PEARL: A Lightweight Prompt-based Feature Interpreter Framework for Real-Time, Anonymous, and Heterogeneous Collaborative Perception

Supplementary Material

Table 4. Detailed configurations for agent type encoders.

Encoder	Variation	Voxel Resolution	2D / 3D CNN Layers	Half LiDAR Range (x,y)	Feature size (C×H × W)
PointPillar [11]	pp8	0.8, 0.8, 4	4 / 0	102.4, 51.2	64×64 × 128
	pp6	0.6, 0.6, 4	4 / 0	153.6, 76.8	64×128 × 256
	pp4	0.4, 0.4, 4	4 / 0	102.4, 51.2	64×128 × 256
VoxelNet [40]	vn6	0.6, 0.6, 0.4	0 / 3	153.6, 76.8	128×256×512
	vn4	0.4, 0.4, 0.4	0 / 3	102.4, 51.2	128×256×512
SECOND [34]	sd2	0.2, 0.2, 0.2	4 / 12	102.4, 51.2	64×64 × 128
	sd1	0.1, 0.1, 0.1	4 / 13	102.4, 51.2	64×128 × 256

1. Experimental Setting Details

LiDAR Encoders. Tab. 4 summarizes the LiDAR encoder configurations, including the voxel resolution, the 3D CNN layers that process the encoder input, and the 2D BEV backbone that converts 3D features into a BEV feature map with the sizes shown in the last column. Unless otherwise specified, all encoders operate over a LiDAR range of $[-102.4, 102.4]$ m along x and $[-51.2, 51.2]$ m along y. The pp6 and vn6 variants instead use an extended range of $[-153.6, 153.6]$ m along x and $[-76.8, 76.8]$ m along y. For PointPillar [11], we use two variants. The pp8 variant serves as the ego encoder and produces a BEV feature map of size $64 \times 64 \times 128$. The pp4 variant is used as a heterogeneous neighbor in both Stage-1 and Stage-2. For VoxelNet [40], we use vn4 as a heterogeneous neighbor in Stage-1 and vn6 as a new heterogeneous agent in Stage-2 for adaptation. At last, for SECOND [34], we use sd2 as a heterogeneous neighbor in Stage-1 and sd1 as a new heterogeneous neighbor in the adaptation stage. All encoder variations are followed by a lightweight compressor that maps their BEV features to the ego feature size of $64 \times 64 \times 128$.

Multi-scale Extractor/Decoder. Our multi-scale extractor comprises three ResNeXt [28] stages operating on the ego BEV feature map. The first stage uses three blocks with 64 channels and stride 1, preserving the spatial resolution and outputting features of size $64 \times 64 \times 128$. The second stage uses five blocks with 128 channels and stride 2, downsampling the BEV features by a factor of 2 to $128 \times 32 \times 64$. The third stage uses eight blocks with 256 channels and stride 2, further downsampling by a factor of 2 to $256 \times 16 \times 32$. All Pipeline-1 and Pipeline-2 modules, *i.e.*, channel cross-attention, spatial attention, the foreground estimator, LWSD, LWDDI, and the domain classifier, operate on these three feature scales. The multi-scale decoder maps each scale back to a common feature space of size $128 \times 64 \times 128$, using stride 1 for the first scale, stride 2 for the second, and stride 4 for the third. The resulting three feature maps are concatenated along the channel dimension and passed through the detection head to produce the final detection outputs.

2. Number of Trainable Parameters Comparison

We further compare the number of trainable parameters of PEARL and PolyInter [26] in offline training. As shown in Tab. 5, in Stage-1 PEARL consistently uses fewer trainable parameters than PolyInter, reducing the parameter count by 1.1 M–4.4 M across different heterogeneous bases. In Stage-2, PEARL has a comparable number of trainable parameters to PolyInter for pp4 and sd1, and higher number of parameters for the other cases. Overall, and by considering both stages, PEARL has lower numbers of parameters for all scenarios (except for the EfficientNet case) while achieving better detection performance as reported in Sec. 4, and PEARL is more efficient and more viable for real-time deployment. The total number of parameters of the deployed models directly reflects the runtime complexity, and PEARL maintains a more compact overall model than PolyInter due to its lightweight prompt-based design.

