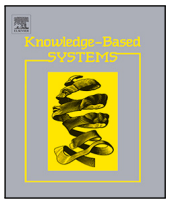
Knowledge-Based Systems 307 (2025) 112741

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys



Mutual information-driven self-supervised point cloud pre-training

Weichen Xu [a,1], Tianhao Fu [a,1], Jian Cao [a,*], Xinyu Zhao [a], Xinxin Xu [a], Xixin Cao [a], Xing Zhang [a,b]

[a] School of Software and Microelectronics, Peking University, Beijing 100871, China
[b] Key Lab of Integrated Microsystems, Peking University Shenzhen Graduate School, Shenzhen 518055, China

ARTICLE INFO

Dataset link: https://gpicture-page.github.io/

Keywords:
Self-supervised learning
Autonomous driving
Point cloud scene understanding
Mutual information
High-level features

ABSTRACT

Learning universal representations from unlabeled 3D point clouds is essential to improve the generalization and safety of autonomous driving. Generative self-supervised point cloud pre-training with low-level features as pretext tasks is a mainstream paradigm. However, from the perspective of mutual information, this approach is constrained by spatial information and entangled representations. In this study, we propose a generalized generative self-supervised point cloud pre-training framework called GPICTURE. High-level features were used as an additional pretext task to enhance the understanding of semantic information. Considering the varying difficulties caused by the discrimination of voxel features, we designed inter-class and intra-class discrimination-guided masking (I^2Mask) to set the masking ratio adaptively. Furthermore, to ensure a hierarchical and stable reconstruction process, centered kernel alignment-guided hierarchical reconstruction and differential-gated progressive learning were employed to control multiple reconstruction tasks. Complete theoretical analyses demonstrated that high-level features can enhance the mutual information between latent features and high-level features, as well as the input point cloud. On Waymo, nuScenes, and SemanticKITTI, we achieved a 75.55% mAP for 3D object detection, 79.7% mIoU for 3D semantic segmentation, and 18.8% mIoU for occupancy prediction. Specifically, with only 50% of the fine-tuning data required, the performance of GPICURE was close to that of training from scratch with 100% of the fine-tuning data. In addition, consistent visualization with downstream tasks and a 57% reduction in weight disparity demonstrated a better fine-tuning starting point. The project page is hosted at https://gpicture-page.github.io/.

1. Introduction

LiDAR has received widespread attention because of its ability to simulate the depth and spatial distribution of the environment. In outdoor autonomous driving (AD), Advanced supervised learning algorithms are implemented based on LiDAR to achieve 3D perception and improve vehicle safety [1], such as PointPillars [2], PV-RCNN [3], and SphereFormer [4]. However, obtaining a large amount of carefully annotated 3D data, such as 3D boxes or semantic categories, is expensive and time-consuming [5]. The limited labeled point clouds restrict further improvements in the performance of supervised learning and constrain advanced models to fit specific scenarios [6]. Learning universal representations of 3D point clouds from extensive unlabeled data is a promising solution to alleviate these problems.

Generative self-supervised point cloud pre-training, represented by masked autoencoders [8–12], has attracted attention in autonomous driving. Similar to images [8], self-supervised representations are utilized for weight initialization in downstream supervised learning. The fundamental challenge in this topic is identifying signals to learn features from unlabeled point clouds. Previous studies have explored meaningful directions. Occupancy-MAE [13] predicts whether each masked voxel is occupied. This enhances scene understanding by inferring the compositions of 3D scenes. GD-MAE [14] reconstructs the 3D coordinates of point clouds in each masked voxel. GeoMAE [7] further introduces geometric features to enhance spatial understanding. MV-JAR [15] combines jigsaw and masked autoencoders to understand the spatial relationships between voxels. Overall, previous methods used low-level features as pretext tasks, such as unimodal features, simple physical properties, or spatial composition information.

Because representations are transferred across pretext tasks, the pre-training process, and downstream tasks, mutual information is suitable for measuring the amount of information shared between the features. Relevant studies have utilized mutual information as a measure of representation learning in various fields such as image [16], speech [17], video [18], and anomaly detection [19]. Therefore, we employed mutual information to analyze issues within the existing paradigm. We

* Corresponding author.
E-mail addresses: xuweichen1999@stu.pku.edu.cn (W. Xu), tianhaofu1@stu.pku.edu.cn (T. Fu), caojian@ss.pku.edu.cn (J. Cao).
[1] The authors contributed equally to this work.

https://doi.org/10.1016/j.knosys.2024.112741
Received 28 August 2024; Received in revised form 23 October 2024; Accepted 12 November 2024
Available online 20 November 2024

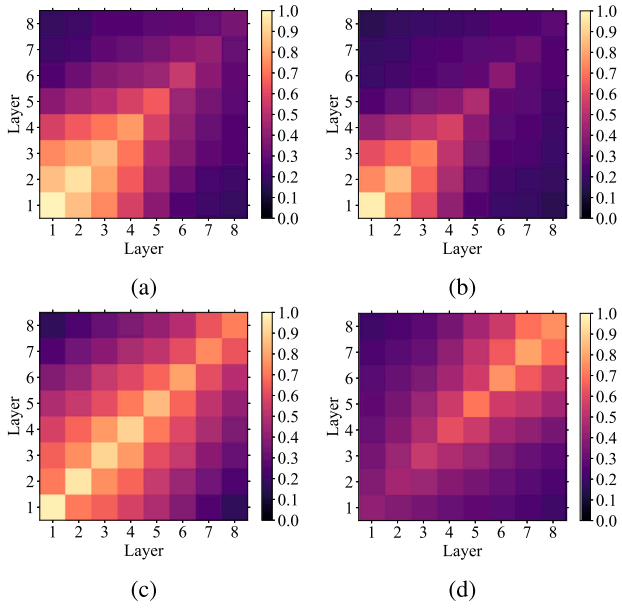


Fig. 1. CKA between the encoder from the SSL method GeoMAE [7] and the encoder from SL for (a) 3D object detection and (b) 3D semantic segmentation for the same input and CKA between two frames from the (c) same scene and (d) different scenes when using the same encoder.

identified two issues with the current paradigm in generative self-supervised point cloud pre-training: 1. The pre-training process, guided by low-level features, focuses on learning the discrimination of spatial information while neglecting semantic information; 2. Low information content in pretext tasks leads to entangled representations. Specifically, we employed the centered kernel alignment (CKA) [20] proposed by Hinton as a metric for measuring mutual information. Without loss of generality, we considered the non-empty voxel output by each DSVT [21] encoder layer as the subject. First, we computed the layer-wise CKA between the outputs of two identical encoders obtained using different methods for the same input, as shown in Fig. 1(a) and (b). These represent the CKA between the encoder from the self-supervised learning (SSL) method GeoMAE [7] and the encoder from supervised learning (SL) for 3D object detection and 3D semantic segmentation. It can be observed that CKA is stronger in the shallow layers but drops sharply in the deeper layers. Moreover, this decline is more pronounced for semantic segmentation. Related studies [9,22] indicate that point cloud encoders follow hierarchical feature learning, in which shallow layers learn local geometric features and spatial information, whereas deep layers capture high-level semantic information related to object categories. This suggests that the current paradigm primarily learns spatial information and neglects semantic information, which is detrimental to downstream tasks that require semantic information. We believe that the low-level features represented by **PoInt Cloud** reconsTruction are insUfficient for learning 3D **RE**presentations in autonomous driving (dubbed **PICTURE**). To address this issue, the mutual information between latent features and semantic features needs to be enhanced. On the other hand, we computed the layer-wise CKA for different inputs after they passed through the encoder with the same network, as shown in Fig. 1(c) and (d). They represent the CKA between two frames from the same and different scenes when using the encoder from the SSL method GeoMAE [7]. The features across all layers exhibit high similarity between two frames from the same scene. For the two frames from different scenes, the low CKA values in the shallow layers indicate significant differences between the two input scenes. However, the similarity of the latent features in deeper layers are comparable to those in the same scene. This suggests that various point clouds are encoded as entangled representations. To address this issue, the mutual information between latent features and the input point cloud needs to be enhanced.

Related studies [23–26] have indicated that multitask optimization [27] and multiform optimization [28] in point cloud processing can effectively leverage knowledge from different tasks and features to achieve better optimization. In contrast to low-level features, high-level features are defined as classical or frequency domain features, multimodal, or semantic features. SLidR [29] and Seal [30] explore collaborative learning on point cloud-image pairs obtained from superpixels [31] or SAM [32]. CLIP^2 [33] follows the architecture of CLIP [34] and performs triple contrastive learning between images, point clouds, and text. Our previous research PICTURE [35] formulates generative point cloud self-supervised pre-training as a multiform optimization problem and supplements the high-level feature reconstruction task. To better transfer semantic information and learn disentangled representations, we further propose a generalized generative self-supervised point cloud pre-training framework, **GPICTURE**, to systematically learn 'what', 'how', 'where', 'when', and 'why' to exploit high-level features. First, MinkUNet pre-trained by Seal [30] acted on raw point clouds. The point cloud features within the identical voxel were aggregated to obtain the Seal voxel features of the masked voxel. We considered reconstructing Seal voxel features as an additional pretext task, which benefited from image supervision and encouraged the model to learn high-level vision concepts. Second, previous mask sampling strategies have neglected the varying difficulties caused by the discrimination of voxel features. The blocked information flow between latent features and high-level features affects the learning of semantic information discrimination. Thus, we proposed inter-class and intra-class discrimination-guided masking (I^{2}Mask). Inter-class discriminative cues and the intra-class consistency coefficient of the high-level voxel features were used to set the masking ratio. Third, to ensure that the pre-training process focuses on learning semantic information under the condition of hierarchical feature learning, we proposed CKA-guided hierarchical reconstruction (CHR) to weight the low-level and high-level reconstruction losses for each layer. Thus, the reasonable task configuration avoids confusion in multiform optimization. Fourth, inspired by curriculum learning [36] and two-stage knowledge transfer strategy [23], we proposed differential-gated progressive learning (DGPL) to adjust the initiation timing of high-level feature reconstruction based on the completion of low-level feature reconstruction. This approach makes the high-level feature reconstruction process more stable. Finally, we provided a comprehensive theoretical analysis demonstrating that supplementing high-level features with deep supervision as pretext tasks could enhance the mutual information between latent features and high-level features, as well as the input point cloud.

We conducted experiments on Waymo [37], nuScenes [38], and SemanticKITTI [39] to validate promotion for 3D object detection, 3D semantic segmentation, and occupancy prediction. The results demonstrated state-of-the-art performance, such as a 58.8% relative increase in 3D object detection. Comprehensive analyses, including hyperparameter tuning, ablation studies, multiple downstream task studies, cross-dataset evaluations, and time cost assessments, demonstrated the generalizability of GPICTURE and the importance of all components. In addition, visualization and exploratory model analysis intuitively showcased the effectiveness of I^{2}Mask, the learning process, and the optimization impact of high-level features on model weights.

The main contributions of this paper are as follows:

- We proposed a generalized generative self-supervised point cloud pre-training framework, GPICTURE, to systematically learn to exploit high-level features. Seal voxel features with strong semantic information served as additional reconstruction targets and guided the mask sampling strategy. Moreover, mutual information-driven CHR and DGPL were utilized to ensure the hierarchical and progressive reconstruction of high-level features.

- We conducted a theoretical analysis showing that incorporating high-level features as pretext tasks, along with I^{2}Mask, CHR, and DGPL, effectively strengthened the mutual information between latent features and high-level features, as well as the input point cloud.
- We achieved relative improvements of 58.8%, 39.1%, and 66.7% compared with advanced self-supervised methods in downstream tasks such as 3D object detection, 3D semantic segmentation, and occupancy prediction, respectively. We demonstrated the benefits of this method through extensive experimentation.

The remainder of this paper is organized as follows. In Section 2, we discuss related studies. The proposed framework GPICTURE and the theoretical analysis are presented in Sections 3 and 4. Section 5 provides comprehensive quantitative and qualitative results. Sections 6 and 7 present limitations and future work of our methods. Finally, Section 8 summarizes this paper.

2. Related work

2.1. Architectures for point cloud processing

Data sparsity and geometric irregularity of the point cloud present unique challenges in architectural design [40]. Existing point cloud architectures are divided into point-based and voxel-based methods with regard to intermediate representations [41]. Point-based approaches derive features from raw data using point-wise feedforward networks [40,42–47] or transformers [48–55]. However, neighborhood search limits the efficiency of these methods [21]. Voxel-based methods map unstructured point clouds onto structured voxels. Owing to the sparsity of point clouds, sparse convolution [3,56–61] or sparse voxel transformers [21,62–66] are employed to process non-empty voxels. For point-based methods such as PointNet++ [42] and sparse convolution-based methods such as SECOND [56] and Minkowski networks [58], deployment in real-world applications is challenging. Modules like submanifold sparse convolution [57] require custom CUDA operators. In contrast, sparse voxel transformer-based methods such as SST [63] and DSVT [21] aggregate features by computing the attention between voxels, which is the deep architecture that this study focuses on.

2.2. Self-supervised learning in autonomous driving

Learning universal features from large-scale point clouds in a self-supervised manner attracts attention in the autonomous driving community [9–12]. Similar to 2D images, contrastive and generative self-supervised learning are the two mainstream approaches [9,11]. In contrastive self-supervised learning [29,30,33,67–69], ProposalContrast [67] applies random geometric transformations under two views to construct training samples. SLidR [29] and Seal [30] perform contrastive learning between the corresponding images and point clouds using SLIC [31] or SAM [32]. CLIP^2 [33] follows the architecture of CLIP [34] and performs triple contrastive learning between images, point clouds, and text. In generative self-supervised learning [7,14,15,70–72], point cloud reconstruction is considered a crucial avenue to understand point cloud scenes. 3D coordinates [14,15] and other physical properties [7,71,72] are used as reconstruction targets. This method encourages the model to understand the details and distribution of 3D point clouds, which is the focus of this study. Recently, drawing inspiration from language models and content generation, some prediction-based and multi modality-based approaches [11] have been proposed. ViDAR [73] forces the model to forecast upcoming events based on past data and monitors dynamic environmental flow and object trajectory. UniPAD [74] synthesizes 2D RGB-D images from raw sparse point clouds. In contrast to fields such as image [16] and speech [17], mutual information-driven design has never been introduced in point cloud in autonomous driving.

2.3. Pretext tasks for generative SSL in autonomous driving

Pretext tasks are exploited by generative SSL methods to extract information from unlabeled datasets [9,10]. The most common pretext task is reconstructing the 3D coordinates in each masked voxel [14,15,72,75]. Some studies [7,13,76,77] have predicted whether each masked voxel is occupied, which improves the ability to reconstruct the complete point cloud from partial observations. In addition, other spatial information have been used for pretext tasks, such as geometric features [7], jigsaw [15], occupation type [71,72], and number of point clouds [72,75]. However, current pretext tasks in autonomous driving still focus on reconstructing low-level spatial information. Semantic features in high-level, such as multimodal or semantic features, have not been introduced as pretext tasks. We believe that spatial information in low-level is insufficient.

2.4. Mask sampling strategy in generative self-supervised learning

The mask sampling strategy determines the location of the pretext tasks [9,10]. Most studies [7,15,72,77] have used random masking for 3D point clouds in autonomous driving. However, this simple yet successful method for 2D images does not perform well when directly applied to 3D point clouds because of the different sparsity levels. MAELi [71] propose that the masking ratio should decrease with distance from the ego, which ensures a consistent sparsity to improve the generalization. BEV-MAE [75] applies a lower masking ratio in sparse areas to ensure a more stable reconstruction process. GD-MAE [14] implements separate masking strategies at different granularities to maintain a consistent masking scope. However, mask sampling strategies based on feature attributes are not generally applied. Hard samples block the information flow between unmasked features and masked features. In 2D images, AttMask [78] employs the attention map to mask highly attended areas are masked, making the reconstruction more efficient. SemMAE [79] and AMT [80] use iteration and throwing to focus the attention map more on objects. Inspired by this, we developed a mask sampling strategy based on the attributes of the target high-level voxel features to adjust the difficulty and focus. Theoretical evidence demonstrated that the smoother information flow between the unmasked and masked features increased the mutual information between latent features and high-level features.

