The 1st-Place Solution for CVPR 2024 Autonomous Grand Challenge Track on Predictive World Model

Jun Yu[1], Shuoping Yang[1], Yinan Shi[2], Zhihao Yan[1], Zhao Yang[3], Ruiyu Liu[1], Fengzhao Sun[1], Yunxiang Zhang[1]
[1]University of Science and Technology of China, [2]Technische Universität München, [3]Xian Jiaotong University
harryjun@ustc.edu.cn, yangsp@mail.ustc.edu.cn, syn1211@outlook.com,
{yanzhihao0822, lingxi.yz950701}@gmail.com, {liury, sunfz, mesa}@mail.ustc.edu.cn

Abstract

*This paper describes our team **USTC_IAT_United**'s 1st-place solution for CVPR 2024 Autonomous Grand Challenge Track on Predictive World Model. The objective of this challenge is to introduce a world model which can predict future states based on the current state. To achieve this, we first utilize high-quality autonomous driving datasets with multiple camera views for self-supervised training. Next, we improve the competition baseline to predict future point clouds. Specifically, we use a pre-trained BEV encoder as a feature extractor and enhance the temporal alignment module within the BEV encoder. We then utilize a Latent Rendering operator to extract more distinctive and representative features and improve the attention mechanism within the Transformer Decoder. Finally, we output the predicted future point clouds. Our ViDAR++ achieves a CD@overall (Chamfer Distance) of 0.6615 on the the OpenScene Private-Test set.*

1. Introduction

Autonomous driving applications [11, 17] require integrated perception, prediction, and planning, which involve features of semantics, 3D geometry, and temporal information. However, traditional pre-training methods face significant challenges because they rely on costly manual annotations (such as semantic class labels, bounding boxes, and trajectories) or require high-precision city HD maps, limiting their scalability on large-scale unlabelled datasets. To address these issues, researchers have proposed a new pre-training task: visual point clouds forecasting [16], which aims to predict future point clouds from historical visual inputs. This is essential for planning and decision-making in autonomous driving systems. Visual point clouds forecasting offers two main advantages: (1) Collaborative Learning [6]: The task requires the model to learn semantics, 3D structure, and temporal dynamics simultaneously. This collaborative learning enables the model to perform better on various downstream tasks. (2) Self-Supervised Training [1]: Visual point clouds forecasting does not require expensive annotated data but instead uses self-supervised training on unlabelled LiDAR [9] sequences, making it more scalable.

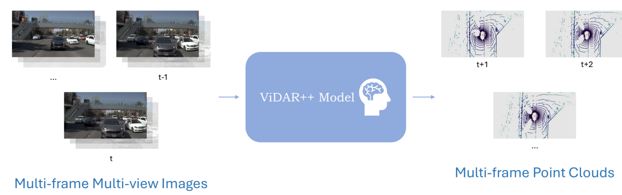

Figure 1. The overall pipeline of point prediction, which builds on ViDAR [16] and outputs point clouds.

The purpose of this challenge is to use world model to predict future frames. Serving as an abstract spatio-temporal representation of reality, the world model can predict future states based on the current state. The learning process of world models has the potential to provide a pre-trained foundation model for autonomous driving. Given vision-only inputs, the neural network outputs point clouds in the future to testify its predictive capability of the world. As shown in Fig. 1, given a visual observation of the world for the past 3 seconds, predict the point clouds in the future 3 seconds based on the designated future ego-vehicle pose.

In this work, we present a multi-stage framework consisting of two stages: *Self-Supervised Training* and *Point Clouds Prediction*. Our contributions are outlined below: ***(1) Multi-View Self-Supervised Training.*** We believe that for visual BEV extractors, contrastive learning methods can initially capture potential features from multi-view images. To enhance BEV feature extraction, we propose a multi-view self-supervised training method. ***(2) Temporal Attention.*** We improve the traditional temporal cross-attention mechanism by aligning and fusing BEV feature maps at two different times based on spatial positions.

