GhostPoint: Self-Supervised Representation Learning by Hallucinating Occluded LiDAR Structure

Mohamed Abdelsamad[1,2⋆], Bin Yang[1,3⋆], Michael Ulrich[1], Miao Zhang[1], Yakov Miron[1], Alexandru Paul Condurache[1,3], and Abhinav Valada[2]

[1] Bosch Center for AI
[2] University of Freiburg
[3] University of Luebeck

Abstract. 3D object detection from LiDAR point clouds is a core problem in autonomous driving. Recent advances in self-supervised learning (SSL) enable scalable pretraining and transfers well to per-point tasks such as semantic and panoptic segmentation, but transfer to 3D detection remains weaker. We analyze recent SSL methods and find that most objectives are defined only on measured LiDAR returns from visible surfaces, leaving occluded and unobserved regions unconstrained. This visible-surface bias can be sufficient for point-wise prediction, but 3D detection requires robustness to missing structure. To address this gap, we propose *GhostPoint*, an SSL framework that hallucinates latent features in local neighborhoods around discovered instances, generated via a novel instance voxel dilation. In *GhostPoint*, an encoder processes observed returns, and an additional predictor infers neighborhood representations from observed context. In addition to standard encoder-level supervision, we introduce a predictor-level supervision scheme on sampled voxels from generated neighborhoods. Specifically, observed (visible/masked) voxels match teacher-encoder targets, while unobserved voxels match teacher-predictor hallucinations. This design encourages the learned representation to explicitly model structure beyond observed returns. Extensive evaluations on nuScenes and Waymo demonstrate that our method achieves state-of-the-art performance, consistently improving downstream 3D detection, especially under sparse scans and limited labels.

Keywords: Self-supervised Representation Learning · Object Detection · Autonomous Driving

1 Introduction

A robust 3D object detector is central to autonomous driving as it provides explicit object extents, position, and orientation for safety-critical behaviors such as collision avoidance, motion planning, and tracking in dynamic traffic scenes [3, 9, 14]. Its relevance remains high even in the era of end-to-end autonomous

⋆ Equal contribution.

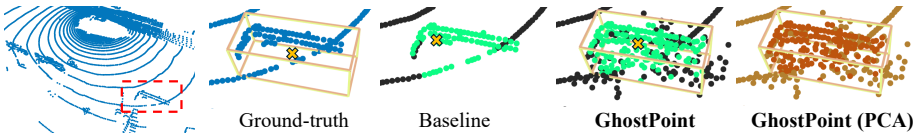


Fig. 1: *GhostPoint* learns beyond visible LiDAR surfaces. Raw scans observe only sparse object returns, causing discovered pseudo-instance centers to be biased toward visible surfaces. From these pseudo instances, *GhostPoint* samples adjacent unobserved regions and hallucinates their latent representations, producing features whose centers better align with the ground-truth object centers, marked by yellow crosses, while recovering object-level structure as illustrated by the PCA visualization.

driving, where explicit 3D perception still offers strong geometric priors and safety interpretability [12, 42]. Accordingly, recent LiDAR-centric detection research continues to improve robustness and label efficiency in realistic driving regimes [13, 21,34,41]. However, these advances still rely heavily on large-scale 3D annotations, which remain expensive and difficult to obtain [33].

Self-supervised learning (SSL) therefore offers a scalable path to pretrain LiDAR representations without annotations [2, 19, 28], reducing reliance on expensive 3D labels and improving data efficiency in autonomous driving and robotics, where unlabeled sensor logs are abundant but dense annotation is costly. A recurring observation in LiDAR SSL is that pretrained representations transfer strongly to per-point tasks such as semantic and panoptic segmentation [2, 32]. Transfer to 3D object detection, however, is often substantially less effective. This gap is particularly relevant for autonomous driving, where reliable 3D detection is a key precursor for downstream planning and multi-object tracking pipelines [6].

We analyze recent SSL methods and identify a shared limitation: the learning signal is predominantly defined on **measured LiDAR returns**, i.e., points captured from visible surfaces. Self-distillation approaches, such as Sonata [28] and DOS [2], explicitly enforce feature matching over scanned points only. Masked autoencoding variants, such as Occ-MAE [19] and NOMAE [1], typically treat voxels with no point returns as empty, rather than modeling them as potentially occluded or unobserved space. Even recent methods that introduce instance-aware cues, e.g., PointINS [32], still inherit this constraint: they are trained on visible returns and often define instance centroids using only the observed points, which can be biased under occlusion. As a result, pretraining emphasizes representations of observed surfaces, while the structure of **occluded, unobserved regions** is not explicitly incorporated into the objective, as shown in Figure 1. This bias can be adequate for tasks defined on measured returns, but detection requires robustness to missing structure since object extent and localization must be inferred from incomplete measurements.

In this work, we propose ***GhostPoint***, an SSL framework that learns beyond visible surfaces by hallucinating latent features in **local neighborhoods** around discovered instances, including unobserved regions. *GhostPoint* applies an encoder to measured LiDAR returns, after inducing pseudo-occlusion by ran-

domly masking returns, and retains standard encoder-level supervision with two complementary signals: **Softmaps** for semantics and **center-offset prediction** for instance discovery. To extend learning into unobserved space, we introduce an efficient instance voxel dilation procedure that expands each discovered instance into a neighborhood spanning both observed and no-return regions, and a lightweight neighborhood predictor that infers representations for sampled neighborhood voxels conditioned on encoded measured context. The predictor is supervised under the same Softmap and center-offset objectives, with targets for observed and masked voxels drawn from the teacher encoder and targets for unobserved voxels hallucinated by the teacher predictor. Since masked observed voxels and true no-return voxels both lack point evidence, learning to reverse pseudo-occlusion transfers naturally to genuinely unobserved regions, directly reducing occlusion bias.

We evaluate *GhostPoint* across several benchmarks and observe consistent improvements in downstream 3D detection, with the largest gains under sparse scans and limited labels. On nuScenes and Waymo, *GhostPoint* improves pretrained backbones out of the box when training only a lightweight detector, and further surpasses prior state of the art after full-parameter fine-tuning. On data-efficient benchmarks, it surpasses prior methods and matches full-label supervision with only 10% labels. On nuScenes, we additionally verify that these improvements do not come at the expense of segmentation transferability, confirming the scalability of *GhostPoint* across diverse downstream tasks. To summarize, this work contains the following contributions:

- We characterize and analyze a systematic transfer gap in LiDAR SSL: pretrained representations that perform well on segmentation transfer substantially less effectively to 3D object detection, and we attribute this to a representational mismatch between sparse surface-aligned SSL objectives and the dense voxel space in which detection labels are annotated.
- We propose ***GhostPoint***, an SSL framework that closes this gap by hallucinating latent features in local neighborhoods generated by instance voxel dilation, with an additional predictor and predictor-level supervision on generated voxels.
- We demonstrate consistent improvements in downstream 3D detection across standard LiDAR benchmarks, with especially strong gains under sparse sensing and limited label budgets, while preserving strong performance on other downstream tasks.

2 Related Works

LiDAR-based 3D Object Detection: 3D object detection from LiDAR point clouds has advanced rapidly with efficient voxel- and pillar-based representations. Earlier works such as VoxelNet [45] and SECOND [31] discretize point clouds into regular 3D grids and apply sparse convolutions, while PointPillars [17] reduces computation by collapsing the vertical axis into pillar features. Anchor-free formulations such as CenterPoint [39] simplify detection by predicting object centers

and offsets, yielding strong performance on standard benchmarks [4, 8, 25]. More recently, transformer-based detectors [7, 10, 18, 20, 29] have shown improved long-range interaction and global context modeling. In parallel with these architectural advances, end-to-end autonomous driving has gained momentum [12, 42]. Yet explicit 3D detection remains a key component as it provides geometric structure and interpretable safety cues that are difficult to replace in planning-critical scenarios [6]. Despite these advances, these methods remain fully supervised and require large volumes of precisely annotated 3D bounding boxes, motivating self-supervised pretraining to leverage abundant unlabeled LiDAR data and reduce annotation dependency.