3. Proposed method

In this section, we introduce the generalized generative self-supervised point cloud pre-training framework GPICTURE, as shown in Fig. 2. Here, we elaborate on ‘what’ and ‘how’ to exploit high-level features in the high-level voxel feature generation module (Section 3.2) and inter-class and intra-class discrimination-guided masking (Section 3.3). Then, we introduce the mutual information-driven CKA-guided hierarchical reconstruction (Section 3.4) and differential-gated progressive learning (Section 3.5) to demonstrate ‘where’ and ‘when’ to exploit high-level features.

3.1. Preliminaries of generative self-supervised point cloud pre-training

Voxelization and Masking. The point cloud can be defined as $\boldsymbol{P} = \left\{ \boldsymbol{c}_\ell, \boldsymbol{f}_\ell \middle| \ell = 1, \ldots, N_\mathrm{p} \right\}$, where $\boldsymbol{c}_\ell$ and $\boldsymbol{f}_\ell$ represent coordinates and features. Voxel Feature Encoding (VFE) [56] was used for voxelization. The coordinates and features of all non-empty voxels are defined as $\boldsymbol{C}_\mathrm{all} \in \mathbb{R}^{N_\mathrm{v} \times 3}$, $\boldsymbol{F}_\mathrm{all} \in \mathbb{R}^{N_\mathrm{v} \times C}$. The masking sampling strategy is described in Section 3.3. The coordinates of the mask voxels are denoted as $\boldsymbol{C}_\mathrm{m}$. The coordinates and features of unmasked voxels are denoted as $\boldsymbol{C}_\mathrm{um}$, $\boldsymbol{F}_\mathrm{um}$.

Sparse Encoder and Decoder. To reduce the computational complexity of attention, sparse voxel transformer-based methods SST [63] and DSVT [21] design grouping mechanisms for sparse voxels. The

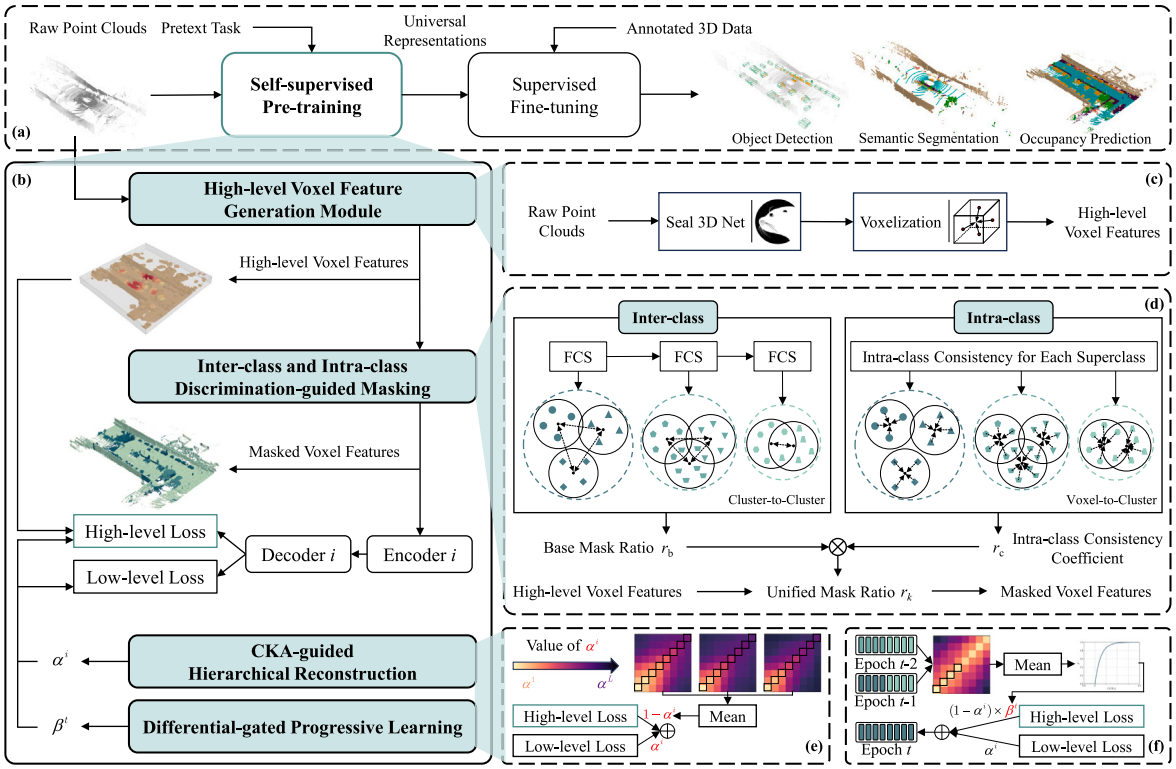


Fig. 2. (a) The self-supervised learning process in 3D point clouds. (b) Architecture overview of the generalized generative self-supervised point cloud pre-training framework GPICTURE. The raw 3D point clouds are fed into (c) the high-level voxel feature generation module to obtain high-level voxel features. (d) Inter-class and intra-class discrimination-guided masking strategy is applied based on the attributes of high-level voxel features. (e) CKA-guided hierarchical reconstruction and (f) differential-gated progressive learning are used to weight the low-level and high-level reconstruction losses spatially and temporally.

coordinates $\boldsymbol{C}_{\text{um}}$ and feature $\boldsymbol{\mathcal{F}}_{\text{um}}$ of unmasked voxels were input into DSVT to extract features. The features of masked voxels were replaced by mask tokens and input into the decoder together with the output of the encoder. We also used DSVT as the decoder. Thus, the features of the masked voxels can be reconstructed.

3.2. High-level voxel feature generation module

The reconstruction of spatial features is insufficient for downstream tasks requiring semantic information. Related studies [23–26] indicate that multiform optimization [28,81] in point cloud processing can effectively leverage knowledge from different forms to achieve better optimization. Therefore, we modeled generative point cloud self-supervised pre-training as a multiform optimization problem and supplemented the high-level feature reconstruction task to transfer semantic information. Seal [30] transfers image features from pre-trained vision foundation models to sparse convolution-based network MinkUNet [58]. This provides favorable semantic feature reconstruction signals. The Seal feature heatmaps [30] for example point cloud scene in nuScenes [38] are shown in Fig. 3(a). At the same time, the ground truth of the same scene in three downstream tasks are visualized in Fig. 3(b). It can be seen Seal features and the real scene have high semantic consistency. The Seal feature is strong for road-related objects. In addition, the feature decreases with distance from ego. The visualization demonstrates that Seal features have high mutual information with semantic annotations and serve as an effective encoding of semantic information. Based on this observation, we considered reconstructing high-level features, specifically Seal voxel features, as an additional pretext task, as shown in Fig. 2(c).

Specifically, all raw non-voxelized point clouds $\boldsymbol{\mathcal{P}}$ were fed into MinkUNet (Res16UNet34C) [58] pre-trained by Seal. Each point cloud aggregated spatial and feature information from other point clouds.

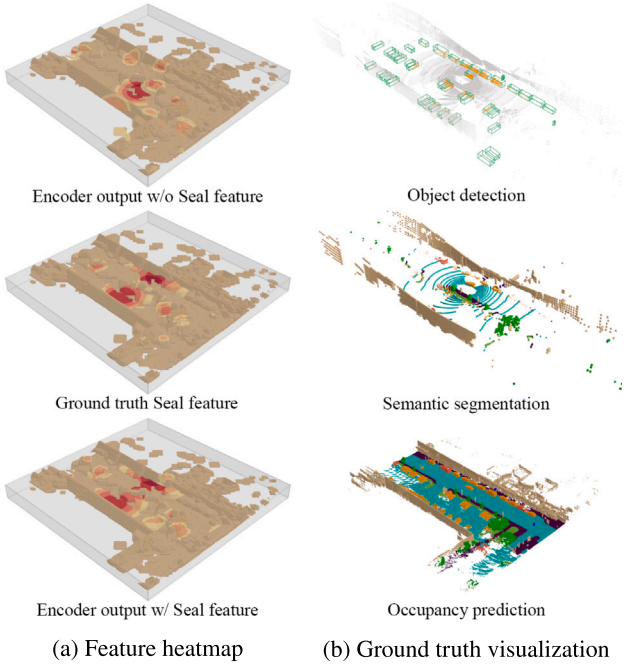


Fig. 3. (a) Heatmap for ground truth Seal features and encoder output pre-trained w/o and w/ Seal features. (b) Ground truth visualization for 3D object detection, 3D semantic segmentation, and occupancy prediction.

Thus, we obtained the Seal point features $\boldsymbol{\mathcal{P}}_{\text{s}}$. Subsequently, voxelization and average pooling were employed to aggregate sparse features located in the same voxel. Finally, we obtained the Seal voxel features $\boldsymbol{S}_{\text{all}} = \left\{ \boldsymbol{s}_j \middle| j = 1, \ldots, N_{\text{v}} \right\}$ of all non-empty voxels. This process can be

completed offline in advance to accelerate training. For masked voxels, the loss between the reconstructed and target Seal voxel features is defined as:

$$L_{\text{High}} = \text{SmoothL1}\left(\mathcal{S}_{\text{m}}^{\text{pred}}, \mathcal{S}_{\text{m}}^{\text{target}}\right). \tag{1}$$

Fig. 3(a) shows the heatmap extracted by the encoder before and after introducing Seal voxel features. After introducing Seal voxel features as pretext tasks, the latent features extracted by the encoder have higher mutual information with high-level features, which is crucial for downstream tasks. Furthermore, the output features are more aligned with downstream tasks than Seal features. In addition, I-JEPA [82] show that reconstructing high-level features can alleviate representation collapse.

3.3. Inter-class and intra-class discrimination-guided masking

The mask sampling strategy influences the information flow from unmasked features to masked features, further impacting the mutual information between latent features and pretext tasks. Previous mask sampling strategies neglect the varying difficulties caused by the discrimination of voxel features. The blocked information flow between latent features and high-level features affects the learning of semantic information discrimination. We developed a mask sampling strategy based on the attributes of the target high-level voxel features to adjust the difficulty and focus, as shown in Fig. 2(d). For 22 categories in SemanticKITTI [39], 17 categories in nuScenes Lidarseg [83], and 22 categories in Waymo Semantic Segmentation [37], we first classified them into eight semantic classes: ground-related, structures, vehicle, two-wheeled vehicle, nature, human, object, and outlier. Semantic labels are agnostic in self-supervised learning. Therefore, we clustered high-level voxel features $\mathcal{S}_{\text{all}}$ using k-means [84], and the number was set to 8. Thus, we obtained eight superclasses k_i. Here, the superclasses are not semantic classes but different research regions in 3D space. Although it is impossible to determine the correspondence between k_i and semantic classes, this partition considered the semantic attributes of the high-level voxel features.

We set the base masking ratio for each superclass based on the inter-class discrimination. First, we computed the cluster center $\boldsymbol{\mu}^{k_i}$, i.e., the mean of high-level voxel features:

$$\boldsymbol{\mu}^{k_i} = \frac{\sum_{j=1}^{N_{k_i}} \boldsymbol{s}_j^{k_i}}{N_{k_i}}, \tag{2}$$

where N_{k_i} is the number of voxels of k_i. Next, we performed Algorithm 1 Fastest Class Sampling on clustering centers $\boldsymbol{\mu}^k$. We defined the set $\mathcal{K} = \{k_i \mid i = 1, 2, \dots, 8\}$. Then we selected any n_1 superclasses from $\mathcal{K}$ and calculated the distance between the cluster centers. The set $\mathcal{D}_{\text{inter}}$ composed of average inter-class distances is:

$$\mathcal{D}_{\text{inter}} = \left\{ \frac{1}{\binom{n_1}{2}} \sum_{m=1}^{n_1-1} \sum_{n=m+1}^{n_1} d\left(\boldsymbol{\mu}^{k_m}, \boldsymbol{\mu}^{k_n}\right) \;\middle|\; \left\{k_1, k_2, \dots, k_{n_1}\right\} \in \binom{\mathcal{K}}{n_1} \right\}, \tag{3}$$

where $d\left(\boldsymbol{\mu}^{k_m}, \boldsymbol{\mu}^{k_n}\right)$ represents the distance between the clustering centers of the two superclasses, e.g., cosine similarity. Next, we identified the n_1 superclass partition with the fastest average inter-class distance, denoted as $\mathcal{K}_1$. Subsequently, the difference between the original set $\mathcal{K}$ and $\mathcal{K}_1$ was used as the input for the next cycle. Then, we selected any n_2 superclasses from the remaining set, repeated the above process, and finally obtained the n_2 superclass partition with the second fastest average inter-class distance, denoted as $\mathcal{K}_2$. The average inter-class distance of the remaining set was the smallest, denoted as $\mathcal{K}_3$.

The average inter-class distance reflects inter-class discrimination. Larger inter-class differences facilitate a more straightforward distinction and reconstruction, and are typically associated with dynamic objects such as vehicles. We set a higher base mask ratio to increase the difficulty and focus on reconstruction. Based on this principle, we set the base mask ratio r_{b}^1, r_{b}^2, r_{b}^3 for each set.

Algorithm 1 Fastest Class Sampling

Input: The set of all superclasses: $\mathcal{K}$
Parameter: Expected number of partition n_1, n_2, n_3
Output: The set of superclass partition $\mathcal{K}_1, \mathcal{K}_2, \mathcal{K}_3$

```
Let K_1, K_2, K_3 = {}, t = 1.
while t ≤ 3 do
    {k_1, k_2, ..., k_{n_t}} ← select any n_t superclass from K.
    compute the distance between clustering centers of any two superclasses: d(μ^{k_m}, μ^{k_n}) = ½(1 − μ^{k_m} μ^{k_n}).
    compute set of average inter-class distances D_inter.
    K_t ← select superclass partition {k_1, k_2, ..., k_{n_t}} with the fastest average inter-class distance max_{d∈D} d.
    K = K \ K_t.
    t = t + 1.
end while
return K_1, K_2, K_3
```

Intra-class discrimination reflects the difficulty of reconstructing masked voxel features from unmasked voxels. We defined the set of average intra-class distance $\mathcal{D}_{\text{intra}}$:

$$\mathcal{D}_{\text{intra}} = \left\{ \frac{\sum_{j=1}^{N_{k_i}} \mathbb{1}\left\{ d\left(\boldsymbol{\mu}^{k_i}, \boldsymbol{s}_j^{k_i}\right) - \lambda > 0 \right\} d^2\left(\boldsymbol{\mu}^{k_i}, \boldsymbol{s}_j^{k_i}\right)}{\sum_{j=1}^{N_{k_i}} \mathbb{1}\{ d\left(\boldsymbol{\mu}^{k_i}, \boldsymbol{s}_j^{k_i}\right) - \lambda > 0 \}} \;\middle|\; i \in \{1, 2, \dots, 8\} \right\}, \tag{4}$$

where $d\left(\boldsymbol{\mu}^{k_i}, \boldsymbol{s}_j^{k_i}\right)$ represents the distance between the high-level features of a voxel and its clustering center. The $\mathbb{1}\{\}$ stands for indicator function. The denominator of Eq. (4) represents the number of voxels whose distance from the center exceeds threshold λ. The meaning of $\mathcal{D}_{\text{intra}}$ is average distance that is too far from the center. This reflects the consistency between high-level voxel features in a superclass.