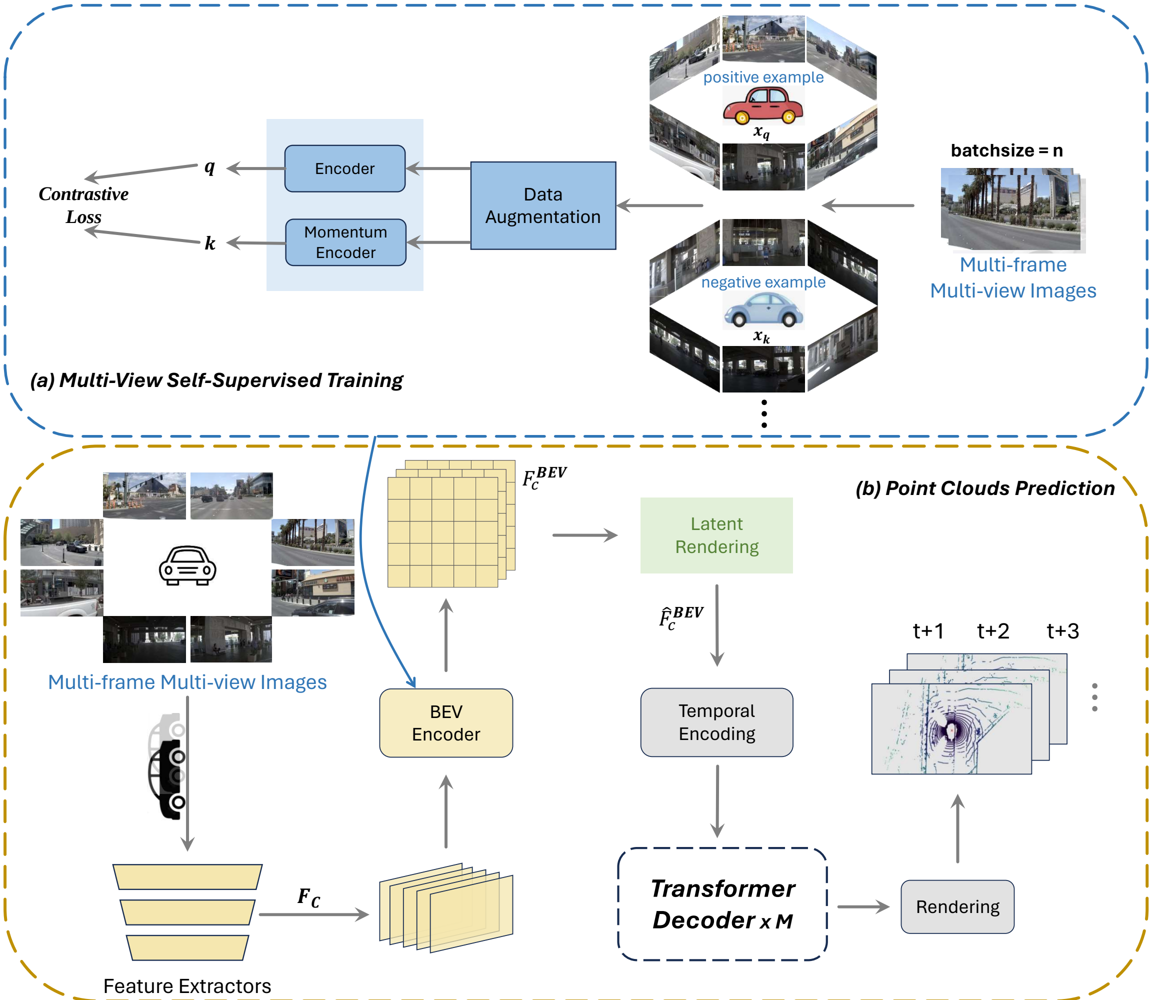

Figure 2. The architecture of our proposed method. We present a two-stage framework. **(a) Multi-View Self-Supervised Training**: We use contrastive learning to train multi-view camera images; **(b) Point Clouds Prediction**: We input the multi-view image to be trained, we read the BEV Encoder pre-trained in the previous stage, and output the final point clouds prediction through each subsequent module.

2. Method

This section introduces the details of our 1st-place method. The overall architecture of our approach is shown in Fig. 2. First, we introduce the self-supervised training process, detailing our training methods and the improvements made to various modules within the encoder. These enhancements help the model better extract BEV features. Next, we describe the point clouds prediction process, focusing on the improvements made to the baseline model ViDAR [16] in its various modules. Our approach involves improvements based on the baseline method, named as ***ViDAR++***. The more technical details are presented in this section.

