Supplementary Material

7. Overview

This supplementary material provides additional technical details and extended results supporting the proposed Visual-referred Probabilistic Prompt Learning (**VirPro**) framework. We first introduce the computation details of RoI contrastive learning objective (Sec. 8), which improves semantic coherence within scenes and enhances inter-scene discriminability in the latent space. We then summarize the pseudo-label generation pipelines of the weakly supervised baselines, WeakM3D and GGA (Sec. 9), clarifying their geometric, semantic, and alignment constraints. Implementation details for all components in these two baselines are provided in Sec. 10, followed by definitions of the clustering metrics used for latent space analysis (Sec. 11). We present additional quantitative (Sec. 12) and qualitative (Sec. 13) results. Finally, we demonstrate additional ablations in Sec. 14.

8. RoI Contrastive Learning

To reinforce the semantic coherence among co-occurring objects within the same scene in the latent space while discriminating scene-specific traits, we follow CAW3D [24] to adopt an object-level matching paradigm based on the traditional image-text contrastive learning. The associated loss is defined below.

Let $\mathbf{e}_i^{\text{txt}}$ denote the text embeddings of the i-th object normalized from prompt distributions $\hat{z}_{i,j}^{(t)}$ by maxpooling, and $\mathbf{e}_j^{\text{img}}$ denote the image embeddings of the j-th object, extracted from the Monocular 3D Encoder and spatially aligned using a 2D detector. The cosine similarity between these embeddings, along with the corresponding contrastive loss for the i-th sample, is formulated as follows:

$$\text{sim}_{ij} = \frac{\left\langle \mathbf{e}_i^{\text{txt}}, \mathbf{e}_j^{\text{img}} \right\rangle}{\|\mathbf{e}_i^{\text{txt}}\|_2 \cdot \|\mathbf{e}_j^{\text{img}}\|_2}, \ell_i = -\log \frac{\exp(\text{sim}_{ij}/\tau)}{\sum_{k=1}^{N} \exp(\text{sim}_{ik}/\tau)}, \tag{12}$$

$$\mathcal{L}_{\text{contrast}} = \frac{1}{N} \sum_{i=1}^{N} \ell_i, \tag{13}$$

where $\langle \cdot, \cdot \rangle$ denotes the inner product, l_i denotes the Cross-Entropy Loss between $\mathbf{e}_i^{\text{txt}}$ and $\mathbf{e}_i^{\text{img}} \tau$ is a temperature scaling factor. N is the total number of objects in the batch.

9. Pseudo-Labels in Baselines

9.1. WeakM3D

WeakM3D [31] generates pseudo 3D labels by projecting LiDAR point clouds onto the corresponding 2D object masks of each image, thereby extracting Region-of-Interest (RoI) points. These RoI points are subsequently aligned with the predicted 3D bounding boxes for loss calculation. To handle the inherent challenges in this process, WeakM3D incorporates essential loss functions as follows:

Geometric Alignment Loss aims to minimize the discrepancy caused by using center loss alone to determine the center of predicted 3D bounding box. The formulation is given as:

$$\begin{aligned} \mathcal{L}_{\text{geo}} &= \|\mathbf{p}_i - \hat{\mathbf{p}}_i\|_1 \\ &= \left\| \mathbf{p}_i - \text{Intersect}\left(\overrightarrow{\mathbf{c} \to \mathbf{p}_i}, \ \hat{b}_{3d} \right) \right\|_1, \end{aligned} \tag{14}$$

where $\mathbf{p}i$ denotes the i-th RoI point, and $\hat{\mathbf{p}}_i$ is computed as the intersection between the ray from predicted 3D center $\mathbf{c}$ to $\mathbf{p}i$ and the surface of predicted 3D bounding box $\hat{b}3d$.

