f1 = \text{fold_w}, f2 = \text{fold_h})$
- 15: **end if**
- 16: Propose centers: $\text{centers} \leftarrow \text{self.centers_proposal}(x)$
- 17: Compute center features:
- 18: $\text{value_centers} \leftarrow \text{rearrange}(\text{self.centers_proposal}(\text{value}),$
- 19: $\text{"b c w h"} \rightarrow \text{"b (w h) c"})$
- 20: Compute pair-wise cosine similarity:
- 21: $\text{sim} \leftarrow \sigma(\text{self.sim_beta} + \text{self.sim_alpha} \cdot \text{pairwise_cos_sim}(\text{value_centers}.\text{reshape}(b, c, -1).\text{permute}(0, 2, 1),$
- 22: $x.\text{reshape}(b, c, -1).\text{permute}(0, 2, 1)))$
- 23: $x.\text{reshape}(b, c, -1).\text{permute}(0, 2, 1)))$
- 24: Generate mask:
- 25: $(\text{sim_max}, \text{sim_max_idx}) \leftarrow \text{sim}.\text{max}(\text{dim} = 1, \text{keepdim} = \text{True})$
- 26: $\text{mask} \leftarrow \text{zeros_like}(\text{sim})$
- 27: $\text{mask}.\text{scatter_}(1, \text{sim_max_idx}, 1.)$
- 28: $\text{sim} \leftarrow \text{sim} \times \text{mask}$
- 29: Rearrange value for aggregation: $\text{value2} \leftarrow \text{rearrange}(\text{value}, \text{"b c w h"} \rightarrow \text{"b (w h) c"})$
- 30: Aggregate features:
- 31: $\text{out} \leftarrow \frac{(\text{value2}.\text{unsqueeze}(1) \times \text{sim}.\text{unsqueeze}(-1)).\text{sum}(\text{dim}=2) + \text{value_centers}.\text{sum}(\text{dim}=-1, \text{keepdim}=\text{True}) + 1.0}{\text{sim}.\text{sum}(\text{dim}=-1, \text{keepdim}=\text{True}) + 1.0}$
- 32: **if** $\text{self.return_center}$ **then**
- 33: Rearrange output to center format:
- 34: $\text{out} \leftarrow \text{rearrange}(\text{out}, \text{"b (w h) c"} \rightarrow \text{"b c w h"}, w = ww)$
- 35: **else**
- 36: Dispatch features to each point:
- 37: $\text{out} \leftarrow (\text{out}.\text{unsqueeze}(2) \times \text{sim}.\text{unsqueeze}(-1)).\text{sum}(\text{dim} = 1)$
- 38: $\text{out} \leftarrow \text{rearrange}(\text{out}, \text{"b (w h) c"} \rightarrow \text{"b c w h"}, w = w)$
- 39: **end if**
- 40: **if** $\text{fold_w} > 1$ **and** $\text{fold_h} > 1$ **then**
- 41: Merge folded regions back:
- 42: $\text{out} \leftarrow \text{rearrange}(\text{out}, \text{"(b f1 f2) c w h"} \rightarrow \text{"b c (f1 w) (f2 h)"},$
- 43: $f1 = \text{fold_w}, f2 = \text{fold_h})$
- 44: **end if**
- 45: Rearrange back to multi-head format:
- 46: $\text{out} \leftarrow \text{rearrange}(\text{out}, \text{"(b e) c w h"} \rightarrow \text{"b (e c) w h"}, e = \text{heads})$
- 47: Project output: $\text{out} \leftarrow \text{self.proj}(\text{out})$
- 48: **return** out



Figure 1: Reconstructed visualization: (a) ground truth, (b) Reconstructed rgb images, (c) Depth maps, (d) dynamic object reconstruction, and (e) static object reconstruction (**zoom in for the best view.**)