3. Details for Real-Time Implementation Results

3.1. Anonymous New Joining Agent

We evaluated PEARL under real-time constraints, as described in Sec. 4.3. At deployment, the ego applies each pretrained LWDDI to the new agent's intermediate BEV features, computes the cosine similarity to its own domain-invariant features, and assigns the interpreter with the highest similarity, without requiring any sensor or model metadata. We operate in the domain-invariant space because domain-specific features from other agents are not available at deployment; producing them would require access to their encoders, reveal sensor/model details, and incur substantial extra computation. We use three scales (stride 2 per scale, with channels doubling across scales) and weight similarities by $[1, 2, 4]$ when averaging across scales.

For fairness, we report both detection performance and cosine similarities across all pretrained interpreters (from Stage-1 and Stage-2), to quantify the effectiveness of the selection process. As there is no established baseline for anonymous interpreter assignment, we compare against the

Table 5. Number of trainable parameters comparison in million (M) with CoBEVT fusion for PolyInter. Stage-1: heterogeneous bases pp4-vn4, vn4-sd2, vn4-ResNet, pp4-EfficientNet with pp8 as ego. Stage-2: new-agent adaptation per base (LiDAR-only: pp4, sd1, vn6; cross-modality: EfficientNet, ResNet). For each method, the first row corresponds to Stage-1 and the second row to Stage-2.

Stage-1 Scenarios		pp4-vn4			vn4-sd2		vn4-ResNet	pp4-EfficientNet
Stage-2 Scenarios	pp4	sd1	vn6	pp4	sd1	vn6	EfficientNet	ResNet
PEARL (ours)		12.4			12.4		12.4	12.4
	4.1	4.1	4.1	4.1	4.1	4.1	4.1	4.1
PolyInter [26]		15.7			16.8		13.5	15.7
	4.3	3.2	1.1	4.3	3.2	1.1	1.1	1.1

average detection AP obtained by randomly selecting an interpreter. We consider a pp4 agent joins without exposing metadata.

A summary of the real-time implementation results is provided in Sec. 4 and Fig. 2. Here, we present the detailed results for pp4 in Tab. 6. For pp4 modality, an anonymous agent initiates collaboration with the ego agent. First, the ego shares a compressor weights with the new agent, specified only by the BEV feature size. The new agent then sends only the compressed BEV features back to the ego. This design reduces communication bandwidth while preserving the new agent's privacy and anonymity. The ego then selects a suitable interpreter by computing the similarity between the new agent's domain-invariant features, generated by each pretrained LWDDI, and its own domain-invariant features.

For pp4, the detailed results are shown in Tab. 6. Interpreters that were trained with pp4 (shown in bold) achieve the highest average similarities (S_{avg}) and the best detection performance. Interpreters pretrained on pp4 gain up to **+24.5%** AP@0.5 and **+31.2%** AP@0.7. On average, the same pretrained interpreters on pp4 gain **+7.0%** AP@0.5 and **+13.0%** AP@0.7 over the average performance across all interpreters (i.e., under random selection). The mean similarity between the pp4 domain-invariant features and the ego features for interpreters pretrained on pp4 exceeds the all-interpreter average by 0.06 (with similarity threshold τ=0.85).

3.2. Real-Time Complexity

Detailed timing breakdowns are reported in Tab. 7. To evaluate PEARL's real-time complexity, we measure the per-sample GPU time for interpreter selection and collaborative detection, where each sample contains both an ego and a neighbor agent. On average, the Pipeline-2 real-time selection step takes **1.67 ms**, and the MS extractor takes **47.0 ms** per sample. This yields an end-to-end selection time of **48.67 ms** per sample. Here, the Pipeline-2 latency (**1.67 ms**) corresponds to the similarity-based interpreter selection in the domain-invariant feature space, while the additional cost arises from applying the multi-scale (MS) feature extractor for each candidate interpreter. The fastest interpreters and heterogeneous bases are those based on `pp4` (PointPillar), while the slowest ones involve camera-based backbones. Importantly, this selection overhead is incurred only on the ego side and does not require querying or executing any additional encoders, which makes the design practical for deployment.

After the ego connects with the neighbor, it first performs the end-to-end selection process to choose a suitable pretrained detection interpreter, and subsequently uses only the selected interpreter for cooperative detection.

4. The V2XSet Dataset Results

In addition to the results reported on OPV2V [33] in Sec. 4 and in this supplementary material, we further evaluate PEARL on V2XSet [32] for offline training and real-time deployment. V2XSet is a large-scale open V2X dataset incorporating vehicle-to-everything cooperation.

