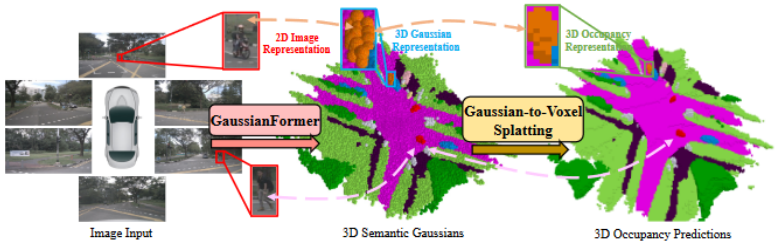
CVPR 2025

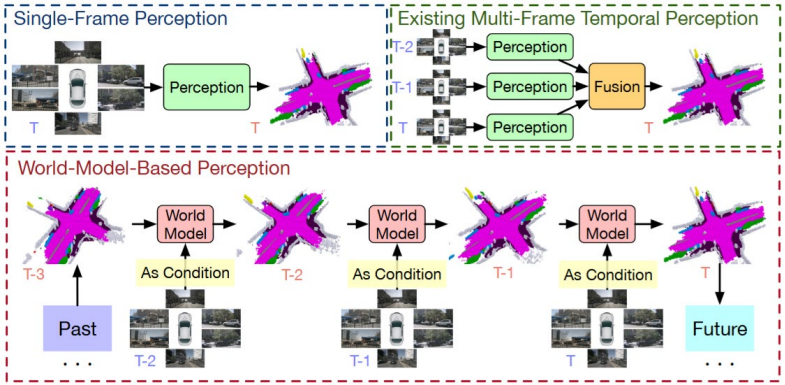
GaussianWorld: Gaussian World Model for Streaming 3D Occupancy Prediction

Sicheng Zuo* Wenzhao Zheng*,† Yuanhui Huang Jie Zhou Jiwen Lu
Department of Automation, Tsinghua University, China
zsc23@mails.tsinghua.edu.cn; wenzhao.zheng@outlook.com

- Problem/Objective
  - 3D occ + GS **+ Temporal**
- Contribution/Key Idea
  - 4D occupancy prediction을 재정의
    - by GaussianWorld 모델
  - Scene evolution 이라는 표현
    - Additional computational 없이 성능 향상



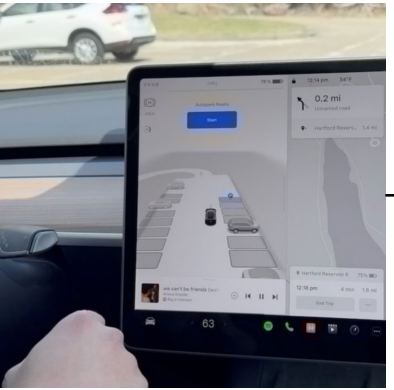
GaussianFormer (ECCV 2024)

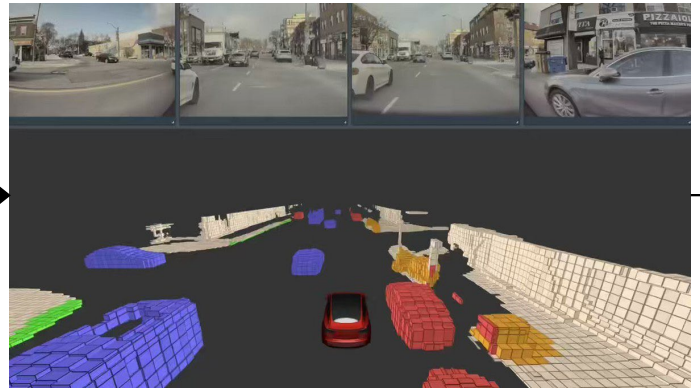


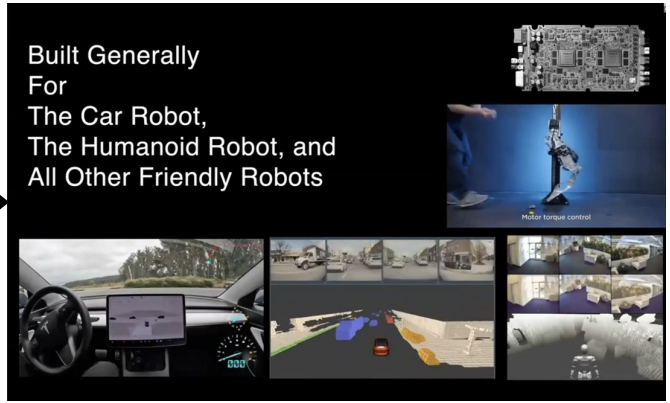
GaussianWorld (CVPR 2025)