We then defined the intra-class consistency coefficient r_{c_i} to reflect the intra-class discrimination:

$$r_{\text{c}_i} = 1 - \frac{\mathcal{D}_{\text{intra}}^i}{\max_{d \in \mathcal{D}_{\text{intra}}} d}. \tag{5}$$

The intra-class consistency coefficient r_{c_i} exhibits a trend opposite to average intra-class distance $\mathcal{D}_{\text{intra}}^i$. It is easier to reconstruct masked voxel features from unmasked voxels in regions with high intra-class consistency coefficient r_{c_i}. Therefore, we used r_{c_i} to modulate the base masking ratio to obtain the unified masking ratio r_{k_i} for each superclass:

$$r_{k_i} = \left(r_{\text{b}}^1 \cdot \mathbb{1}\left\{k_i \in \mathcal{K}_1\right\} + r_{\text{b}}^2 \cdot \mathbb{1}\left\{k_i \in \mathcal{K}_2\right\} + r_{\text{b}}^3 \cdot \mathbb{1}\left\{k_i \in \mathcal{K}_3\right\}\right) \cdot r_{\text{c}_i}. \tag{6}$$

The unified mask ratio was calculated for each superclass based on its inter-class and intra-class discrimination. Compared to attention-guided methods such as AttMask [78], SemMAE [79], and AMT [80], our approach allows high-level features to determine the masking ratio, which is consistent with reconstruction process and interpretable. Moreover, by adjusting the difficulty and focus, the smoother information flow between unmasked and masked features enhances the mutual information between latent features and high-level features.

3.4. CKA-guided hierarchical reconstruction

Because the shallower layers of the encoder receive weaker information feedback from the supervision signal, several studies [85,86] have introduced deep supervision to provide explicit guidance to the shallow layers. Inspired by this, we added deep supervision to the generative self-supervised point cloud pre-training. For the reconstruction of non-final layers, only half the number of decoder layers were employed to avoid high computational load. Additionally, related studies [9,

22] have indicated that point cloud encoders follow hierarchical feature learning, where shallow layers learn local geometric features and spatial distribution information, while deep layers capture high-level semantic information. It is unsuitable for all layers to perform the same reconstruction task and knowledge transfer. To tackle this problem, we proposed a multi-task weighting strategy, CKA-guided hierarchical reconstruction, to weight the low-level and high-level reconstruction losses for each layer, as shown in Fig. 2(e). In this way, the reasonable task configuration avoids confusion during knowledge transfer.

Specifically, we computed the layer-wise CKA between the encoder from the self-supervised learning method GeoMAE [7] and the encoder from supervised learning for 3D object detection, 3D semantic segmentation, and occupancy prediction. This represents the contribution of the low-level feature reconstruction to the desired weight. Thus, we defined the layer-wise mean of the CKA for the three downstream tasks as the proportion of low-level information α^i $(i = 1, 2, \ldots, L)$. Accordingly, we assigned the weights for low-level and high-level reconstruction as $w_{\mathrm{L}}^i = \alpha^i$ and $w_{\mathrm{H}}^i = 1 - \alpha^i$. Our approach fully considers the deficiencies in mutual information between the existing paradigm GeoMAE and supervised learning. This ensures that the shallow layers primarily learn low-level features, such as local geometric features, while the deep layers focus on learning high-level features, such as semantic features. The pre-training process with CHR not only focuses on learning semantic information, but also avoids confusion in multiform optimization. In addition, CHR accelerates convergence owing to reasonable hierarchical learning.

3.5. Differential-gated progressive learning

The two reconstruction tasks exhibit different levels of difficulty and progressive objectives. The reconstruction of high-level features fundamentally depends on the accurate reconstruction of low-level features, such as local geometric features. The error amplification caused by parallel learning leads to instability in the reconstruction of high-level features. Inspired by curriculum learning [36] and two-stage knowledge transfer strategy [23], we proposed a multi-task switching strategy, differential-gated progressive learning, to adjust the initiation timing of high-level feature reconstruction based on the completion of low-level feature reconstruction, as shown in Fig. 2(f). This enhances the stability and efficiency of the high-level feature reconstruction.

The activation differential between two adjacent epochs can reflect the degree of training completion. A higher activation differential indicates that convergence has not yet been achieved, whereas a lower activation differential signifies that the weight updates are stabilizing. Therefore, we used the measure of mutual information, CKA, between epochs $t-2$ and $t-1$ to calculate the activation differential. In hierarchical feature learning [22], shallow layers learn local geometric features and other low-level features. Therefore, we calculated the average CKA of the first four layers as the completion degree of low-level feature reconstruction (CLFR) c_{r}^t at the current epoch t. Specifically, we defined the CLFR for the first two epochs as 0. Subsequently, we defined the initiation timing factor β^t:

$$\beta^t = \begin{cases} 0, & 0 \le c_{\mathrm{r}}^t < \delta \\ 1 - \mathrm{e}^{-\gamma\left(c_{\mathrm{r}}^t - \delta\right)}, & \delta \le c_{\mathrm{r}}^t \le 1 \end{cases}, \tag{7}$$

where γ and δ are the slope factor and gating factor. The initiation timing factor β^t is positively correlated with the completion degree of low-level feature reconstruction and is within the range of 0 to 1. We utilized β^t to further adjust the weight of the high-level feature reconstruction loss w_{H}^i. High-level feature reconstruction is initiated only when the CLFR c_{r}^t exceeds the gating factor δ, which enhances the stability of the training process.

4. Theoretical analysis

Information theory [87] provides a foundational framework for quantifying information and understanding systems. Mutual information, a key concept in this field, measures the amount of information shared between variables. Therefore, we provided a theoretical analysis from the perspective of the InfoMax principle [88] and mutual information. We conclude that supplementing high-level features with deep supervision and the masking sampling strategy will increase the mutual information between latent features and high-level features. Meanwhile, high-level features with deep supervision will also enhance the lower bound of mutual information between latent features and the input point cloud. Moreover, the multi-task control strategies CHR and DGPL lead to a greater impact of high-level features with deep supervision on deep layers.

Problem Formulation. The generative self-supervised learning network with deep supervision consists of encoders, decoders, and head networks. Let $\mathcal{X}$ denote the set of input point clouds. The encoder extracts the unmasked input point cloud $\mathcal{X}_{\mathrm{um}}$ into unmasked latent features $\mathcal{Z}_{\mathrm{um}}^i$ (with i being the layer index, $i = 1, 2, \ldots, L$). The decoder reconstructs masked features $\mathcal{Z}_{\mathrm{m}}^i$ based on unmasked features $\mathcal{Z}_{\mathrm{um}}^i$. Then, the masked features are utilized to reconstruct pretext tasks $\mathcal{Y}$ through the head network. The pretext tasks $\mathcal{Y}$ consist of two parts, which include reconstructing low-level features $\mathcal{Y}_{\mathrm{L}}$ and high-level features $\mathcal{Y}_{\mathrm{H}}$. The information flow is $\mathcal{X} \to \mathcal{Z}_{\mathrm{um}}^i \to \mathcal{Z}_{\mathrm{m}}^i \to \widehat{\mathcal{Y}} \Longleftrightarrow \mathcal{Y}$. Once the generative self-supervised learning is completed, the decoder and head network are discarded, and the encoder along with the latent features $\mathcal{Z}^i$ are used for fine-tuning in downstream tasks. Here, we primarily investigate the impact of reconstructing high-level features $\mathcal{Y}_{\mathrm{H}}$ as a pretext task on pre-training.

Proposition 1. *The high-level features $\mathcal{Y}_{\mathrm{H}}$ and the input point cloud $\mathcal{X}$ exhibit higher mutual information.*

We use mutual information to represent the correlation between $\mathcal{Y}$ and $\mathcal{X}$:

$$I(\mathcal{Y}; \mathcal{X}) = \iint p(y|x)\, p(x) \log \frac{p(y|x)}{p(y)} \mathrm{d}x\mathrm{d}y, \tag{8}$$

where $p(y)$ is the distribution of $\mathcal{Y}$ given $p(y|x)$:

$$p(y) = \int p(y|x)\, p(x)\, \mathrm{d}x. \tag{9}$$

We further transform Eq. (8) into:

$$\begin{aligned} I(\mathcal{Y}; \mathcal{X}) &= \iint p(y|x)\, p(x) \log \frac{p(y|x)\, p(x)}{p(y)\, p(x)} \mathrm{d}x\mathrm{d}y \\ &= D_{\mathrm{KL}}\left(p(y|x)\, p(x) \parallel p(y)\, p(x)\right), \end{aligned} \tag{10}$$

where D_{KL} represents the Kullback–Leibler (KL) divergence [89]. Eq. (10) indicates that the mutual information $I(\mathcal{Y}; \mathcal{X})$ is positively correlated with the distance between $p(y|x)\, p(x)$ and $p(y)\, p(x)$. However, the KL divergence is theoretically unbounded, and maximizing it may not provide sufficient constraints to ensure convergence to a suitable solution. We use another bounded metric: the Jensen–Shannon (JS) divergence [90], which is defined as:

$$D_{\mathrm{JS}}(A \parallel B) = \frac{1}{2} D_{\mathrm{KL}}\left(A \parallel \frac{A+B}{2}\right) + \frac{1}{2} D_{\mathrm{KL}}\left(B \parallel \frac{A+B}{2}\right). \tag{11}$$

Mutual information $I(\mathcal{Y}; \mathcal{X})$ is positively correlated with the JS divergence between $p(y|x)\, p(x)$ and $p(y)\, p(x)$:

$$I(\mathcal{Y}; \mathcal{X}) \propto D_{\mathrm{JS}}\left(p(y|x)\, p(x) \parallel p(y)\, p(x)\right). \tag{12}$$

The estimation of JS divergence can be obtained by the local variation inference of the general f-divergence [91]:

$$\begin{aligned} D_{\mathrm{JS}}(A \parallel B) = {} & \mathbb{E}_{x \sim a(x)}\left[\log \sigma(T(x))\right] \\ & + \mathbb{E}_{x \sim b(x)}\left[\log\left(1 - \sigma(T(x))\right)\right]. \end{aligned} \tag{13}$$

Therefore, mutual information $I(\mathcal{Y}; \mathcal{X})$ can ultimately be expressed as:

$$\begin{aligned} I(\mathcal{Y}; \mathcal{X}) \propto {} & \mathbb{E}_{(y,x) \sim p(y|x)p(x)}\left[\log \sigma(T(y,x))\right] \\ & + \mathbb{E}_{(y,x) \sim p(y)p(x)}\left[\log\left(1 - \sigma(T(y,x))\right)\right], \end{aligned} \tag{14}$$

where $\sigma(T(y,x))$ is a discriminative network, with x and its corresponding y considered as a positive pair, and x with the y of other samples considered as negative pairs.

Therefore, the mutual information $I(\mathcal{Y};\mathcal{X})$ between the pretext task $\mathcal{Y}$ and the set of input point clouds $\mathcal{X}$ depends on the discrimination of the pretext task. Low-level features $\mathcal{Y}_{\mathrm{L}}$ such as 3D coordinates, geometric features (e.g., normals), and the number of point clouds are not distinctive from the input point clouds $\mathcal{X} = \mathcal{P} = \left\{ \boldsymbol{c}_\ell, \boldsymbol{f}_\ell \middle| \ell = 1, \ldots, N_{\mathrm{p}} \right\}$. When reconstructing 3D coordinates, the pretext task involves relative coordinates normalized to the voxel center. Different voxels may share identical 3D coordinates, geometric features, and other low-level features. The confusion between positive and negative sample pairs leads to lower mutual information $I(\mathcal{Y}_{\mathrm{L}};\mathcal{X})$ between the low-level features and the input point cloud. On the contrary, high-level features such as Seal features contain rich semantic information, which may include the semantic class and the collective information from contextual point clouds. Therefore, the high-level features $\mathcal{Y}_{\mathrm{H}}$ are more suitable to be supplemented as a pretext task due to the higher mutual information $I(\mathcal{Y}_{\mathrm{H}};\mathcal{X})$ with the input point cloud.

Proposition 2. *High-level features with deep supervision will increase the mutual information between latent features $\mathcal{Z}^i_{\mathrm{um}}$ $(i = 1, 2, \ldots, L)$ and high-level features $\mathcal{Y}_{\mathrm{H}}$.*

The process of reconstructing the high-level features $\mathcal{Y}_{\mathrm{H}}$ from masked features $\mathcal{Z}^i_{\mathrm{m}}$ can be regarded as feature distillation [92–94], which ensures that masked features can capture the semantic components within high-level features. According to the information flow $\mathcal{Z}^i_{\mathrm{um}} \to \mathcal{Z}^i_{\mathrm{m}} \to \hat{\mathcal{Y}} \Longleftrightarrow \mathcal{Y}$, feature distillation will further affect unmasked features. The mutual information $I(\mathcal{Z}^i_{\mathrm{um}};\mathcal{Y}_{\mathrm{H}})$ between unmasked features $\mathcal{Z}^i_{\mathrm{um}}$ and high-level features $\mathcal{Y}_{\mathrm{H}}$ is defined as:

$$I(\mathcal{Z}^i_{\mathrm{um}};\mathcal{Y}_{\mathrm{H}}) = H(\mathcal{Z}^i_{\mathrm{um}}) + H(\mathcal{Y}_{\mathrm{H}}) - H(\mathcal{Z}^i_{\mathrm{um}},\mathcal{Y}_{\mathrm{H}}), \tag{15}$$

where $H(\mathcal{Z}^i_{\mathrm{um}},\mathcal{Y}_{\mathrm{H}})$ represents the joint entropy between $\mathcal{Z}^i_{\mathrm{um}}$ and $\mathcal{Y}_{\mathrm{H}}$. Feature distillation will make $\mathcal{Z}^i_{\mathrm{um}}$ converge towards $\mathcal{Y}_{\mathrm{H}}$. Furthermore, due to the greater mutual information $I(\mathcal{Y}_{\mathrm{H}};\mathcal{X})$ between the high-level features and the input point cloud, the joint entropy $H(\mathcal{Z}^i_{\mathrm{um}},\mathcal{Y}_{\mathrm{H}})$ between latent features and the high-level features will be further reduced. $H(\mathcal{Z}^i_{\mathrm{um}})$ and $H(\mathcal{Y}_{\mathrm{H}})$ will remain stable after feature normalization. Referring to Eq. (15), the mutual information between latent features $\mathcal{Z}^i_{\mathrm{um}}$ $(i = 1, 2, \ldots, L)$ and high-level features $\mathcal{Y}_{\mathrm{H}}$ will increase.

