VWS-PFE: MODELING VERTICALLY WEAKLY ORDERED STRUCTURES FOR PILLAR-BASED 3D OBJECT DETECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Pillar-based LiDAR detectors compress the points within each vertical column into a compact feature for efficient bird's-eye-view detection. Conventional Pillar Feature Network (PFN) encoders apply shared point-wise mappings and symmetric pooling, leaving cross-point vertical relations implicit. Gravity, ground constraints, object geometry, and LiDAR sampling provide a natural organizing direction, while the observed returns remain sparse and irregularly spaced. We introduce the Vertical Weakly-Ordered Structure-Aware Pillar Feature Encoder (VWS-PFE), which realizes height-induced weak ordering through a sorted traversal and relative height-rank descriptors. Context extracted from these features is restored to each source point and reused for feature recalibration before the shared PFN projection and aggregation after projection. This connects how points are represented with how they contribute to the pillar feature, without height bins or auxiliary intra-pillar grids. With downstream detectors fixed, VWS-PFE improves mean moderate 3D AP_{R40} on KITTI from 58.66 to 61.86. On nuScenes, it improves PointPillars by 4.27 NDS and 5.22 mAP, and CenterPoint-Pillar by 0.90 NDS and 1.40 mAP. KITTI ablations show benefits from contextual encoding and a further 1.43-point gain from ascending-height over random traversal. Conv1D, Transformer, and Mamba provide different accuracy–latency trade-offs within the same encoder design.

1 INTRODUCTION

Pillar-based 3D object detectors compress all points in a vertical column into one feature before spatial reasoning in bird's-eye view (BEV). This enables efficient two-dimensional backbones, but makes the point-to-pillar encoder responsible for representing vertical geometry before compression. In PointPillars (Lang et al., 2019), a Pillar Feature Network (PFN) decorates points with coordinates and offsets, applies shared point-wise transformations, and takes a channel-wise maximum. The encoder therefore determines which intra-pillar relationships are explicitly available to subsequent BEV processing.

Height provides a physically meaningful relation among these points. Gravity-aligned coordinates, ground constraints, object geometry, and LiDAR sampling produce recurring vertical patterns. Max pooling selects the strongest response in each channel; other responses can contribute through interactions learned before pooling (Figure 1). These interactions help represent vertically distributed returns from pedestrians, cyclists, and other road users. LiDAR sampling leaves unequal height gaps and incomplete surface coverage. We describe the observed points as a *vertically weakly ordered structure*: height organizes the returns from bottom to top, while relative positions and spacing describe the geometry of this sparse traversal.

Our objective is to make vertical relations available to both feature learning and aggregation at this compression stage. PointNet (Qi et al., 2017) and Deep Sets (Zaheer et al., 2017) provide effective permutation-invariant representations, and conventional PFNs already include height coordinates and offsets. A height-organized context can connect how each point is transformed with how the transformed points are combined into one feature.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

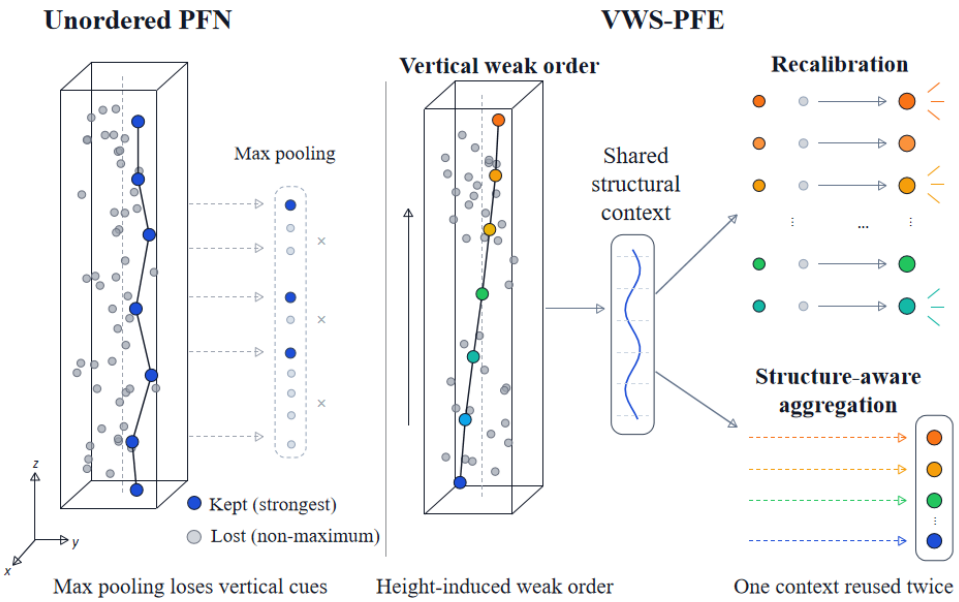

Figure 1: **Vertical context before pillar compression.** Conventional PFNs aggregate shared point-wise features by channel-wise max pooling. VWS-PFE extracts context from the height-organized points and uses it for feature recalibration and weighted aggregation, giving non-maximum responses an additional path into the pillar feature.

We propose VWS-PFE, a PFN-compatible encoder that uses one vertical context to guide both feature formation and aggregation. Relative Height-Rank Encoding (RHRE) describes relative height, rank, adjacent gaps, and occupied height span. A compact context operator processes the height-sorted point features, and an inverse permutation restores each context vector to its source point. The context recalibrates input channels before the shared PFN projection and supplies weights for a complementary aggregation branch after projection. The resulting pillar feature combines channel-wise maxima with a weighted summary of the projected points. The BEV scatter, backbone, head, and detection objective retain their existing interfaces.

This design differs from preserving height through discrete histograms (Zhou et al., 2025) or spatio-temporal virtual grids (Park et al., 2024). It operates on the observed points and reuses a common relational signal across feature formation and compression. Conv1D, Transformer (Vaswani et al., 2017), and Mamba (Gu & Dao, 2023) instantiate the same context interface, allowing us to examine the representation principle separately from a particular operator.