Ray Tracing Loss is designed to mitigate surface uncertainty associated with RoI points by enforcing their accurate correspondence to the correct object surface, thereby enhancing geometric consistency and localization precision. The loss is formulated as:

$$\mathcal{L}_{\text{ray}} = \begin{cases} \left\| \mathbf{p}_i - \mathbf{p}_i^{(r)} \right\|_1, & \text{if Ray}(\mathbf{p}_{\text{cam}} \to \mathbf{p}_i) \cap \hat{b}_{3d} \neq \emptyset, \\ 0, & \text{otherwise}, \end{cases} \tag{15}$$

and $\mathbf{p}_i^{(r)}$ denotes the intersection point on the predicted 3D bounding box $\hat{b}_{3d}$ that is closest to the camera along the ray from the camera center $\mathbf{p}_{\text{cam}}$ through the RoI point $\mathbf{p}_i$.

Point-wise Balancing Loss compensates for non-uniform point cloud distributions by ensuring that sparse yet significant points are not overlooked, thereby improving the completeness of the overall object detection process. For each point $\mathbf{p}_i$, we compute its local neighborhood density as:

$$w_i = \left| \left\{ \mathbf{p}_j \mid \|\mathbf{p}_i - \mathbf{p}_j\|_2 < R, \ j \neq i \right\} \right|, \tag{16}$$

where w_i is the neighborhood count of point $\mathbf{p}_i$, and R is a predefined distance threshold for determining neighborhood connectivity within the RoI point set.
The final 3D supervision loss is then weighted inversely by this density, and formulated as:

$$\mathcal{L}_{\text{3D}} = \frac{1}{M} \sum_{i=1}^{M} \frac{1}{w_i} \left(\mathcal{L}_{\text{geo},i} + \mathcal{L}_{\text{ray},i} + \lambda \, \mathcal{L}_{\text{center},i} \right), \tag{17}$$

where M denotes the total number of RoI points and λ is a scalar hyperparameter used to balance the contribution of the center loss term.
By incorporating this series of 3D loss functions, the monocular detector is guided to acquire enhanced spatial awareness through supervision from 3D pseudo labels, thereby improving the accuracy of 3D object detection.

9.2. GGA

GGA [55] presents a unified weakly supervised 3D detection framework that integrates geometric constraints, 2D–3D consistency, and static textual prompts. For each 2D frustum, point clouds are cropped as *In-Box Points* and fed into a point-cloud backbone with a proposal head to estimate 3D bounding boxes, class scores, and auxiliary pseudo-scores. To strengthen geometric reliability, GGA incorporates the following components:
Boundary Projection Loss (BPL) enforces 2D-3D consistency by aligning each predicted 3D bounding box with its corresponding 2D annotation. Given the calibrated camera model, the eight corners of a predicted 3D box are first projected onto the image plane, and the minimum enclosing rectangle of these projected points forms a predicted 2D box. Formally, let
$\mathbf{C}^p = \mathrm{Proj}(\mathrm{Corners}(B^p_{3d}))$ denote the set of projected corners, and define the predicted 2D bounds as $\mathbf{b}^p_{2d} = [\min(\mathbf{C}^p_x), \min(\mathbf{C}^p_y), \max(\mathbf{C}^p_x), \max(\mathbf{C}^p_y)]$. The BPL then minimizes the discrepancy between $\mathbf{b}^p_{2d}$ and the ground-truth 2D box $\mathbf{b}_{2d} = [x_{\min}, y_{\min}, x_{\max}, y_{\max}]$ through an L_1 penalty:

$$\mathcal{L}_{\mathrm{BPL}} = \left\| \mathbf{b}^p_{2d} - \mathbf{b}_{2d} \right\|_1 . \tag{18}$$