2.1. Self-Supervised Training

Elaboration. To better extract BEV features, we propose a multi-view self-supervised training method for this competition. Similar to MoCo [7], we believe that contrastive learning can initially capture the latent features of multi-view images. This approach makes the network more stable and converges more rapidly compared to networks initialized randomly.

Our pre-training framework is illustrated in the diagram shown in Fig. 2 (a). Firstly, we apply data augmentation techniques including RandomResizedCrop, ColorJitter, RandomGrayscale, and GaussianBlur [13] to the input unlabeled images. Additionally, both our training encoder and momentum encoder utilize our proposed BEV encoder. We set the training batch size as a multiple of 8. In contrastive learning, the positive samples are images taken from different perspectives by eight cameras at the same frame, while the negative samples are the remaining samples in the batch, i.e., multi-view images from different frames. The positive and negative samples undergo two different data augmentations to obtain x_q and x_k, respectively. Then, x_q and x_k are fed into the encoder and momentum encoder to obtain features q and k. Subsequently, the similarity between the features of the positive and negative samples is calculated. Finally, the network is updated based on these similarities. Throughout the training process, we use Noise Contrastive Estimation (NCE) [5] loss to ensure the maximization of output similarity between each image and its augmented version.

BEV Encoder. Similar to BEVFormer [10], our BEV encoder consists of six layers, each following the traditional Transformer structure but with three custom designs: BEV queries, spatial cross-attention, and temporal self-attention. Specifically, BEV queries are grid-like learnable parameters aimed at querying features in BEV space from multi-camera views through an attention mechanism. Spatial cross-attention and temporal self-attention are attention layers used in conjunction with BEV queries to locate and aggregate spatial features from multi-camera images and temporal features from historical BEVs, respectively. We mainly improve the temporal attention module to achieve BEV temporal alignment and fusion.

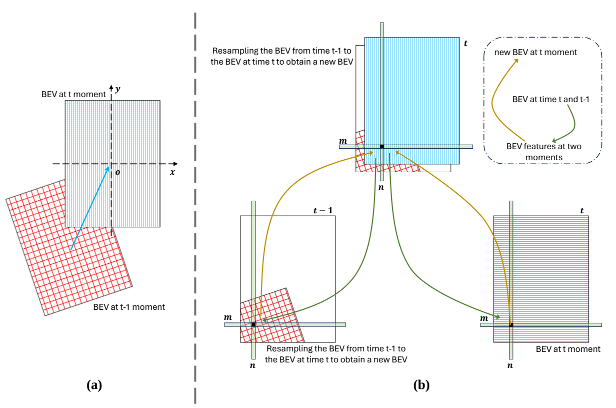

Figure 3. Our improvement plan for the temporal block.

Temporal Attention Module. Fig. 3 (a) shows BEV maps at time $t-1$ and time t, where the two BEV maps differ in angle and have spatial displacement. Fig. 3 (b) illustrates the principle of temporal alignment. Specifically, the BEV map at time $t-1$ is resampled at the spatial positions of the BEV map at time t to obtain a new BEV map at time $t-1$, corresponding to the spatial positions at time t. Then, the new BEV map at time $t-1$ and the BEV map at time t are concatenated along the channel dimension to obtain a concatenated BEV. This concatenated BEV has completed spatiotemporal alignment. Next, the channel vector value at each spatial position is updated to get a new fused BEV map at time t. For example, at spatial position (n, m), to find the channel vector value at (n, m) in the new BEV map at time t, first, the channel vector value at (n, m) is obtained from the concatenated BEV map. Then, k relative feature index positions and weights are calculated. In the second step, the required feature 1 is obtained directly at (n, m) in the BEV feature map at time t based on the relative feature index positions and weights. In the third step, the required feature 2 is obtained at (n, m) in the new BEV feature map at time $t-1$ based on the relative feature index positions and weights. Finally, the two features are averaged to get the channel vector value at (n, m) in the new BEV map at time t. This completes the BEV temporal alignment and fusion.