VWS-PFE improves aggregate accuracy in KITTI PointPillars and nuScenes PointPillars and CenterPoint-Pillar. On KITTI, the Mamba instantiation gains 3.20 mean moderate AP_{R40} points. A random-traversal variant with height descriptors already gains 1.77 points; ascending height adds 1.43 points. The results identify benefits from contextual encoding and from physically organizing its traversal. Across the three operators, added parameters remain below 0.74% of the KITTI detector, with detector latency increasing from 5.68 to 8.44–13.37 ms.

Our contributions are:

- We introduce a height-organized pillar representation that combines ordinal position and continuous vertical spacing to contextualize irregularly sampled returns before compression.
- We develop a PFN-compatible encoder in which one point-aligned structural context guides both feature recalibration and aggregation, without height bins or auxiliary intra-pillar grids.
- Experiments with PointPillars and CenterPoint-Pillar show accuracy gains; KITTI ablations identify the additional value of height traversal and context-derived pooling.

2 Related Work

Set representation and geometric context. PointNet (Qi et al., 2017) uses shared transformations and symmetric pooling, while Deep Sets (Zaheer et al., 2017) studies permutation-invariant representations more generally. GeoFormer (Jin et al., 2025) introduces intra-voxel geometric relations through a graph-based Transformer. VWS-PFE exploits the vertical organization of points within a pillar and reuses the resulting context during both feature formation and compression.

Pillar architectures and attention. PointPillars (Lang et al., 2019) learns pillar features for BEV detection, and CenterPoint (Yin et al., 2021) provides a center-based detection framework. PillarNet (Shi et al., 2022) and PillarNeXt (Li et al., 2023) advance pillar-based detection through encoder and downstream network design. TANet (Liu et al., 2020) uses point-, channel-, and voxel-wise attention. SwiftPillars (Jin et al., 2024) combines point/channel attention in its encoder with a multi-scale decoder, while AA-Pillars (Jiang et al., 2026) couples an agent-attention PFN with a depthwise-separable network for pseudo-image processing. VWS-PFE derives both channel recalibration and pooling weights from the same height-organized point context before BEV processing.

Height-aware representations. PillarHist (Zhou et al., 2025) encodes height-aware histograms, and FG-PFE (Park et al., 2024) constructs spatio-temporal virtual grids. SVP (Ao et al., 2026) extracts stratified vertical features in a parallel branch and fuses them with horizontal features at multiple scales. These methods provide distinct ways to preserve vertical information. VWS-PFE operates directly on the observed points: relative ranks and continuous height gaps organize a point-aligned context, which guides both recalibration and aggregation. The representation carries vertical relationships through the PFN without height bins or auxiliary intra-pillar grids.

Point-cloud ordering and serialization. FlatFormer (Liu et al., 2023) uses window-based sorting to form equal-sized attention groups, and Point Transformer V3 (Wu et al., 2024) constructs serialized neighborhoods with space-filling curves. PointMamba (Liang et al., 2024) uses space-filling curves for point tokenization, while Point Cloud Mamba (Zhang et al., 2025) introduces six coordinate-order variants through Consistent Traverse Serialization. These approaches organize point-cloud backbone computation. VWS-PFE applies height-induced organization at the point-to-pillar compression stage. Rank and irregular vertical spacing describe the local geometry, and the same point-aligned context participates in the operations on both sides of the shared PFN projection.

3 Height-Organized Pillar Representation

3.1 Problem formulation and meaning of weak order

Consider a non-empty pillar containing N valid points and at most T entries after padding. Let $X \in \mathbb{R}^{T\times C}$ denote the usual PFN inputs, $z \in \mathbb{R}^T$ their heights, and $m \in \{0,1\}^T$ the validity mask. PFN inputs include raw point attributes and offsets from the point mean and pillar center. A conventional encoding has the form

$$p_{\mathrm{PFN}} = \max_{i:m_i=1} \psi(x_i), \tag{1}$$

where ψ is a shared point-wise projection and the maximum is channel-wise. The representation is invariant to the enumeration of valid points, but cross-point interaction at the projection stage is limited to the information already present in the decorated features.

Height induces the weak order $i \preceq_z j$ when $z_i \le z_j$, with equal-height points sharing a level. We realize this relation computationally by sorting valid points into a bottom-up traversal; the sorting implementation orders ties. RHRE augments this traversal with ordinal position, continuous relative height, and neighboring gaps. Consecutive ranks can therefore describe unequal height intervals, and the observed points can cover only part of a surface. Weak ordering specifies the geometric organization, while the sorted features and descriptors are the inputs processed by the context operator.