This loss encourages the projected 3D box to tightly align with its 2D counterpart, thereby constraining the 3D box location and scale from the perspective of the image space.
Semantic Ratio Loss (SRL) leverages simple yet effective shape priors derived from GPT-4 to regularize the predicted 3D box dimensions. Instead of relying on handcrafted geometric rules or synthetic statistics, SRL uses the observation that the bird's-eye-view width–height ratio provides sufficient semantic cues for constraining object shapes. Let the predicted 3D box be parameterized by $(x, y, z, l, w, h, \alpha)$. We compute the predicted ratio using the shorter side over the longer side:

$$r^p = \frac{\min(l, w)}{\max(l, w)}. \tag{19}$$

Given a category-level prior ratio r obtained from GPT-4, SRL penalizes deviations between the predicted and prior ratios using an L_1 loss:

$$\mathcal{L}_{\mathrm{SRL}} = L(r^p,\, r)\,, \tag{20}$$

where $L(\cdot)$ denotes the L_1 distance. By providing a lightweight semantic constraint on object shape, SRL helps the model converge faster and improves the stability of 3D box estimation.
Points-to-Box Alignment Loss (PAL) exploits the spatial relationship between the predicted box and the In-Box-Points to impose geometric supervision on predicted 3D boxes in the absence of full annotations. Since a valid 3D box should enclose the corresponding foreground points in the BEV space, PAL first computes the distances from each point to the four edges of the predicted BEV box. Let (l, w) be the predicted length and width, and let $(d^1_i, d^2_i, d^3_i, d^4_i)$ denote the distances from point i to the left, right, top, and bottom edges, respectively. A soft constraint encourages points to lie inside the predicted box through a ReLU activation $\phi(\cdot)$, yielding:

$$\mathcal{L}_{\mathrm{PAL}_1} = \sum_{i=1}^{N} \left(\sum_{j\in\{1,2\}} \phi(d^j_i - \tfrac{l}{2}) + \sum_{k\in\{3,4\}} \phi(d^k_i - \tfrac{w}{2}) \right). \tag{21}$$

However, RGB-D and LiDAR observations often capture only one visible side of an object, causing points to cluster around box boundaries. To leverage this property for implicit supervision, PAL further minimizes the shortest edge-wise distance for each point, producing a tighter alignment:

$$\mathcal{L}_{\mathrm{PAL}_2} = \sum_{i=1}^{N} \min\left(d^1_i,\, d^2_i,\, d^3_i,\, d^4_i\right). \tag{22}$$

Together, $\mathcal{L}_{\mathrm{PAL}_1}$ and $\mathcal{L}_{\mathrm{PAL}_2}$ constrain the predicted BEV box to geometrically align with the foreground point distribution, providing effective supervision for learning accurate 3D box dimensions and positions.
The overall training objective combines the proposed geometric, semantic, and alignment constraints with standard detection losses. Specifically, the final loss is defined as:

$$\begin{aligned} \mathcal{L} = {} & \lambda_1 \mathcal{L}_{\mathrm{BPL}} + \lambda_2 \mathcal{L}_{\mathrm{SRL}} + \lambda_3 (\mathcal{L}_{\mathrm{PAL}_1} + \mathcal{L}_{\mathrm{PAL}_2}) \\ & + \lambda_4 \mathcal{L}_{\mathrm{score}} + \lambda_5 \mathcal{L}_{\mathrm{cls}}, \end{aligned} \tag{23}$$

where $\lambda_{1\text{–}5}$ are balancing weights. $\mathcal{L}_{\mathrm{score}}$ denotes the objectness heatmap regression loss used in CenterPoint and the centerness loss in FCAF3D, while $\mathcal{L}_{\mathrm{cls}}$ is the cross-entropy loss for classification. The predicted 3D boxes are subsequently treated as pseudo labels to train the final 3D detector PGD [42] in a fully supervised manner.