Point Cloud Self-supervised Learning (SSL): SSL has emerged as a prominent paradigm for learning robust 3D representations from LiDARs without human annotations. Contrastive learning methods typically enforce representation consistency across augmented views and separate features of different samples, instances, or regions [22, 23, 30, 38, 43], while masked modeling and reconstruction-based methods pretrain encoders by recovering geometric information from masked inputs, such as occupancy or point coordinates [11, 19, 26, 36]. More recent work extends this line by reconstructing occupancy only within local neighborhoods and at multiple scales, improving robustness on large-scale point clouds in the autonomous driving domain [1]. With the rapid adoption of Transformer-based backbones in various 3D perception tasks [10, 16, 29, 44], recent SSL approaches have also incorporated such architectures into point cloud representation learning [2, 28]. These methods typically employ a teacher–student framework with shared architectures, where the teacher predictions are regularized to maintain distributional sharpness and supervise the student on corrupted inputs. Despite effectiveness, all of these approaches share a critical limitation that they formulate pretraining objectives exclusively over the visible points, while ignoring occluded and sparse regions, where LiDAR fails to return. *GhostPoint* departs from this paradigm by explicitly targeting unobserved local neighborhoods as the primary objective during pretraining, encouraging the encoder to internalize object geometry beyond measured surfaces.

Point Cloud Completion: The idea of inferring unobserved 3D geometry has been extensively studied in the context of point cloud completion, where the goal is to reconstruct a full object shape from a partial observation [35]. Methods such as PCN [40] and FoldingNet [37] learn to map partial inputs to complete point clouds using encoder-decoder architectures trained on paired partial-complete data. Implicit representations and occupancy prediction networks [5, 24, 27] take a complementary approach, learning continuous signed distance or occupancy that can be queried at arbitrary 3D locations. Despite their ability to model unobserved geometry, these methods are fundamentally supervised where they rely on complete ground-truth shapes or bounding boxes, and are applied at inference time as standalone reconstruction tools. *GhostPoint* shares the intuition that modeling unobserved structure is valuable, but targets self-supervised representation learning on sparse LiDAR scans rather than supervised shape completion.

3 Method

Building on the completion intuition, we study why current LiDAR SSL still transfers weakly to 3D detection and show that this gap stems from a representation mismatch between surface-only pretraining and box-level supervision under occlusion. We then introduce *GhostPoint*, which extends self-distillation with dilated instance neighborhoods and a predictor-level distillation loss on non-visible voxels (masked and no-return) to encourage reasoning beyond observed returns.

3.1 The SSL Transfer Gap to 3D Object Detection

Empirical Transfer Gap We begin with a direct empirical observation. We evaluate two representative SSL methods on two downstream tasks, semantic segmentation and 3D object detection, on nuScenes [4]. Using a probing protocol, we freeze the pretrained encoder and train only a lightweight task decoder. For segmentation, we use the standard lightweight decoder probe [28, 32]. For detection, we attach a CenterPoint BEV backbone and head [39] directly after the 3D encoder. All methods use PTv3 [29] for architectural comparability. As shown in Table 1, SSL methods transfer strongly to segmentation, achieving *near-supervised* mIoU. For 3D detection, however, the same pretrained encoders show a substantial gap to the supervised baseline in both mAP and NDS. PointINS [32] introduces instance-aware cues to improve instance-localization learning and achieves a modest gain over DOS, however, the detection gap remains large.

Table 1: Empirical gap of existing SSL approaches between semantic segmentation and object detection.

Method	**Sem. Seg**	**Obj. Det**	
	mIoU	mAP	NDS
PTv3 (sup.)	80.3	63.8	68.4
DOS [2]	79.2	55.4	61.6
PointINS [32]	80.0	56.7	62.5
PointINS + Box Reg.	-	56.2	62.4

Figure 1 illustrates the reason. Even when SSL correctly groups visible points into an instance, occlusion yields only a partial object view, so the pseudo-instance centroid is biased toward visible surfaces instead of the true geometric center. We also test whether adding box regression resolves this. As shown in the fourth row of Table 1, extending PointINS with box fitting and a box regression head [22] does not improve over the baseline and slightly degrades performance. The reason is that the regression target is still derived from the same occluded point cluster, so geometric objectives defined only on visible points inherit the bias they aim to correct. This negative result leads to a key conclusion: instance-aware pretraining is necessary but not sufficient. The limitation is not missing geometric learning, but the incompleteness of the observed scene.

Representation Mismatch in SSL We formalize this observation as a fundamental mismatch between how SSL pretraining and downstream detection define and use representations. Let $\mathcal{V} = \mathcal{V}^o \cup \mathcal{V}^u$ be the full scene voxel set, where $\mathcal{V}^o$ contains measured returns and $\mathcal{V}^u$ has no observations. In prior SSL, transformer

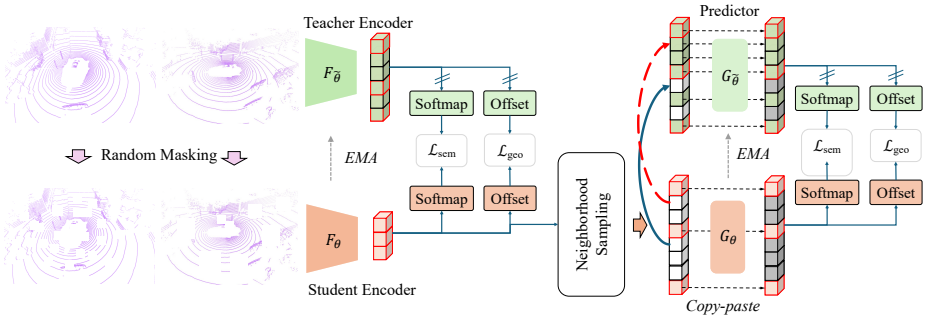


Fig. 2: Overview of *GhostPoint*: A scan is augmented into two views. The student encodes a masked version of each view F_θ, while the teacher encoder $F_{\tilde{\theta}}$ (EMA) processes the unmasked views. Teacher features are used to discover instances. Discovered instances drive Neighborhood Sampling, which dilates their occupancy to form a neighborhood voxel set $\mathcal{Q}$ (visible, masked, and unobserved voxels). Teacher and student predictors $G_{\tilde{\theta}}$ and G_θ operate on tokens over $\mathcal{Q}$: the teacher hallucinates only at no-return voxels (overwriting occupied voxels with encoder features), while the student predicts over all non-visible voxels $\mathcal{Q} \setminus \mathcal{Q}_{\text{vis}}$. Softmap and offset heads produce targets and predictions at two levels: visible voxels use encoder-level features, while the second-level supervision on non-visible voxels is described in Section 3.2.

encoders (e.g., PTv3 [29]) operate only on $\mathcal{V}^o$, producing no representation for $v \in \mathcal{V}^u$. This sparsity enables efficiency in large outdoor scenes. Accordingly, SSL objectives are naturally defined over occupied voxels $\mathcal{V}^o$, encouraging representations aligned with observed measurements. This works well for per-point tasks such as semantic segmentation, whose labels are also defined on $\mathcal{V}^o$ and do not require reasoning about unobserved regions.

Detection is annotated differently: boxes $\mathcal{B}_k$ specify the full object extent in $\mathcal{V}$, including partially occluded regions. During supervised training, the model must reason beyond sparse observed returns, effectively extrapolating sparse encoder features into unobserved space for object-level predictions. This creates a **representation mismatch** between SSL pretraining and detection fine-tuning. SSL constrains the encoder only on occupied voxels, whereas detection depends on representation quality over the full object extent. Thus, an encoder trained only with SSL may be valid on $\mathcal{V}^o$ yet inconsistent in unobserved regions $\mathcal{V}^u$. The issue is especially severe in outdoor LiDAR scenes with heavy occlusion. This exposes a core limitation of existing SSL for 3D detection and motivates pretraining that explicitly models object-level structure beyond observed measurements.