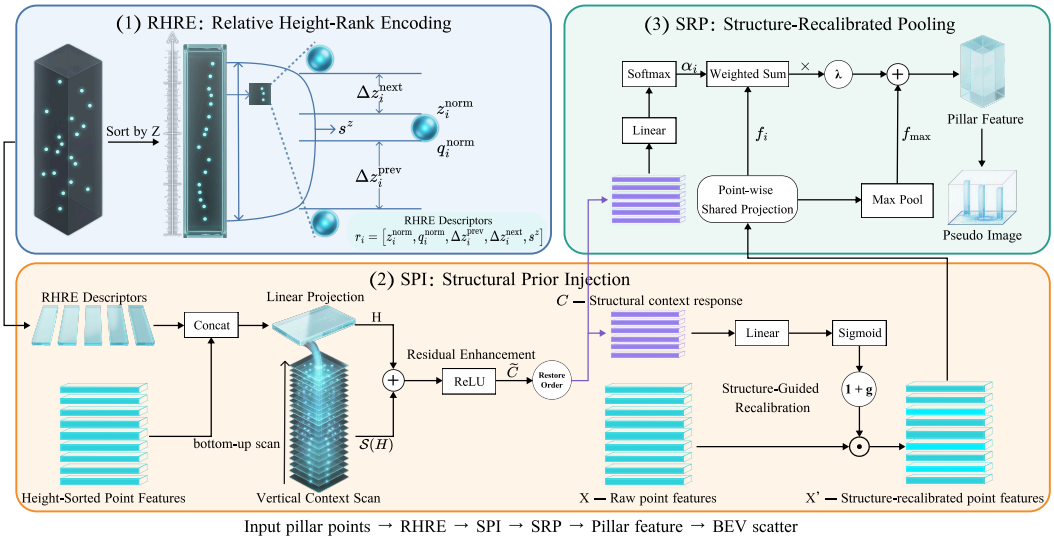

Figure 2: **One vertical context, two uses.** RHRE augments height-sorted points; Vertical Context Scan (VCS) extracts a point-level context. After inverse permutation, Structure-Guided Recalibration (SGR) uses it before the shared PFN projection, and Structure-Recalibrated Pooling (SRP) uses it to weight projected features. The weighted branch complements max pooling. Only the point-to-pillar encoder is redesigned.

Figure 2 summarizes the resulting encoder. RHRE assigns a vertical identity to each point; Structural Prior Injection (SPI) extracts and injects point-aligned context; Structure-Recalibrated Pooling (SRP) reuses that context during compression. Together, these stages carry height-induced structure through the full point-to-pillar encoding process.

3.2 Relative Height-Rank Encoding

Let π sort valid points by ascending height, with padded entries placed last by assigning them height $+\infty$ for sorting only. Write $\widetilde{z}_i = z_{\pi(i)}$ and $\widetilde{x}_i = x_{\pi(i)}$. For the valid points, define

$$\delta = \widetilde{z}_N - \widetilde{z}_1, \qquad s = \max(\delta, 10^{-6}), \qquad q_i = \frac{i-1}{\max(N-1, 1)}. \tag{2}$$

The five-dimensional descriptor is

$$r_i = \left[\frac{\widetilde{z}_i - \widetilde{z}_1}{s},\ q_i,\ \frac{d_i^-}{s},\ \frac{d_i^+}{s},\ \frac{\delta}{H_{\text{range}}} \right], \tag{3}$$

where $d_i^- = \widetilde{z}_i - \widetilde{z}_{i-1}$ for $i > 1$ and zero otherwise, and $d_i^+ = \widetilde{z}_{i+1} - \widetilde{z}_i$ for $i < N$ and zero otherwise. H_{range} is the configured vertical extent, 4 m for KITTI and 8 m for nuScenes. Padded descriptors are zero.

Relative height describes position within the observed span, whereas rank describes position within the observed population. Their combination provides complementary geometric and ordinal information under irregular sampling. The two gap terms encode local vertical spacing, and the final term retains the occupied span relative to the configured vertical extent. Thus, RHRE provides each valid point with a compact vertical identity that is richer than its raw height alone. The denominator clamp and boundary conventions keep the descriptor well-defined for singleton and zero-span pillars.

3.3 CONSTRUCTING AND INJECTING A SHARED CONTEXT

We project each sorted point together with its RHRE descriptor into a context space of dimension $D = 64$:

$$H = \phi([\widetilde{X}; R]) \odot \widetilde{m}, \tag{4}$$

$$\widetilde{C} = \mathrm{ReLU}\left(H + \mathcal{S}(H)\right) \odot \widetilde{m}, \qquad C = \mathcal{G}_{\pi^{-1}}(\widetilde{C}) \odot m. \tag{5}$$

Here ϕ is a bias-free linear projection, $\mathcal{G}$ denotes gathering, and masks broadcast across channels. $\mathcal{S}$ includes the contextual block and its normalization; Appendix A specifies the instantiations. Inverse gathering associates each context vector c_i with its source point x_i before recalibration and aggregation.

The context first drives Structure-Guided Recalibration:

$$x'_i = x_i \odot (1 + \sigma(W_g c_i + b_g))\, m_i. \tag{6}$$

The bounded multiplier in $(1, 2)$ provides context-dependent multiplicative enhancement, with a separate amplification factor for each input channel. Context extraction and recalibration together form SPI. Each point retains its correspondence and its own feature vector, allowing the original shared PFN projection to learn discriminative responses from structurally enhanced inputs.

3.4 REUSING CONTEXT DURING AGGREGATION

Let $f_i = \mathrm{PFN}_{\mathrm{pw}}(x'_i)$, where the shared point-wise stage uses Linear–BatchNorm–ReLU. We retain its salient-response path, $p_{\max} = \max_{i:m_i=1} f_i$, and add a context-weighted summary:

$$\alpha_i = \frac{\exp(w_\alpha^\top c_i + b_\alpha)}{\sum_{j:m_j=1} \exp(w_\alpha^\top c_j + b_\alpha)}, \quad p_{\mathrm{str}} = \sum_{i:m_i=1} \alpha_i f_i, \quad p = p_{\max} + \lambda p_{\mathrm{str}}. \tag{7}$$

Invalid entries are excluded from the softmax and reductions. The learned scalar λ is initialized to zero, so aggregation initially uses max pooling over the recalibrated point features; training learns the contribution of the weighted branch. This residual fusion retains salient maximum responses while providing a complementary path for vertically distributed cues.

SRP separates the features being aggregated from the representation used to predict their weights. The projected feature f_i carries the point response after context-guided recalibration, while c_i supplies the structural signal for the pooling weight. Using c_i at both stages links the channel-wise enhancement of a point to its contribution during compression.

3.5 CONTEXT OPERATORS AND COMPUTATIONAL SCOPE

Context operators. The context interface supports Conv1D, Transformer (Vaswani et al., 2017), and Mamba (Gu & Dao, 2023). Conv1D captures local vertical continuity through neighboring points in height order. Transformer models pairwise dependencies through self-attention, with RHRE supplying the vertical positional description. Mamba propagates selective context in one causal scan from lower to higher points. Thus, Transformer receives vertical structure through RHRE, while Conv1D and causal Mamba additionally exploit the explicit sorted traversal. All three operate independently within each pillar and share the same RHRE–SPI–SRP framework. Their exact configurations are given in Appendix A.

