MoST: Multi-modality Scene Tokenization for Motion Prediction

Supplementary Material

Figure 1. Additional qualitative comparison between MoST and Wayformer [3] baseline. The agent boxes are colored by their types: gray for vehicle, red for pedestrian, and cyan for cyclist. The predicted trajectories are ordered temporally from green (+0s) to blue (+8.0s). For each modeled agent, the models predict 6 trajectory candidates, whose confidence scores are illustrated by transparency: the more confident, the more visible. Ground truth trajectory is shown as red dots. Note that the vehicle indicated by the red arrow is entering a plaza which has no map coverage. Since our model has access to the rich visual signals, it correctly predicts the vehicle's possible trajectory which includes follows the arrow and turn right. Wayformer, on the other hand, completely missed this possibility due to the lack of road graph information in that region.

1. Additional Qualitative Results

An additional qualitative comparison can be found in Figure 1. In this scenario, the model is asked to predict the future trajectory of a vehicle entering a plaza which is not mapped by the road graph. Our model with access to visual information correctly predicts several trajectories following the arrow painted on the ground and turning right.

2. WOMD Camera Embeddings

VQGAN Embedding To extract VQGAN embedding for an image, we first resize the image into shape of 256×512. Then we horizontally split the image into two patches and apply pre-trained ViT-VQGAN [4] model on each patch respectively. Each patch contains 16×16 tokens so each camera image can be represented as 512 tokens. The code-book size is 8192.

SAM-H Embedding For each camera we extract SAM ViT-H [1] embedding of size $64 \times 64 \times 256$. Compared to VQGAN embeddings, SAM features are less spatially compressed due to its high-resolution feature map. The visualization of SAM Embedding can be found in Figure 3. We release the SAM features pooled per-scene-element.

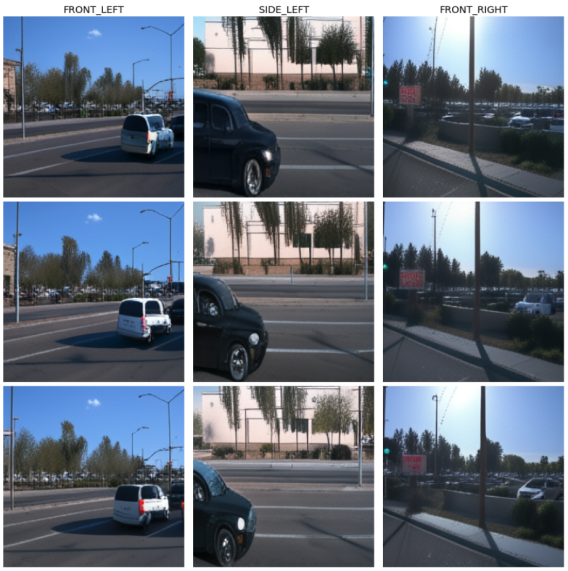

Figure 2. Examples of reconstructed driving images from ViT-VQGAN codes. We show 3 cameras at 3 consecutive timestamps. We are able to decode high quality images from VQGAN codes.

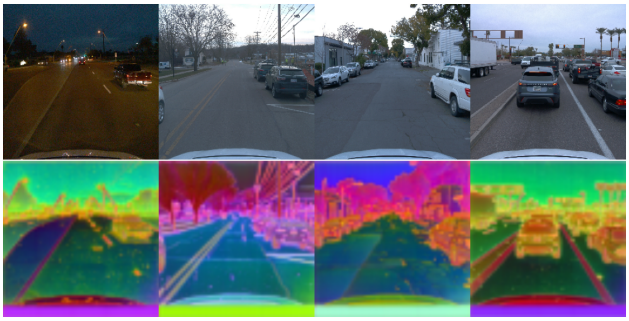

Figure 3. Examples of SAM feature. The first row shows camera images and the second row illustrates the SAM feature map visualized by PCA reduction from 256 to 3 dimensions.

3. Implementation Details

Model Detail We use $N_{\text{elem}}^{\text{agent}} = 128$, $N_{\text{elem}}^{\text{open-set}} = 384$, $N_{\text{elem}}^{\text{gnd}} = 256$, and $N_{\text{pts}} = 65536$ in our experiments. We use sensor data from past 10 frames that correspond to the 1 second history and the current frame (i.e. $T = 11$). Following Wayformer [3], we train our model to output K modes for the Gaussian mixture, where we experiment with $K = \{6, 64\}$. During inference, we draw 2048 samples from the predicted Gaussian mixture distribution, and use

Variable name	Description	Tensor Shape
$\mathbf{N}_{\text{pts}}$	The total number of LiDAR points after down-sampling.	1
$\mathbf{N}_{\text{elem}}$	The total number of scene elements.	1
T	The total number of frames.	1
D	The feature dimension.	1
$\mathbf{P}_{\text{xyz}}$	The aggregated LiDAR points from all frames after downsampling	$N_{\text{pts}} \times 3$
$\mathbf{P}_{\text{ind}}$	The scene element index and frame index for each LiDAR point	$N_{\text{pts}} \times 2$
$\mathbf{F}_{\text{pts}}$	The per point image feature.	$N_{\text{pts}} \times D$
$\mathbf{B}$	The box attributes, including box center, box size, and box heading.	$N_{\text{elem}} \times T \times 7$
$\mathbf{F}_{\text{img}}$	The per scene-element image feature.	$N_{\text{elem}} \times T \times D$
$\mathbf{F}_{\text{geo}}$	The per scene-element geometry feature.	$N_{\text{elem}} \times T \times D$
$\mathbf{f}_{\text{temporal}}$	The learnable temporal embedding.	$1 \times T \times D$

Table 1. Descriptions for variables used in the main paper.

K-Means clustering to aggregate those 2048 samples into 6 final trajectory predictions.

Training Detail For all experiments, we train our model using AdamW [2] on 64 Google Cloud TPUv4 cores[1] with a global batch size of 512. We use a cosine learning rate schedule, where the learning rate is initialized to 3×10^{-4} and ramps up to 6×10^{-4} after 1,000 steps. The training finishes after 500,000 steps.

Notations Please refer to Table 1 for a summary of the notations used in the main paper.

References

[1] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. *arXiv preprint arXiv:2304.02643*, 2023. 1

[2] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2017. 2

[3] Nigamaa Nayakanti, Rami Al-Rfou, Aurick Zhou, Kratarth Goel, Khaled S Refaat, and Benjamin Sapp. Wayformer: Motion forecasting via simple & efficient attention networks. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2980–2987. IEEE, 2023. 1

[4] Jiahui Yu, Xin Li, Jing Yu Koh, Han Zhang, Ruoming Pang, James Qin, Alexander Ku, Yuanzhong Xu, Jason Baldridge, and Yonghui Wu. Vector-quantized image modeling with improved vqgan. *arXiv preprint arXiv:2110.04627*, 2021. 1

[1] https://cloud.google.com/tpu