Proposition 3. *High-level features with deep supervision will increase the lower bound of mutual information between latent features $\mathcal{Z}^i_{\mathrm{um}}$ $(i = 1, 2, \ldots, L)$ and input point cloud $\mathcal{X}$.*

By definition, the mutual information between the unmasked latent features $\mathcal{Z}^i_{\mathrm{um}}$ of the encoder and the input point cloud $\mathcal{X}$ is:

$$I(\mathcal{Z}^i_{\mathrm{um}};\mathcal{X}) = H(\mathcal{Z}^i_{\mathrm{um}}) - H(\mathcal{Z}^i_{\mathrm{um}}|\mathcal{X}), \tag{16}$$

where $H(\mathcal{Z}^i_{\mathrm{um}})$ represents the entropy of $\mathcal{Z}^i_{\mathrm{um}}$, which remains stable after feature normalization. $H(\mathcal{Z}^i_{\mathrm{um}}|\mathcal{X})$ is the conditional entropy of $\mathcal{Z}^i_{\mathrm{um}}$ given the input point cloud $\mathcal{X}$. We further investigate the impact of high-level features with deep supervision on conditional entropy $H(\mathcal{Z}^i_{\mathrm{um}}|\mathcal{X})$. Based on the definition of conditional entropy, we get:

$$H(\mathcal{Z}^i_{\mathrm{um}}|\mathcal{X}) = -\mathbb{E}_{P(\mathcal{X},\mathcal{Z}^i_{\mathrm{um}})}\left[\log P(\mathcal{Z}^i_{\mathrm{um}}|\mathcal{X})\right]. \tag{17}$$

According to variational inference (VI) [95], we introduce an approximate distribution $Q(\mathcal{Z}^i_{\mathrm{um}}|\mathcal{X})$ to approximate the true conditional distribution $P(\mathcal{Z}^i_{\mathrm{um}}|\mathcal{X})$. The KL divergence between them can be expressed as:

$$\begin{aligned} D_{\mathrm{KL}}\left(P(\mathcal{Z}^i_{\mathrm{um}}|\mathcal{X}) \parallel Q(\mathcal{Z}^i_{\mathrm{um}}|\mathcal{X})\right) &= \mathbb{E}_{P(\mathcal{Z}^i_{\mathrm{um}},\mathcal{X})}\left[\log \frac{P(\mathcal{Z}^i_{\mathrm{um}}|\mathcal{X})}{Q(\mathcal{Z}^i_{\mathrm{um}}|\mathcal{X})}\right] \\ &= \mathbb{E}_{P(\mathcal{Z}^i_{\mathrm{um}},\mathcal{X})}\left[\log P(\mathcal{Z}^i_{\mathrm{um}}|\mathcal{X})\right] - \mathbb{E}_{P(\mathcal{Z}^i_{\mathrm{um}},\mathcal{X})}\left[\log Q(\mathcal{Z}^i_{\mathrm{um}}|\mathcal{X})\right]. \end{aligned} \tag{18}$$

Therefore, the conditional entropy $H(\mathcal{Z}^i_{\mathrm{um}}|\mathcal{X})$ can be reformulated as:

$$\begin{aligned} H(\mathcal{Z}^i_{\mathrm{um}}|\mathcal{X}) = &-\left(\mathbb{E}_{P(\mathcal{Z}^i_{\mathrm{um}},\mathcal{X})}\left[\log Q(\mathcal{Z}^i_{\mathrm{um}}|\mathcal{X})\right]\right. \\ &\left. + D_{\mathrm{KL}}\left(P(\mathcal{Z}^i_{\mathrm{um}}|\mathcal{X}) \parallel Q(\mathcal{Z}^i_{\mathrm{um}}|\mathcal{X})\right)\right). \end{aligned} \tag{19}$$

Since the KL divergence $D_{\mathrm{KL}}\left(P(\mathcal{Z}^i_{\mathrm{um}}|\mathcal{X}) \parallel Q(\mathcal{Z}^i_{\mathrm{um}}|\mathcal{X})\right)$ is always non-negative, it follows that:

$$H(\mathcal{Z}^i_{\mathrm{um}}|\mathcal{X}) \le \mathbb{E}_{P(\mathcal{Z}^i_{\mathrm{um}},\mathcal{X})}\left[-\log Q(\mathcal{Z}^i_{\mathrm{um}}|\mathcal{X})\right]. \tag{20}$$

Optimizing the reconstruction loss $L_i(\mathcal{Z}^i_{\mathrm{m}},\mathcal{Y})$ will indirectly reduce the log-likelihood $-\log Q(\mathcal{Z}^i_{\mathrm{um}}|\mathcal{X})$ and minimize the upper bound of the conditional entropy $H(\mathcal{Z}^i_{\mathrm{um}}|\mathcal{X})$. Moreover, due to the greater mutual information $I(\mathcal{Y}_{\mathrm{H}};\mathcal{X})$ between high-level features and the input point cloud, the loss of reconstruction will be lower, which will further reduce the upper bound of the conditional entropy $H(\mathcal{Z}^i_{\mathrm{um}}|\mathcal{X})$. According to Eq. (16), the lower bound of the mutual information $I(\mathcal{Z}^i_{\mathrm{um}};\mathcal{X})$ between latent features $\mathcal{Z}^i_{\mathrm{um}}$ and the input point cloud $\mathcal{X}$ will increase, which is beneficial for downstream tasks.

Proposition 4. *The masking sampling strategy based on high-level features will increase the mutual information between latent features $\mathcal{Z}^i_{\mathrm{um}}$ and high-level features $\mathcal{Y}_{\mathrm{H}}$.*

Our proposed masking sampling strategy adjusts the reconstruction difficulty based on the discrimination of high-level features, which primarily affects $\mathcal{Z}^i_{\mathrm{um}} \to \mathcal{Z}^i_{\mathrm{m}}$ in the information flow. For regions with small inter-class discrimination, we set a lower masking ratio to simplify the reconstruction and reduce focus. Overall, Our proposed I^{2}Mask enhances the mutual information $I(\mathcal{Z}^i_{\mathrm{um}};\mathcal{Z}^i_{\mathrm{m}})$ between unmasked features $\mathcal{Z}^i_{\mathrm{um}}$ and masked features $\mathcal{Z}^i_{\mathrm{m}}$. In addition, in the information flow $\mathcal{Z}^i_{\mathrm{um}} \to \mathcal{Z}^i_{\mathrm{m}} \to \hat{\mathcal{Y}} \Longleftrightarrow \mathcal{Y}$, $\mathcal{Z}^i_{\mathrm{um}}$ and $\mathcal{Z}^i_{\mathrm{m}}$ are ultimately used to reconstruct $\mathcal{Y}_{\mathrm{H}}$, which aligns $\mathcal{Z}^i_{\mathrm{um}}$, $\mathcal{Z}^i_{\mathrm{m}}$, and $\mathcal{Y}_{\mathrm{H}}$ in the feature space. Therefore, it is reasonable to determine the masking sampling strategy based on the discrimination in $\mathcal{Y}_{\mathrm{H}}$. Based on the chain rule of mutual information [96], we get:

$$I(\mathcal{Z}^i_{\mathrm{um}};\mathcal{Y}_{\mathrm{H}}) = I(\mathcal{Z}^i_{\mathrm{um}};\mathcal{Z}^i_{\mathrm{m}}) + I(\mathcal{Z}^i_{\mathrm{m}};\mathcal{Y}_{\mathrm{H}}|\mathcal{Z}^i_{\mathrm{um}}). \tag{21}$$

Thanks to the concise design of the head network and the Chamfer distance [97], $I(\mathcal{Z}^i_{\mathrm{m}};\mathcal{Y}_{\mathrm{H}}|\mathcal{Z}^i_{\mathrm{um}})$ remains stable. Therefore, $I(\mathcal{Z}^i_{\mathrm{um}};\mathcal{Y}_{\mathrm{H}})$ will increase, which is beneficial for downstream tasks that require semantic information.

Proposition 5. *CKA-guided Hierarchical Reconstruction leads to greater impact of high-level features with deep supervision on deeper layers.*

According to Proposition 2, feature distillation between high-level features $\mathcal{Y}_{\mathrm{H}}$ and masked features $\mathcal{Z}^i_{\mathrm{m}}$ will further affect unmasked features $\mathcal{Z}^i_{\mathrm{um}}$. Therefore, the joint entropy $H(\mathcal{Z}^i_{\mathrm{um}},\mathcal{Y}_{\mathrm{H}})$ between $\mathcal{Z}^i_{\mathrm{um}}$ and $\mathcal{Y}_{\mathrm{H}}$ is positively correlated with the loss $L_i(\mathcal{Z}^i_{\mathrm{m}},\mathcal{Y}_{\mathrm{H}})$ between $\mathcal{Z}^i_{\mathrm{m}}$ and $\mathcal{Y}_{\mathrm{H}}$. Based on Eq. (15), the mutual information $I(\mathcal{Z}^i_{\mathrm{um}};\mathcal{Y}_{\mathrm{H}})$ between $\mathcal{Z}^i_{\mathrm{um}}$ and $\mathcal{Y}_{\mathrm{H}}$ is negatively correlated with the loss $L_i(\mathcal{Z}^i_{\mathrm{m}},\mathcal{Y}_{\mathrm{H}})$. Furthermore, in Section 3.4, we propose CKA-guided Hierarchical Reconstruction. The proportion of low-level information α^i is used to weight the low-level and high-level reconstruction losses. Therefore, the relationship between the mutual information $I(\mathcal{Z}^i_{\mathrm{um}};\mathcal{Y}_{\mathrm{H}})$ and the loss $L_i(\mathcal{Z}^i_{\mathrm{m}},\mathcal{Y}_{\mathrm{H}})$ can be expressed as:

$$I(\mathcal{Z}^i_{\mathrm{um}};\mathcal{Y}_{\mathrm{H}}) \propto -\left(1-\alpha^i\right) L_i(\mathcal{Z}^i_{\mathrm{m}},\mathcal{Y}_{\mathrm{H}}). \tag{22}$$

As the depth of layers increases, α^i gradually decreases, and the scaling factor $\left(1-\alpha^i\right)$ gradually increases. The decrease in the loss leads to a greater increase in the mutual information. Therefore, due to CKA-guided Hierarchical Reconstruction, the increase in mutual information between latent features $\mathcal{Z}^i_{\mathrm{um}}$ and high-level features $\mathcal{Y}_{\mathrm{H}}$ will be more significant in deeper layers.

According to Proposition 3, the reconstruction loss $L_i(\mathcal{Z}^i_{\mathrm{m}},\mathcal{Y})$ can be expressed as:

$$L_i(\mathcal{Z}^i_{\mathrm{m}},\mathcal{Y}) = \alpha^i L_i(\mathcal{Z}^i_{\mathrm{m}},\mathcal{Y}_{\mathrm{L}}) + \left(1-\alpha^i\right) L_i(\mathcal{Z}^i_{\mathrm{m}},\mathcal{Y}_{\mathrm{H}}). \tag{23}$$

Due to the greater mutual information $I(\mathcal{Y}_{\mathrm{H}};\mathcal{X})$ and larger scaling factor $\left(1-\alpha^i\right)$, optimizing the reconstruction loss has a more significant effect on enhancing the lower bound of mutual information between latent features and the input point cloud in deeper layers.

Table 1
Pre-training settings on Waymo.

Parameter	Value
Point cloud range	[−74.88, −74.88, −2, 74.88, 74.88, 4.0]
Voxel size	[0.32, 0.32, 0.1875]
Voxel grid shape	[468, 468, 32]
Augmentor	×
Sparse point encoder	MinkUNet (Res16UNet34C)
Feature dimension of point encoder	64
Feature dimension of voxel encoder	192
Number of DSVT encoder	8
Number of DSVT decoder	4
High-level feature loss	SmoothL1Loss
Optimizer	AdamW
Weight decay	0.05
Epochs	30

Table 2
Pre-training settings on nuScenes.

Parameter	Value
Point cloud range	[−51.2, −51.2, −5, 51.2, 51.2, 3]
Voxel size	[0.2, 0.2, 0.2]
Voxel grid shape	[512, 512, 40]
Augmentor	×
Sparse point encoder	MinkUNet (Res16UNet34C)
Feature dimension of point encoder	64
Feature dimension of voxel encoder	256
Number of DSVT/SST encoder	8
Number of DSVT decoder	4
High-level feature loss	SmoothL1Loss
Optimizer	AdamW
Epochs	72

Proposition 6. *Differential-gated Progressive Learning leads to more stable training and greater impact of high-level features with deep supervision on deep layers.*

Differential-gated Progressive Learning influences the learning sequence. The learning of high-level features is conducted after low-level features have been well-learned. We take the conditional entropy $H\left(\mathcal{Z}_{\mathrm{um}}^{i}|\mathcal{Z}_{\mathrm{um}}^{j}\right)(j<i)$ between the deep latent feature $\mathcal{Z}_{\mathrm{um}}^{i}$ and the shallow latent feature $\mathcal{Z}_{\mathrm{um}}^{j}$ as the research objective. During parallel learning, $\mathcal{Z}_{\mathrm{um}}^{i}$ is inferred from random features $\mathcal{Z}_{\mathrm{um}}^{j}$, leading to a large $H\left(\mathcal{Z}_{\mathrm{um}}^{i}|\mathcal{Z}_{\mathrm{um}}^{j}\right)$. After incorporating DGPL, the shallow latent feature, which is close to low-level features, provides a reasonable foundation for the deep latent feature. Network training becomes more stable. Meanwhile, $H\left(\mathcal{Z}_{\mathrm{um}}^{i}|\mathcal{Z}_{\mathrm{um}}^{j}\right)$ is significantly reduced. Based on the information flow $\mathcal{X}\rightarrow\mathcal{Z}_{\mathrm{um}}^{j}\rightarrow\mathcal{Z}_{\mathrm{um}}^{i}$ and the chain rule, we obtain:

$$I\left(\mathcal{Z}_{\mathrm{um}}^{i};\mathcal{X}\right)=I\left(\mathcal{Z}_{\mathrm{um}}^{j};\mathcal{X}\right)+I\left(\mathcal{Z}_{\mathrm{m}}^{i};\mathcal{Z}_{\mathrm{um}}^{j}|\mathcal{X}\right). \tag{24}$$

Therefore, the mutual information between the deep latent features $\mathcal{Z}_{\mathrm{um}}^{i}$ and the input point cloud $\mathcal{X}$ increases. In addition, the shallow layer is capable of extracting local geometric features and spatial information. On this basis, the deep latent feature will be closer to the high-level feature $\mathcal{Y}_{\mathrm{H}}$. This will increase the mutual information between the deep latent feature $\mathcal{Z}_{\mathrm{um}}^{i}$ and the high-level feature $\mathcal{Y}_{\mathrm{H}}$.

5. Experiments

5.1. Experimental settings

Dataset. Consistent with previous methods [7,13,14,71], we demonstrated the performance of GPICTURE in 3D object detection on Waymo Open Dataset [37] and nuScenes [38], 3D semantic segmentation on nuScenes [38] and SemanticKITTI [39], and occupancy prediction on OpenOccupancy [98]. To mitigate the risk of data leakage, we performed pre-training and fine-tuning on the training set, and obtained scores on the validation and test set.

- **Waymo Open Dataset** [37]. A leading benchmark for 3D perception in autonomous driving comprising 1150 sequences (over 200k frames) with 798 for training, 202 for validation, and 150 for test. Each frame captures a 150 m × 150 m scene using a primary 64-beam LiDAR, supported by four short-range LiDARs, producing approximately 180k points every 0.1 s. This extensive dataset, featuring urban and suburban scenes, has been widely adopted in state-of-the-art methods owing to its challenging and comprehensive coverage. The 3D detection evaluation metrics included the standard 3D mean average precision (mAP) and mAP weighted by heading accuracy (mAPH). These metrics use an intersection of union (IoU) threshold of 0.7 for vehicles and 0.5 for other categories, with performance evaluated at two difficulty levels: LEVEL 1 for boxes with more than five LiDAR points and LEVEL 2 for boxes with at least one LiDAR point. We adopted L2 mAP and L2 mAPH as the main evaluation metrics.
- **nuScenes** [38]. A large-scale dataset for autonomous driving comprising 1000 driving sequences with 700 for training, 150 for validation, and 150 for test. Each 20-s sequence was recorded using a 32-beam LiDAR, producing approximately 30 000 points per frame. The dataset was recorded in Boston and Singapore and is characterized by its diversity, including night-time and rainy conditions. The dataset includes aligned vehicle pose data for each frame, with bounding box labels every 0.5 s. In total, there were 28 130 training instances, 6019 validation instances, and various annotations for downstream tasks such as 3D semantic segmentation. The standard evaluation metrics included mAP and nuScenes detection score (NDS). Unlike standard box overlap, mAP employs top-down central distances ranging from 4 m, 2 m, 1 m, and 0.5 m. NDS is an integrated metric that considers mAP alongside other attribute metrics, encompassing average translation error (ATE), average scale error (ASE), average orientation error (AOE), average velocity error (AVE), and average attribute error (AAE) to provide a more holistic evaluation. For segmentation tasks, mIoU was used.