3.2 *GhostPoint* Framework

To address the mismatch identified above, we propose *GhostPoint*, which extends self-distillation [2, 32] with an explicit objective over unobserved regions as illustrated in Figure 2.

Given an input point cloud, we generate two independently augmented views. For each augmented view $\mathcal{P} = \{(x_i, f_i)\}_{i=1}^{N}$, both student and teacher process the

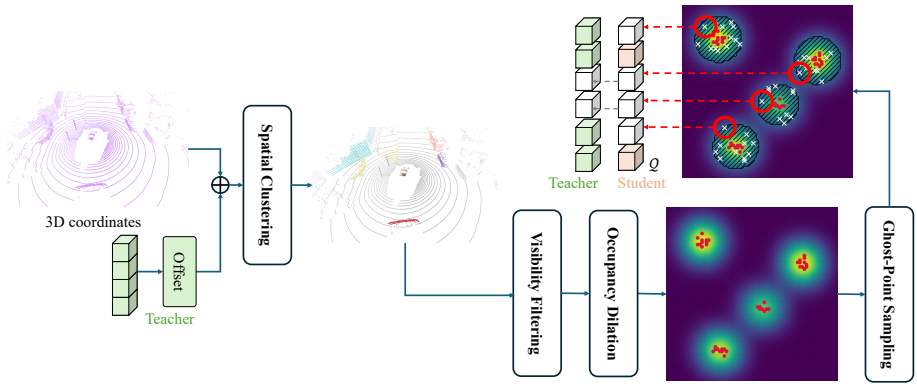


Fig. 3: Neighborhood Sampling. Teacher encoder features are passed to an offset head to predict center-offsets and cluster non-ground points into instance proposals. We drop proposals fully masked in the student view (visibility filtering), then dilate each remaining instance occupancy (kernel size $ks > 1$) to form a neighborhood voxel set $\mathcal{Q}$ containing visible, masked, and newly activated no-return voxels. $\mathcal{Q}$ defines predictor queries and Softmap/offset supervision's scope for predictor-level distillation.

same view: the teacher receives the unmasked set $\mathcal{P}$, while the student receives its randomly masked subset $\mathcal{P}_{\text{vis}} \subset \mathcal{P}$. Let $\mathcal{V}^o$ be the set of occupied voxels from voxelizing the teacher view $\mathcal{P}$, and let $\mathcal{V}^{\text{vis}} \subseteq \mathcal{V}^o$ be the occupied voxels obtained by voxelizing the corresponding masked student input $\mathcal{P}_{\text{vis}}$. The student encoder receives $\mathcal{V}^{\text{vis}}$, while the teacher encoder receives $\mathcal{V}^o$, and the masked occupied voxels are $\mathcal{V}^{\text{mask}} = \mathcal{V}^o \setminus \mathcal{V}^{\text{vis}}$. Student and teacher share the same architecture, with teacher parameters updated as an EMA of the student. Both encoders output sparse per-voxel features. As in prior self-distillation [32], Softmap and offset heads produce semantic assignments and center-offset predictions to form pseudo-instances and provide the base learning signal over matched visible voxels of the student. *GhostPoint* augments this setup with a prediction branch that extends supervision into occluded space, described next.

New Voxels Sampling The challenge of modeling unobserved regions without labels is knowing *where* to place new queries in unoccupied space. Uniform sampling in free space is computationally prohibitive at outdoor scene scale and yields mostly background locations with little structure to predict [1]. *GhostPoint* instead generates queries *near objects*: LiDAR returns are spatially clustered, and missing object surfaces are most likely to occur adjacent to observed instance occupancy (e.g., due to occlusions or no-return gaps), while voxels far from any instance are predominantly background. We implement this instance-centric sampling in two steps: filtering student-visible instances and dilating their occupancy, as illustrated in Figure 3.

In the first step, we drop pseudo-instances that are fully masked in the student view, keeping an instance if it contains at least one visible voxel. Let $\mathcal{V}^o_{\text{inst}} \subseteq \mathcal{V}^o$ denote the union of occupied voxels from the remaining instances. We then dilate

the binary occupancy grid of $\mathcal{V}^o_{\text{inst}}$ with a 3D kernel of size $ks > 1$ to obtain a candidate neighborhood set $\mathcal{Q}^{\text{dil}} \supseteq \mathcal{V}^o_{\text{inst}}$. For efficiency, we sample a fixed-size subset $\mathcal{Q} \subseteq \mathcal{Q}^{\text{dil}}$ and use this sampled set for all subsequent predictor queries and losses. With respect to the student masking, $\mathcal{Q}$ decomposes into $\mathcal{Q}_{\text{vis}} = \mathcal{Q} \cap \mathcal{V}^{\text{vis}}$, $\mathcal{Q}_{\text{mask}} = \mathcal{Q} \cap \mathcal{V}^{\text{mask}}$, and $\mathcal{Q}_{\text{unobs}} = \mathcal{Q} \setminus \mathcal{V}^o$.

Token Initialization and Refinement The student predictor operates on the neighborhood voxel set $\mathcal{Q}$, which includes both occupied voxels and newly activated no-return voxels. Tokens are available from the student encoder only on the visible occupied subset $\mathcal{Q}_{\text{vis}} = \mathcal{Q} \cap \mathcal{V}^{\text{vis}}$. For all remaining queried voxels $\mathcal{Q} \setminus \mathcal{Q}_{\text{vis}}$ (including masked occupied voxels $\mathcal{Q}_{\text{mask}}$ and no-return voxels $\mathcal{Q}_{\text{unobs}}$), the student must initialize tokens explicitly before applying the predictor G_θ. By default, we initialize each such voxel q by distance-weighted interpolation from its k nearest occupied voxel neighbors in $\mathcal{V}^{\text{vis}}$ (we use $k = 3$). Here, $\mathcal{N}_k(q)$ returns the k nearest occupied voxels $v \in \mathcal{V}^{\text{vis}}$ to voxel q, and $c(\cdot)$ denotes a voxel's 3D center coordinate:

$$\mathbf{f}_q = \sum_{v \in \mathcal{N}_k(q)} w_v \, \mathbf{f}_v, \qquad w_v = \frac{1/d_v}{\sum_{v' \in \mathcal{N}_k(q)} 1/d_{v'}}, \tag{1}$$

where $d_v = \|c(v) - c(q)\|_2$ and $\mathbf{f}_v$ denotes the student encoder feature at visible occupied voxel v. As an alternative, we also evaluate a simpler *mask-token* initialization, where all voxels in $\mathcal{Q} \setminus \mathcal{Q}_{\text{vis}}$ share a learnable token $\mathbf{m} \in \mathbb{R}^d$. The initialized tokens, together with encoder tokens on $\mathcal{Q}_{\text{vis}}$, are then processed by the lightweight predictor G_θ to aggregate information from visible voxels into the new tokens across $\mathcal{Q}$. To refine only newly queried tokens while using self-attention over all tokens for simplicity, we copy visible-token features from before the predictor to after the predictor (identity overwrite on $\mathcal{Q}_{\text{vis}}$), and keep predictor outputs only on $\mathcal{Q} \setminus \mathcal{Q}_{\text{vis}}$. Softmap and offset heads then map these features to semantic and geometric predictions over $\mathcal{Q}$.

New Voxels Target Generation To supervise these predictions, we construct asymmetric *teacher targets* on $\mathcal{Q}$ that are anchored on measured occupied voxels and hallucinate only in no-return space, following the same visible-token identity overwrite described above. For occupied voxels ($\mathcal{Q} \cap \mathcal{V}^o$, including $\mathcal{Q}_{\text{mask}}$), we compute target Softmaps and offsets from the teacher *encoder* representations, since they are directly supported by LiDAR returns. For unobserved voxels $\mathcal{Q}_{\text{unobs}}$, we instead compute targets from the teacher *predictor* output, providing context-based semantic and geometric predictions where no encoder token exists. Because the teacher predictor is an EMA of the student predictor and is conditioned on the full (unmasked) voxel set, these hallucinated targets become progressively stronger as training proceeds. We apply predictor-level distillation losses on $\mathcal{Q} \setminus \mathcal{Q}_{\text{vis}} = \mathcal{Q}_{\text{mask}} \cup \mathcal{Q}_{\text{unobs}}$, thereby training masked occupied voxels and unobserved voxels under a unified objective and encouraging the same semantic and geometric completion behavior to transfer from pseudo-occluded to truly unobserved regions without any annotations.