김범준

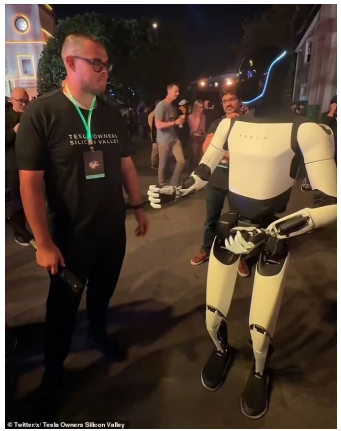
- **3D occupancy prediction이란?**





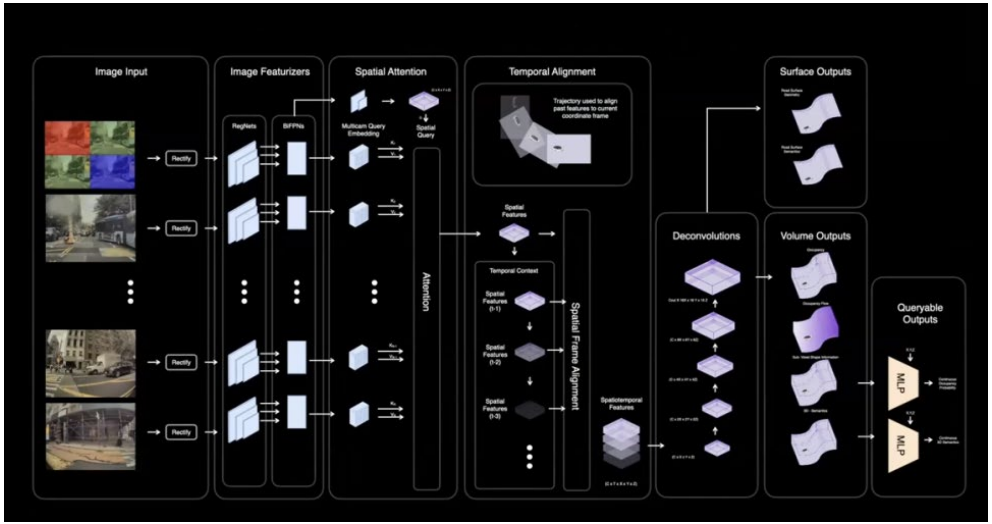


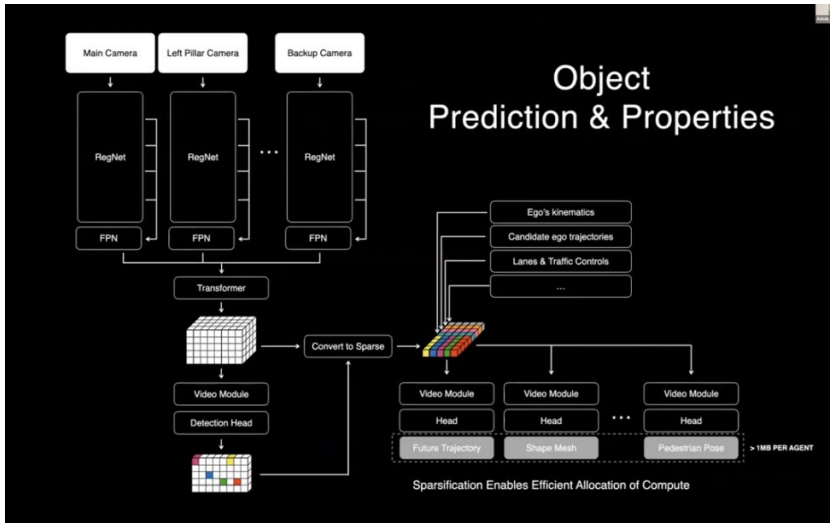
- Tesla가 2022년 CVPR에서 공개한 내용
- 이후 CVPR challenge 등 큰 관심
- 2023, 2024년 CVPR에서는 tesla 휴머노이드 로봇 옵티머스에도 같은 방법이 적용중이라고 밝힘



- Voxel 단위 3차원 BEV segmentation
  - 공간에 대한 점유 여부 + 점유한 공간이 어떤 class인지 분류하는 task

- **OCC by tesla**





[1] 유튜브 링크 - [CVPR'23 WAD] Keynote - Ashok Elluswamy, Tesla
[2] 한글 번역 - [CVPR'23 WAD] Keynote - Ashok Elluswamy, Tesla

