Supplementary Material

In this supplementary material, we present the implementation details of the proposed LaPred method.

1. System Setup and Training Details

- Time steps of past trajectory
  - nuScenes dataset: $\tau = 4$ (2 seconds)
  - Argoverse dataset: $\tau = 20$ (2 seconds)
- Time steps of future trajectory
  - nuScenes dataset: $h = 12$ (6 seconds)
  - Argoverse dataset: $h = 30$ (3 seconds)
- Number of lane candidates: $N = 6$
- Setting for lane candidates
  - nuScenes dataset
    - Length of lane candidate: 130m (Forward: 100m, Backward: 30m)
    - Distance between adjacent coordinate points: 0.5m
    - Total number of the coordinate points for each lane candidate: $M = 260$
  - Argoverse dataset
    - Length of lane candidate: 80m (Forward: 50m, Backward: 30m)
    - Distance between adjacent coordinate points: 1.0m
    - Total number of the coordinate points for each lane candidate: $M = 80$
- Adam optimizer with initial learning rate 0.0003
- Learning rate decay: Reduced by half when the validation loss is plateaued for more than three epochs
- Batch size: $B = 32$
- Loss parameters: $\alpha = 0.3$ and $\beta = 0.7$

2. Network Architecture

Table 1 to 3 present the detailed network architectures of the TFE block, LA block, and MTP block, respectively.

Input	$V^{(P)}$		L^i		V^i	
1D CNN	Input size	$B \times \tau \times 2$	Input size	$B \times M \times 2$	Input size	$B \times \tau \times 2$
	Specification	$u64 - k2 - s1 - p0$	Specification	$u64 - k3 - s1 - p1$	Specification	$u64 - k2 - s1 - p0$
	Output size	$B \times (\tau - 1) \times 64$	Output size	$B \times M \times 64$	Output size	$B \times (\tau - 1) \times 64$
	Input size	$B \times (\tau - 1) \times 64$	Input size	$B \times M \times 64$	Input size	$B \times (\tau - 1) \times 64$
	Specification	$u64 - k2 - s1 - p0$	Specification	$u64 - k3 - s1 - p1$	Specification	$u64 - k2 - s1 - p0$
	Output size	$B \times (\tau - 2) \times 64$	Output size	$B \times M \times 64$	Output size	$B \times (\tau - 2) \times 64$
	Input size	$B \times (\tau - 2) \times 64$	Input size	$B \times M \times 64$	Input size	$B \times (\tau - 2) \times 64$
	Specification	$u96 - k3 - s1 - p1$	Specification	$u96 - k3 - s1 - p1$	Specification	$u96 - k3 - s1 - p1$
	Output size	$B \times M \times 96$	Output size	$B \times M \times 96$	Output size	$B \times M \times 96$
	Input size	$B \times M \times 96$	Input size	$B \times M \times 96$	Input size	$B \times M \times 96$
	Specification	$u96 - k3 - s1 - p1$	Specification	$u96 - k3 - s1 - p1$	Specification	$u96 - k3 - s1 - p1$
	Output size	$B \times M \times 64$	Output size	$B \times M \times 64$	Output size	$B \times M \times 64$
LSTM	Input size	$B \times (\tau - 2) \times 64$	Input size	$B \times M \times 96$	Input size	$B \times (\tau - 2) \times 64$
	Specification	$u512$	Specification	$u2048$	Specification	$u512$
	Output size	$B \times 512$	Output size	$B \times 2048$	Output size	$B \times 512$
Output	ξ_{V_p}		ξ_{L^i}		ξ_{V^i}	
	Concatenation					
FC	Input size		$B \times 3072$			
	Specification		$u2048$			
	Output size		$B \times 2048$			
	Input size		$B \times 2048$			
	Specification		$u2048$			
	Output size		$B \times 2048$			
	Input size		$B \times 2048$			
	Specification		$u1024$			
	Output size		$B \times 1024$			
	Input size		$B \times 1024$			
	Specification		$u1024$			
	Output size		$B \times 1024$			
Output	ξ^i					

Table 1: Detailed network architecture of TFE block. $ux - ky - sz - pw$ represents a layer with the number of unit x, kernel size y, stride z, and padding on all side w.

Input	$\xi^{1:N}$	
	Concatenation	
FC	Input size	$B \times (1024 \times N)$
	Specification	$u512$
	Output size	$B \times 512$
	Input size	$B \times 512$
	Specification	$u512$
	Output size	$B \times 512$
	Input size	$B \times 512$
	Specification	$u256$
	Output size	$B \times 256$
	Input size	$B \times 256$
	Specification	$u256$
	Output size	$B \times 256$
	Input size	$B \times 256$
	Specification	$u64$
	Output size	$B \times 64$
	Input size	$B \times 64$
	Specification	$u64$
	Output size	$B \times 64$
	Input size	$B \times 64$
	Specification	$u(N)$
	Output size	$B \times N$
	Softmax	

Table 2: Detailed network architecture of LA block.

Input	ξ	ξ_{V_p}
	Concatenation	
FC(k)	Input size	$B \times 1536$
	Specification	$u512$
	Output size	$B \times 512$
	Input size	$B \times 512$
	Specification	$u512$
	Output size	$B \times 512$
	Input size	$B \times 512$
	Specification	$u256$
	Output size	$B \times 256$
Shared FC	Input size	$B \times 256$
	Specification	$u256$
	Output size	$B \times 256$
	Input size	$B \times 256$
	Specification	$u\,(h \times 2)$
	Output size	$B \times (h \times 2)$
Output	$\hat{V}^{(f,k)}$	

Table 3: Detailed network architecture of MTP block.

3. Single-agent vs. Multi-agent features for employing nearby agents.

Table 4 presents the comparison of the prediction performance among different methods to employ nearby agents in the TFE block. Specifically, we compare our proposed method $LaPred_{SL}$, which considers a single agent per lane, against two different methods $LaPred_{ML}$ and $LaPred_{M}$, which aggregate multiple nearby agents per lane by using max-pooling. In $LaPred_{ML}$, the closest lane to each agent is identified then the agent features that share the same closest lane are aggregated to their corresponding closest lane. On the other hand, $LaPred_{M}$ aggregates every agent feature to all lanes regardless of their distances to lanes.

Method	ADE_5	FDE_5	ADE_{10}	FDE_{10}
$LaPred_{SL}$	1.53	3.37	1.21	2.61
$LaPred_{ML}$	1.56	3.42	1.22	2.63
$LaPred_{M}$	1.60	3.56	1.25	2.68

Table 4: Performance of different methods to aggregate nearby agents evaluated on nuScenes validation set.