Training objectives *GhostPoint* applies distillation at two complementary stages: encoder-level supervision on voxels visible to the student, and predictor-level supervision on non-visible voxels in the sampled neighborhood (including masked and unobserved regions).

Encoder-level distillation. We apply standard self-distillation on visible occupied voxels $\mathcal{V}^{\text{vis}}$. Teacher and student encoder outputs are mapped to (i) Softmap distributions via the prototype head and (ii) center-offset vectors via the offset head. The resulting semantic and geometric losses, $\mathcal{L}_{\text{sem,vis}}$ and $\mathcal{L}_{\text{geo,vis}}$, are computed over $\mathcal{V}^{\text{vis}}$ following prior work [2, 32].

Predictor-level distillation. Using the sampled neighborhood voxel set for the kept instances $\mathcal{Q}$ and its student-visible subset $\mathcal{Q}_{\text{vis}} = \mathcal{Q} \cap \mathcal{V}^{\text{vis}}$, predictor-level supervision is applied on $\mathcal{Q}_{\neg\text{vis}} = \mathcal{Q} \setminus \mathcal{Q}_{\text{vis}}$. We retain the same semantic (Softmap) and geometric (offset) objectives used at the encoder level, following prior self-distillation findings that these two signals are complementary for representation learning :

$$\mathcal{L}_{\text{sem},\neg\text{vis}} = \sum_{q \in \mathcal{Q}_{\neg\text{vis}}} \text{KL}\big(\pi^{G_{\tilde{\theta}}}(q) \parallel \pi^{G_\theta}(q)\big)\,, \tag{2}$$

$$\mathcal{L}_{\text{geo},\neg\text{vis}} = \sum_{q \in \mathcal{Q}_{\neg\text{vis}}} \left(\Big\| \|\delta^{G_\theta}(q)\| - \|\tilde{\delta}^{G_{\tilde{\theta}}}(q)\| \Big\|_1 + 1 - \frac{\delta^{G_\theta}(q) \cdot \tilde{\delta}^{G_{\tilde{\theta}}}(q)}{\|\delta^{G_\theta}(q)\| \, \|\tilde{\delta}^{G_{\tilde{\theta}}}(q)\|} \right), \tag{3}$$

where $\pi^{G_{\tilde{\theta}}}(q)$ and $\pi^{G_\theta}(q)$ denote the Softmap distributions at voxel q, and $\tilde{\delta}^{G_{\tilde{\theta}}}(q)$ and $\delta^{G_\theta}(q)$ denote the corresponding teacher and student predictor offsets. The full objective is

$$\mathcal{L} = \mathcal{L}_{\text{sem,vis}} + \lambda\, \mathcal{L}_{\text{geo,vis}} + \mathcal{L}_{\text{sem},\neg\text{vis}} + \lambda\, \mathcal{L}_{\text{geo},\neg\text{vis}}, \tag{4}$$

with λ balancing semantic and geometric terms.

4 Experiments

Building on the *GhostPoint* design introduced in Section 3.2, we evaluate how well the resulting pretrained representations transfer to 3D object detection. We first describe implementation details and benchmarks, then report main detection results under probing and full fine-tuning, followed by component ablations and additional analyses on label efficiency, segmentation transfer, cross-dataset transferability, and architectural transferability.

4.1 Implementation Details

We evaluate *GhostPoint* with two encoder backbones for F_θ: PTv3 [29] and the sparse convolutional backbone of CenterPoint [39]. In both cases, we use the

Table 2: Main results of SSL pretraining for LiDAR 3D object detection with PTv3 as backbone. *probe*: freeze the backbone F_θ and train the remaining detector components from scratch; *ft*: end-to-end fine-tuning. Best results in each setting are in **bold**.

Method	nuScenes		Waymo (L2 mAP)			
	mAP	NDS	Vehicle	Pedestrian	Cyclist	mAP
PTv3 (sup.)	63.8	68.4	69.5	67.5	66.7	67.9
PSA [22] (prob.)	41.3	52.8	45.6	42.4	40.8	42.4
SONATA [28] (prob.)	44.6	55.0	51.2	46.2	44.9	47.4
NOMAE [1] (prob.)	53.5	60.1	55.3	51.5	50.3	52.4
DOS [2] (prob.)	55.4	61.6	59.4	56.6	55.2	57.1
PointINS [32] (prob.)	56.7	62.5	60.2	56.6	55.7	57.5
GhostPoint (prob.)	**59.5**	**64.2**	**62.3**	**59.2**	**58.5**	**60.0**
PSA [22] (ft.)	63.9	68.9	69.9	67.5	67.0	68.1
SONATA [28] (ft.)	64.2	69.0	70.8	67.9	67.0	68.6
NOMAE [1] (ft.)	65.8	69.8	71.4	68.5	68.5	69.5
DOS [2] (ft.)	65.5	69.7	71.4	68.2	68.0	69.2
PointINS [32] (ft.)	66.3	70.1	71.6	68.5	68.3	69.5
GhostPoint (ft.)	**67.5**	**71.2**	**72.0**	**69.2**	**69.0**	**70.1**

same lightweight predictor G, implemented as two PTv3-style transformer blocks. Training uses a two-stage warmup before joint optimization: for the first 10% of epochs we train only the encoder Softmap branch to stabilize representations; for the next 10% we activate the encoder offset branch and predictor branches, detaching their gradients from the student encoder for stability. We then optimize all components end-to-end. Performance is robust to these hyperparameters as long as the warmup order is preserved (see Appendix). Unless specified otherwise, we set $ks = 5$ for occupancy dilation and $\lambda = 0.1$ for loss weighting. For downstream evaluation, the pretrained encoder serves as the backbone of a CenterPoint detector [39]. Additional details are provided in the Appendix.

4.2 Dataset

We evaluate *GhostPoint* on two widely adopted LiDAR 3D object detection benchmarks. **nuScenes** [4] is a large-scale autonomous driving dataset comprising 700 training, 150 validation, and 150 test scenes captured with a 32-beam LiDAR sensor. It provides 1.4M annotated 3D bounding boxes across 10 object categories, with performance measured by mean Average Precision (mAP) and the nuScenes Detection Score (NDS). **Waymo Open Dataset** [25] is a large-scale benchmark containing 798 training and 202 validation sequences, annotated across 3 object categories. Following prior works [22, 46], we use 20% of full training set for pretraining and report Average Precision (AP) at Level 2 difficulty.