Complexity and integration. For P pillars with at most T points, sorting costs $O(PT \log T)$. With channel widths fixed, descriptor construction, gathering, gating, and pooling are linear in PT. Conv1D and Mamba have linear dependence on T, while Transformer attention has quadratic dependence. The principal context tensor has shape $P \times T \times 64$, with $T \leq 32$ on KITTI and $T \leq 20$ on nuScenes. Context modeling is therefore confined to short intra-pillar lists. The output remains one 64-dimensional feature per pillar, preserving the BEV scatter interface and avoiding auxiliary vertical-grid tensors or detection objectives.

4 EXPERIMENTS

4.1 EVALUATION PROTOCOL

Datasets and metrics. KITTI (Geiger et al., 2012) uses the standard 3,712/3,769 training/validation split. We report moderate 3D AP_{R40} at IoU 0.7 for Car and 0.5 for Pedestrian and Cyclist. Mean Mod. is the arithmetic mean over these three classes. nuScenes (Caesar et al., 2020) uses the official train/validation split, ten LiDAR sweeps, and its standard NDS and mAP metrics. All AP and NDS numbers are expressed on a 0–100 scale; differences are percentage points. We use validation results throughout the main comparisons.

Implementation details. Our implementation is based on OpenPCDet (OpenPCDet Development Team, 2020), and all experiments use a single NVIDIA GeForce RTX 4090 GPU. KITTI models are trained for 80 epochs with batch size 4. The nuScenes multi-head PointPillars models are trained for 20 epochs with batch size 2, and the CenterPoint comparison follows the training and evaluation protocol of its corresponding CenterPoint-Pillar baseline. Unless otherwise specified, we use Mamba as the context operator. Appendix C provides the preprocessing, optimization, and timing settings.

Controlled PFN replacement. The primary comparison measures the accuracy and latency change from replacing the PFN within a fixed detector configuration. For each baseline–replacement pair, preprocessing, pillarization, BEV backbone, detection head, training schedule, and post-processing are held fixed. This attributes the measured change to point-to-pillar encoding. Detector inference latency is measured on the same GPU with batch size 1, using 50 warm-up and 500 measured iterations of the detector call, including detector-side post-processing.

4.2 DETECTION ACCURACY AND COMPUTATIONAL TRADE-OFFS

Table 1: **KITTI validation: replacing the PFN.** Moderate 3D AP_{R40}, full-detector parameters, and detector inference latency on an RTX 4090. All context variants use $D = 64$. Best accuracy in each column is bold.

Encoder	Params (M)	Latency (ms)	Car	Ped.	Cyc.	Mean Mod.
PointPillars PFN	4.8349	5.68	75.24	41.14	59.61	58.66
VWS-PFE–Conv1D	4.8492	8.44	75.67	42.74	59.72	59.38
VWS-PFE–Transformer	4.8703	12.85	75.66	44.58	**62.71**	60.98
VWS-PFE–Mamba	4.8695	13.37	**77.19**	**45.83**	62.55	**61.86**

KITTI results. Table 1 compares the three VWS-PFE instantiations under the same PointPillars detector. All context operators improve Mean Mod., demonstrating the effectiveness of incorporating vertical structure into the PFN. Mamba reaches 61.86, a 3.20-point gain over the baseline, with improvements of 1.95, 4.69, and 2.94 points for Car, Pedestrian, and Cyclist. The largest gain is on Pedestrian, where moderate AP increases from 41.14 to 45.83. This result highlights the benefit of contextualizing sparse, vertically distributed returns before pillar compression.

The operators offer distinct accuracy–latency trade-offs. Conv1D gains 0.72 Mean Mod. points at 8.44 ms, providing the fastest VWS-PFE configuration. Transformer and Mamba gain 2.32 and 3.20 points at 12.85 and 13.37 ms, respectively, compared with 5.68 ms for the baseline. Mamba is the accuracy-oriented option: it adds 7.69 ms and achieves the highest Mean Mod., while Transformer gives the highest Cyclist AP. Relative to Transformer, Mamba gains 0.88 Mean Mod. points for an additional 0.52 ms. Parameter increments are 14,348, 35,404, and 34,572, each below 0.74% of the detector.

Table 2: **nuScenes validation: controlled PFN replacement.** VWS-PFE improves both detector settings while preserving their downstream architectures. Gains are computed within each pair; per-class AP is given in Appendix D.

Detector	NDS	mAP	ΔNDS	ΔmAP	Latency (ms)
PointPillars	55.16	41.06	–	–	17.56
+ VWS-PFE	59.43	46.28	+4.27	+5.22	23.20
CenterPoint-Pillar	60.70	50.03	–	–	21.75
+ VWS-PFE	61.60	51.43	+0.90	+1.40	23.41

nuScenes results across detector configurations. Table 2 evaluates VWS-PFE with two nuScenes detectors. Replacing the PointPillars PFN increases NDS from 55.16 to 59.43 and mAP from 41.06 to 46.28. AP improves for all ten classes, including Bus (+8.67), Motorcycle (+7.82), Truck (+6.14), and Pedestrian (+5.20), covering both large vehicles and smaller road users. On the stronger CenterPoint-Pillar baseline, VWS-PFE gains 0.90 NDS and 1.40 mAP, with improvements in eight classes. Bicycle gains 6.56 AP points; Trailer and Construction Vehicle decrease by 1.92 and 1.30 points. Detector inference latency changes from 17.56 to 23.20 ms for PointPillars and from 21.75 to 23.41 ms for CenterPoint-Pillar. Thus, the same encoder improves aggregate accuracy under both downstream detection designs.