2.2. Point Clouds Prediction

The Point Clouds Prediction stage is an end-to-end training process. After completing self-supervised training, we utilize the pre-trained weights to initialize the BEV Encoder. Next, we proceed with supervised end-to-end training using labeled data. Our method is based on ViDAR [16], with improvements made to the Future Decoder. These improvements enhance the temporal coherence of the predicted point clouds. The overall architecture comprises three components: **(a)** a *BEV Encoder*, which serves as the target structure for pre-training, extracts BEV embeddings F_{bev} from visual sequence inputs $\mathcal{I}$. The encoder we use refers to the structure of BEVFormer [10]; **(b)** a *Latent Rendering* operator, this component has the same structure as in ViDAR, which simulates the volume rendering process in latent space to derive the geometric embedding $\hat{F}_{bev}$ from F_{bev}; **(c)** a *Transformer Decoder*, which predicts future BEV features $\hat{F}_t$ at timestamps $t \in \{1, 2, ...\}$ in an auto-regressive manner. Subsequently, a prediction head is utilized to project $\hat{F}_t$ into a 3D occupancy volume P_t, ensuring accurate spatial representation of the environment. In the Temporal Cross-Attention within the Transformer Encoder, we also apply previous improvements to the temporal module to better integrate temporal relationships.

3. Experiment

In this section, we first provide a brief introduction to the relevant dataset adopted, then state the details of the experimental implementation. Finally, we present the corresponding experimental results.

3.1. Dataset and Metrics

Dataset. During the *Self-Supervised Training* phase, we primarily utilize the large-scale public datasets Waymo [14] and nuScenes [3], leveraging numerous unlabeled multi-view images for pre-training. Specifically, the Waymo Open Dataset is a high-resolution sensor dataset collected by Waymo autonomous vehicles under a variety of driving conditions. Currently, the dataset includes 1,950 segments,

Table 1. Ablation experiment results on the OpenScene mini set.

Methods	CD@0.5s ↓	CD@1.0s ↓	CD@1.5s ↓	CD@2.0s ↓	CD@2.5s ↓	CD@3.0s ↓	**CD@overall ↓**
Baseline	0.7970	0.9425	1.0726	1.2022	1.3497	1.5122	1.1460
Baseline+Temporal Attention Module	0.7134	0.8903	0.9583	1.1447	1.2806	1.4793	1.0778
Self-Supervised+Baseline	0.6278	0.7570	0.8602	0.9403	1.0803	1.1366	0.9004
Proposed	**0.5531**	**0.6590**	**0.7604**	**0.8588**	**0.9659**	**1.0939**	**0.8152**

Table 2. The final leaderboard of Predictive World Model Challenge. We ranked 1st on the OpenScene Private-Test set.

rank	**id**	CD@0.5s ↓	CD@1.0s ↓	CD@1.5s ↓	CD@2.0s ↓	CD@2.5s ↓	CD@3.0s ↓	**CD@overall ↓**
1	USTC_IAT_United	0.5448	0.5883	0.6316	0.6874	0.7289	0.7878	0.6615
2	Huawei-Noah & CUHK-SZ	0.5596	0.6903	0.7858	0.8485	0.8909	0.9996	0.7958
3	mcchi	0.7669	0.8211	0.8842	0.9574	1.04	1.1677	0.9396

each containing 20 seconds of continuous driving footage. It features multimodal sensor data such as LiDAR, cameras, GPS, and IMU. These sensors capture a wealth of information across diverse environments, including day and night, dusk and dawn, sunny and rainy conditions, as well as urban centers and suburban areas. nuScenes is the first large-scale dataset to provide data from the entire sensor suite of an autonomous vehicle, including six cameras, one LiDAR, GPS, and IMU. The nuScenes dataset contains 1,000 driving scenes, each lasting 20 seconds, and annotated at a frequency of 2Hz. The nuScenes dataset is widely used for research and development in autonomous driving technology, particularly for challenging urban driving scenarios. In the subsequent *Point Clouds Prediction* phase, we extensively validate the proposed ViDAR++ on challenge dataset OpenScene [12].

Metrics. The *Chamfer Distance* [15] is used to measure the similarity between two sets of points, which represent the shapes or contours of two scenes. It compares the predicted shape to the ground truth shape by calculating the average nearest neighbor distance from each point in one set to the other set (and vice versa). For this challenge, we compare the *Chamfer Distance* between the predicted point clouds and the ground truth point clouds within the range of -51.2 meters to 51.2 meters.