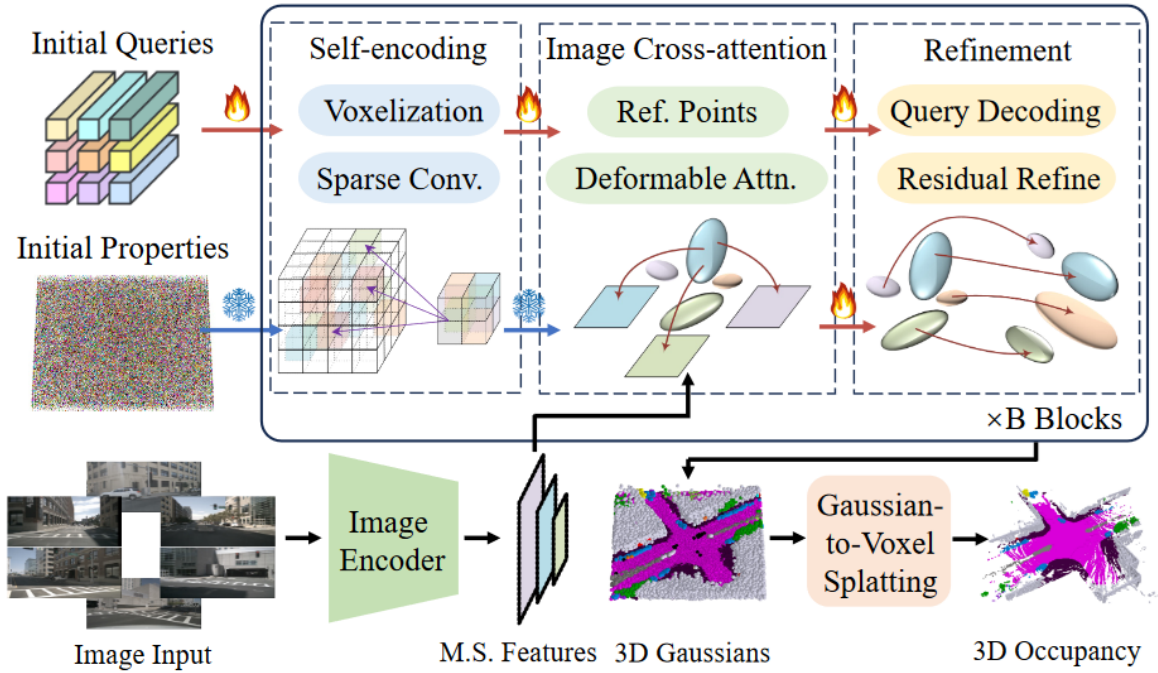
- **(Recap) GaussianFormer**

Query: high-dimensional vector

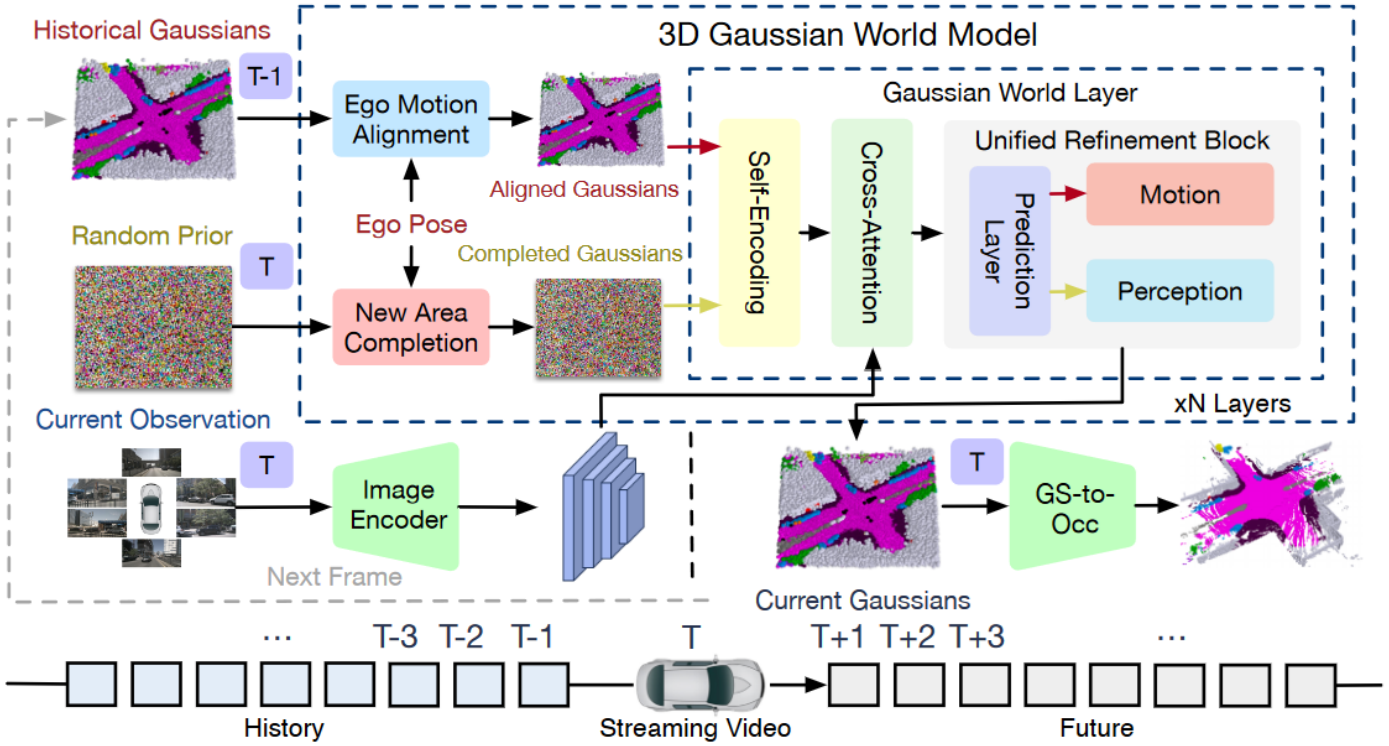
Property: (mean, scale, rotation, semantic)

