Occupancy Learning with Spatiotemporal Memory

Supplementary Material

A. Implementation Details

In this section, we provide implementation details of the proposed ST-Occ and experimental setup.

A.1. Spatiotemporal Memory

Our spatiotemporal memory $\mathbf{M}$ has a channel size of $C_G = 101$, which includes historical representation with 80 channels, historical class activations $\mathbf{c}$ with 18 channels, occupancy flow with 2 channels, and averaged log variance with 1 channel.

The spatial dimension H_G, W_G is determined at the beginning of each scene according to the ego-motion, and $Z_G = Z = 8$. From an analytical perspective, recurrent-based or stack-based approaches require $O(kHWZ)$ memory to store the historical frame representations, where k is the number of timesteps used (e.g., $k = 40$ for 20s at FPS 2, or $k = 200$ at FPS 10). In contrast, our temporal modeling needs $O\left((1+\Delta)HWZ\right)$ memory, where Δ represents the relative volume change and it is much smaller than k. For example, in all nuScenes samples, $\Delta_{\text{mean}} = 3.25, \Delta_{\text{max}} = 16.75$, and $\Delta_{\text{min}} = 0.56$ over a 20-second duration and it is FPS-*agnostic*. In other words, our method requires, on average, only $\frac{1}{10}$ of the memory compared to queue-based approaches and at most $\frac{1}{2}$ of the memory in the worst case.

A.2. Feature Sampling

The feature sampling operation $\chi[\cdot]$ is used to extract the ego vehicle-centered representation at timestamp t from our spatiotemporal memory $\mathbf{M}_t$ given the ego vehicle pose T_t. For the voxel grid $\mathcal{G} = \left\{\mathbf{p} \in \mathbb{R}^3 \mid \mathbf{p} = (x, y, z), x \in [0, W), y \in [0, H), z \in [0, D)\right\}$ with the ego vehicle in the center, we use the ego pose matrix $T_t \in \mathbb{R}^{4\times4}$ provided in the dataset for transformation. We first express $\mathbf{p}$ in homogeneous coordinates as

$$\tilde{\mathbf{p}} = \begin{bmatrix} x & y & z & 1 \end{bmatrix}^T. \tag{19}$$

We then transform the ego vehicle-centered grid into our spatiotemporal memory coordinate system according to

$$\mathcal{G}_{\mathcal{M}} = \left\{ \left[\mathbf{T} \cdot \begin{bmatrix} x & y & z & 1 \end{bmatrix}^T \right]_{1:3} \mid (x, y, z) \in \mathcal{G} \right\}. \tag{20}$$

We use this transformed grid for sampling via trilinear interpolation. The sampled representation is then used for downstream occupancy prediction.

When updating the spatiotemporal memory with ego vehicle-centered representation or temporal attributes, the process follows

$$\chi[\mathbf{M}_{t+1}\langle\text{value}\rangle, T_t] = \text{value}_t. \tag{21}$$

We first determine the corresponding region of interest (RoI) in scene-centered coordinates. For each voxel in the RoI, its location is transformed into ego vehicle-centered coordinates using the inverse ego pose T_t^{inv}. Then we perform grid sampling with bilinear interpolation to ensure accurate feature retrieval. This process ensures that each historically traversed location in the spatiotemporal memory is updated, thereby mitigating misalignments caused by coordinate transformations.

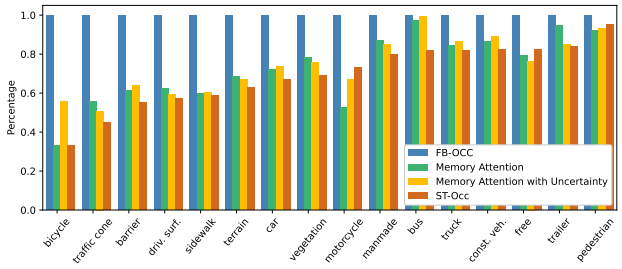


Figure A. The temporal consistency evaluation results of each class. FB-OCC is set as the baseline, followed by different settings of our method. We compute the relative mSTCV reduction with respect to baseline (the lower the better). Thus the results for FB-OCC are all 1s.