4.3 Main Results

Table 2 compares *GhostPoint* against recent SSL methods under two evaluation protocols: decoder probing, where the pretrained PTv3 encoder is frozen and other CenterPoint components are trained from scratch, and full fine-tuning,

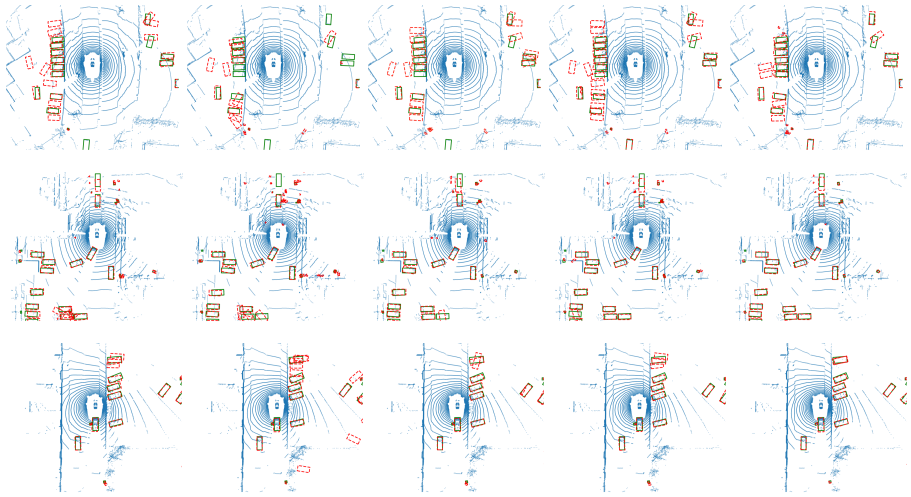


Fig. 4: Qualitative results on the nuScenes dataset [4]. All self-supervised models are evaluated under the decoder-probing protocol. Green boxes denote ground-truth annotations and red boxes denote predictions. *GhostPoint* produces predictions that align more closely with ground-truth boxes in both center localization and spatial extent, particularly for partially observed and occluded objects at mid-to-long range.

where the encoder is initialized from pretrained weights and the full detector is optimized end-to-end. *GhostPoint* consistently achieves the best performance. Under decoder probing, *GhostPoint* surpasses the second-best SSL method by 2.8 mAP and 1.7 NDS on nuScenes, and by 2.5 mAP on Waymo. Under full fine-tuning, gains are further amplified, with *GhostPoint* reaching 67.5 mAP and 71.2 NDS on nuScenes and 70.1 mAP on Waymo, surpassing the supervised baseline by 3.7 mAP on nuScenes. These results demonstrate that explicitly modeling unobserved object geometry during pretraining yields representations that are substantially better aligned with 3D object detection. Additionally, Figure 4 highlights these gains: in the first row, *GhostPoint* is the only method that correctly localizes the partially observed car at the top right. In the second row, it reduces false positives and improves center/orientation accuracy, although it can still hallucinate occasional boxes in empty space. In the third row, it is the only method that consistently produces a single prediction per object with the correct orientation.

4.4 Influence of Key Components

Table 3 ablates the key components of *GhostPoint*. Starting from the baseline [32], we first add the Neighborhood Sampling module with direct distance-weighted KNN interpolation, assigning features to ghost positions from their nearest observed neighbors without any additional predictor. This already yields a

Table 3: Ablation study on *GhostPoint*'s key components. We evaluate on nuScenes [4] under decoder probing protocol. ⋆: no predictor, features of newly sampled voxels are interpolated from observed nearest neighbors.

Model	mAP	NDS
Baseline [32]	56.7	62.5
+ ⋆Neighborhood Sampling	57.3	62.7
+ Predictor (no warmup)	57.2	63.0
+ Predictor (one-stage warmup)	58.5	63.6
+ Predictor (two-stage warmup)	59.5	64.2
+Box Regression	59.1	63.9

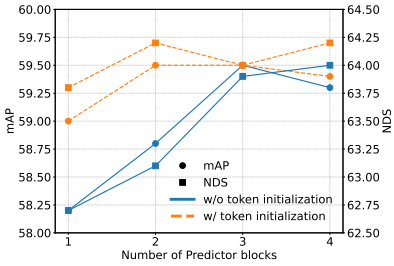


Fig. 5: Effect of predictor depth and token initialization. We ablate number of blocks in G_θ and KNN feature interpolation for teacher token initialization.

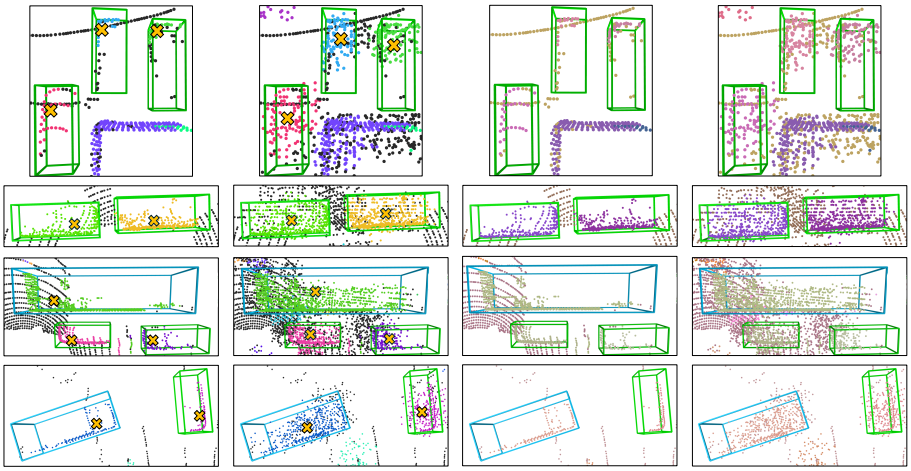
consistent improvement, which confirms that extending the supervision signal into occluded regions is beneficial for SSL even without additional model capacity. Adding the predictor with the warmup schedule produces a substantially larger gain. This improvement is not merely a capacity effect: the predictor enables active context propagation between observed and unobserved positions, requiring the encoder to produce features that are aware of object geometry beyond directly measured surfaces. Finally, consistent with the analysis in Section 3.1, augmenting *GhostPoint* with an additional box regression target leads to a performance drop. This observation reinforces our conclusion that additional geometric supervision cannot resolve the representational mismatch in SSL.

Figure 5 reports the effect of predictor depth and token initialization scheme. Without KNN-based initialization of newly sampled tokens, performance increases consistently with predictor depth up to 3 blocks, then plateaus. This suggests that additional capacity helps propagate context between observed and ghost positions but yields diminishing returns beyond a moderate depth. With the token initialization, performance is consistently higher and remains stable from 2 blocks onward, which indicates that providing the teacher with spatially coherent ghost initializations reduces the burden on the predictor and makes training less sensitive to predictor depth. Based on these results, we adopt 2 predictor blocks as the default, balancing performance and computational cost. Additional ablations on other hyperparameters are presented in the Appendix.

Disentangling hallucination from regularization. Table 4 isolates the source of gains through three controlled ablations. First, restricting the predictor to masked observed points (*No hallucination*) slightly improves over PointINS (+0.6 mAP / +0.6 NDS), suggesting mild regularization from predictor capacity. Second, keeping ghost-point neighborhoods but replacing their targets with zero offsets and uniform Softmaps (*Zero targets*) yields only marginal further

Table 4: Disentangling hallucination from regularization on nuScenes (decoder probing).

Model	mAP	NDS
PointINS	56.7	62.5
+ No hallucination	57.3	63.1
+ Zero targets	57.6	63.3
+ Swap masked → hallucinated	59.4	64.1
GhostPoint (ours)	**59.5**	**64.2**



(a) orig. Instance **(b)** halluc. Instance **(c)** orig. PCA **(d)** halluc. PCA

Fig. 6: Visualizations of original and hallucinated point clouds. Ground-truth bounding boxes are shown for reference. (a)/(b): pseudo instances obtained by clustering predicted offsets; (c)/(d): PCA projections of point features.

change. This shows that ghost-point locations alone contribute little. Third, swapping the same number of masked points with hallucinated points (*Swap*) matches full *GhostPoint*, ruling out any supervision-count advantage. Together, these results attribute the gains specifically to **informative hallucination targets** in object-adjacent unobserved regions, rather than to predictor regularization or extra supervised positions.

4.5 Visualizations

To further analyze what is learned within *GhostPoint*, we provide visualizations in Fig. 6. We apply clustering to predicted center offsets and visualize the resulting pseudo-instance assignments in color, examining whether hallucinated points meaningfully extend object geometry into occluded regions. We additionally visualize PCA-projected features to assess the semantic coherence of hallucinated points. Both views consistently show that *GhostPoint* produces object-consistent representations rather than arbitrary fillers, and that the hallucinated neighborhoods reduce visible-surface bias under occlusion.

Nonetheless, the visualizations also expose clear failure modes. When objects are extremely sparse due to severe occlusion, as in the first example, the hallucination fails to recover the complete object shape. Additionally, hallucination can be sensitive to background outliers, leading to blurred or incorrectly extended instance boundaries.