3.2. Implementation Details

We implement our proposed model using the PyTorch framework. Here are the details of the training process: *(1) Self-Supervised Training Stage*. During the self-supervised training phase, we train the BEV Encoder using 16 NVIDIA A100 GPUs. The batch size is set at 64 with Stochastic Gradient Descent (SGD) [2] and a base learning rate of 0.05. The training comprises 100 epochs, and the queue size for the momentum encoder is 3,276,800. Similar to the enhancements described in MoCoV2 [4], we utilize the same loss function and data augmentation techniques; *(2) Point Clouds Prediction Stage*. During this phase, we train the model on 32 NVIDIA A100 GPUs. The training involves using 5 frames of historical multi-view images and iterating the Transformer Decoder 6 times to predict point clouds for the upcoming 3 seconds, with each frame spaced 0.5 seconds apart. To save GPU memory, we detach gradients of other predictions at each training step. The system undergoes 8 epochs of pre-training using the AdamW [8] optimizer with an initial learning rate of 2e-4, which is adjusted via a cosine annealing strategy.

3.3. Ablations Studies

The ViDAR [16] model is used to form the baseline network. In order to verify the effectiveness of the our proposed method, we design a set of ablation experiments and evaluate them on the OpenScene dataset. Considering the training and testing sensor datasets total approximately 2TB, we conduct our ablation experiments by training and validating on a mini set. The combination method is as follows: *(1) Baseline*: original ViDAR model; *(2) Self-Supervised+Baseline*: On the basis of (1), use pre-trained BEV encoder; *(3) Baseline+Temporal Attention Module*: On the basis of (1), use improved timing modules in BEV Encoder and Transformer Decoder.

3.4. Final Result

The final result of the Predictive World Model Challenge is shown in Tab. 2. Our method shows a much stronger performance compared to the 2nd solution.

References

[1] Prarthana Bhattacharyya, Chengjie Huang, and Krzysztof Czarnecki. Ssl-lanes: Self-supervised learning for motion forecasting in autonomous driving. In *Conference on Robot Learning*, pages 1793–1805. PMLR, 2023. 1

[2] Léon Bottou. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010: 19th International Conference on Computational StatisticsParis France, August 22-27, 2010 Keynote, Invited and Contributed Papers*, pages 177–186. Springer, 2010. 4

[3] 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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631, 2020. 3

[4] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020. 4

[5] Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 297–304. JMLR Workshop and Conference Proceedings, 2010. 3

[6] Yushan Han, Hui Zhang, Huifang Li, Yi Jin, Congyan Lang, and Yidong Li. Collaborative perception in autonomous driving: Methods, datasets, and challenges. *IEEE Intelligent Transportation Systems Magazine*, 2023. 1

[7] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020. 2

[8] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 4

[9] You Li and Javier Ibanez-Guzman. Lidar for autonomous driving: The principles, challenges, and trends for automotive lidar and perception systems. *IEEE Signal Processing Magazine*, 37(4):50–61, 2020. 1

[10] 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 via spatiotemporal transformers. In *European conference on computer vision*, pages 1–18. Springer, 2022. 3

[11] Liangkai Liu, Sidi Lu, Ren Zhong, Baofu Wu, Yongtao Yao, Qingyang Zhang, and Weisong Shi. Computing systems for autonomous driving: State of the art and challenges. *IEEE Internet of Things Journal*, 8(8):6469–6486, 2020. 1

[12] Songyou Peng, Kyle Genova, Chiyu Jiang, Andrea Tagliasacchi, Marc Pollefeys, Thomas Funkhouser, et al. Openscene: 3d scene understanding with open vocabularies. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 815–824, 2023. 4

[13] Connor Shorten and Taghi M Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of big data*, 6(1):1–48, 2019. 2

[14] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2446–2454, 2020. 3

[15] Tong Wu, Liang Pan, Junzhe Zhang, Tai Wang, Ziwei Liu, and Dahua Lin. Balanced chamfer distance as a comprehensive metric for point cloud completion. *Advances in Neural Information Processing Systems*, 34:29088–29100, 2021. 4

[16] Zetong Yang, Li Chen, Yanan Sun, and Hongyang Li. Visual point cloud forecasting enables scalable autonomous driving. *arXiv preprint arXiv:2312.17655*, 2023. 1, 2, 3, 4

[17] Ekim Yurtsever, Jacob Lambert, Alexander Carballo, and Kazuya Takeda. A survey of autonomous driving: Common practices and emerging technologies. *IEEE access*, 8:58443–58469, 2020. 1