A.3. Memory Attention

Our memory attention consists of 3 temporal self-attention (TSA) layers [15], each performing operations in the sequence of *self-attention*, *normalization*, *feedforward*, and *normalization*. A 3D learnable position embedding is added to the query. For the deformable attention in the *self-attention* operation, we use four sampling points for each reference point corresponding to the query.

The network used to encode temporal attributes is a 4-layer Multi-Layer Perceptron (MLP) with hidden sizes 64, 32, 16, and 1, respectively. The encoded temporal attributes u and occupancy flow $\mathbf{f}$ are shared across all three TSA layers.

Our method applied a three-layer convolutional network on ego vehicle-centered representation for occupancy flow prediction. It achieves an mAVE of 0.618, which is on par with existing methods [22] and ensures its reliability and efficiency.

B. Temporal Consistency

Our memory attention, equipped with uncertainty awareness, also results in better temporal consistency of the occupancy prediction. Results in Tab. 2 show that uncertainty awareness contributes an additional 2% improvement in temporal consistency by reducing mSTCV. This highlights the effectiveness of uncertainty modeling in mitigating noise accumulation.

To further demonstrate the effectiveness of our design in reducing temporal inconsistencies in occupancy prediction, we evaluate temporal consistency across individual classes in Fig. A. Results reveal a 40% reduction in temporal inconsistency for static classes with our memory attention. With the uncertainty and dynamic awareness incorporated, our ST-Occ can further reduce inconsistency for certain classes. The decrease in temporal inconsistency is consistent with the increase in occupancy prediction regarding various object classes. Notably, classes such as *barrier*, *traffic cone*, and *drivable surface*, which exhibit lower temporal inconsistency, also achieve higher occupancy prediction accuracy than the baseline. These findings not only verify the effectiveness of our method but also highlight the importance of reducing temporal inconsistency in occupancy prediction, thereby providing more reliable and robust predictions for downstream tasks.

C. Historical Occupancy Prediction

To evaluate the models' performance in preserving and utilizing historical information, we extend the original occupancy prediction evaluation scope while maintaining the mIoU metric unchanged. During the evaluation, we included not only the visible voxels in the current timestamp but also any invariant voxels visible in the previous timestamp. Voxels corresponding to dynamic objects in historical frames are excluded to ensure evaluation consistency. Furthermore, these historically visible invariant voxels can be incorporated during training to enhance occupancy learning.

The results in Tab. A demonstrate that including historically visible voxels in the evaluation leads to a lower mIoU score than the original setting, as accurately predicting these voxels is inherently more challenging. Despite this harder evaluation, our proposed ST-Occ outperforms the FB-OCC by a margin of 5%. Additionally, when historically visible voxels are incorporated during training, our method achieves the highest performance on the extended evaluation scope. The observed performance improvement in FB-OCC indicates that training with historically visible voxels benefits the occupancy prediction.

Method	mIoU	mIoU†
FB-OCC	39.11	33.71
ST-Occ	**42.13**	35.34
FB-OCC‡	40.06	35.78
ST-Occ ‡	41.62	**36.96**

Table A. Historical occupancy prediction results on the extended Occ3D benchmark. † denotes evaluation with historically visible voxels included. ‡ incorporates historically visible voxels during training.

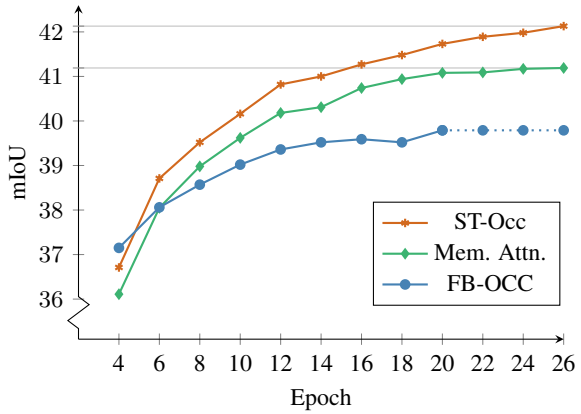


Figure B. Training curve of our ST-Occ and Memory Attention compared to FB-OCC. Our approach demonstrates faster convergence while achieving impressive performance.