The two nuScenes configurations show a baseline-dependent return: PointPillars gains more accuracy, while CenterPoint-Pillar gains 1.40 mAP with 1.66 ms added latency. We next use KITTI ablations to examine the roles of context, traversal, and pooling.

4.3 Ablation analysis

Table 3: **KITTI ablations with Mamba context.** Every row reports moderate AP_{R40}. Each block changes the named component, traversal order, or aggregation weight source. The full configuration is repeated to make within-block comparisons explicit.

Study	Variant	Car	Ped.	Cyc.	Mean Mod.
Components	PFN baseline	75.24	41.14	59.61	58.66
	Without RHRE	76.31	44.78	60.36	60.48
	Without VCS	75.64	42.40	59.64	59.23
	Without SGR	75.77	42.38	64.24	60.80
	Without SRP	75.85	42.44	62.66	60.32
	Full	77.19	45.83	62.55	**61.86**
Traversal order	Ascending x	76.41	41.90	59.80	59.37
	Ascending y	76.40	41.57	61.49	59.82
	Descending height	76.14	42.57	62.25	60.32
	Random	76.34	43.75	61.20	60.43
	Original	75.94	42.78	60.30	59.67
	Ascending height	77.19	45.83	62.55	**61.86**
Pooling weights	Max only	75.85	42.44	62.66	60.32
	Uniform	76.04	43.45	61.75	60.41
	Raw input x_i	76.06	44.42	62.19	60.89
	Projected feature f_i	76.22	42.02	59.37	59.20
	Shared context c_i	77.19	45.83	62.55	**61.86**

Cross-point context and its two uses. Removing VCS reduces Mean Mod. from 61.86 to 59.23, the largest drop in the component study (Table 3). Retaining the descriptors and point-wise projection recovers less of the gain than contextual processing across points. Removing RHRE, SGR, or SRP reduces Mean Mod. by 1.38, 1.06, or 1.54 points, respectively. These comparisons connect the three design choices: vertical descriptors organize the inputs, VCS models cross-point relations, and applying the resulting context in both recalibration and pooling gives the highest mean accuracy.

Ascending height improves on random traversal. All traversal variants include RHRE, Mamba-based VCS, SGR, and SRP. Random traversal reaches 60.43 Mean Mod., exceeding the PFN baseline by 1.77 points with this contextual encoder and height descriptors. Ascending height adds 1.43 points, reaching 61.86; it also exceeds descending height by 1.54 points and ascending x/y by 2.49/2.04 points. These comparisons separate the improvement already present under random traversal from the additional benefit of height organization. In causal Mamba, traversal determines each point's available context. The bottom-up result supports aligning this directional context with height in the evaluated encoder.

Structural context guides effective pooling. The pooling study compares sources for the aggregation weights. Uniform weighting reaches 60.41 Mean Mod., close to max-only pooling at 60.32. Raw-input and projected-feature weights reach 60.89 and 59.20. Weights from the shared context c_i yield 61.86, gaining 1.54 points over max-only pooling and 0.97 points over raw-input weighting. Adding a weighted branch is therefore most effective when its weights come from the structural context. Together with the SGR result, this shows that the same context is useful both for shaping point features and for combining them into a pillar representation.

4.4 Qualitative Results

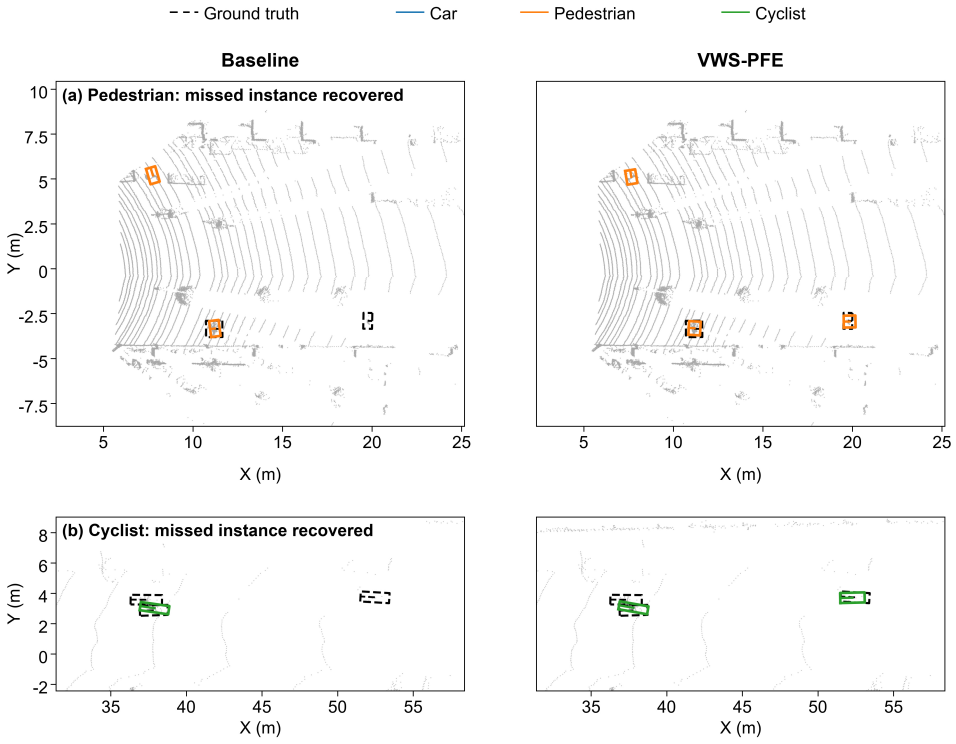

Figure 3: **Detection comparisons on KITTI.** Both models use identical point clouds, a score threshold of 0.35, and the same rendering range for each case. Black dashed boxes denote ground truth; colored boxes denote predictions. Top: VWS-PFE recovers a Pedestrian missed by the baseline in frame 004425. Bottom: VWS-PFE recovers a distant Cyclist missed by the baseline in frame 002404.