Random

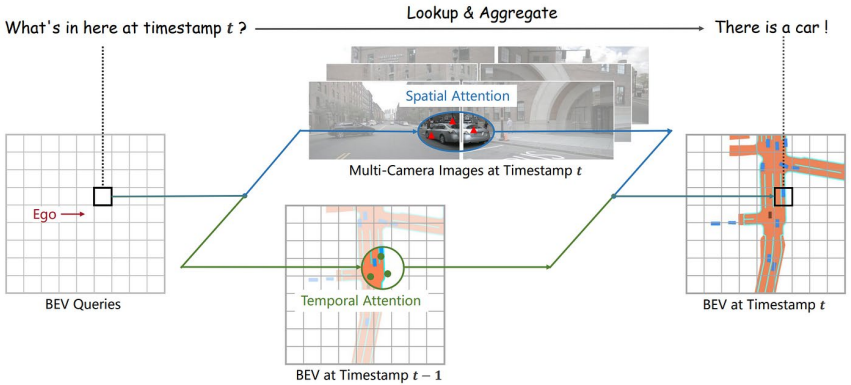
Random

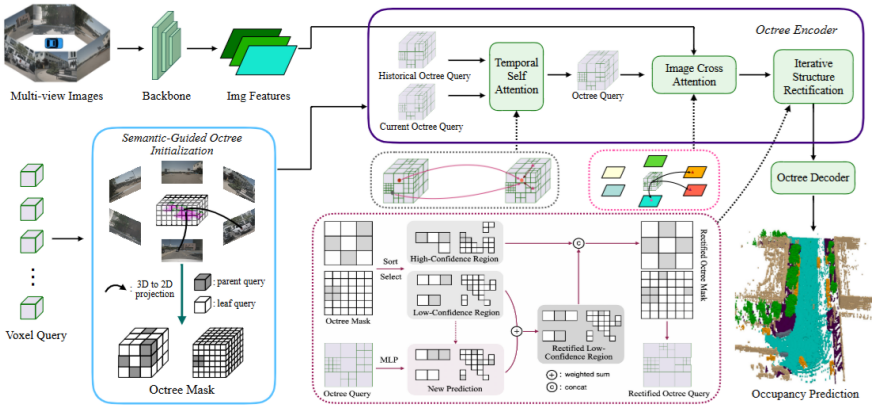


- **GaussianWorld**



Introduction

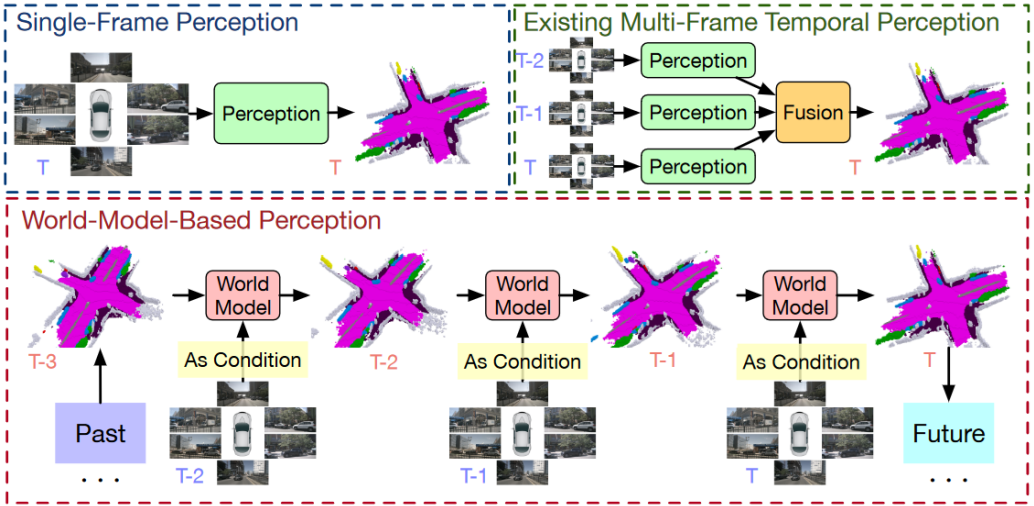




- 기존의 Temporal fusion 방식들
  - Representation들 간의 fusion (ex. BEV feature, voxel feature, cost-volume feature 등등)
  - 단점1) → driving scenarios에 내재된 continuity를 전혀 고려x
  - 단점2) → ignores **strong prior**
  - 단점3) → complexity & computational cost ⬆
- 본 논문
  - World 모델 기반 패러다임
    - explore scene evolution

GaussianWorld: Gaussian World Model for Streaming 3D Occupancy Prediction

- **Method**



prediction sensor input ego position

$$\mathbf{y}^T = A(\{\mathbf{x}^T, ..., \mathbf{x}^{T-t}\}, \{\mathbf{p}^T, ..., \mathbf{p}^{T-t}\}),$$

representation

$$\mathbf{z}^n = P_{cr}(\mathbf{x}^n), \mathbf{a}^n = T_{rans}(\mathbf{z}^n, \mathbf{p}^n),$$
$$\mathbf{y}^T = F_{use}(\mathbf{a}^T, ..., \mathbf{a}^{T-t}),$$