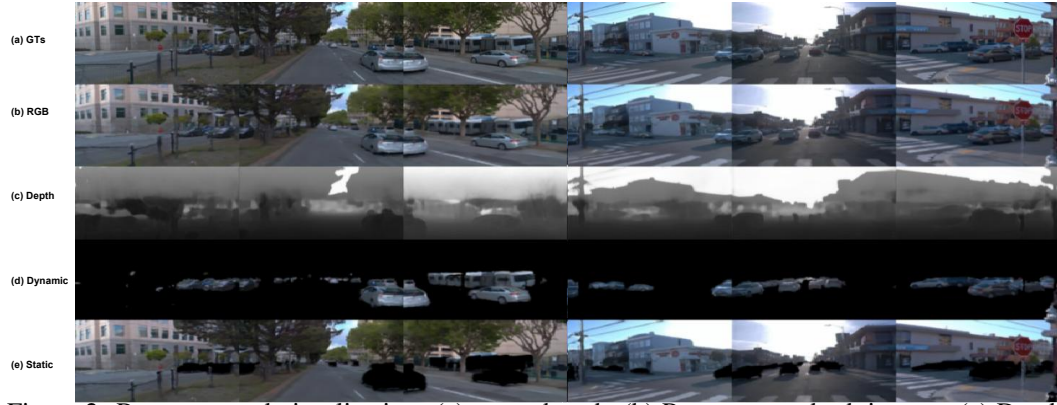


Figure 2: Reconstructed visualization: (a) ground truth, (b) Reconstructed rgb images, (c) Depth maps, (d) dynamic object reconstruction, and (e) static object reconstruction (**zoom in for the best view.**)

3 Segmentation

Segmentation supervision can help the network better understand the semantics of the scene and can also decompose static objects for cross-temporal view supervision. We utilize the DeepLabv3plus to produce three kinds of masks: dynamic objects (various vehicles and people), static objects, and the sky¹. Additionally, we project a 3D box onto a 2D plane as a prompt to use SAM to generate more accurate dynamic object masks. The masks of two dynamic objects are fused using "or" logic to ensure that all dynamic objects are masked.

4 Further Discussion

4.1 Social Impact

The development of DrivingRecon represents a meaningful advancement in the field of autonomous driving by addressing key challenges in large-scale scene reconstruction. By introducing a feed-forward spatial-temporal framework capable of handling both static and dynamic components in surround-view settings, our work enhances the robustness and reliability of perception systems in autonomous vehicles. This improvement directly contributes to safer and more predictable autonomous navigation, especially in complex urban environments where real-time scene understanding is critical. Additionally, the model's compatibility with pre-training and scene editing opens doors for applications beyond driving, such as simulation environments, digital twin creation, and augmented reality. By supporting more efficient and accurate digital reconstructions, DrivingRecon may accelerate progress across sectors like urban planning, robotics, and intelligent transportation systems. Moreover, our approach reduces reliance on costly and time-intensive iterative reconstruction methods,

¹<https://github.com/VainF/DeepLabV3Plus-Pytorch>

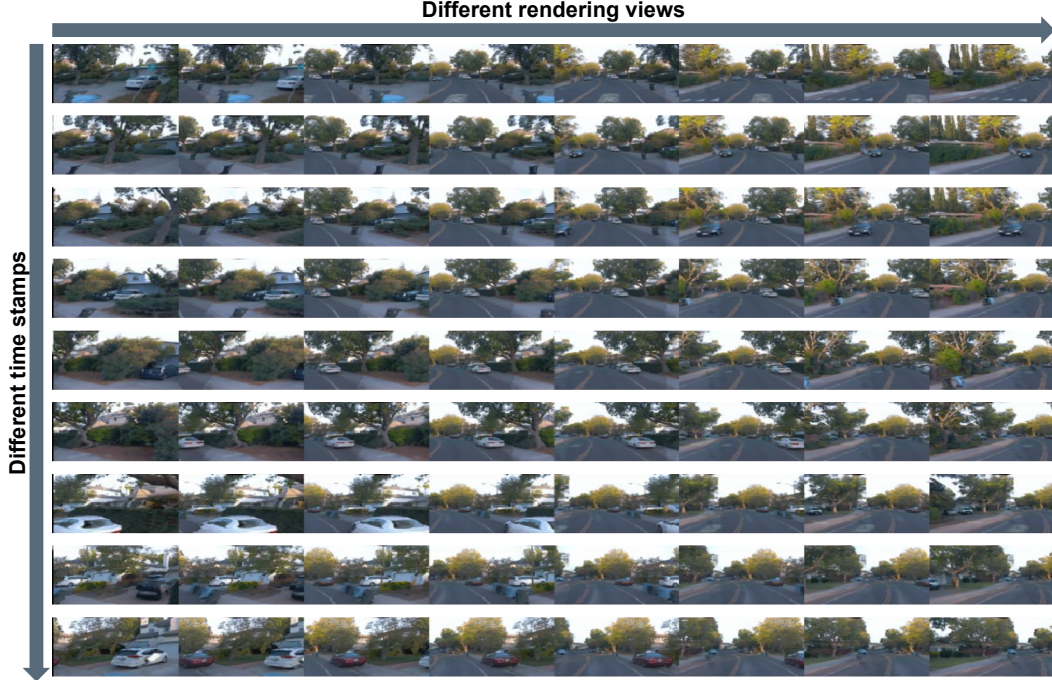


Figure 3: Novel view rendering. Based on the predicted Gaussians, we render different views at different times. The novel views are of very high quality and very high spatio-temporal consistency (zoom in for the best view.)