Detection under sparse observations. Figure 3 shows a Pedestrian and a distant Cyclist recovered by VWS-PFE but missed by the PointPillars baseline. The comparisons use identical point clouds and display settings. Both objects have sparse returns, and the recovered detections provide examples of the Pedestrian and Cyclist improvements in Table 1.

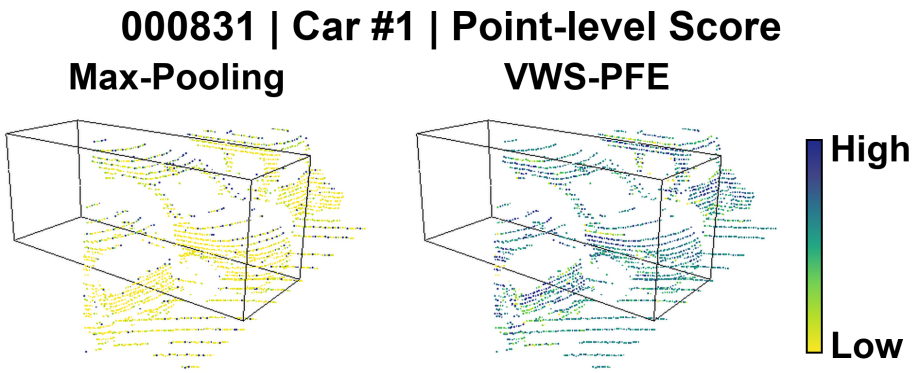

Figure 4: **Point-level aggregation responses on KITTI.** The same Car instance in frame 000831 is shown for both models. Max-pooling response measures the fraction of output channels for which a point supplies the maximum; VWS-PFE response shows the SRP weight relative to uniform weighting, $N\alpha_i$. Colors are normalized for visualization. VWS-PFE distributes aggregation responses across vertically related object points.

Complementary aggregation across vertical structure. Figure 4 shows how points contribute to aggregation for the same Car instance. Max pooling concentrates its channel-wise selections on a small subset of points. SRP assigns weights across vertically related object points, providing a complementary summary alongside these maxima. The weighted branch gives non-maximum responses a direct contribution to the pillar feature, illustrating the aggregation design evaluated in Table 3. Appendix F defines the displayed scores.

5 Discussion

The results identify two useful choices at the point-to-pillar compression stage: learning cross-point context and organizing that context by height. Random traversal already improves on the PFN, and ascending height adds a further gain. Reusing the resulting context for recalibration and pooling then connects point transformation with point contribution. This coupling is the central representation choice in VWS-PFE; Conv1D, Transformer, and Mamba provide different ways to compute the context.

The observed gains span PointPillars and CenterPoint-Pillar, with their magnitude and runtime cost depending on the detector configuration. Conv1D gives the lowest VWS-PFE latency; Transformer and Mamba trade more computation for larger accuracy gains. The geometric prior is suited to gravity-aligned scenes, where height has a consistent physical interpretation. Rank and neighboring gaps describe the observed returns and change with sampling, so mixed surfaces and incomplete observations can weaken local vertical relationships.

6 Conclusion

VWS-PFE uses height-induced weak ordering to couple contextual feature learning with pillar aggregation. Ordinal and metric height descriptors organize the points, and a shared point-aligned context guides both recalibration and pooling. Matched replacements improve PointPillars and CenterPoint-Pillar, while KITTI ablations distinguish the benefit of contextual encoding from the additional gain of ascending-height traversal. These results establish the value of carrying vertical relationships through feature formation and compression before BEV processing.

AI USE STATEMENT

Generative AI tools were used to assist with manuscript organization, language editing, literature search, LaTeX preparation. They were not used to generate experimental data or measurements. The authors reviewed and verified all AI-assisted content and take full responsibility for the paper's text, claims, references, and results.

REPRODUCIBILITY STATEMENT

Section 3 specifies the encoder, including RHRE, context extraction, masking, and aggregation. Appendix A provides tensor shapes and operator configurations; Appendix C gives the preprocessing, augmentation, optimization, and latency measurement settings. Appendix E defines each ablation, and Appendix F describes the visualization protocol. These details specify how VWS-PFE integrates with the baseline detectors and how the experimental comparisons are conducted.

ETHICS STATEMENT

This work uses established public driving benchmarks to study LiDAR representation learning. Improved detection of pedestrians, cyclists, and other road users can contribute to more capable perception systems. Deployment in autonomous systems requires validation under the intended operating conditions and compliance with dataset licenses and applicable safety requirements.

REFERENCES

Lei Ao, Wenkang Wan, Nan Ouyang, Jianzhao Li, Qingqing Li, Maoguo Gong, and Kai Sheng. SVP: Stratified Vertical Priors for LiDAR-Based 3D Object Detection. *Neurocomputing*, 659: 131737, January 2026. doi: 10.1016/j.neucom.2025.131737.

Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuScenes: A Multimodal Dataset for Autonomous Driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? The KITTI vision benchmark suite. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3354–3361. IEEE, 2012. doi: 10.1109/cvpr.2012.6248074.

Albert Gu and Tri Dao. Mamba: Linear-Time Sequence Modeling with Selective State Spaces, 2023. URL https://arxiv.org/abs/2312.00752.

Junzhao Jiang, Honghao Gao, Zijian Zhou, Bofu Wu, Yaoye Wu, Zehao Li, Yichen Wang, and Shuhua Li. AA-Pillars: Pillar-Based 3D object detection with refined pillar features. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, January 2026. doi: 10.1177/09544070251408158.

Xin Jin, Kai Liu, Cong Ma, Ruining Yang, Fei Hui, and Wei Wu. SwiftPillars: High-Efficiency Pillar Encoder for Lidar-Based 3D Detection. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(3):2625–2633, March 2024. doi: 10.1609/aaai.v38i3.28040.

Xin Jin, Haisheng Su, Cong Ma, Kai Liu, Wei Wu, Fei Hui, and Junchi Yan. GeoFormer: Geometry Point Encoder for 3D Object Detection with Graph-based Transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 26879–26889, 2025.