10. Implementation Details

WeakM3D optimized using the Adam optimizer [20] with an initial learning rate of 10^{-4}. The network is trained for

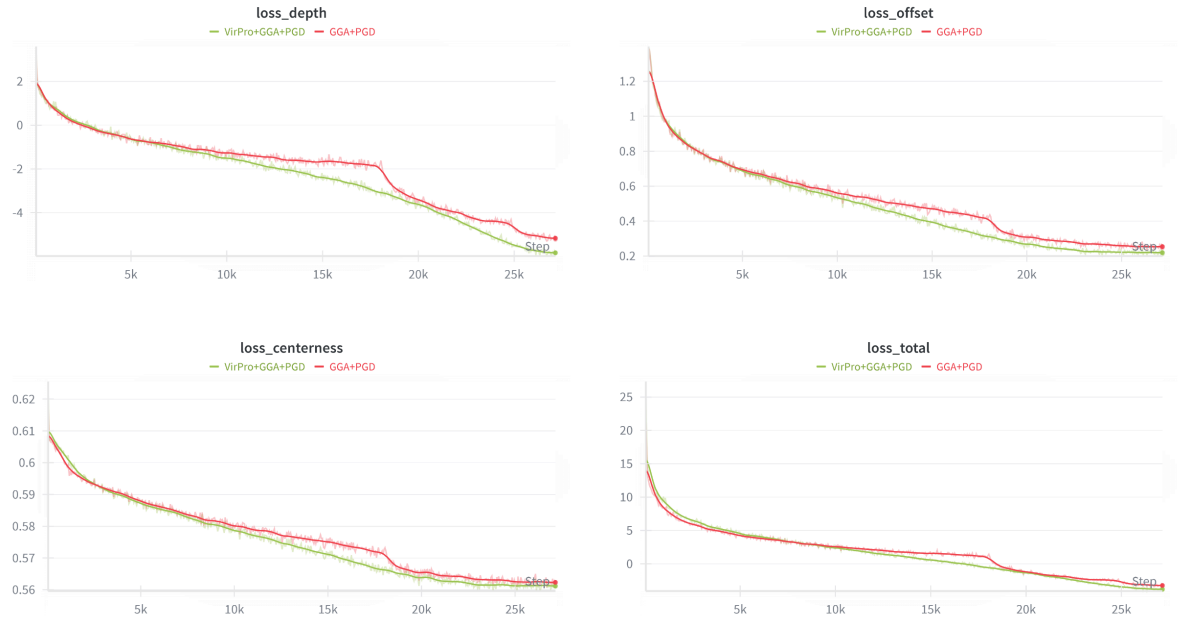

Figure 6. **Training loss comparison between VirPro+GGA+PGD and GGA+PGD**. The depth loss supervises the predicted 3D depth of the object to ensure accurate distance estimation from the camera. The offset loss constrains the projected 2D center offset. The centerness loss encourages confident predictions near object centers while suppressing noisy peripheral responses. The total loss is the weighted sum of all objectives.

50 epochs. To initialize the object point cloud, WeakM3D adopts an off-the-shelf 2D detector FPN [32]. For car-sized objects, the frozen dimensions are empirically set to height 1.6 m, width 1.8 m, and length 4.0 m. The point density threshold in Eq. 17 is fixed to 0.4. Following the 2D-3D alignment strategy of Brazil and Liu [2], the y-coordinate adjustment is applied to improve geometric consistency. The image backbone is ResNet34 [14].

GGA adopts CenterPoint [54] as the backbone networks. The framework is implemented in MMDetection3D [10] and optimized using the AdamW optimizer. The RANSAC thresholds for plane fitting are set to 0.2. Following the configuration of CenterPoint on KITTI, GGA omits $\mathcal{L}_{\text{cls}}$ and assigns $\lambda_{1-4} = 0.3,\ 0.1,\ 0.1,\ 5$. The framework is trained for 120 epochs. The image backbone of PGD [42] is the ResNet101 [14].

11. Latent Space Evaluation Metrics

To qualitatively assess the impact of **VirPro** on the structure of latent embeddings, we perform a clustering-based analysis of RoI visual embeddings produced by our model and CAW3D [24], a deep semantic supervision work with hand-crafted prompts, after stage 1 training. Specifically, we extract all RoI features from the validation set and treat each as an individual point, grouped by its originating scene. We adopt two standard clustering metrics: the **Calinski–Harabasz (CH)** index and the **average Silhouette Score** ($\bar{s}$). The CH index measures the ratio of between-cluster dispersion to within-cluster dispersion, with higher values indicating better cluster separation and compactness. It is defined as:

$$\text{CH} = \frac{\text{Tr}(B_k)}{\text{Tr}(W_k)} \cdot \frac{n-k}{k-1} \tag{24}$$

where $\text{Tr}(B_k)$ and $\text{Tr}(W_k)$ denote the between-cluster and within-cluster dispersion, respectively; n is the number of samples and k is the number of clusters.