52 promoting greater accessibility and scalability for smaller research institutions and startups. This
 53 democratization of high-quality reconstruction technologies fosters innovation across a broader range
 54 of contributors.

55 4.2 Limitations

56 Despite its contributions, DrivingRecon has several limitations that warrant attention:

- 57 1. Gaussian Redundancy and Fixed Budgeting: Although the PD-Block helps manage overlapping
 58 Gaussian points, the feed-forward prediction of a fixed number of points per region can still limit
 59 expressiveness, particularly in highly complex or cluttered scenes.
- 60 2. Sparse View Supervision: Our method still relies on limited-view supervision at specific time steps.
 61 In scenarios where view diversity is insufficient or occlusion is prevalent, the quality of reconstructed
 62 geometry and texture may degrade.
- 63 3. Dynamic Object Generalization: While our static-dynamic decoupling improves rendering of
 64 dynamic objects, generalization to unseen moving patterns or novel object classes remains a challenge,
 65 particularly under domain shifts.
- 66 4. Computational Load: Though faster than iterative approaches, the full model inference, including
 67 depth estimation, feature extraction, and multi-view fusion, can be resource-intensive, potentially
 68 limiting real-time deployment on low-power edge devices.

69 —

70 4.3 Future Work

71 Several promising directions can be pursued to extend the capabilities of DrivingRecon:

- 72 1. Adaptive Gaussian Budgeting: Future iterations could implement adaptive point allocation based
 73 on scene complexity and object structure, enhancing flexibility and improving reconstruction fidelity
 74 without overwhelming memory resources.

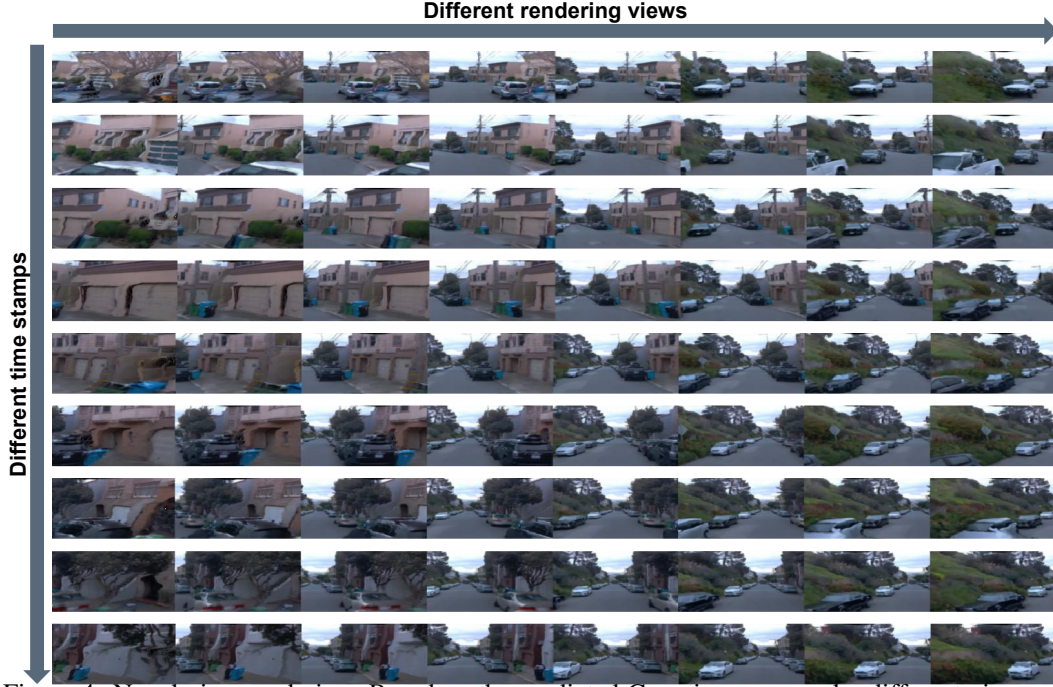


Figure 4: Novel view rendering. Based on the predicted Gaussians, we render different views at different times. The novel views are of very high quality and very high spatio-temporal consistency (zoom in for the best view.)

- 75 2. Real-Time Deployment Optimization: Through model compression, quantization, and neural
76 architecture search, DrivingRecon could be optimized for real-time execution on embedded systems
77 used in commercial autonomous vehicles.
- 78 3. Broader Scene Understanding: Integrating semantic segmentation and object-level reasoning into
79 the pipeline could enable richer scene interpretation, making the model more useful for planning and
80 decision-making tasks.
- 81 By addressing these limitations and expanding on its current strengths, DrivingRecon can further
82 evolve into a foundational system for next-generation spatial-temporal understanding in intelligent
83 systems.