4.6 Additional Results

Label Efficiency We evaluate *GhostPoint* on label-efficient benchmarks on both datasets. Results are shown in Table 5 and Table 7. *GhostPoint* consistently

Table 5: Label efficiency results on nuScenes.

Method	0.1%		1%		10%	
	mAP	NDS	mAP	NDS	mAP	NDS
PTv3 (sup.)	18.1	35.3	37.3	48.0	57.9	65.0
PSA [22]	25.2	42.8	41.0	51.9	58.0	65.2
SONATA [28]	26.4	44.3	43.2	53.6	58.4	65.8
NOMAE [1]	30.5	46.8	45.7	56.0	59.7	66.6
DOS [2]	34.0	49.1	49.0	58.0	61.4	67.4
PointINS [32]	34.8	50.8	50.8	60.2	62.1	67.8
GhostPoint	**36.9**	**52.1**	**53.1**	**62.5**	**63.5**	**68.8**

Table 6: Wy → nS Transfer Probing

Method	mAP	NDS
PSA [22]	28.2	43.9
SONATA [28]	32.2	47.3
NOMAE [1]	35.6	50.5
DOS [2]	41.0	52.3
PointINS [32]	41.6	52.4
GhostPoint	**43.4**	**54.1**

Table 7: Label efficiency results on Waymo.

Method	0.1%				1%			
	Vehicle	Pedestrian	Cyclist	mAP	Vehicle	Pedestrian	Cyclist	mAP
PTv3 (sup.)	32.1	15.2	12.6	20.0	42.1	27.2	33.5	34.3
NOMAE [1]	34.8	20.9	18.8	24.8	43.6	29.8	35.0	36.1
DOS [2]	38.3	23.2	20.0	27.2	45.1	33.2	36.5	38.3
PointINS [32]	39.0	22.8	19.6	27.1	46.5	34.3	37.2	39.3
GhostPoint	**40.5**	**23.5**	**21.6**	**28.5**	**47.2**	**36.8**	**39.0**	**41.0**

outperforms all prior SSL methods across every annotation regime on both datasets. On nuScenes, with only 0.1% of labels, *GhostPoint* achieves 36.9 mAP and 52.1 NDS, surpassing the second-best method by 2.1 mAP and 1.3 NDS. At 1% labels, the margin widens to 2.3 mAP and 2.3 NDS over PointINS, and at 10% labels *GhostPoint* reaches 63.5 mAP and 68.8 NDS. On Waymo, *GhostPoint* similarly leads across all categories at both 0.1% and 1% labels, with particularly strong gains on pedestrian and cyclist detection where occlusion is most severe.

Segmentation Performance To verify that *GhostPoint* does not sacrifice per-point transferability in favor of detection, we evaluate semantic and panoptic segmentation on nuScenes under a linear-probing protocol (Table 9). On semantic segmentation, *GhostPoint* reaches **74.2** mIoU, essentially matching the strongest SSL baseline (PointINS: **74.4**), which indicates that surface-level feature quality is preserved. On panoptic segmentation, *GhostPoint* achieves the best SSL performance with **62.8** PQ, improving over PointINS by +**0.6** and over DOS by +**5.4**; it also attains the top SQ/RQ among SSL methods (**84.6**/**73.2**). Overall, these results show that *GhostPoint* strengthens object-level transfer for detection while maintaining strong per-point semantics.

Cross-dataset Transferability To assess cross-domain generalization, we pretrain on Waymo and transfer to nuScenes under the decoder-probing protocol. As shown in Tab. 6, *GhostPoint* achieves the best cross-dataset performance with **43.4** mAP and **54.1** NDS. This improves over the strongest prior SSL baseline (PointINS: 41.6 mAP, 52.4 NDS) by +**1.8** mAP and +**1.7** NDS, and over DOS (41.0 mAP, 52.3 NDS) by +**2.4** mAP and +**1.8** NDS, indicating better robustness to domain-specific sensor characteristics and scene layouts.

Table 8: Architectural transferability on Waymo (L2 mAP). Replacing PTv3 with the CenterPoint sparse CNN encoder, distillation from a frozen *GhostPoint* PTv3 teacher consistently outperforms training the CNN directly (probing and fine-tuning).

SPUNet	Veh.	Ped.	Cyc.	mAP
From scratch	64.9	62.3	66.3	64.5
Pretrain (prob.)	48.5	43.4	50.2	48.0
Distill (prob.)	58.3	54.7	59.0	57.3
Pretrain (ft.)	65.8	63.5	66.9	65.4
Distill (ft.)	**68.8**	**64.8**	**68.0**	**67.2**

Table 9: Linear probing performance on nuScenes Semantic and Panoptic Segmentation.

Method	**Sem.**	**Pan.**		
	mIoU	PQ	SQ	RQ
PTv3 (sup.)	80.3	69.9	86.3	80.5
PSA [22]	44.5	30.1	73.9	38.6
SONATA [28]	59.2	50.7	79.8	61.6
NOMAE [1]	64.7	45.5	77.0	56.4
DOS [2]	74.1	57.4	82.8	68.5
PointINS [32]	**74.4**	62.2	84.5	72.8
GhostPoint	74.2	**62.8**	**84.6**	**73.2**

Architectural Transferability Table 8 reports *GhostPoint* results on **Waymo** when replacing the PTv3 encoder with the **convolutional encoder from CenterPoint**. Training *GhostPoint* directly with this CNN backbone transfers less effectively than with a transformer: under probing, the CNN variant reaches only **48.0** L2 mAP overall, compared to **57.3** when distilling from a frozen *GhostPoint* PTv3 teacher (**PTv3→CNN**). With full fine-tuning, distillation further improves the CNN student to **67.2** overall, outperforming both the CNN trained with *GhostPoint* directly (**65.4**) and the supervised CNN baseline (**64.5**). This suggests *GhostPoint* pretrained PTv3 features are broadly useful beyond the PTv3 architecture, offering a practical path to transferring object-aware representations into lightweight CNN backbones where transformer encoders may be computationally prohibitive. This trend also aligns with observations in earlier work that pretraining a transformer and distilling to a CNN can outperform pretraining the CNN directly [2].

5 Conclusion

We presented *GhostPoint*, which addresses a key limitation of LiDAR SSL: objectives defined primarily on measured returns leave occluded and no-return regions largely unconstrained, contributing to a systematic transfer gap to 3D object detection. We attributed this gap to a representation mismatch between surface-aligned SSL supervision and detection labels defined over full object extents. *GhostPoint* mitigates this mismatch by generating instance-centric neighborhood voxels via instance voxel dilation and training a predictor to infer ghost representations from observed context. A predictor-level supervision scheme matches student predictions to teacher targets on non-visible neighborhood voxels, promoting features that support reasoning beyond visible surfaces. Experiments on nuScenes and Waymo show consistent improvements across evaluation protocols and label budgets, while maintaining strong transfer to segmentation. Our work has two limitations: under probing, performance still trails fully supervised training, and we have not evaluated *GhostPoint* on other sparse sensing modalities affected by occlusion (e.g., radar). We leave both to future work.