The Silhouette Score $\bar{s}$ evaluates the consistency within clusters by comparing the intra-cluster distance $a(i)$ and the nearest-cluster distance $b(i)$ for each sample i:

$$s(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))} \qquad \bar{s} = \frac{1}{n}\sum_{i=1}^{n} s(i), \tag{25}$$

where $s(i)$ is the silhouette coefficient of sample i, $a(i)$ is the average distance to all other points in the same cluster, and $b(i)$ is the average distance to points in the nearest neighboring cluster.

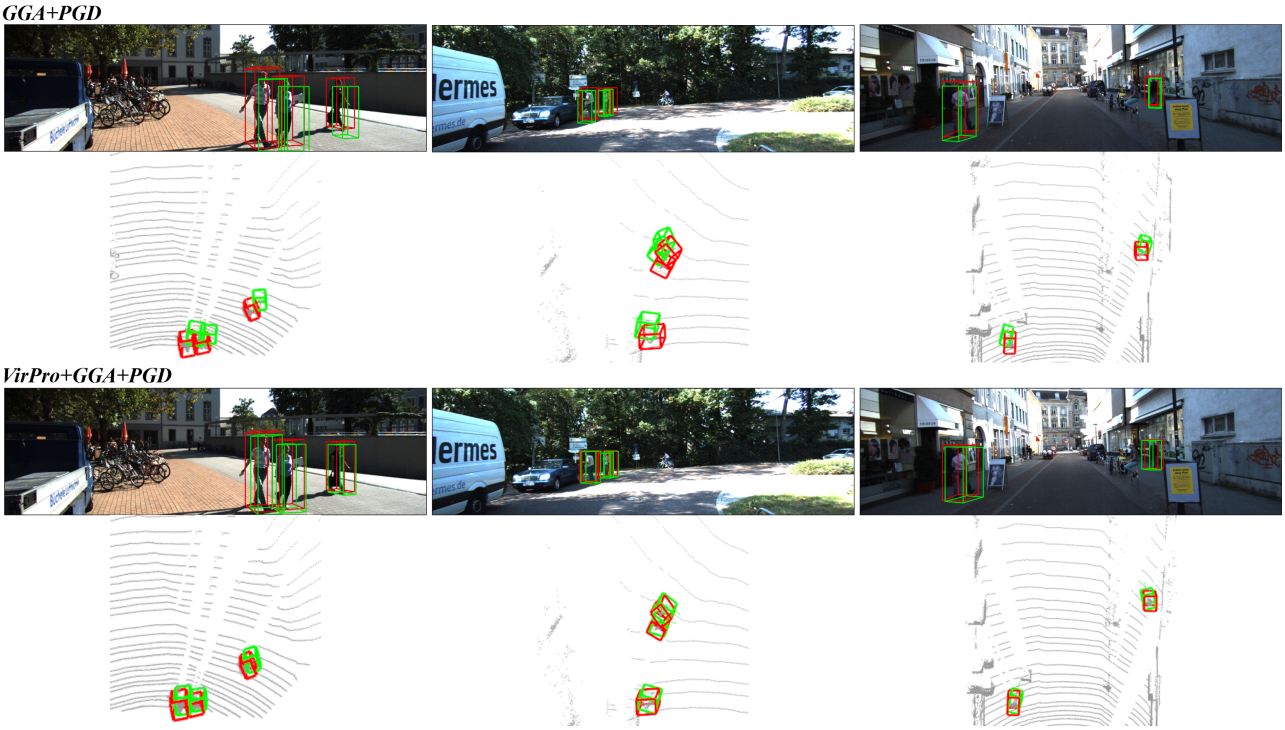

Figure 7. **Qualitative results on the KITTI *validation* set comparing ours on the "Pedestrian" category.** Predicted boxes are rendered in green, and ground-truth boxes are shown in red.