Alex H. Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. PointPillars: Fast Encoders for Object Detection From Point Clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

Jinyu Li, Chenxu Luo, and Xiaodong Yang. PillarNeXt: Rethinking Network Designs for 3D Object Detection in LiDAR Point Clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 17567–17576, 2023.

Dingkang Liang, Xin Zhou, Wei Xu, Xingkui Zhu, Zhikang Zou, Xiaoqing Ye, Xiao Tan, and Xiang Bai. PointMamba: A Simple State Space Model for Point Cloud Analysis. In *Advances in Neural Information Processing Systems*, volume 37, pp. 32653–32677. Curran Associates, Inc., 2024. doi: 10.52202/079017-1026.

Zhe Liu, Xin Zhao, Tengteng Huang, Ruolan Hu, Yu Zhou, and Xiang Bai. TANet: Robust 3D Object Detection from Point Clouds with Triple Attention. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(07):11677–11684, 2020. doi: 10.1609/aaai.v34i07.6837.

Zhijian Liu, Xinyu Yang, Haotian Tang, Shang Yang, and Song Han. FlatFormer: Flattened Window Attention for Efficient Point Cloud Transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1200–1211, June 2023.

OpenPCDet Development Team. OpenPCDet: An Open-Source Toolbox for 3D Object Detection from Point Clouds. `https://github.com/open-mmlab/OpenPCDet`, 2020.

Konyul Park, Yecheol Kim, Junho Koh, Byungwoo Park, and Jun Won Choi. Fine-Grained Pillar Feature Encoding Via Spatio-Temporal Virtual Grid for 3D Object Detection. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4259–4265. IEEE, 2024. doi: 10.1109/icra57147.2024.10611414.

Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

Guangsheng Shi, Ruifeng Li, and Chao Ma. PillarNet: Real-Time and High-Performance Pillar-Based 3D Object Detection. In *Computer Vision – ECCV 2022*, pp. 35–52. Springer Nature Switzerland, 2022. doi: 10.1007/978-3-031-20080-9_3.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is All you Need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

Xiaoyang Wu, Li Jiang, Peng-Shuai Wang, Zhijian Liu, Xihui Liu, Yu Qiao, Wanli Ouyang, Tong He, and Hengshuang Zhao. Point Transformer V3: Simpler Faster Stronger. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4840–4851, June 2024.

Tianwei Yin, Xingyi Zhou, and Philipp Krahenbuhl. Center-Based 3D Object Detection and Tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11784–11793, 2021.

Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Russ R Salakhutdinov, and Alexander Smola. Deep Sets. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

Tao Zhang, Haobo Yuan, Lu Qi, Jiangning Zhang, Qianyu Zhou, Shunping Ji, Shuicheng Yan, and Xiangtai Li. Point Cloud Mamba: Point Cloud Learning via State Space Model. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(10):10121–10130, April 2025. doi: 10.1609/aaai.v39i10.33098.

Sifan Zhou, Zhihang Yuan, Dawei Yang, Xing Hu, Jian Qian, and Ziyu Zhao. PillarHist: A Quantization-aware Pillar Feature Encoder based on Height-aware Histogram. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 27336–27345, 2025.

A Implementation Details

Tensor interface. For P non-empty pillars, the decorated input has shape $P \times T \times C$ and RHRE has shape $P \times T \times 5$. The sorted and restored contexts have shape $P \times T \times 64$. SGR maps the context to C gates per point. The final point-wise PFN features have width 64, and SRP produces one scalar weight per valid point. The output is a $P \times 64$ tensor with the original pillar coordinates, which the existing scatter maps into the unchanged BEV canvas.

Table 4: Context operator configurations. All variants use one contextual layer and $D = 64$. The residual and output ReLU in Equation 5 surround the contextual transformation.

Operator	Configuration
Conv1D	Kernel size 3, padding 1, no convolution bias; BatchNorm, ReLU, and LayerNorm.
Transformer	Four attention heads; feed-forward width 128; GELU; pre-norm; dropout 0; key-padding mask; no extra positional encoding beyond RHRE.
Mamba	State dimension 16, convolution width 4, expansion factor 2, dropout 0; LayerNorm before the block and a second LayerNorm after the block/dropout; one causal ascending-height scan.

Masking and numerical conventions. Padded points receive $+\infty$ only as the sorting key. Their features and descriptors are zero before contextual modeling, and context outputs are masked again before inverse gathering and reuse. Transformer keys exclude padding. Conv1D and Mamba receive zeroed padded inputs. Valid points precede padding, so a causal Mamba scan does not use padded suffix tokens when processing valid points. All reductions in Equation 7 use the valid-point mask. Empty pillars are not encoded. A denominator clamp of 10^{-6} prevents division by zero in RHRE.

Initialization and information paths. The context input projection has no bias. SGR uses the standard PyTorch linear initialization, and its sigmoid gate produces a multiplier between 1 and 2. Only the scalar SRP coefficient is explicitly zero-initialized. At that point, the output still depends on context through SGR. Once the weighted path contributes, its softmax gives nonnegative point weights summing to one; the learned fusion coefficient is unconstrained. The weighted branch is a convex combination of point features, scaled by λ and added to the maximum response.

Algorithmic outline. For each non-empty pillar: (1) construct the usual PFN features and validity mask; (2) sort valid points by height and build RHRE; (3) project point–descriptor pairs and apply the chosen context operator; (4) restore context to the source-point indexing; (5) amplify input channels using SGR and apply the shared PFN projection; (6) compute masked maximum and context-weighted aggregation; (7) emit their learned sum at the original pillar coordinate. Only steps inside the point-to-pillar mapping change.

B Descriptor and Ordering Properties

For a fixed set of points with distinct heights, height sorting produces the same ordered inputs regardless of their initial enumeration. Inverse gathering restores point correspondence, and symmetric aggregation yields the same pillar feature under deterministic evaluation. Equal-height ties follow the sorting implementation, with no secondary key.