References

1. Abdelsamad, M., Ulrich, M., Gläser, C., Valada, A.: Multi-scale neighborhood occupancy masked autoencoder for self-supervised learning in lidar point clouds. In: CVPR. pp. 22234–22243 (2025)
2. Abdelsamad, M., Ulrich, M., Yang, B., Zhang, M., Miron, Y., Valada, A.: Dos: Distilling observable softmaps of zipfian prototypes for self-supervised point cloud representation learning. In: AAAI (2026)
3. Büchner, M., Valada, A.: 3d multi-object tracking using graph neural networks with cross-edge modality attention. IEEE Robotics and Automation Letters **7**(4), 9707–9714 (2022)
4. Caesar, H., Bankiti, V., Lang, A.H., Vora, S., Liong, V.E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., Beijbom, O.: nuscenes: A multimodal dataset for autonomous driving. In: CVPR. pp. 11621–11631 (2020)
5. Chibane, J., Alldieck, T., Pons-Moll, G.: Implicit functions in feature space for 3d shape reconstruction and completion. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2020)
6. Contreras, C., de Charette, R., Foresti, G.L., Micheloni, C.: A survey on deep-learning methods for 3d object detection in autonomous driving. Frontiers in Big Data **7**, 1212070 (2024)
7. Dong, S., Ding, L., Wang, H., Xu, T., Xu, X., Wang, J., Bian, Z., Wang, Y., Li, J.: Mssvt: Mixed-scale sparse voxel transformer for 3d object detection on point clouds. In: Advances in Neural Information Processing Systems (NeurIPS) (2022)
8. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? the kitti vision benchmark suite. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2012)
9. Gosala, N., Kiran, B.R., Yogamani, S., Valada, A.: Sparse3dtrack: Monocular 3d object tracking using sparse supervision. arXiv preprint arXiv:2603.18298 (2026)
10. He, C., Li, R., Li, S., Zhang, L.: Voxel set transformer: A set-to-set approach to 3d object detection from point clouds. In: CVPR. pp. 8417–8427 (2022)
11. Hess, G., Jaxing, J., Svensson, E., Hagerman, D., Petersson, C., Svensson, L.: Masked autoencoder for self-supervised pre-training on lidar point clouds. In: Proceedings of the IEEE/CVF Winter conference on Applications of Computer Vision (WACV). pp. 350–359 (2023)
12. Hu, Y., Yang, J., Chen, L., Sima, C., Li, S., Zhu, X., Chai, T., Du, S., Lin, W., Wang, W., et al.: Planning-oriented autonomous driving. In: CVPR. pp. 17853–17862 (2023)
13. Käppeler, M., Çiçek, Ö., Cattaneo, D., Gläser, C., Miron, Y., Valada, A.: Bridging perspectives: Foundation model guided bev maps for 3d object detection and tracking. arXiv preprint arXiv:2510.10287 (2025)
14. Käppeler, M., Çiçek, Ö., Miron, Y., Valada, A.: Leveraging previous-traversal point cloud map priors for camera-based 3d object detection and tracking. arXiv preprint arXiv:2604.25405 (2026)
15. Kong, L., Liu, Y., Li, X., Chen, R., Zhang, W., Ren, J., Pan, L., Chen, K., Liu, Z.: Robo3d: Towards robust and reliable 3d perception against corruptions. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 19994–20006 (2023)
16. Lai, X., Chen, Y., Lu, F., Liu, J., Jia, J.: Spherical transformer for lidar-based 3d recognition. In: CVPR. pp. 17545–17555 (2023)

17. Lang, A.H., Vora, S., Caesar, H., Zhou, L., Yang, J., Beijbom, O.: Pointpillars: Fast encoders for object detection from point clouds. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR). pp. 12697–12705 (2019)
18. Lang, C., Braun, A., Schillingmann, L., Valada, A.: A point-based approach to efficient lidar multi-task perception. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 13261–13268 (2024)
19. Min, C., Xiao, L., Zhao, D., Nie, Y., Dai, B.: Occupancy-mae: Self-supervised pre-training large-scale lidar point clouds with masked occupancy autoencoders. IEEE Transactions on Intelligent Vehicles **9**(7), 5150–5162 (2023)
20. Misra, I., Girdhar, R., Joulin, A.: An end-to-end transformer model for 3d object detection. In: Proceedings of the IEEE/CVF international conference on computer vision (ICCV). pp. 2906–2917 (2021)
21. Mohan, R., Cattaneo, D., Drews, F., Valada, A.: Progressive multi-modal fusion for robust 3d object detection. In: 8th Annual Conference on Robot Learning (2024)
22. Nisar, B., Waslander, S.L.: Psa-ssl: Pose and size-aware self-supervised learning on lidar point clouds. In: CVPR. pp. 6670–6679 (2025)
23. Nunes, L., Marcuzzi, R., Chen, X., Behley, J., Stachniss, C.: Segcontrast: 3d point cloud feature representation learning through self-supervised segment discrimination. IEEE Rob. and Auto. Letters **7**(2), 2116–2123 (2022)
24. Peng, S., Niemeyer, M., Mescheder, L., Pollefeys, M., Geiger, A.: Convolutional occupancy networks. In: european conference on computer vision. pp. 523–540 (2020)
25. Sun, P., Kretzschmar, H., Dotiwalla, X., Chouard, A., Patnaik, V., Tsui, P., Guo, J., Zhou, Y., Chai, Y., Caine, B., Vasudevan, V., Han, W., Ngiam, J., Zhao, H., Timofeev, A., Ettinger, S., Krivokon, M., Gao, A., Joshi, A., Zhang, Y., Shlens, J., Chen, Z., Anguelov, D.: Scalability in perception for autonomous driving: Waymo open dataset. In: CVPR. pp. 2446–2454 (2020)
26. Tian, X., Ran, H., Wang, Y., Zhao, H.: Geomae: Masked geometric target prediction for self-supervised point cloud pre-training. In: CVPR. pp. 13570–13580 (2023)
27. Wang, X., Zhang, Y., Liu, T., Liu, X., Xu, K., Wan, J., Guo, Y., Wang, H.: Topnet: Transformer-efficient occupancy prediction network for octree-structured point cloud geometry compression. In: Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR). pp. 27305–27314 (2025)
28. Wu, X., DeTone, D., Frost, D., Shen, T., Xie, C., Yang, N., Engel, J., Newcombe, R., Zhao, H., Straub, J.: Sonata: Self-supervised learning of reliable point representations. In: CVPR (2025)
29. Wu, X., Jiang, L., Wang, P.S., Liu, Z., Liu, X., Qiao, Y., Ouyang, W., He, T., Zhao, H.: Point transformer v3: Simpler faster stronger. In: CVPR. pp. 4840–4851 (2024)
30. Xie, S., Gu, J., Guo, D., Qi, C.R., Guibas, L., Litany, O.: Pointcontrast: Unsupervised pre-training for 3d point cloud understanding. In: ECCV. pp. 574–591. Springer (2020)
31. Yan, Y., Mao, Y., Li, B.: Second: Sparsely embedded convolutional detection. Sensors (2018)
32. Yang, B., Abdelsamad, M., Zhang, M., Condurache, A.P.: Towards foundation models for 3d scene understanding: Instance-aware self-supervised learning for point clouds. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2026)
33. Yang, B., Condurache, A.P.: Collaborative learning for semi-supervised lidar semantic segmentation. In: International Conference on Machine Learning (ICML) (2026)

34. Yang, B., Condurache, A.P.: Flares: fast and accurate lidar multi-range semantic segmentation. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV). pp. 3451–3461 (2026)
35. Yang, B., Pfreundschuh, P., Siegwart, R., Hutter, M., Moghadam, P., Patil, V.: Tulip: Transformer for upsampling of lidar point clouds. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 15354–15364 (2024)
36. Yang, H., He, T., Liu, J., Chen, H., Wu, B., Lin, B., He, X., Ouyang, W.: Gd-mae: generative decoder for mae pre-training on lidar point clouds. In: CVPR. pp. 9403–9414 (2023)
37. Yang, Y., Feng, C., Shen, Y., Tian, D.: Foldingnet: Point cloud auto-encoder via deep grid deformation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 206–215 (2018)
38. Yin, J., Zhou, D., Zhang, L., Fang, J., Xu, C.Z., Shen, J., Wang, W.: Proposalcontrast: Unsupervised pre-training for lidar-based 3d object detection. In: European conference on computer vision (ECCV). pp. 17–33 (2022)
39. Yin, T., Zhou, X., Krähenbühl, P.: Center-based 3d object detection and tracking. CVPR (2021)
40. Yuan, W., Khot, T., Held, D., Mertz, C., Hebert, M.: Pcn: Point completion network. In: 2018 international conference on 3D vision (3DV). pp. 728–737 (2018)
41. Zhan, Y., Li, J., Sun, B., Wang, Y., Jia, K.: Csot: Cross-scan object transformer for 3d object detection in point clouds. In: European Conference on Computer Vision (ECCV). pp. 245–262 (2024)
42. Zhang, B., Song, N., Jin, X., Zhang, L.: Bridging past and future: End-to-end autonomous driving with historical prediction and planning. In: CVPR. pp. 6854–6863 (2025)
43. Zhang, Z., Girdhar, R., Joulin, A., Misra, I.: Self-supervised pretraining of 3d features on any point-cloud. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 10252–10263 (2021)
44. Zhao, H., Jiang, L., Jia, J., Torr, P.H., Koltun, V.: Point transformer. In: ICCV. pp. 16259–16268 (2021)
45. Zhou, Y., Tuzel, O.: Voxelnet: End-to-end learning for point cloud based 3d object detection. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR). pp. 4490–4499 (2018)
46. Zhu, H., Dong, Z., Topollai, K., Sha, B., Choromanska, A.: Self-supervised representation learning with joint embedding predictive architecture for automotive lidar object detection. In: Annual AAAI Conference on Artificial Intelligence (AAAI) (2026)