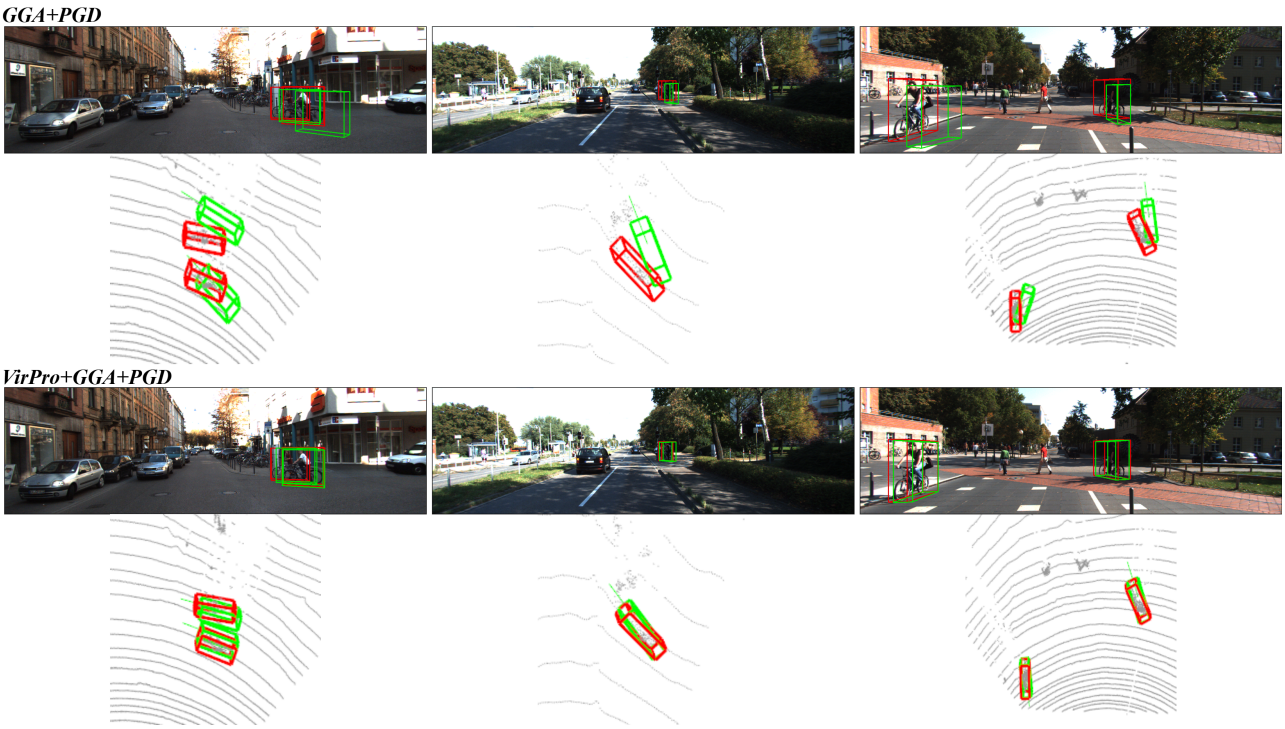

Figure 8. **Qualitative results on the KITTI *validation* set comparing ours on the "Cyclsit" category.** Predicted boxes are rendered in green, and ground-truth boxes are shown in red.

Table 7. **Comparison on the KITTI *validation* set (Pedestrian category).** We report validation performance using the AP_{40} at an IoU threshold of 0.5. The best results are highlighted in red.