For a fixed point set, vertical translation leaves all five RHRE components unchanged. Relative height and gaps describe geometry within the observed span, rank describes ordinal position, and the final component records occupied span relative to the configured vertical extent. Adding or removing points can change ranks and gaps, reflecting the sampling of the observed structure.

C EXPERIMENTAL PROTOCOL

Table 5: Preprocessing and training settings. All spatial quantities are in meters.

Setting	KITTI	nuScenes
LiDAR sweeps	1	10
Minimum (x, y, z)	$(0, -39.68, -3)$	$(-51.2, -51.2, -5)$
Maximum (x, y, z)	$(69.12, 39.68, 1)$	$(51.2, 51.2, 3)$
Pillar size	$(0.16, 0.16, 4)$	$(0.2, 0.2, 8)$
Maximum points per pillar	32	20
Maximum pillars, train/test	16,000 / 40,000	30,000 / 30,000
Epochs, local training	80	20
Batch size, local training	4	2
Learning rate	0.003	0.001

Data augmentation. The KITTI training recipe uses database sampling with at least five points per sampled object, sample groups of 15 for each of Car, Pedestrian, and Cyclist, and the road-plane option. Additional augmentations are random flipping along the x axis, rotation in $[-0.78539816, 0.78539816]$, and scaling in $[0.95, 1.05]$. The nuScenes recipe uses class-balanced resampling and database sample groups car:2, truck:3, construction vehicle:7, bus:4, trailer:6, barrier:2, motorcycle:6, bicycle:6, pedestrian:2, and traffic cone:2, with at least five points per sampled object. It also uses flipping along both horizontal axes, rotation in $[-0.3925, 0.3925]$, and scaling in $[0.95, 1.05]$.

Optimization. The training recipe uses Adam one-cycle, weight decay 0.01, momentum 0.9, one-cycle momenta $[0.95, 0.85]$, a start fraction of 0.4, division factor 10, no learning-rate warm-up, gradient-norm clipping at 10, and four data-loader workers. Final checkpoints are used for locally trained models: epoch 80 on KITTI and epoch 20 on nuScenes. The context width and operator settings are fixed across the reported datasets and ablations.

Environment and timing. The experiments use one NVIDIA GeForce RTX 4090 with 24 GB memory, PyTorch 2.4.0/CUDA 12.1, OpenPCDet 0.6.0 with rebuilt local CUDA operations, `mamba_ssm` 2.3.0, and `causal-conv1d` 1.6.1. Timing uses evaluation/inference mode, batch size 1, CUDA events, synchronization around each measurement, 50 warm-up iterations, and 500 timed iterations. Reported detector inference latency covers the GPU detector call, including the encoder, scatter, backbone, head, and detector-side post-processing. The measurement excludes data loading and preprocessing outside that call.

Parameter accounting. Full-detector parameter counts are 4,834,888 for PFN, 4,849,236 for Conv1D, 4,870,292 for Transformer, and 4,869,460 for Mamba. The respective increments are 14,348, 35,404, and 34,572 parameters, approximately 0.297%, 0.732%, and 0.715%. All three operators add less than 0.74% to the baseline detector.

D nuSCENES PER-CLASS RESULTS

Table 6: nuScenes validation AP. PP denotes PointPillars, CP denotes CenterPoint-Pillar, and VWS denotes replacing the pillar encoder with VWS-PFE.

Class	PP	PP + VWS	CP	CP + VWS
Car	78.14	81.14	83.14	83.46
Truck	44.84	50.98	50.58	51.88
Bus	55.86	64.53	62.58	63.21
Trailer	30.43	35.44	33.92	32.00
Construction vehicle	8.94	12.89	11.76	10.46
Pedestrian	68.43	73.63	78.79	80.21
Motorcycle	25.69	33.51	45.27	48.45
Bicycle	5.01	8.92	19.30	25.86
Traffic cone	45.12	50.87	57.03	59.41
Barrier	48.11	50.94	57.96	59.37
mAP	41.06	46.28	50.03	51.43
NDS	55.16	59.43	60.70	61.60

E ABLATION CONFIGURATIONS

Components. Without RHRE, points retain ascending-height sorting but the five descriptor channels are omitted. Without VCS, RHRE and its point-wise input projection remain, but the context does not mix points. Without SGR, the original PFN inputs enter the shared projection while context still drives SRP. Without SRP, context still drives SGR but aggregation is masked max pooling. The unchanged modules retain the full-model configuration.

Traversal. The x and y variants sort valid points in ascending horizontal-coordinate order. Descending height reverses the vertical traversal. Random uses a random permutation of valid points. Original follows the voxelization/sample order. All ordering variants retain RHRE, Mamba-based VCS, SGR, and SRP, including the height descriptors in the Random variant. Ascending height is the full VWS-PFE configuration.

Aggregation. Max only removes the weighted branch. Uniform uses $\alpha_i = 1/N$. Raw input predicts scores from the decorated PFN input before SGR. Projected feature predicts scores from the post-projection point features. Shared context uses c_i in Equation 7. All weighted variants retain a learned fusion coefficient initialized to zero.

F QUALITATIVE ANALYSIS PROTOCOL

Detection comparisons. Figure 3 uses KITTI validation frames 004425 and 002404 for the Pedestrian and Cyclist comparisons. Within each case, both models use the same point cloud, rendering range, box styles, and score threshold of 0.35. Black dashed boxes denote ground truth, and colored boxes denote predictions.

Point-level responses. Figure 4 uses the same Car instance from frame 000831 and the same point crop for both models. For max pooling, point response is the fraction of PFN output channels for which that point supplies the maximum. For VWS-PFE, the displayed response is $N\alpha_i$, the SRP weight relative to the uniform weight $1/N$. Scores are normalized for visualization. The two measures describe the spatial distribution of channel-wise selection and structure-weighted aggregation, respectively.