1) perception
2) transformation
3) fusion

$$\mathbf{z}^T = \mathbf{w}(\mathbf{z}^{T-1}, \mathbf{x}^T).$$

$$\mathbf{y}^T = \mathbf{A}(\mathbf{z}^{T-1}, \mathbf{x}^T) = \mathbf{h}(\mathbf{w}(\mathbf{z}^{T-1}, \mathbf{x}^T)),$$

<Perception World Model>
Prior representation + Current Sensor input(condition)
Representation Head

$$\mathbf{g}^T = \mathbf{w}(\mathbf{g}^{T-1}, \mathbf{x}^T).$$

<Gaussian World model>
Prior Gaussian + Current sensor input(condition)

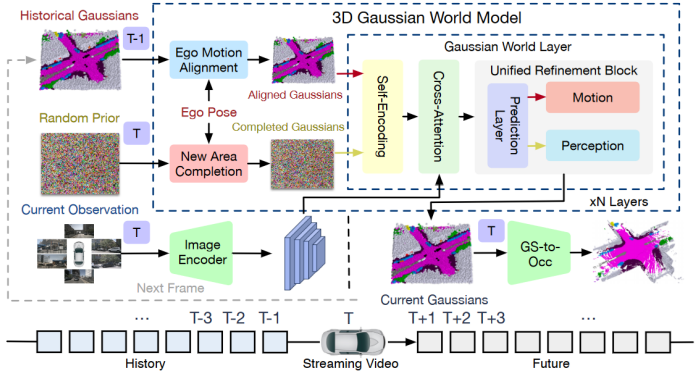
GaussianWorld: Gaussian World Model for Streaming 3D Occupancy Prediction

CVPR 2025

- **Method**

$$\mathbf{g} = \{\mathbf{p}, \mathbf{s}, \mathbf{r}, \mathbf{c}, \mathbf{f}\},$$

position
scale
rotation
semantic probability
temporal feature



1) Ego Motion Alignment of Static Scenes

Gaussian Ego Pose

$$\begin{aligned}\mathbf{g}_A^T &= A_{lign}(\mathbf{g}^{T-1}, \mathbf{M}_{ego}), \\ &= R_{ef}(\mathbf{g}^{T-1}; \mathbf{M}_{ego} \cdot A_{ttr}(\mathbf{g}^{T-1}; \mathbf{p}); \mathbf{p}),\end{aligned}$$

Rotation/Translation으로 global하게 align

2) Local Movements of Dynamic Objects

Aligned Gaussian current observation (2D feature)

$$\begin{aligned}\mathbf{g}_M^T &= M_{ove}(\mathbf{g}_A^T, \mathbf{x}_T), \\ &= R_{ef}(\mathbf{g}_A^T; E_{nc}(\mathbf{g}_A^T, \mathbf{x}_T) \cdot I(\mathbf{g}_A^T \in \{\mathbf{g}_D\}); \mathbf{p}),\end{aligned}$$

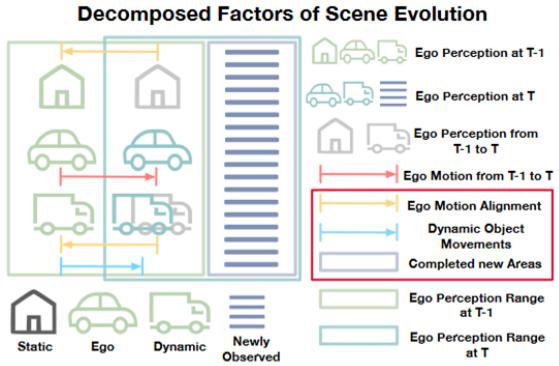
gaussian의 semantic probability로 이동 객체 분리

3) Completion of Newly-Observed Areas

새로 추가된 gaussian

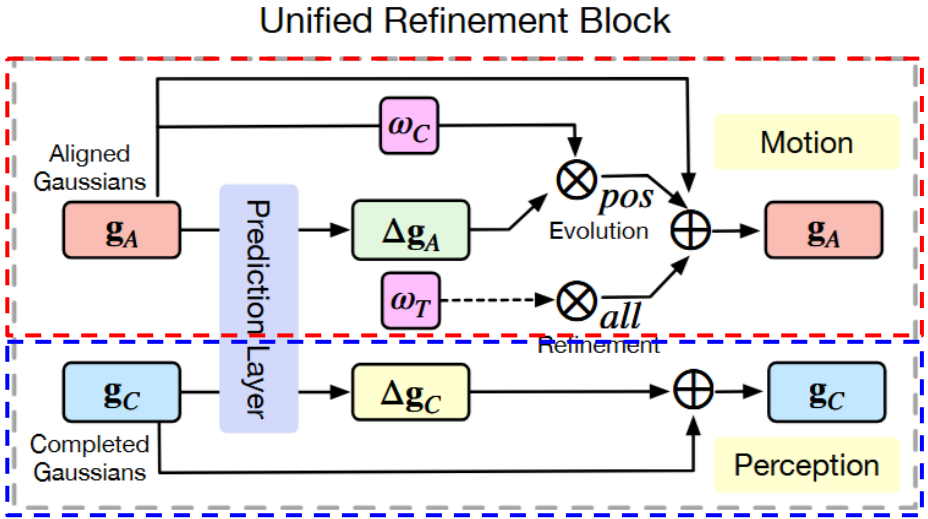
$$\begin{aligned}\mathbf{g}_C^T &= P_{er}(\mathbf{g}_I^T, \mathbf{x}_T), \\ &= R_{ef}(\mathbf{g}_I^T; E_{nc}(\mathbf{g}_I^T, \mathbf{x}_T); \{\mathbf{p}, \mathbf{s}, \mathbf{r}, \mathbf{c}, \mathbf{f}\}),\end{aligned}$$