Pedestrian Validation	AP_{BEV}			AP_{3D}		
	Easy	*Mod*	*Hard*	*Easy*	*Mod*	*Hard*
GGA+PGD [42, 55]	3.94	2.94	2.45	2.48	1.86	1.33
VirPro+GGA+PGD	5.19	4.11	3.15	3.38	2.52	1.91

Table 8. **Comparison on the KITTI *validation* set (Cyclist category).** We report validation performance using the AP_{40} at an IoU threshold of 0.5. The best results are highlighted in red.

Cyclist Validation	AP_{BEV}			AP_{3D}		
	Easy	*Mod*	*Hard*	*Easy*	*Mod*	*Hard*
GGA+PGD [42, 55]	2.32	1.06	0.73	2.23	0.77	0.73
VirPro+GGA+PGD	4.13	2.03	1.67	3.11	1.86	1.51

Table 9. **Comparison on the KITTI *test* set (Pedestrian category).** GGA+PGD is the baseline method using weak 2D-3D alignment and textual prompts generated from LLM for weak supervision. The best results are highlighted in red.

Pedestrian Test	AP_{BEV}			AP_{3D}		
	Easy	*Mod*	*Hard*	*Easy*	*Mod*	*Hard*
GGA+PGD [42, 55]	0.87	0.61	0.46	0.57	0.36	0.27
VirPro+GGA+PGD	1.59	1.09	0.91	1.07	0.70	0.51

Table 10. **Comparison on the KITTI *test* set (Cyclist category).** GGA+PGD is the baseline method using weak 2D-3D alignment and textual prompts generated from LLM for weak supervision. The best results are highlighted in red.

Cyclist Test	AP_{BEV}			AP_{3D}		
	Easy	*Mod*	*Hard*	*Easy*	*Mod*	*Hard*
GGA+PGD [42, 55]	0.84	0.28	0.30	0.69	0.25	0.28
VirPro+GGA+PGD	1.29	0.52	0.43	1.08	0.45	0.49

Table 11. **Performances on nuScenes *val* set of "Car" class.**

Method	AP↑	ATE↓	ASE↓	AAE↓
WeakM3D [31]	0.214	0.814	0.234	0.682
SKD-WM3D [16]	0.242	0.795	0.231	0.659
GGA+PGD [55]	0.227	0.661	0.173	0.238
Virpro+GGA+PGD	0.239	0.626	0.158	0.206

12. Quantitative Results

Tabs. 9, 7, 10, and 8 present that consistent with the trends observed on the "Car" category, our VirPro pretraining paradigm delivers clear and steady improvements over the baseline GGA+PGD [42, 55] on both "Pedestrian" and "Cyclist" categories under all difficulty levels. These results demonstrate that the proposed visually referred probabilistic prompts provide contextual and more informative supervisory signals, enabling stronger 3D localization and shape estimation for both Pedestrian and Cyclist instances as well. In addition, we evaluate our method on nuScenes dataset. As shown in Tab. 11, VirPro yields consistent gains across nuScenes' diverse scenes which demonstrates improved robustness beyond KITTI. We follow SKD-WM3D [16] to evaluate "Car" on validation set, since test is unreported. We train and validate our model on "CAM_FRONT" split.