- **SemanticKITTI** [39]. A comprehensive dataset designed for point cloud semantic segmentation in autonomous driving scenarios. Using a Velodyne-HDLE 64 LiDAR sensor, the dataset comprises 22 sequences and over 40 000 point clouds for training and validation. It is labeled into 19 classes after merging classes. We used mIoU to evaluate the performance of relevant methods across all classes.
- **OpenOccupancy** [98]. An extensive benchmark designed to assess semantic occupancy perception in the surroundings. It enhances nuScenes [38] by integrating dense semantic annotations via an augmenting and purifying (AAP) pipeline. The resulting OpenOccupancy offers a significantly enriched dataset with approximately ∼400K occupied voxels annotated per frame, encompassing 28 130 training and 6019 validation frames with 17 distinct semantic labels. The benchmark employs mIoU as a key semantic evaluation metric to ensure robust accuracy quantification.

Model. We utilized the popular frameworks OpenPCDet [99] and MMDetection3D [100]. We fine-tuned the sparse voxel transformer-based encoders SST, DSVT, and Cylinder3D SST. We employed four DSVTs (with eight attention layers) as encoders, two DSVTs (with four attention layers) as decoders for the final layer, and one DSVT (with two attention layers) as decoder for non-final layers. Please refer to the Training Details for model hyperparameters. For Waymo, we set the voxel size to [0.32, 0.32, 0.1875]. For nuScenes (semantic segmentation and occupancy prediction), we set it to [0.2, 0.2, 0.2]. For SemanticKITTI, we set it to [0.32, 0.32, 0.125].

Training Details. Tables 1 and 2 provide the pre-training settings on Waymo and nuScenes. We pre-trained on Waymo and report the results of 3D object detection on Waymo val and test set. We pre-trained on nuScenes and report the results of 3D object detection, 3D semantic segmentation, and occupancy prediction on nuScenes val set. We pre-trained on SemanticKITTI and reported the results of 3D semantic

Table 3
Comparisons of 3D object detection between GPICTURE and other self-supervised learning methods on Waymo val set. 'Epochs' and 'Fraction' denote the pre-training epochs and dataset fraction used for pre-training. The improvement compared to training from scratch is indicated with red superscripts.

Method	Epochs	Fraction	L2 (AP/APH)↑			
			Overall	Vehicle	Pedestrian	Cyclist
SST[a] [63]	–	–	68.50/65.54	64.96/64.56	72.38/64.89	68.17/67.17
Occupancy-MAE (SST)[b] [13]	30	100%	70.15$^{+1.65}$/67.16$^{+1.62}$	68.46/67.86	72.53/65.12	69.47/68.50
MV-JAR (SST)[b] [15]	30	100%	70.38$^{+1.88}$/67.37$^{+1.83}$	68.59/68.05	72.77/65.23	69.78/68.82
GD-MAE (SST)[a] [14]	30	100%	70.62$^{+2.12}$/67.64$^{+2.10}$	68.72/68.29	72.84/65.47	70.30/69.16
PICTURE (SST) [35]	30	100%	71.02$^{+2.52}$/68.03$^{+2.49}$	69.37/68.84	73.44/66.15	70.26/69.09
GPICTURE (SST)	30	20%	70.03$^{+1.53}$/67.03$^{+1.49}$	68.31/67.65	72.45/64.95	69.32/68.48
GPICTURE (SST)	30	100%	**71.46**$^{+2.96}$ / **68.51**$^{+2.97}$	**69.81/69.32**	**73.88/66.63**	**70.70/69.57**
DSVT[a] [21]	–	–	73.20/71.00	70.90/70.50	75.20/69.80	73.60/72.70
Occupancy-MAE (DSVT)[b] [13]	30	100%	73.86$^{+0.66}$/71.78$^{+0.78}$	71.53/71.21	76.02/70.69	74.02/73.44
MV-JAR (DSVT)[b] [15]	30	100%	74.37$^{+1.17}$/72.01$^{+1.01}$	72.15/71.53	76.44/70.93	74.52/73.58
GeoMAE (DSVT)[b] [7]	30	100%	74.46$^{+1.26}$/72.05$^{+1.05}$	72.13/71.62	76.61/71.02	74.64/73.52
GD-MAE (DSVT)[b] [14]	30	100%	74.68$^{+1.48}$/72.22$^{+1.22}$	72.39/71.81	76.77/71.23	74.88/73.63
PICTURE (DSVT) [35]	30	100%	75.13$^{+1.93}$/72.69$^{+1.69}$	72.93/72.45	77.18/71.66	75.27/73.96
GPICTURE (DSVT)	30	20%	73.84$^{+0.64}$/71.75$^{+0.75}$	71.55/71.22	75.99/70.61	73.98/73.42
GPICTURE (DSVT)	30	100%	**75.55**$^{+2.35}$ / **73.13**$^{+2.13}$	**73.38/72.87**	**77.52/72.01**	**75.75/74.51**

[a] Indicates the results are from the original paper.
[b] Presents re-implemented by OpenPCDet.

Table 4
Comparisons of 3D object detection between GPICTURE and other self-supervised learning methods on nuScenes val set.

Method	Epochs	Fraction	mAP↑	NDS↑	mATE↓	mASE↓	mAOE↓	mAVE↓	mAAE↓
DSVT† [21]	–	–	66.4	71.1	27.0	24.8	27.2	22.6	18.9
MV-JAR (DSVT)* [15]	72	100%	67.6$^{+1.2}$	72.2$^{+1.1}$	26.5	24.4	26.8	22.1	18.3
GD-MAE (DSVT)* [14]	72	100%	67.7$^{+1.3}$	72.2$^{+1.1}$	26.4	24.4	26.9	21.9	18.2
PICTURE (DSVT) [35]	72	100%	68.1$^{+1.7}$	72.6$^{+1.5}$	25.8	24.2	26.5	21.6	17.7
GPICTURE (DSVT)	72	100%	**68.6**$^{+2.2}$	**73.0**$^{+1.9}$	**25.5**	**23.8**	**25.8**	**20.7**	**17.4**

segmentation on SemanticKITTI val set. In inter-class discrimination-guided masking, we set the expected number of superclass partition (n_1, n_2, n_3) as (3, 3, 2) and the base mask ratio (r_b^1, r_b^2, r_b^3) as (0.9, 0.45, 0), which implies some superclasses were not masked. The distance threshold λ in intra-class discrimination-guided masking was set to 0.6. In CKA-guided hierarchical reconstruction, the proportion of low-level information $\alpha^i (i = 1, 2, \ldots, 8)$ was [0.97, 0.88, 0.78, 0.64, 0.55, 0.44, 0.36, 0.31]. In differential-gated progressive learning, the slope factor γ and gating factor δ were set to 10 and 0.2. For fine-tuning, we trained for 12 epochs on Waymo, 24 epochs on nuScenes, and 30 epochs on 100% labeled SemanticKITTI [77]. All experiments were conducted on 8 NVIDIA A100-SXM4-40 GB GPUs.

5.2. Comparison with state-of-the-art methods

First, we compared the performance of our method with other self-supervised learning methods in 3D object detection. Tables 3 and 9 present the performance on the Waymo val set and test set for the leaderboard, respectively. We employed two sparse voxel transformer blocks, SST and DSVT, as encoders. The compared low-level features include 3D point cloud coordinates (Occupancy-MAE [13], GD-MAE [14]), geometric features (GeoMAE [7]), and the jigsaw puzzle (MV-JAR [15]). Our proposed GPICTURE outperforms training from scratch, leading to improvements of 2.35% and 2.13% in L2 mAP/mAPH. Moreover, compared to the best self-supervised learning method, GD-MAE, our proposed GPICTURE achieves improvements of 0.87% and 0.91%. This is attributed to the introduction of high-level features, which increases the mutual information between latent features and semantic features, and the input point cloud. More semantically meaningful and disentangled representations promote fine-tuning performance. Specifically, the 0.42% improvement over our previous study, PICTURE, highlights the importance of multi-task control strategies. Furthermore, as the amount of pre-training data increased, the performance improves significantly from 73.84% to 75.55%, opening up possibilities for leveraging large-scale unlabeled point clouds.

For 3D object detection, in addition to the Waymo dataset [37], we also report results on the val set of the nuScenes [38] in Table 4. Despite the robust DSVT being utilized as an encoder, our proposed method, GPICTURE, achieves improvements of 2.2% in mAP and 1.9% in NDS, respectively. Employing high-level features for reconstruction targets, as opposed to relying solely on low-level features such as 3D point cloud coordinates and jigsaw puzzles, yields respective improvements of 0.9% and 1.0% in mAP. This indicates that our proposed 3D self-supervised reconstruction target exhibits strong generalization across various autonomous driving datasets.

Table 5 provides a comparison of 3D semantic segmentation on the nuScenes val set. We considered low-level features such as geometric features and the occupation type specifically designed for segmentation (ALSO [77]). GPICTURE outperforms them by 1.1% and 0.9%, respectively, and exhibits better advantages in categories such as bicycle and car. Therefore, high-level features have a positive impact on segmentation tasks that rely on semantic information. Furthermore, we directly compared it with the Seal after alignment. Our generative self-supervised approach further exploits the Seal features, resulting in a 0.9% improvement. Table 6 compares 3D semantic segmentation on the SemanticKITTI val set. MAELi [71] uses occupancy type as the pretext task and employs range-aware random masking. The lower amount of information in the pretext task and the uniform attention to all objects limit its performance. By contrast, our adaptive mask sampling strategy provides higher attention to the reconstruction of semantic objects and promotes the mutual information between latent features and high-level features. Segmentation tasks that rely on semantic information show improvements across all categories.

Table 7 provides a comparison with other self-supervised methods in occupancy prediction. Because the Seal feature is aligned with both images and text, reconstructing the Seal voxel features can enhance the occupancy prediction performance. The proposed GPICTURE surpasses GD-MAE by 1.0%. Furthermore, our proposed I^{2}Mask focuses more on road-related objects, and improvement in occupancy prediction accuracy is crucial for driving safety.

Table 5
Comparisons of 3D semantic segmentation between GPICTURE and other self-supervised methods on nuScenes val set.

Method	mIoU↑	Bicycle	Bus	Car	Motorcycle	Pedestrian	Trailer	Truck
Cylinder3D-SST† [63]	76.5	40.0	91.8	94.2	78.1	80.1	62.5	84.7
GeoMAE (Cylinder3D-SST)† [7]	78.6$^{+2.1}$	42.6	93.5	95.8	79.8	83.5	65.6	87.3
Seal (Cylinder3D-SST)* [30]	78.8$^{+2.3}$	42.1	93.5	95.7	80.1	84.0	65.5	87.2
ALSO (Cylinder3D-SST)* [77]	78.8$^{+2.3}$	42.4	93.8	95.5	80.2	83.7	65.6	86.8
PICTURE (Cylinder3D-SST) [35]	79.4$^{+2.9}$	43.2	94.5	96.3	80.6	84.1	65.7	87.5
GPICTURE (Cylinder3D-SST)	**79.7**$^{+3.2}$	**43.6**	**94.8**	**96.5**	**81.0**	**84.4**	**65.8**	**87.7**

Table 6
Comparisons of 3D semantic segmentation between GPICTURE and other self-supervised methods on SemanticKITTI val set.

Method	mIoU↑	Car	Bicycle	Motorcycle	Truck	Other-vehicle	Person
Cylinder3D-SST* [63]	62.9	95.6	48.7	67.6	83.1	49.6	71.3
ALSO† [77]	63.6	95.8	49.9	68.6	83.4	50.1	71.9
MAELi† [71]	64.2	96.1	50.1	69.3	84.1	50.5	72.1
GPICTURE	**64.7**	**96.4**	**50.3**	**69.8**	**84.8**	**50.9**	**72.4**

5.3. Study of multiple downstream tasks

Table 8 shows the performance under different task weights in the multi-task case of object detection and occupancy prediction on nuScenes (OpenOccupancy) val set. Unlike the pre-training, we configured the multi-task settings in fine-tuning to enable the model to share representations between different downstream tasks [24,25]. Therefore, the multi-task settings during the fine-tuning belong to the multitask optimization problem. It can be observed that with appropriate task weights for multiple downstream tasks, multitask optimization outperforms single downstream tasks owing to shared features and knowledge transfer.

5.4. Study of hyperparameters

Table 10 provides a study of $(r_{\mathrm{b}}^1, r_{\mathrm{b}}^2, r_{\mathrm{b}}^3)$ in inter-class discrimination-guided masking. Through the study, (0.9, 0.45, 0) was determined as a suitable base mask ratio. Table 11 presents the expected number (n_1, n_2, n_3) in Algorithm Fastest Class Sampling. Using this hyperparameter, the eight superclasses were divided into three sets. The results show that among the two divisions, (3, 3, 2) yielded superior results to (4, 2, 2). Table 12 presents the distance threshold λ in intra-class discrimination-guided masking. This threshold was used to determine whether the high-level features of a voxel were too far away from its cluster center $\boldsymbol{\mu}^{k_i}$. Through experiments, 0.6 was selected as the default value. Table 13 shows the performance under different slope factors γ and gating factors δ in differential-gated progressive learning. The slight performance fluctuations demonstrate the stability and effectiveness of progressive learning.

The performance does not vary significantly with these hyperparameters, indicating that the default values maintain good generalization.

5.5. Ablation studies

Table 14 shows the ablation studies on the reconstruction target, mask sampling strategy, CHR, and DGPL. First, when exclusively using high-level features, the performance is inferior to that when only reconstructing coordinates (73.96 vs. 74.05), indicating that low-level features are necessary. Second, with random masking, supplementation with high-level voxel features as a pretext task results in an improvement of 0.69% and 0.48% in L2 mAP compared to various low-level features. This indicates that the high-level feature exhibits a significant positive impact owing to its higher mutual information with the input point cloud. Third, the performance further improves when random masking is replaced with I^2Mask. Inter-class and intra-class discrimination-guided masking further contribute to an additional improvement of 0.08% and 0.11%. When both distance metrics were considered together, distinct mask ratios were assigned to each superclass, resulting in an ultimate improvement of 0.39%. Therefore, the two metrics have a synergistic effect. With the further introduction of CHR and DGPL, a reasonable multi-task control strategy unlocks the potential of the high-level feature reconstruction task and multiform optimization, resulting in a 0.42% improvement in L2 mAP.

In addition to Seal features, we also explore alternative high-level features that can be used for pretext tasks in Table 15. We first utilized the simplest high-level feature in the dataset: intensity, which contains information about the material of objects and it brings a 0.19% gain (74.05 vs. 74.24). Additionally, all complex high-level features yield a minimum improvement of 0.81% compared with coordinates only. This indicates that enhancing the mutual information between latent features and high-level features is necessary, as proven in Proposition 2. On the other hand, although all point cloud encoders are aligned with the semantic information of images or texts, Seal features exhibit the highest mutual information with the input point cloud by locating image-point cloud pairs via SAM. Seal features yield the best performance, with improvements ranging from 0.31% to 1.31% compared to other high-level features.

The comparison of mask sampling strategies is presented in Table 16. We compared our approach with random masking [7], range-aware random masking [13], and FPS-based masking [15]. The key distinction is that only our proposed I^2Mask is based on feature attributes rather than voxel positions. Compared to random masking, our method achieves an improvement of 0.51% in L2 mAP. This indicates that a smoother information flow between unmasked and masked features enhances the mutual information between latent features and high-level features, as proven in Proposition 4.

We compare other multi-task weighting strategies in Table 17. In our previous study PICTURE [35], we used a fixed weighting scheme without deep supervision where the ratio of w_{L}^L to w_{H}^L was $1:3$. On the one hand, deep supervision can improve model performance (75.21 vs. 75.13 and 75.55 vs. 75.33) thanks to the higher mutual information between latent features and the input point cloud at each layer, as proven in Proposition 3. On the other hand, fixed weighting with deep supervision can cause confusion in hierarchical learning. Our proposed CKA-guided hierarchical reconstruction weights multiform optimization based on hierarchical feature learning and improves L2 mAP by 0.34% (75.55 vs. 75.21). This is attributed to higher mutual information between deep latent features and high-level features, as proven in Proposition 5.