D. Training Analysis

Fig. B illustrates the training curve of our method compared with the baseline. While our method slightly underperforms FB-OCC [16] during the initial epochs, this is attributed to the more complex network architecture deployed for temporal fusion. However, our method demonstrates faster convergence and performs comparable to FB-OCC in nearly half the training epochs. In the end, the ST-Occ delivers impressive performance with significant performance improvements.

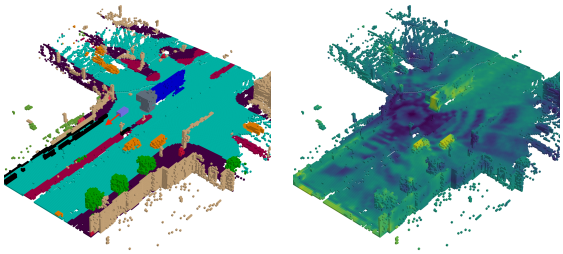
E. Visualization

E.1. Uncertainty

We visualize uncertainty estimated by our method in Fig. C, where dynamic, occluded, and unobserved voxels have higher uncertainty while observed regions show lower uncertainty.

E.2. Occupancy Prediction

We present visualizations of ego vehicle-centered and scene-level occupancy prediction done by the proposed method on additional large-scale scenes in Fig. D. Our