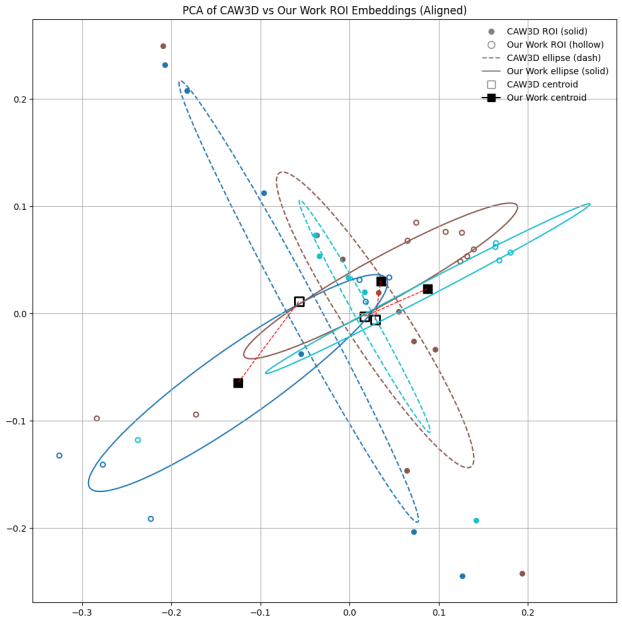

Figure 9. **PCA Visualization of RoI Embeddings from CAW3D and Our Proposed VirPro.** We compare the RoI embeddings distribution projected via PCA from CAW3D and our proposed *VirPro*, where our work exhibits better-separated clusters across scenes, indicating stronger scene discrimination and improved latent space structuring.

Table 12. **Ablations on Gaussian Sampling of Prompts.** G.S. denotes Gaussian Sampling. Both prompts are generated by the same prompt bank and visual conditioning.

Gaussian sampling	AP_{BEV}			AP_{3D}		
	Easy	*Mod*	*Hard*	*Easy*	*Mod*	*Hard*
Prompts w/o G.S.	57.36	40.39	36.76	52.17	37.86	31.49
Prompts w/ G.S.	60.11	42.95	37.50	54.72	39.49	33.32

Table 13. **Ablations on the Quality of 2D Annotation.**

2D GT Annotation	AP_{BEV}			AP_{3D}		
	Easy	*Mod*	*Hard*	*Easy*	*Mod*	*Hard*
w/o 2D GT finetune	59.31	43.32	37.17	52.62	38.65	33.51
w/ 2D GT finetune	60.11	42.95	37.50	54.72	39.49	33.32

13. Qualitative Results

3D Visualizations for Cyclist and Pedestrian. We provide qualitative 3D visualizations on the KITTI *validation* set for the Pedestrian and Cyclist categories. As illustrated in Fig. 7 and 8, VirPro+GGA+PGD generates noticeably more accurate and spatially coherent 3D bounding boxes than the GGA+PGD baseline. Across diverse urban scenes, our predictions exhibit improved scale estimation, orientation stability, and depth reasoning, yielding tighter alignment between predicted boxes (green) and ground-truth annotations (red). The gains are particularly clear for small, heavily occluded, and cluttered instances, highlighting the effectiveness of visually enriched probabilistic prompts.

Latent Space Distribution. As show in Fig. 9, PCA on RoI embeddings from three randomly selected scenes in KITTI validation split shows more clearly separated clusters under VirPro. Specifically, the CAW3D embeddings (solid markers) form highly overlapping clusters, with large covariance ellipses indicating weak scene discrimination and significant intra-scene variation. In contrast, the RoI embeddings generated by VirPro (hollow markers) exhibit sharper and well-separated clusters across scenes. Moreover, the centroids of VirPro embeddings align more distinctly between scenes, suggesting improved inter-scene separability and a more structured latent space. The result verifies that visually referred probabilistic prompts yield a more structured and semantically discriminative latent space.

Train Loss Curve. Fig. 6 shows that VirPro demonstrates a noticeably smoother and more stable optimization trajectory, with reduced oscillation and consistently lower loss values throughout pretraining. This reflects VirPro's strong guidance on geometric reasoning and convergence behavior, enabling a more stable and steady learning compared to the baseline. This improvement is mainly attributed to that our proposed VirPro encourages smoother modality alignment. Therefore, the model receives soft, probabilistic guidance rather than rigid supervision, which leads to steady convergence and mitigates training noise.

14. Ablation Experiments

Quality of 2D Annotation VirPro is designed to be robust because visually injected probabilistic prompts capture cross-scene diversity and uncertainty for each RoI, reducing sensitivity to imperfect 2D boxes. Empirically, Tab. 13 shows additional fine-tuning with 2D GT after Stage 1 yields only moderate effects.

Effectiveness of Gaussian Sampling. We added a controlled ablation on the KITTI benchmark by removing Gaussian sampling. Tab. 12 validates the Gaussian sampling with consistent gains across BEV and 3D.