In Table 18, we compare other multi-task switching strategies. First, compared to continuously enabling high-level feature reconstruction, all switching strategies yield improvements. This indicates that following progressive learning in multiform optimization is beneficial for knowledge transfer. In addition, the derivative-based method [102] utilizes the gradient of the pre-task loss to determine whether to initiate the high-level task. The achievement-based method [103] defines the ratio of pre-task accuracy to individual task as achievement and uses this to trigger the high-level task. The proposed DGPL delves deeper into the mutual information of features instead of relying on objective metrics such as gradients and accuracy. Our more accurate switching strategy results in a 0.21% improvement in L2 mAP (75.55 vs. 75.34). This is due to the lower conditional entropy between the deep latent feature and the shallow latent feature, as proven in Proposition 6.

Table 7
Comparisons of occupancy prediction between GPICTURE and other self-supervised methods on OpenOccupancy val set.

Method	mIoU↑	Bicycle	Bus	Car	Motorcycle	Pedestrian	Trailer	Truck
DSVT* [21]	16.3	6.4	13.8	18.5	5.6	10.1	13.8	14.2
Occupancy-MAE (DSVT)* [13]	17.2$^{+0.9}$	6.7	15.1	19.3	6.6	11.3	14.5	15.3
GD-MAE (DSVT)* [14]	17.8$^{+1.5}$	7.2	15.1	20.1	6.8	11.9	15.3	15.5
PICTURE (DSVT) [35]	18.4$^{+2.1}$	8.1	15.6	20.7	7.4	12.3	15.7	15.8
GPICTURE (DSVT)	**18.8**$^{+2.5}$	**8.4**	**16.2**	**21.1**	**7.9**	**12.8**	**15.9**	**16.3**

Table 8
Comparisons of different task weights in the multi-task case of object detection and occupancy prediction on nuScenes val set.

Pre-training	Task weight	3D object detection		Occupancy prediction		
		mAP↑	NDS↑	mIoU↑	Bicycle	Bus
None	–	66.4	71.1	16.3	6.4	13.8
GPICTURE (DSVT)	None	68.6	73.0	18.8	8.4	16.2
	1:1	68.8	73.3	19.0	8.7	16.6
	1:2	68.8	73.2	**19.2**	**8.9**	**16.9**
	2:1	**69.1**	**73.4**	19.1	8.7	16.7

Table 9
Comparisons of 3D object detection on Waymo test set.

Method	L2 (AP/APH)↑			
	Overall	Vehicle	Pedestrian	Cyclist
CenterPoint [60]	73.38/71.93	73.42/72.99	74.56/71.52	72.17/71.28
SST [63]	74.41/72.81	73.08/72.74	76.93/73.51	73.22/72.17
DSVT [21]	74.65/73.02	75.11/74.10	75.62/72.60	73.21/72.35
PV-RCNN++ [101]	75.00/73.52	76.31/75.92	76.63/73.55	72.06/71.09
GD-MAE (SST) [14]	76.47/73.37	75.83/75.46	77.10/71.28	76.48/73.37
PICTURE (DSVT) [35]	77.52/76.10	**77.70/77.30**	78.65/75.79	76.20/75.20
GPICTURE (DSVT)	**77.93/76.47**	76.79/76.39	**78.90/75.91**	**78.09/77.10**

Table 10
Impacts of the base mask ratio for superclass partition $\mathcal{K}_1$, $\mathcal{K}_2$, $\mathcal{K}_3$ in inter-class discrimination-guided masking.

(r_b^1, r_b^2, r_b^3)	L2 (AP/APH)↑			
	Overall	Vehicle	Pedestrian	Cyclist
(1.0, 0.5, 0)	75.34/72.92	73.14/72.54	77.34/71.90	75.55/74.32
(0.9, 0.45, 0)	**75.55/73.13**	**73.38/72.87**	**77.52/72.01**	**75.75/74.51**
(0.8, 0.4, 0)	75.28/72.85	73.12/72.56	77.21/71.79	75.52/74.21
(0.6, 0.3, 0)	75.25/72.82	73.10/72.50	77.18/71.78	75.46/74.17

Table 11
Impacts of expected number of superclass partition n_1, n_2, n_3 in inter-class discrimination-guided masking.

(n_1, n_2, n_3)	L2 (AP/APH)↑			
	Overall	Vehicle	Pedestrian	Cyclist
(3, 3, 2)	**75.55/73.13**	**73.38/72.87**	**77.52/72.01**	**75.75/74.51**
(4, 2, 2)	75.25/72.84	73.01/72.45	77.28/71.85	75.46/74.23

Table 12
Impacts of distance threshold λ in intra-class discrimination-guided masking.

λ	L2 (AP/APH)↑			
	Overall	Vehicle	Pedestrian	Cyclist
0.8	75.19/72.75	73.05/72.53	77.14/71.58	75.38/74.14
0.7	75.37/72.94	73.23/72.65	77.38/71.87	75.51/74.30
0.6	**75.55/73.13**	**73.38/72.87**	**77.52/72.01**	**75.75/74.51**
0.5	75.26/72.84	73.10/72.60	77.18/71.77	75.49/74.16

Table 13
Impacts of the slope factor γ and gating factor δ in differential-gated progressive learning.

(γ, δ)	L2 (AP/APH)↑			
	Overall	Vehicle	Pedestrian	Cyclist
(5, 0.2)	75.11/72.66	72.98/72.46	77.09/71.47	75.26/74.05
(10, 0.3)	75.34/72.93	73.15/72.62	77.40/71.88	75.47/74.29
(10, 0.2)	**75.55/73.13**	**73.38/72.87**	**77.52/72.01**	**75.75/74.51**
(20, 0.2)	75.22/72.79	73.08/72.57	77.12/71.70	75.46/74.10

Table 14
Ablation study of pre-training, reconstruction target, mask sampling strategy, CHR, and DGPL on the Waymo val set.

Pre-training	Reconstruction target	I^{2}Mask	CHR	DGPL	L2 mAP↑	L2 mAPH↑
None	–	–	–	–	73.20	71.00
GPICTURE	Coord.	×	×	×	74.05	72.11
	Coord. + Geo.	×	×	×	74.26	72.25
	Seal	×	×	×	73.96	71.82
	Seal	✓	×	×	74.14	72.17
	Coord. + Seal	×	×	×	74.74	72.56
	Coord. + Seal	Only inter-class	×	×	74.82	72.60
	Coord. + Seal	Only intra-class	×	×	74.85	72.58
	Coord. + Seal	✓	×	×	75.13	72.69
	Coord. + Seal	✓	✓	×	75.26	72.85
	Coord. + Seal	✓	×	✓	75.32	72.87
	Coord. + Seal	✓	✓	✓	**75.55**	**73.13**

Table 15
Comparisons of different high-level features for pretext tasks in 3D object detection on Waymo val set.

Pre-training	Reconstruction target	L2 mAP↑	L2 mAPH↑
None	–	73.20	71.00
GPICTURE	Coord.	74.05	72.11
	Coord. + Intensity	74.24	72.22
	Coord. + SLidR [29]	74.86	72.60
	Coord. + CLIP^2 [33]	75.12	72.66
	Coord. + CLIP2Scene [104]	75.24	72.78
	Coord. + Seal [30]	**75.55**	**73.13**

Table 16
Comparisons of different mask sampling strategies.

Pre-training	Mask sampling strategy	L2 mAP↑	L2 mAPH↑
None	–	73.20	71.00
GPICTURE	Random masking	75.04	72.63
	Range-aware random masking	75.14	72.74
	FPS-based masking	75.18	72.75
	I^{2}Mask	**75.55**	**73.13**

5.6. Data efficiency

Comparison with training from scratch on different data scales for fine-tuning is presented in Table 19. On the one hand, at all data scales, the encoder pre-trained with GPICTURE outperforms the encoder without pre-training. The advantage of pre-training becomes more pronounced when there are fewer fine-tuning data. For instance, when fine-tuning with 10% of the data, there is a 5.36% increase in L2 mAPH. On the other hand, with only 50% fine-tuning data required, the performance of DSVT w/GPICURE is close to that of training from

Table 17
Comparisons of different multi-task weighting strategies.

Pre-training	Deep supervision	Weighting strategy	L2 mAP↑	L2 mAPH↑
None	–	–	73.20	71.00
GPICTURE	✓	$w_{\mathrm{L}}^{i} : w_{\mathrm{H}}^{i} = 1 : 1$	74.83	72.59
	×	$w_{\mathrm{L}}^{i} : w_{\mathrm{H}}^{i} = 1 : 3$	75.13	72.69
	✓	$w_{\mathrm{L}}^{i} : w_{\mathrm{H}}^{i} = 1 : 3$	75.21	72.77
	×	CKA-guided Hierarchical Reconstruction	75.33	72.87
	✓	CKA-guided Hierarchical Reconstruction	**75.55**	**73.13**

Table 18
Comparisons of different multi-task switching strategies.

Pre-training	Switching strategy	L2 mAP↑	L2 mAPH↑
None	–	73.20	71.00
GPICTURE	None	75.26	72.85
	Derivative-based method [102]	75.34	72.89
	Achievement-based method [103]	75.30	72.86
	Differential-gated progressive learning	**75.55**	**73.13**

Table 19
Comparison with no pre-training on different data scales in 3D object detection using L2 mAPH on Waymo val set.

DSVT w/ GPICTURE	10%	20%	50%	100%
	52.71	57.46	67.32	71.00
✓	58.07$^{+5.36}$	62.59$^{+5.13}$	71.17$^{+3.85}$	73.13$^{+2.13}$

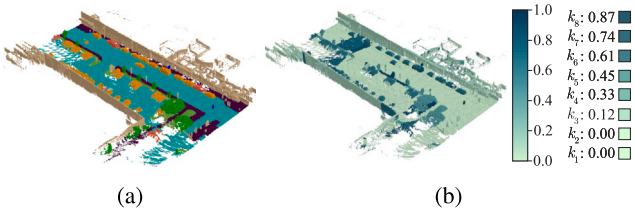


Fig. 4. (a) Ground truth visualization of occupancy prediction. (b) The mask ratio distribution derived by I^{2}Mask.

scratch with 100% fine-tuning data, which is beneficial for autonomous driving research that severely lack annotated data.

5.7. Study of across datasets and joint datasets

In Table 20, transfer learning across datasets and on joint datasets reflects the ability of self-supervised learning to acquire essential and disentangled features. This encompasses differences in vehicle appearances, weather conditions, and architectural styles. When pre-trained on Waymo and fine-tuned on nuScenes, the model exhibits a decrease of 1.07% in mAP compared to when it is solely pre-trained on nuScenes. Nonetheless, an improvement of 1.13% is still achieved compared to no pre-training. When the model is pre-trained on nuScenes and then fine-tuned on Waymo, it shows an improvement of 1.27% compared to training from scratch. This performance difference is attributed to higher mutual information between the latent features and the input point cloud, as proven in Proposition 3. When the model is pre-trained on the joint dataset, it achieves optimal fine-tuning performance on both Waymo and nuScenes datasets. This highlights the value of leveraging abundant unlabeled point cloud data.

5.8. Study of time cost

The time cost of our proposed method, GPICTURE, compared to other self-supervised learning methods, is presented in Table 21. Despite the greater effectiveness of high-level features in contrast to low-level features, it requires approximately 5× time cost (244 h vs. 48 h). This represents a trade-off between performance and time cost. To accelerate the pre-training, we adopted an offline approach to pre-extract high-level features for the entire dataset and save them locally. Thus, high-level features can be reused during pre-training. Following offline processing, the time cost for GPICTURE is approximately 65 h, which is comparable to the time required for low-level features (65 h vs. 61 h, 48 h).

5.9. Visualization

Fig. 4 shows the semantic and mask ratio distribution for a certain scene. Compared to random masking, our proposed I^{2}Mask assigns mask ratios with inverse trends for different regions based on their reconstruction difficulty. Moreover, areas such as vehicles and pedestrians, which are highly focused on in downstream tasks, have a high masking ratio, which forces a complex reconstruction. However, the masking ratio is lower for areas such as roads and constructions, which reduces the attention during the reconstruction. Our adaptive mask sampling strategy facilitates the mutual information between latent features and semantic objects.

Fig. 5(a) illustrates the CKA between the unmasked and masked tokens output by the decoder during pre-training. The curves and shaded bands represent the mean and variance across all samples. We compared I^{2}Mask with random masking, which is used by the majority of studies. Our proposed I^{2}Mask adjusts the difficulty of reconstructing masked tokens from unmasked tokens based on the discrimination of high-level features. For regions with small discrimination, we set a lower masking ratio to simplify the reconstruction. Compared to random masking, the CKA between unmasked and masked tokens increases by approximately 30%. As proven in Proposition 4, a smoother information flow between unmasked and masked tokens enhances the mutual information between latent features and high-level features.

Fig. 5(b) shows the loss curves for high-level feature reconstruction of the final layer. We fitted smooth curves to visualize the loss trend better. First, it can be observed that CHR w/ deep supervision converges the earliest. For pre-training w.o. deep supervision, the learning of shallow spatial information is slow, resulting in slow convergence. Fixed weighting w/ deep supervision causes confusion in hierarchical learning. Second, after supplementing DGPL, the loss oscillation becomes more stable thanks to progressive learning. Finally, because of the clear learning path, CHR + DGPL with deep supervision results in the lowest high-level feature reconstruction loss. This demonstrates efficient knowledge transfer in multiform optimization and more significant mutual information between latent features and high-level features, as proven in Propositions 5, 6.

5.10. Exploratory model analysis

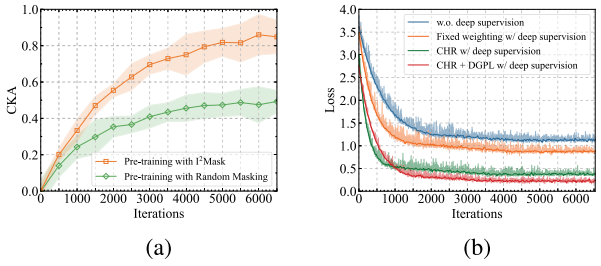
Pretext tasks are essential in generative self-supervised point cloud pre-training. Appropriately designed pretext tasks guide the network towards more semantically meaningful features. Furthermore, the distinction in mutual information among latent features can also be quantitatively represented as model weights. Therefore, we explored the differences between pretext tasks from the perspective of weight distribution. We selected the weights of the value nodes in all attention

Table 20
Self-supervised learning across datasets and joint datasets in 3D object detection.

Pre-training \ Downstream	Waymo		nuScenes	
	L2 mAP↑	L2 mAPH↑	mAP↑	NDS↑
×	73.20	71.00	66.40	71.10
Waymo	75.55$^{+2.35}$	73.13$^{+2.13}$	67.53$^{+1.13}$	72.14$^{+1.04}$
nuScenes	74.47$^{+1.27}$	72.48$^{+1.48}$	68.60$^{+2.20}$	73.00$^{+1.90}$
Waymo + nuScenes	**75.73**$^{+2.53}$	**73.21**$^{+2.21}$	**68.93**$^{+2.53}$	**73.26**$^{+2.16}$

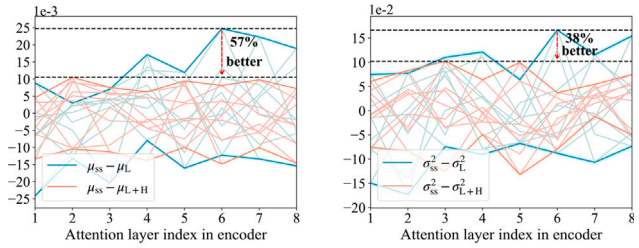
Table 21
Time cost of self-supervised learning methods during pre-training on 8 A100-SXM4-40 GB GPUs on Waymo.