Align 하면서 시야에서 사라진 gaussian 개수만큼 추가

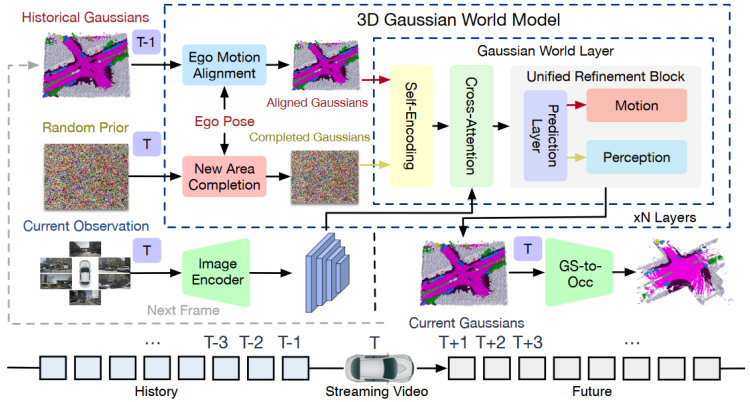


김범준

- **Method**



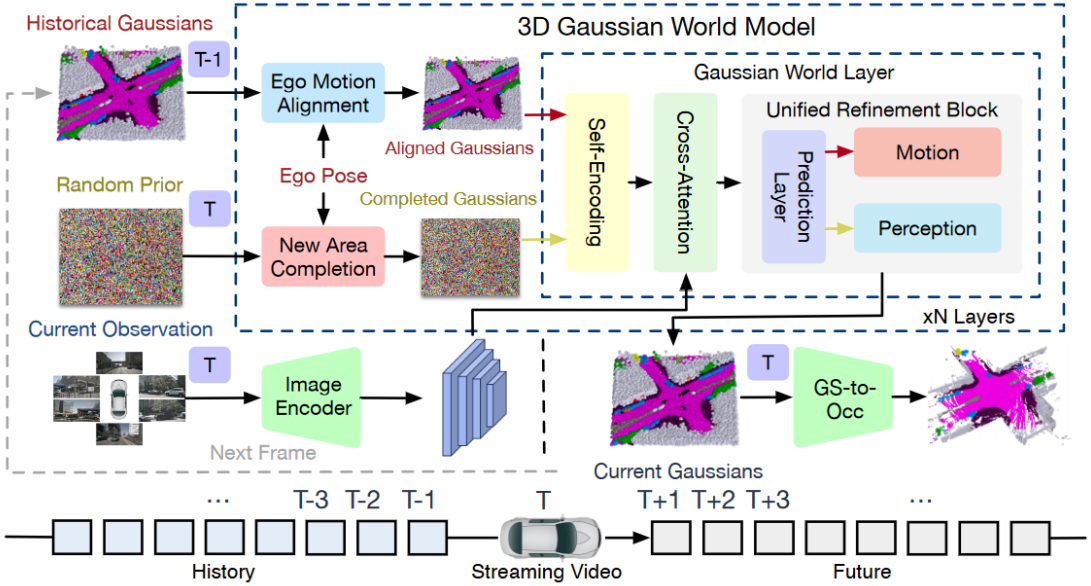
$$\begin{aligned}\mathbf{g}_{n+1}^{T} &= R_{efine}(\mathbf{g}_{n}^{T}, \mathbf{x}_{T}), \\ &= R_{ef}(\mathbf{g}_{n}^{T}; E_{nc}(\mathbf{g}_{n}^{T}, \mathbf{x}_{T}); \{\mathbf{p}, \mathbf{s}, \mathbf{r}, \mathbf{c}, \mathbf{f}\}),\end{aligned}$$



$$\mathbf{g}_{l+1}^{T} = E_{vol}(\mathbf{g}_{l}^{T}, \mathbf{x}_{T}) = \begin{cases} P_{er}(\mathbf{g}_{l}^{T}, \mathbf{x}_{T}), & \text{if new,} \\ M_{ove}(\mathbf{g}_{l}^{T}, \mathbf{x}_{T}), & \text{otherwise,} \end{cases}$$

It is worth noting that the two layers share the same model architecture and parameters, namely the encoder module E_{nc} and the refinement module R_{ef}, allowing them to be integrated into a unified evolution layer E_{vol} and computed in parallel. This design ensures that GaussianWorld maintains model simplicity and computational efficiency.