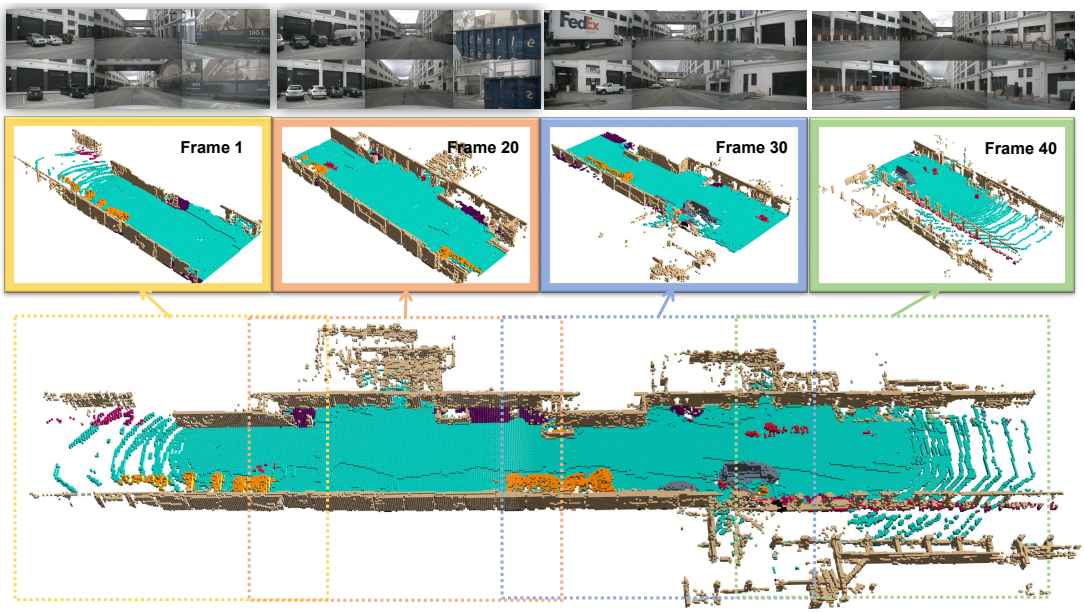


(a) ST-Occ Prediction (Ego Veh.) **(b)** Uncertainty from ST-Occ

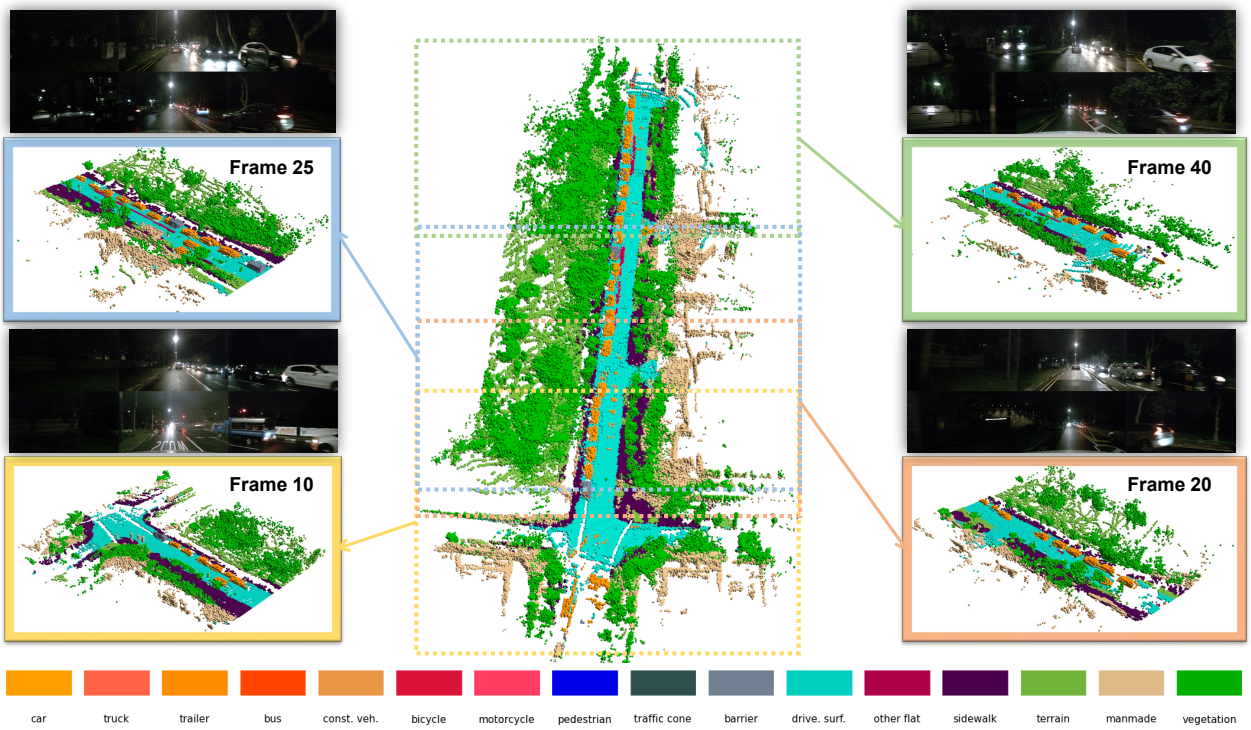
Figure C. Visualization of uncertainty in ST-Occ.

method can produce precise occupancy predictions and construct a comprehensive scene representation.

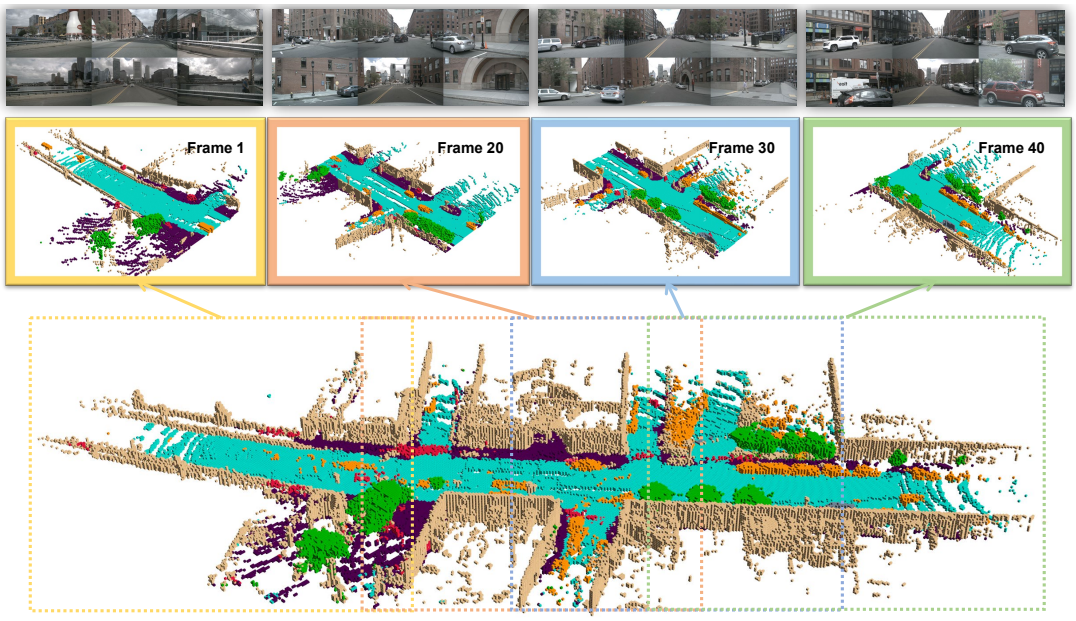
The minor inconsistencies observed between ego vehicle-centered and scene-level prediction (particularly evident in frame 20 of Fig. Diii and Fig. Div) can be attributed to two factors: 1) Continuous Updates. The RoI regarding each frame in the spatiotemporal memory is updated incrementally by subsequent frames with additional observations. 2) Dynamic Instances. Our pipeline does not incorporate explicit dynamic object masking in the spatiotemporal memory. Instead, we rely on memory attention with uncertainty and dynamic awareness to handle dynamic voxels implicitly.