4.1. Performance Comparison

Following Sec. 4.2 of the paper, we compare PEARL with the state-of-the-art PolyInter [26] (one-stage interpreter) under the authors' settings, using CoBEVT [31] for PolyInter's fusion and matching LiDAR ranges for fairness. Tab. 8 shows Stage-1 (heterogeneous base) and Stage-2 (new-agent adaptation) results. PEARL efficiently reduces heterogeneous domain gaps in Stage-1 and further adapts the interpreters in Stage-2. Both Stage-1 and Stage-2 interpreters can be deployed under real-time constraints (see Sec. 4.2).

With pp8 as ego and LiDAR neighbors pp4-vn4, vn4-sd2, PEARL improves AP by **+1.2/+0.8** and **+1.0/+0.6** (AP@0.5/@0.7) over PolyInter, demonstrating consistency and improved AP performance by reducing heterogeneous feature gaps while maintaining privacy of new heterogeneous agents. In Stage-2, it further outperforms PolyInter by **+1.1/+0.7** for pp4, **+1.4/+0.7** for sd1, and **+2.0/+1.7**

Table 6. Detection performance (AP@0.5 / AP@0.7(%)), feature similarity for scales 0–2 (S_0–S_2) and their average (S_{avg}) for real-time anonymous model selection with pp4. Heterogeneous bases: pp4-vn4, vn4-sd2, vn4-ResNet, pp4-EfficientNet with pp8 as ego. Stage-2 (s-2): new-agent adaptation per base (LiDAR-only: pp4, sd1, vn6; cross-modality: EfficientNet, ResNet).

Stage-1 Scenarios	Interpreter	AP@0.5/0.7(%)	S_0	S_1	S_2	S_{avg}
pp4-vn4	**pp4**	**95.6/84.0**	**0.8049**	**0.8543**	**0.8649**	**0.8533**
pp4-vn4	vn4	93.4/72.9	0.8844	0.8464	0.8507	0.8531
pp4-vn4	**pp4 (s-2)**	**95.5/83.7**	**0.9240**	**0.8472**	**0.8913**	**0.8834**
pp4-vn4	sd1 (s-2)	87.2/66.6	0.8569	0.8149	0.7377	0.7767
pp4-vn4	vn6 (s-2)	70.9/52.5	0.8001	0.7835	0.8184	0.8059
vn4-sd2	vn4	84.1/63.6	0.8299	0.7795	0.8327	0.8171
vn4-sd2	sd2	86.4/63.0	0.8951	0.8058	0.8167	0.8248
vn4-sd2	**pp4 (s-2)**	**95.5/84.0**	**0.8403**	**0.8577**	**0.86676**	**0.8604**
vn4-sd2	sd1 (s-2)	88.5/67.1	0.8571	0.8585	0.7653	0.8050
vn4-sd2	vn6 (s-2)	79.7/55.6	0.8001	0.6654	0.6579	0.6805
vn4-ResNet	vn4	88.1/68.6	0.8833	0.8223	0.8376	0.8398
vn4-ResNet	Resnet	88.8/71.7	0.8833	0.8400	0.6085	0.7340
vn4-ResNet	EfficientNet (s-2)	88.6/71.9	0.9006	0.7671	0.6786	0.7356
p4-EfficientNet	**pp4**	**95.2/83.1**	**0.8428**	**0.9066**	**0.9134**	**0.9014**
pp4-EfficientNet	EfficientNet	88.3/71.4	0.8856	0.8733	0.7007	0.8107
pp4-EfficientNet	ResNet (s-2)	88.4/70.5	0.8191	0.8150	0.7920	0.8025
Average over	all	88.4/70.7	0.8568	0.8210	0.7932	0.8142
Minimum AP	all	70.9/52.5	0.8001	0.7835	0.8184	0.8059
Average over	**pretrained pp4**	**95.4/83.7**	**0.8530**	**0.8664**	**0.8841**	**0.8746**

Table 7. Total Pipeline-2 and MS extractor computation time for Stage-2 interpreters (in milliseconds). Heterogeneous bases: pp4-vn4, vn4-sd2, vn4-ResNet, pp4-EfficientNet with pp8 as ego. Stage-2: new-agent adaptation per base (LiDAR-only: pp4, sd1, vn6; cross-modality: EfficientNet, ResNet).