- **GaussianWorld**



- 학습은 안정성을 위해 single frame task로 학습하고 그 weight를 이용해서 → sequential 하게 finetun
- Sequential 학습도 안정성을 고려해서 short sequences → Long sequences
  - 그리고 prior에 대한 의존성을 줄이기 위해 probabilistic modeling을 통해 확률 *P* random하게 이전 frame을 drop
    - 이 확률 *P* 도 진행될수록 값을 줄여 long sequence를 연속적으로 예측하게끔 학습

- **Experiments**

Table 1. **3D semantic occupancy prediction results on nuScenes validation set.** The original TPVFormer [14] is trained with sparse LiDAR segmentation labels, and TPVFormer* is supervised by dense occupancy labels. **GaussianFormer-B**, **GaussianFormer-T** denotes the single-frame and temporal fusion variant of GaussianFormer [15].

Method	IoU	mIoU	barrier	bicycle	bus	car	const. veh.	motorcycle	pedestrian	traffic cone	trailer	truck	drive. suf.	other flat	sidewalk	terrain	manmade	vegetation
MonoScene [4]	23.96	7.31	4.03	0.35	8.00	8.04	2.90	0.28	1.16	0.67	4.01	4.35	27.72	5.20	15.13	11.29	9.03	14.86
Atlas [30]	28.66	15.00	10.64	5.68	19.66	24.94	8.90	8.84	6.47	3.28	10.42	16.21	34.86	15.46	21.89	20.95	11.21	20.54
BEVFormer [22]	30.50	16.75	14.22	6.58	23.46	28.28	8.66	10.77	6.64	4.05	11.20	17.78	37.28	18.00	22.88	22.17	**13.80**	**22.21**
TPVFormer [14]	11.51	11.66	16.14	7.17	22.63	17.13	8.83	11.39	10.46	8.23	9.43	17.02	8.07	13.64	13.85	10.34	4.90	7.37
TPVFormer* [14]	30.86	17.10	15.96	5.31	23.86	27.32	9.79	8.74	7.09	5.20	10.97	19.22	38.87	21.25	24.26	23.15	11.73	20.81
OccFormer [49]	**31.39**	19.03	18.65	10.41	23.92	30.29	10.31	14.19	**13.59**	10.13	12.49	20.77	38.78	19.79	24.19	22.21	13.48	21.35
GaussianFormer [15]	29.83	19.10	19.52	11.26	26.11	29.78	10.47	13.83	12.58	8.67	12.74	21.57	**39.63**	23.28	24.46	22.99	9.59	19.12
SurroundOcc [42]	**31.49**	**20.30**	**20.59**	11.68	**28.06**	**30.86**	**10.70**	15.14	**14.09**	**12.06**	**14.38**	**22.26**	37.29	**23.70**	24.49	22.77	**14.89**	**21.86**
GaussianFormer-B	30.68	19.73	19.36	**13.19**	**26.90**	29.79	10.20	**15.17**	12.55	9.29	**12.96**	21.45	39.55	23.03	**25.07**	**23.65**	12.35	21.18
GaussianFormer-T	31.34	**20.42**	**20.82**	**12.07**	26.89	**30.94**	**10.52**	**16.48**	13.15	**10.46**	12.90	**21.79**	**41.13**	**24.22**	**26.29**	**24.89**	12.80	21.45
GaussianWorld (ours)	**33.40**	**22.13**	**21.38**	**14.12**	**27.71**	**31.84**	**13.66**	**17.43**	**13.66**	**11.46**	**15.09**	**23.94**	**42.98**	**24.86**	**28.84**	**26.74**	**15.69**	**24.74**

batch size 16으로 16개의 4090에서 15h 소요

- **Experiments**

Table 2. **Comparisons of different temporal modeling methods.** The latency and memory consumption for all methods are tested on one NVIDIA 4090 GPU with a batch size of 1 during inference. 3D Gaussian Fusion and Perspective View Fusion denote the temporal fusion in the 3D Gaussian space and the perspective view space, respectively.

Temporal Modeling	Number of Historical Input	Latency	Memory	mIoU	IoU
Single-Frame	0	**225** ms	**6958** M	19.73	30.68
3D Gaussian Fusion	3	379 ms	9993 M	20.24	32.27
Perspective View Fusion	3	382 ms	10019 M	20.42	31.34
GaussianWorld (ours)	**1**	**228** ms	**7030** M	**21.87**	**33.02**

- **Experiments**

Table 3. **Ablation on the decomposed factors of GaussianWorld.** Neglecting the movements of ego-vehicle or other dynamic objects leads to a performance decline. × denotes training collapse without scene completion of newly observed areas.