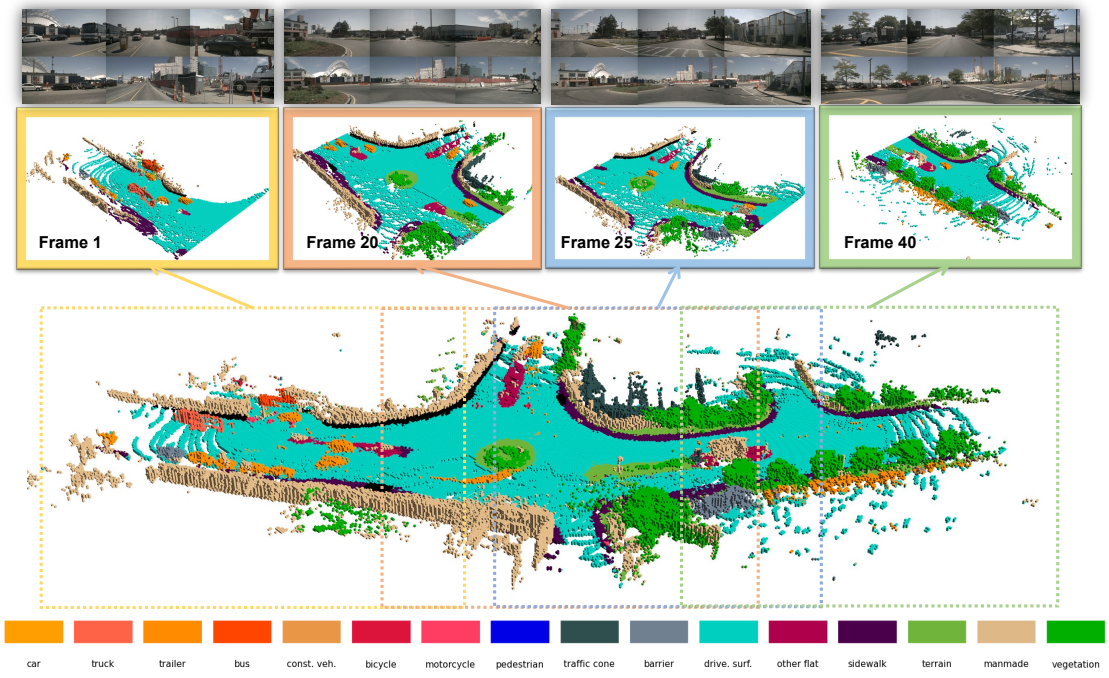
(i) *Scene-0092*



(ii) *Scene-1069*



(iii) *Scene-0103*



(iv) *Scene-0523*

Figure D. Visualization of occupancy prediction results from ST-Occ across four representative scenes. The ego vehicle-centered predictions at different timestamps are displayed within solid-line boxes, with the corresponding input RGB images provided above. Each scene also includes the scene-level occupancy prediction derived from our spatiotemporal memory aggregated over all frames.