Stage-1 Scenarios	Interpreter	Pipeline-2	Ms extractor	Pipeline-2+Ms extractor
pp4-vn4	pp4	1.60	27.7	29.3
pp4-vn4	sd1	1.64	58.0	59.70
pp4-vn4	vn6	1.65	50.5	52.22
vn4-sd2	pp4	1.61	19.9	21.51
vn4-sd2	sd1	1.65	59.3	61.01
vn4-sd2	vn6	1.67	54.5	56.14
vn4-ResNet	EfficientNet	1.76	53.6	55.46
pp4-EfficientNet	ResNet	1.80	52.5	54.54
Average over	all	1.67	47.0	48.67

for vn6 (AP@0.5/@0.7). The largest gains are observed on vn6, where the longer LiDAR range and coarser voxels increase the domain shift. Overall, PEARL exceeds PolyInter by **+1.1/+0.7** in Stage-1 and **+1.5/+1.0** in Stage-2 (AP@0.5/@0.7).

Table 8. Detection performance comparison (AP@0.5 / AP@0.7) on V2XSet with CoBEVT fusion for PolyInter. Stage-1: heterogeneous bases pp4-vn4 and vn4-sd2 with pp8 as ego. Stage-2: new-agent adaptation per base (LiDAR-only: pp4, sd1, vn6).

Stage-1 Scenarios		pp4-vn4			vn4-sd2	
Stage-2 Scenarios	pp4	sd1	vn6	pp4	sd1	vn6
PEARL (ours)		**86.7/61.3**			**87.8/65.4**	
	86.4/59.9	**88.5/68.8**	**74.9/56.4**	**85.9/59.2**	**88.7/69.0**	**75.5/57.6**
PolyInter [26]		85.5/60.5			86.8/64.8	
	85.4/59.2	86.8/68.0	73.1/54.5	84.6/58.7	87.3/68.4	73.4/56.0

Table 9. Detection performance (AP@0.5 / AP@0.7, %), feature similarity for scales 0–2 (S_0–S_2) and their average (S_{avg}) for real-time anonymous model selection with pp4 on V2XSet dataset. Heterogeneous bases: pp4-vn4, vn4-sd2, with pp8 as ego. Stage-2 (s-2): new-agent adaptation per base (LiDAR-only: pp4, sd1, vn6).

Stage-1 Scenarios	Interpreter	AP@0.5/0.7(%)	S_0	S_1	S_2	S_{avg}
pp4-vn4	**pp4**	**85.4/56.8**	**0.8539**	**0.9002**	**0.9355**	**0.9138**
pp4-vn4	vn4	47.4/16.0	0.8006	0.8733	0.9053	0.8812
pp4-vn4	**pp4 (s-2)**	**86.3/58.9**	**0.8991**	**0.8822**	**0.9000**	**0.8948**
pp4-vn4	sd1 (s-2)	43.2/14.4	0.7778	0.8427	0.7933	0.8052
pp4-vn4	vn6 (s-2)	33.5/10.4	0.8415	0.8307	0.8062	0.8182
vn4-sd2	vn4	14.5/4.0	0.7495	0.8189	0.8861	0.8474
vn4-sd2	sd2	22.5/7.4	0.7419	0.8435	0.9095	0.8667
vn4-sd2	**pp4 (s-2)**	**84.7/57.0**	**0.8946**	**0.8938**	**0.8930**	**0.8935**
vn4-sd2	sd1 (s-2)	21.5/6.3	0.8394	0.8354	0.8965	0.8709
vn4-sd2	vn6 (s-2)	41.2/10.9	0.8104	0.8531	0.8457	0.8427
Average over	all	48.0/24.2	0.8209	0.8574	0.8771	0.8634
Average over	**pretrained pp4**	**85.5/57.6**	**0.8825**	**0.8921**	**0.9095**	**0.9007**

4.2. V2XSet Real-Time Results

We also evaluated our real-time interpreter selection process on V2XSet. Following Sec. 3.1, we test the anonymous selection protocol using pp4 as an anonymous neighbor. The trends match those observed on OPV2V: the cosine-similarity-based selection reliably identifies the most suitable interpreter among the pretrained models, improving AP over random interpreter selection while maintaining a small selection overhead on the ego side.

For pp4, the detailed results are shown in Tab. 9. Interpreters that were trained with pp4 (shown in bold) achieve the highest average similarities (S_{avg}) and the best detection performance. Interpreters pretrained on pp4 gain **+37.4%** AP@0.5 and **+33.4%** AP@0.7 over the average performance across all interpreters (i.e., under random selection). The mean similarity between the pp4 domain-invariant features and the ego features for interpreters pretrained on pp4 exceeds the all-interpreter average by 0.04 (with similarity threshold τ=0.89).