Ego	Dynamics	Completion	mIoU	IoU
	✓	✓	18.47	28.88
✓		✓	21.17	32.49
✓	✓		×	×
✓	✓	✓	**21.50**	**32.72**

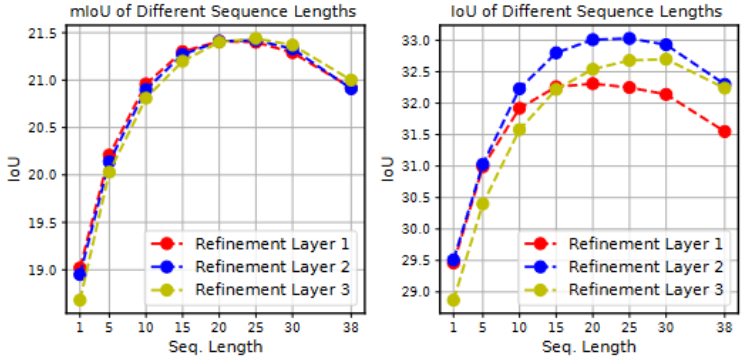


Figure 5. **Performance of streaming occupancy prediction with different sequence lengths.** We also show the performance of using different numbers of refinement layers.

- **Experiments**

Table 4. **Ablation on the schedules of streaming training.** Seq. Lengths and Epochs denote the increasing sequence length and the corresponding number of epochs during training. Sum and Mean represent gradient accumulation by summation and averaging, and None indicates no gradient accumulation. Prob. denotes whether to use probabilistic modeling.

Index	Seq. Lengths	Epochs	Grad. Acc.	Prob.	Training Time	mIoU	IoU
A	[5, 10, 20, 30, 38]	[80, 40, 20, 20, 20]	Sum	N	30 h	19.77	30.90
B	[5, 10, 20, 30, 38]	[80, 40, 20, 20, 20]	Mean	N	30 h	19.81	30.93
C	[5, 10, 20, 30, 38]	[40, 20, 10, 5, 20]	None	N	20 h	18.63	28.69
D	[5, 10, 20, 30, 38]	[0, 0, 0, 0, 30]	None	N	**15** h	19.82	30.96
E	[5, 10, 20, 30, 38]	[5, 5, 5, 5, 20]	None	Y	**15** h	**20.24**	**31.22**

- **Experiments**

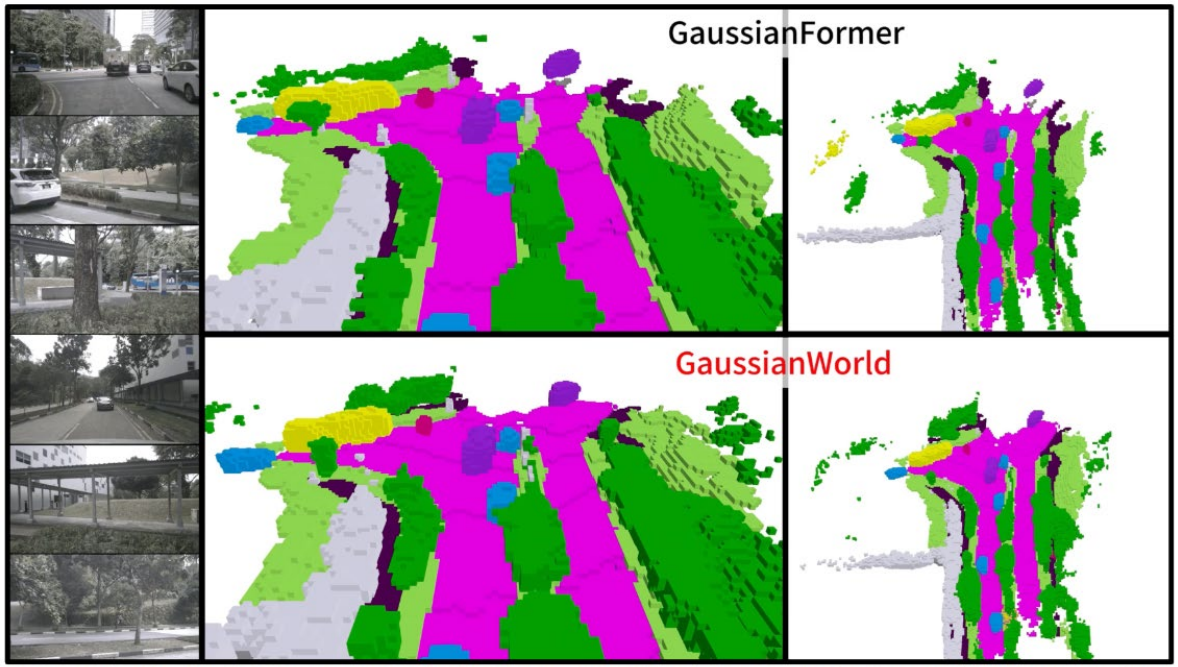


Figure 7. **Visualizations of the proposed GaussianWorld compared to GaussianFormer [15] for 3D semantic occupancy prediction on the nuScenes [3] validation set.** We visualize the six surrounding camera inputs and the corresponding occupancy prediction results. The upper row shows the predicted occupancy by GaussianFormer in the global view(left) and the bird's eye view(right). The lower row shows the prediction results of GaussianWorld.