A Additional Implementation Details

For pretraining, we use two H200 GPUs in Distributed Data Parallel (DDP) mode, and for downstream fine-tuning, we use a single NVIDIA H200 GPU. We summarize all hyperparameter configurations in Tab. 10. For spatial clustering, k_{nn} and τ_d denote the neighbor count and maximum search radius used to construct a graph over PIT-normalized predicted centroids. A standard BFS algorithm decomposes the graph into connected components; components smaller than τ_{min} are discarded to remove noisy groupings, and each remaining component is treated as a pseudo-instance.

Table 10: Pretraining settings of *GhostPoint*

Config	**Value**
Optimizer	AdamW
Scheduler	Cosine annealing
Learning rate	2e-4
Weight decay	4e-2
Batch size	16
Mask Ratio	0.6
Mask Size	1 m
Warmup ratio	0.05
Teacher temperature	0.035
Student temperature	0.05
Training epochs	50
α_{zipf}	1.3
Warmup ratio of $\mathcal{L}_{\text{geo}}$	0.1
Warmup ratio of activating predictor G	0.1
λ	0.1
ks (Occupancy Dilation)	5
k (Token Initialization)	3
k_{nn} (Spatial Clustering)	20
τ_d (Spatial Clustering)	5 m
τ_{min} (Spatial Clustering)	10
Offset Direction Distribution (PIT-normalization)	Uniform(0, 1)
Offset Magnitude Distribution (PIT-normalization)	LogNormal(μ=1.4, σ=0.8)

B Additional Experiments

B.1 Runtime Analysis

Table 11 reports total pretraining time for *GhostPoint* compared to DOS [2] and PointINS [32], measured on the same hardware under identical training configurations. *GhostPoint* requires 26 hours, compared to 20 hours for PointINS

and 15 hours for DOS. The additional overhead relative to PointINS stems from the Neighborhood Sampling module and the ghost predictor G, which is incurred only during pretraining while downstream fine-tuning cost is identical to prior methods as both components are discarded. We consider this 30% overhead over the strongest prior method acceptable given the consistent improvements in downstream 3D detection reported in the main results.

Table 11: Runtime Analysis: we use the same hardware configuration across all models for fair comparison.

Method	Time
DOS [2]	15 hours
PointINS [32]	20 hours
GhostPoint	26 hours

Table 12: Effect of warmup ratio: we tested different set of warmup ratios for two stages. The performance remains consistent.

Warmup Ratio (first stage, second stage)	mAP	NDS
(0.1, 0.1)	59.5	64.2
(0.2, 0.2)	59.3	64.1
(0.1, 0.2)	59.4	64.2
(0.2, 0.1)	59.2	64.0

B.2 Effect of Warmup Ratio

Table 12 reports the sensitivity of *GhostPoint* to the two-stage warmup ratios. Performance remains stable across all tested configurations, with variations of less than 0.3 mAP and 0.2 NDS. The default configuration of (0.1, 0.1) achieves the best performance, and reducing or increasing either stage ratio has only marginal effect. These results confirm that *GhostPoint* is robust to the choice of warmup schedule as long as the two-stage order is preserved.

B.3 Effect of Subsample Ratio

Table 13 reports the effect of the subsample ratio of unobserved voxels. Performance peaks at 0.5 and degrades slightly at both extremes, revealing a coverage-quality trade-off. At low ratios, newly sampled voxels are concentrated near instance boundaries where dilation first activates, providing reliable but spatially limited supervision to cover the unobserved object extent sufficiently. At high ratios, the number of sampled voxels increases significantly in the unobserved regions, where the distillation targets become noisier. As default, we set the ratio at 0.5 to balance these two effects.

B.4 Out-of-Distribution (OOD) Generalization

Table 14 evaluates robustness to out-of-distribution (OOD) conditions on the Robo3D benchmark [15], which measures resilience to real-world corruptions such as weather effects, sensor noise, and point cloud degradation. We report mean Resilience Rate (mRR) and mean Corruption Error (mCE), where higher mRR and lower mCE indicate greater robustness. *GhostPoint* achieves the best

Table 13: Effect of subsample ratio: we test the different subsample ratio for sampling unobserved voxels in neighborhood sampling module.

Subsample Ratio	mAP	NDS
0.2	58.9	63.9
0.5	59.5	64.2
0.8	59.2	64.1

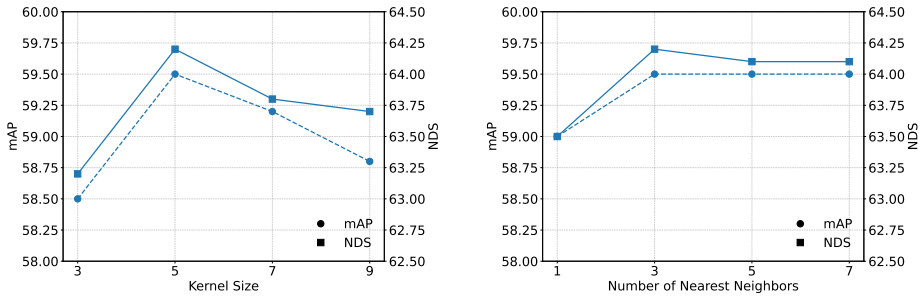
Table 14: OOD robustness on nuScenes-C: comparison of SSL methods under real-world corruptions. We use the models trained under decoder probing protocol and regard CenterPoint [39] as the unified baseline for computing metrics.

Method	mRR ↑	mCE ↓
SONATA [28]	96.1	88.6
DOS [2]	96.3	71.2
PointINS [32]	96.7	70.4
GhostPoint	**97.5**	**68.8**

performance on both metrics, with 97.5 mRR and 68.8 mCE, outperforming all prior SSL methods by a substantial margin on mCE. This result suggests that explicitly modeling unobserved object geometry during pretraining not only improves detection under clean conditions but also produces representations that are more resilient to input corruptions.

B.5 Effect of Kernel Size ks and k for Token Initialization

Figure 7 reports the sensitivity of *GhostPoint* to two key hyperparameters. For the dilation kernel size ks, performance peaks at $ks = 5$ and degrades gradually for larger values, suggesting that moderate dilation captures sufficient unobserved object structure without introducing excessive background voxels that dilute the instance signal. For the neighbor count k, performance improves from $k = 1$ to $k = 3$ and remains stable beyond that, indicating that aggregating features from a small local neighborhood provides sufficient context for token initialization and that additional neighbors offer diminishing returns. Based on these results, we adopt $ks = 5$ and $k = 3$ as default configurations, both of which correspond to the peak or plateau of their respective sensitivity curves.



(a) Occupancy Dilation (ks) **(b)** Token Initialization (k)

Fig. 7: Sensitivity analysis on two key hyperparameters: the dilation kernel size ks, which controls the spatial extent of occupancy dilation around pseudo-instance proposals, and the neighbor count k, which determines how many observed encoder features are aggregated via KNN interpolation to initialize tokens at unobserved voxels.