Method	Epochs	Fraction	Time
MV-JAR (DSVT) [15]	30	100%	61 h
GeoMAE (DSVT) [7]	30	100%	48 h
GPICTURE (online) (DSVT)	30	100%	244 h
GPICTURE (offline) (DSVT)	30	100%	65 h



(a) (b)

Fig. 5. (a) CKA between the unmasked and masked tokens output by the decoder during pre-training. (b) Loss curves for high-level feature reconstruction of the final layer.



(a) (b)

Fig. 6. The disparity in the weight distribution of each attention layer in the encoder. (a) and (b) represent mean and variance, respectively.

layers of the encoder as our research subject, including the weights from multi-heads.

We used DSVT [21] as the encoder. We employed the reconstruction of 3D point cloud coordinates as a low-level pretext task, and the reconstruction of Seal voxel features as a high-level pretext task. To simplify the analysis, we assumed that the weights follow a Gaussian distribution [105] and computed the mean μ_L, μ_{L+H} and variance $\sigma^2_L, \sigma^2_{L+H}$ using maximum likelihood estimation. We further assumed the weight distribution derived from the supervised learning μ_s, σ^2_s served as the ideal weight distribution for self-supervised learning μ_{ss}, σ^2_{ss}. Fig. 6(a) and (b) show the disparity from the ideal weight distribution when using only low-level features and when supplementing with high-level features. We used differences to measure disparity without loss of generality. After supplementation with high-level features, the model weights obtained from self-supervised learning are closer to those of downstream tasks. This serves as the evidence that the mutual information between their latent features increases. Moreover, smaller weight disparity in the deeper layers indicates that the model learned more semantic features during pre-training, as proven in Proposition 2.

6. Limitations

First, we conclude in the Theoretical Analysis that the mutual information between latent features and high-level features, as well as the input point cloud, depends on the mutual information between the high-level features and the input point cloud. However, the weak high-level feature extractor limits performance. Owing to the lack of large-scale point cloud-image-text datasets, high-level feature extractors in autonomous driving scenarios cannot leverage the powerful understanding abilities of large language models and are limited to visual concepts. **Second**, the time cost of extracting Seal features is higher compared to low-level features such as geometric features. We pre-extracted Seal features for all point cloud scenes in the entire dataset. Loading these features offline during pre-training allows the time cost of GPICTURE to be similar to that of other self-supervised methods. However, the process of pre-extracting Seal features incurs a significant time cost. **Third**, the method used in CHR to calculate the proportion of low-level information is not sufficiently refined. We used only the three downstream tasks of the corresponding dataset to calculate mutual information. The combination of more downstream tasks and datasets will yield more comprehensive results. **Finally**, pre-training may be unstable in areas far from ego, where the point clouds are sparse. It is not easy to reconstruct the masked voxels at excessively far distances because of the lack of nearby unmasked voxels.

7. Future work

We propose the following future work directions to improve the learning of universal representations from extensive unlabeled 3D point clouds and their application in practical autonomous driving scenarios. **First**, we will develop more robust high-level feature extractors with semantic information. Utilizing high-level features with higher mutual information with the input point cloud in our framework will provide a good training starting point for downstream tasks. Combining textual descriptions and the powerful understanding abilities of large language models is a promising path to achieve this more robust high-level feature extractor. **Second**, we will explore improvements to the model under various environments and conditions, such as weather conditions, lighting variations, and sensor noise, to enhance the generalizability of the pre-trained model. **Third**, we will investigate online learning or incremental learning strategies to enable the model to adapt to new data and environmental changes without training from scratch. **Finally**, we will combine I^{2}Mask with strategies such as Range-aware Masking to avoid unstable training in areas far from ego.

8. Conclusion

Generative self-supervised point cloud pre-training in autonomous driving is trapped in spatial information and entangled representations. To address these issues, we proposed GPICTURE, a generalized generative self-supervised point cloud pre-training framework that systematically integrated high-level features as pretext tasks. With an adaptive masking sampling strategy I^{2}Mask and well-designed multi-task control strategies CHR and DGPL, GPICTURE learnt to extract semantic and disentangled 3D representations by performing high-level feature reconstruction tasks. The concise information flow definition and theoretical analysis provide strong support for method design. Extensive experiments demonstrated the superiority of GPICTURE in leveraging large-scale unlabeled point clouds. We believe that this mutual information-driven idea will become an important concept in designing algorithms for generative self-supervised point cloud pre-training.

CRediT authorship contribution statement

Weichen Xu: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Tianhao Fu:** Writing – review & editing, Visualization, Investigation, Formal analysis, Conceptualization. **Jian Cao:** Writing – review & editing, Validation, Supervision, Resources, Project administration, Funding acquisition, Conceptualization. **Xinyu Zhao:** Writing – review & editing, Visualization, Validation, Software, Investigation, Formal analysis, Conceptualization. **Xinxin Xu:** Writing – review & editing, Visualization, Software, Investigation, Formal analysis, Data curation, Conceptualization. **Xixin Cao:** Writing – review & editing, Supervision, Resources, Methodology, Investigation, Funding acquisition, Formal analysis, Conceptualization. **Xing Zhang:** Writing – review & editing, Validation, Supervision, Resources, Project administration, Funding acquisition, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (No. 62406324), the Natural Science Foundation of Beijing Municipality, China (No. 4244100), and the Science and Technology Planning Project of Shenzhen Municipality, China (No. KQTD20200820113105004). We thank all reviewers who generously contributed their time and efforts.

Data availability

The project page is hosted at https://gpicture-page.github.io/.

References

[1] T. Huang, R. Fu, Prediction of the driver's focus of attention based on feature visualization of a deep autonomous driving model, Knowl.-Based Syst. 251 (2022) 109006, http://dx.doi.org/10.1016/j.knosys.2022.109006.

[2] A.H. Lang, S. Vora, H. Caesar, L. Zhou, et al., Pointpillars: Fast encoders for object detection from point clouds, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 12697–12705, http://dx.doi.org/10.1109/cvpr.2019.01298.

[3] S. Shi, C. Guo, L. Jiang, Z. Wang, J. Shi, X. Wang, H. Li, Pv-rcnn: Point-voxel feature set abstraction for 3d object detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 10529–10538, http://dx.doi.org/10.1109/cvpr42600.2020.01054.

[4] X. Lai, Y. Chen, F. Lu, J. Liu, J. Jia, Spherical transformer for lidar-based 3d recognition, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 17545–17555, http://dx.doi.org/10.1109/cvpr52729.2023.01683.

[5] J. Mao, M. Niu, C. Jiang, et al., One million scenes for autonomous driving: ONCE dataset, in: Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks, 2021.

[6] L. Kong, N. Quader, et al., Conda: Unsupervised domain adaptation for lidar segmentation via regularized domain concatenation, in: IEEE International Conference on Robotics and Automation, IEEE, 2023, pp. 9338–9345, http://dx.doi.org/10.1109/icra48891.2023.10160410.

[7] X. Tian, H. Ran, et al., GeoMAE: Masked geometric target prediction for self-supervised point cloud pre-training, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 13570–13580, http://dx.doi.org/10.1109/cvpr52729.2023.01304.

[8] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, R. Girshick, Masked autoencoders are scalable vision learners, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 16000–16009, http://dx.doi.org/10.1109/cvpr52688.2022.01553.

[9] C. Zeng, W. Wang, A. Nguyen, Y. Yue, Self-supervised learning for point cloud data: A survey, Expert Syst. Appl. (2023) 121354, http://dx.doi.org/10.1016/j.eswa.2023.121354.

[10] A. Xiao, J. Huang, D. Guan, X. Zhang, S. Lu, L. Shao, Unsupervised point cloud representation learning with deep neural networks: A survey, IEEE Trans. Pattern Anal. Mach. Intell. 45 (9) (2023) 11321–11339, http://dx.doi.org/10.1109/tpami.2023.3262786.

[11] B. Fei, W. Yang, L. Liu, T. Luo, R. Zhang, Y. Li, Y. He, Self-supervised learning for pre-training 3d point clouds: A survey, 2023, arXiv preprint arXiv:2305.04691.

[12] S.S. Sohail, Y. Himeur, A. Amira, F. Fadli, W. Mansoor, S. Atalla, A. Copiaco, Deep transfer learning for 3d point cloud understanding: A comprehensive survey, 2023, Available at SSRN 4348272.

[13] C. Min, L. Xiao, D. Zhao, et al., Occupancy-MAE: Self-supervised pre-training large-scale LiDAR point clouds with masked occupancy autoencoders, IEEE Trans. Intell. Veh. (2023) http://dx.doi.org/10.1109/tiv.2023.3322409.

[14] H. Yang, T. He, J. Liu, et al., GD-MAE: generative decoder for MAE pre-training on lidar point clouds, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 9403–9414, http://dx.doi.org/10.1109/cvpr52729.2023.00907.

[15] R. Xu, T. Wang, W. Zhang, R. Chen, J. Cao, J. Pang, D. Lin, MV-JAR: Masked voxel jigsaw and reconstruction for LiDAR-based self-supervised pre-training, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 13445–13454, http://dx.doi.org/10.1109/cvpr52729.2023.01292.

[16] P. Bachman, R.D. Hjelm, W. Buchwalter, Learning representations by maximizing mutual information across views, Adv. Neural Inf. Process. Syst. 32 (2019) http://dx.doi.org/10.5555/3454287.3455679.

[17] A.H. Liu, S.-L. Yeh, J.R. Glass, Revisiting self-supervised learning of speech representation from a mutual information perspective, in: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP, IEEE, 2024, pp. 12051–12055, ICASSP48485.2024.10447758.

[18] D. Liu, X. Fang, X. Qu, J. Dong, H. Yan, Y. Yang, P. Zhou, Y. Cheng, Unsupervised domain adaptative temporal sentence localization with mutual information maximization, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 38, 2024, pp. 3567–3575, http://dx.doi.org/10.1609/aaai.v38i4.28145.

[19] S. Liu, M. Tian, Mutual information maximization for semi-supervised anomaly detection, Knowl.-Based Syst. 284 (2024) 111196, http://dx.doi.org/10.1016/j.knosys.2023.111196.

[20] S. Kornblith, M. Norouzi, H. Lee, G. Hinton, Similarity of neural network representations revisited, in: Proceedings of the International Conference on Machine Learning, PMLR, 2019, pp. 3519–3529.

[21] H. Wang, C. Shi, S. Shi, M. Lei, S. Wang, D. He, B. Schiele, L. Wang, Dsvt: Dynamic sparse voxel transformer with rotated sets, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 13520–13529, http://dx.doi.org/10.1109/cvpr52729.2023.01299.

[22] Y. Guo, H. Wang, Q. Hu, H. Liu, L. Liu, M. Bennamoun, Deep learning for 3d point clouds: A survey, IEEE Trans. Pattern Anal. Mach. Intell. 43 (12) (2020) 4338–4364, http://dx.doi.org/10.1109/TPAMI.2020.3005434.

[23] Y. Wu, H. Ding, M. Gong, A.K. Qin, W. Ma, Q. Miao, K.C. Tan, Evolutionary multiform optimization with two-stage bidirectional knowledge transfer strategy for point cloud registration, IEEE Trans. Evol. Comput. 28 (1) (2022) 62–76, http://dx.doi.org/10.1109/tevc.2022.3215743.

[24] Y. Wu, H. Ding, B. Xiang, J. Sheng, W. Ma, K. Qin, Q. Miao, M. Gong, Evolutionary multitask optimization in real-world applications: A survey, J. Artif. Intell. Technol. 3 (1) (2023) 32–38, http://dx.doi.org/10.37965/jait.2023.0149.

[25] Y. Wu, J. Sheng, H. Ding, P. Gong, H. Li, M. Gong, W. Ma, Q. Miao, Evolutionary multitasking descriptor optimization for point cloud registration, IEEE Trans. Evol. Comput. (2024) http://dx.doi.org/10.1109/tevc.2024.3417416.

[26] H. Ding, Y. Wu, M. Gong, H. Li, P. Gong, Q. Miao, W. Ma, Y. Duan, X. Tao, Point cloud registration via sampling-based evolutionary multitasking, Swarm Evol. Comput. 89 (2024) 101535, http://dx.doi.org/10.1016/j.swevo.2024.101535.

[27] A. Gupta, Y.-S. Ong, L. Feng, Multifactorial evolution: Toward evolutionary multitasking, IEEE Trans. Evol. Comput. 20 (3) (2015) 343–357, http://dx.doi.org/10.1109/tevc.2015.2458037.

[28] A. Gupta, Y.-S. Ong, L. Feng, Insights on transfer optimization: Because experience is the best teacher, IEEE Trans. Emerg. Top. Comput. Intell. 2 (1) (2017) 51–64, http://dx.doi.org/10.1109/tetci.2017.2769104.

[29] C. Sautier, G. Puy, S. Gidaris, A. Boulch, A. Bursuc, R. Marlet, Image-to-lidar self-supervised distillation for autonomous driving data, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 9891–9901, http://dx.doi.org/10.1109/cvpr52688.2022.00966.

[30] Y. Liu, L. Kong, J. Cen, R. Chen, W. Zhang, L. Pan, K. Chen, Z. Liu, Segment any point cloud sequences by distilling vision foundation models, Adv. Neural Inf. Process. Syst. 36 (2024) http://dx.doi.org/10.5555/3666122.3667739.

[31] R. Achanta, A. Shaji, K. Smith, A. Lucchi, et al., SLIC superpixels compared to state-of-the-art superpixel methods, IEEE Trans. Pattern Anal. Mach. Intell. 34 (11) (2012) 2274–2282, http://dx.doi.org/10.1109/tpami.2012.120.

[32] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, et al., Segment anything, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2023, pp. 4015–4026, http://dx.doi.org/10.1109/ICCV51070.2023.00371.

[33] Y. Zeng, C. Jiang, J. Mao, J. Han, C. Ye, Q. Huang, D.-Y. Yeung, et al., CLIP2: Contrastive language-image-point pretraining from real-world point cloud data, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 15244–15253, http://dx.doi.org/10.1109/cvpr52729.2023.01463.

[34] A. Radford, J.W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al., Learning transferable visual models from natural language supervision, in: Proceedings of the International Conference on Machine Learning, PMLR, 2021, pp. 8748–8763.

[35] W. Xu, J. Cao, T. Fu, R. Ren, Z. Hu, X. Cao, X. Zhang, Point cloud reconstruction is insufficient to learn 3D representations, in: Proceedings of the ACM International Conference on Multimedia, 2024, http://dx.doi.org/10.1145/3664647.3680890.

[36] X. Wang, Y. Chen, W. Zhu, A survey on curriculum learning, IEEE Trans. Pattern Anal. Mach. Intell. 44 (9) (2021) 4555–4576, http://dx.doi.org/10.1109/TPAMI.2021.3069908.

[37] P. Sun, H. Kretzschmar, X. Dotiwalla, A. Chouard, V. Patnaik, P. Tsui, J. Guo, Y. Zhou, Y. Chai, et al., Scalability in perception for autonomous driving: Waymo open dataset, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 2446–2454, http://dx.doi.org/10.1109/cvpr42600.2020.00252.

[38] H. Caesar, V. Bankiti, A.H. Lang, S. Vora, V.E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, O. Beijbom, nuscenes: A multimodal dataset for autonomous driving, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 11621–11631, http://dx.doi.org/10.1109/cvpr42600.2020.01164.

[39] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, C. Stachniss, J. Gall, Semantickitti: A dataset for semantic scene understanding of lidar sequences, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 9297–9307, http://dx.doi.org/10.1109/iccv.2019.00939.

[40] C.R. Qi, H. Su, K. Mo, L.J. Guibas, Pointnet: Deep learning on point sets for 3d classification and segmentation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2017, pp. 652–660, http://dx.doi.org/10.1109/cvpr.2017.16.

[41] H. Wang, H. Tang, S. Shi, A. Li, Z. Li, B. Schiele, L. Wang, UniTR: A unified and efficient multi-modal transformer for bird's-eye-view representation, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2023, pp. 6792–6802, http://dx.doi.org/10.1109/iccv51070.2023.00625.

[42] C.R. Qi, L. Yi, H. Su, L.J. Guibas, Pointnet++: Deep hierarchical feature learning on point sets in a metric space, Adv. Neural Inf. Process. Syst. 30 (2017) http://dx.doi.org/10.5555/3295222.3295263.

[43] Y. Liu, B. Fan, S. Xiang, C. Pan, Relation-shape convolutional neural network for point cloud analysis, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 8895–8904, http://dx.doi.org/10.1109/cvpr.2019.00910.

[44] C.R. Qi, O. Litany, K. He, L.J. Guibas, Deep hough voting for 3d object detection in point clouds, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 9277–9286, http://dx.doi.org/10.1109/iccv.2019.00937.

[45] S. Shi, X. Wang, H. Li, Pointrcnn: 3d object proposal generation and detection from point cloud, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 770–779, http://dx.doi.org/10.1109/cvpr.2019.00086.

[46] Z. Yang, Y. Sun, S. Liu, J. Jia, 3Dssd: Point-based 3d single stage object detector, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 11040–11048, http://dx.doi.org/10.1109/cvpr42600.2020.01105.

[47] B. Cheng, L. Sheng, S. Shi, M. Yang, D. Xu, Back-tracing representative points for voting-based 3d object detection in point clouds, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 8963–8972, http://dx.doi.org/10.1109/cvpr46437.2021.00885.

[48] X. Pan, Z. Xia, S. Song, L.E. Li, G. Huang, 3D object detection with pointformer, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 7463–7472, http://dx.doi.org/10.1109/cvpr46437.2021.00738.

[49] H. Zhao, L. Jiang, J. Jia, P.H. Torr, V. Koltun, Point transformer, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 16259–16268, http://dx.doi.org/10.1109/iccv48922.2021.01595.

[50] Z. Liu, Z. Zhang, Y. Cao, H. Hu, X. Tong, Group-free 3d object detection via transformers, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 2949–2958, http://dx.doi.org/10.1109/iccv48922.2021.00294.

[51] X. Wu, Y. Lao, L. Jiang, X. Liu, H. Zhao, Point transformer v2: Grouped vector attention and partition-based pooling, Adv. Neural Inf. Process. Syst. 35 (2022) 33330–33342, http://dx.doi.org/10.5555/3600270.3602685.

[52] C. Park, Y. Jeong, M. Cho, J. Park, Fast point transformer, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 16949–16958, http://dx.doi.org/10.1109/cvpr52688.2022.01644.

[53] X. Lai, J. Liu, L. Jiang, L. Wang, H. Zhao, S. Liu, X. Qi, J. Jia, Stratified transformer for 3d point cloud segmentation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 8500–8509, http://dx.doi.org/10.1109/cvpr52688.2022.00831.

[54] Y. Shen, L. Hui, FlowFormer: 3D scene flow estimation for point clouds with transformers, Knowl.-Based Syst. 280 (2023) 111041, http://dx.doi.org/10.1016/j.knosys.2023.111041.

[55] G. Wang, Q. Zhai, H. Liu, Cross self-attention network for 3D point cloud, Knowl.-Based Syst. 247 (2022) 108769, http://dx.doi.org/10.1016/j.knosys.2022.108769.

[56] Y. Yan, Y. Mao, B. Li, Second: Sparsely embedded convolutional detection, Sensors 18 (10) (2018) 3337, http://dx.doi.org/10.3390/s18103337.

[57] B. Graham, M. Engelcke, L. Van Der Maaten, 3D semantic segmentation with submanifold sparse convolutional networks, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018, pp. 9224–9232, http://dx.doi.org/10.1109/cvpr.2018.00961.

[58] C. Choy, J. Gwak, S. Savarese, 4D spatio-temporal convnets: Minkowski convolutional neural networks, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 3075–3084, http://dx.doi.org/10.1109/cvpr.2019.00319.

[59] S. Shi, Z. Wang, J. Shi, X. Wang, H. Li, From points to parts: 3d object detection from point cloud with part-aware and part-aggregation network, IEEE Trans. Pattern Anal. Mach. Intell. 43 (8) (2020) 2647–2664, http://dx.doi.org/10.1109/tpami.2020.2977026.

[60] T. Yin, X. Zhou, et al., Center-based 3d object detection and tracking, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 11784–11793, http://dx.doi.org/10.1109/cvpr46437.2021.01161.

[61] J. Deng, S. Shi, P. Li, W. Zhou, Y. Zhang, H. Li, Voxel r-cnn: Towards high performance voxel-based 3d object detection, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 35, 2021, pp. 1201–1209, http://dx.doi.org/10.1609/aaai.v35i2.16207.

[62] J. Mao, Y. Xue, M. Niu, H. Bai, J. Feng, X. Liang, H. Xu, C. Xu, Voxel transformer for 3d object detection, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 3164–3173, http://dx.doi.org/10.1109/iccv48922.2021.00315.

[63] L. Fan, Z. Pang, T. Zhang, et al., Embracing single stride 3d object detector with sparse transformer, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 8458–8468, http://dx.doi.org/10.1109/cvpr52688.2022.00827.

[64] C. He, R. Li, S. Li, L. Zhang, Voxel set transformer: A set-to-set approach to 3d object detection from point clouds, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 8417–8427, http://dx.doi.org/10.1109/cvpr52688.2022.00823.

[65] S. Dong, L. Ding, H. Wang, T. Xu, X. Xu, J. Wang, Z. Bian, Y. Wang, J. Li, Mssvt: Mixed-scale sparse voxel transformer for 3d object detection on point clouds, Adv. Neural Inf. Process. Syst. 35 (2022) 11615–11628, http://dx.doi.org/10.5555/3600270.3601114.

[66] Z. Liu, X. Yang, H. Tang, S. Yang, S. Han, Flatformer: Flattened window attention for efficient point cloud transformer, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 1200–1211, http://dx.doi.org/10.1109/cvpr52729.2023.00122.

[67] J. Yin, D. Zhou, L. Zhang, J. Fang, C.-Z. Xu, et al., Proposalcontrast: Unsupervised pre-training for lidar-based 3d object detection, in: Proceedings of the European Conference on Computer Vision, Springer, 2022, pp. 17–33, http://dx.doi.org/10.1007/978-3-031-19842-7_2.

[68] L. Nunes, L. Wiesmann, R. Marcuzzi, X. Chen, J. Behley, C. Stachniss, Temporal consistent 3d lidar representation learning for semantic perception in autonomous driving, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 5217–5228, http://dx.doi.org/10.1109/cvpr52729.2023.00505.

[69] M. Afham, I. Dissanayake, D. Dissanayake, A. Dharmasiri, K. Thilakarathna, R. Rodrigo, Crosspoint: Self-supervised cross-modal contrastive learning for 3d point cloud understanding, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 9902–9912, http://dx.doi.org/10.1109/cvpr52688.2022.00967.

[70] R. Zhang, Z. Guo, P. Gao, R. Fang, B. Zhao, D. Wang, Y. Qiao, H. Li, Point-m2ae: multi-scale masked autoencoders for hierarchical point cloud pre-training, Adv. Neural Inf. Process. Syst. 35 (2022) 27061–27074, http://dx.doi.org/10.5555/3600270.3602232.

[71] G. Krispel, D. Schinagl, C. Fruhwirth-Reisinger, et al., MAELi: Masked autoencoder for large-scale LiDAR point clouds, in: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2024, pp. 3383–3392, http://dx.doi.org/10.1109/wacv57701.2024.00335.

[72] G. Hess, J. Jaxing, E. Svensson, D. Hagerman, C. Petersson, L. Svensson, Masked autoencoder for self-supervised pre-training on lidar point clouds, in: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2023, pp. 350–359, http://dx.doi.org/10.1109/wacvw58289.2023.00039.

[73] Z. Yang, L. Chen, Y. Sun, H. Li, Visual point cloud forecasting enables scalable autonomous driving, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024, pp. 14673–14684, http://dx.doi.org/10.1109/cvpr52733.2024.01390.

[74] H. Yang, S. Zhang, D. Huang, X. Wu, H. Zhu, T. He, S. Tang, H. Zhao, Q. Qiu, B. Lin, et al., Unipad: A universal pre-training paradigm for autonomous driving, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024, pp. 15238–15250, http://dx.doi.org/10.1109/cvpr52733.2024.01443.

[75] Z. Lin, Y. Wang, et al., BEV-MAE: Bird's eye view masked autoencoders for point cloud pre-training in autonomous driving scenarios, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 38, 2024, pp. 3531–3539, http://dx.doi.org/10.1609/aaai.v38i4.28141.

[76] C. Min, L. Xiao, et al., Multi-camera unified pre-training via 3D scene reconstruction, IEEE Robot. Autom. Lett. (2024) http://dx.doi.org/10.1109/lra.2024.3362635.

[77] A. Boulch, C. Sautier, B. Michele, G. Puy, Also: Automotive lidar self-supervision by occupancy estimation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 13455–13465, http://dx.doi.org/10.1109/cvpr52729.2023.01293.

[78] I. Kakogeorgiou, S. Gidaris, B. Psomas, Y. Avrithis, et al., What to hide from your students: Attention-guided masked image modeling, in: Proceedings of the European Conference on Computer Vision, Springer, 2022, pp. 300–318, http://dx.doi.org/10.1007/978-3-031-20056-4_18.

[79] G. Li, H. Zheng, D. Liu, C. Wang, B. Su, et al., Semmae: Semantic-guided masking for learning masked autoencoders, Adv. Neural Inf. Process. Syst. 35 (2022) 14290–14302, http://dx.doi.org/10.5555/3600270.3601309.

[80] Z. Liu, J. Gui, H. Luo, Good helper is around you: Attention-driven masked image modeling, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 37, 2023, pp. 1799–1807, http://dx.doi.org/10.1609/aaai.v37i2.25269.

[81] H. Ding, Z. Wu, L. Zhao, Whale optimization algorithm based on nonlinear convergence factor and chaotic inertial weight, Concurr. Comput.: Pract. Exper. 32 (24) (2020) e5949, http://dx.doi.org/10.1002/cpe.5949.

[82] M. Assran, Q. Duval, I. Misra, P. Bojanowski, P. Vincent, M. Rabbat, Y. LeCun, N. Ballas, Self-supervised learning from images with a joint-embedding predictive architecture, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 15619–15629, http://dx.doi.org/10.1109/cvpr52729.2023.01499.

[83] W.K. Fong, R. Mohan, J.V. Hurtado, L. Zhou, et al., Panoptic nuscenes: A large-scale benchmark for lidar panoptic segmentation and tracking, IEEE Robot. Autom. Lett. 7 (2) (2022) 3795–3802, http://dx.doi.org/10.1109/lra.2022.3148457.

[84] Y. Cheng, Mean shift, mode seeking, and clustering, IEEE Trans. Pattern Anal. Mach. Intell. 17 (8) (1995) 790–799, http://dx.doi.org/10.1109/34.400568.

[85] H. Wang, Y. Tang, Y. Wang, J. Guo, Z.-H. Deng, K. Han, Masked image modeling with local multi-scale reconstruction, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 2122–2131, http://dx.doi.org/10.1109/cvpr52729.2023.00211.

[86] Y. Liu, M.-M. Cheng, D.-P. Fan, L. Zhang, J.-W. Bian, D. Tao, Semantic edge detection with diverse deep supervision, Int. J. Comput. Vis. 130 (1) (2022) 179–198, http://dx.doi.org/10.1007/s11263-021-01539-8.

[87] E.T. Jaynes, Information theory and statistical mechanics, Phys. Rev. 106 (4) (1957) 620, http://dx.doi.org/10.1103/PhysRev.106.620.

[88] R. Linsker, An application of the principle of maximum information preservation to linear systems, Adv. Neural Inf. Process. Syst. 1 (1988) http://dx.doi.org/10.5555/2969735.2969757.

[89] S. Kullback, R.A. Leibler, On information and sufficiency, Ann. Math. Stat. 22 (1) (1951) 79–86, http://dx.doi.org/10.1214/aoms/1177729694.

[90] J. Lin, Divergence measures based on the Shannon entropy, IEEE Trans. Inform. Theory 37 (1) (1991) 145–151, http://dx.doi.org/10.1109/18.61115.

[91] S. Nowozin, B. Cseke, R. Tomioka, f-gan: Training generative neural samplers using variational divergence minimization, Adv. Neural Inf. Process. Syst. 29 (2016) http://dx.doi.org/10.5555/3157096.3157127.

[92] B. Heo, J. Kim, S. Yun, H. Park, N. Kwak, J.Y. Choi, A comprehensive overhaul of feature distillation, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 1921–1930, http://dx.doi.org/10.1109/iccv.2019.00201.

[93] S. Jung, D. Lee, T. Park, T. Moon, Fair feature distillation for visual recognition, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 12115–12124, http://dx.doi.org/10.1109/cvpr46437.2021.01194.

[94] L. Zhang, Y. Shi, Z. Shi, K. Ma, C. Bao, Task-oriented feature distillation, Adv. Neural Inf. Process. Syst. 33 (2020) 14759–14771, http://dx.doi.org/10.5555/3495724.3496961.

[95] D.M. Blei, A. Kucukelbir, J.D. McAuliffe, Variational inference: A review for statisticians, J. Amer. Statist. Assoc. 112 (518) (2017) 859–877, http://dx.doi.org/10.1080/01621459.2017.1285773.

[96] T.M. Cover, J.A. Thomas, et al., Entropy, relative entropy and mutual information, Elements Inf. Theory 2 (1) (1991) 12–13, http://dx.doi.org/10.1002/0471200611.ch2.

[97] H. Fan, H. Su, L.J. Guibas, A point set generation network for 3d object reconstruction from a single image, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2017, pp. 605–613, http://dx.doi.org/10.1109/cvpr.2017.264.

[98] X. Wang, Z. Zhu, W. Xu, Y. Zhang, Y. Wei, X. Chi, Y. Ye, D. Du, J. Lu, X. Wang, Openoccupancy: A large scale benchmark for surrounding semantic occupancy perception, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2023, pp. 17850–17859, http://dx.doi.org/10.1109/iccv51070.2023.01636.

[99] O.D. Team, OpenPCDet: An open-source toolbox for 3D object detection from point clouds, 2020, https://github.com/open-mmlab/OpenPCDet.

[100] M. Contributors, MMDetection3D: OpenMMLab next-generation platform for general 3D object detection, 2020, https://github.com/open-mmlab/mmdetection3d.

[101] S. Shi, L. Jiang, J. Deng, Z. Wang, C. Guo, J. Shi, X. Wang, H. Li, PV-RCNN++: Point-voxel feature set abstraction with local vector representation for 3D object detection, Int. J. Comput. Vis. 131 (2) (2023) 531–551, http://dx.doi.org/10.1007/s11263-022-01710-9.

[102] Y. Lu, X. Ma, L. Yang, T. Zhang, Y. Liu, Q. Chu, J. Yan, W. Ouyang, Geometry uncertainty projection network for monocular 3d object detection, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 3111–3121, http://dx.doi.org/10.1109/iccv48922.2021.00310.

[103] H. Yun, H. Cho, Achievement-based training progress balancing for multi-task learning, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2023, pp. 16935–16944, http://dx.doi.org/10.1109/iccv51070.2023.01553.

[104] R. Chen, Y. Liu, L. Kong, X. Zhu, Y. Ma, Y. Li, Y. Hou, Y. Qiao, W. Wang, CLIP2Scene: Towards label-efficient 3D scene understanding by CLIP, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 7020–7030, http://dx.doi.org/10.1109/cvpr52729.2023.00678.

[105] X. Glorot, Y. Bengio, Understanding the difficulty of training deep feedforward neural networks, in: Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, JMLR Workshop and Conference Proceedings, 2010, pp. 249–256.