GeoMIM: Towards Better 3D Knowledge Transfer via Masked Image Modeling for Multi-view 3D Understanding

Jihao Liu⁵, Tai Wang⁴, Boxiao Liu², Qihang Zhang¹, Yu Liu², Hongsheng Li²,³,⁴
¹CUHK MMLab ²SenseTime Research ³Shanghai AI Laboratory ⁴CPII under InnoHK

Abstract

Multi-view camera-based 3D detection is a challenging problem in computer vision. Recent works leverage a pretrained LiDAR detection model to transfer knowledge to a camera-based student network. However, we argue that there is a major domain gap between the LiDAR BEV features and the camera-based BEV features, as they have different characteristics and are derived from different sources. In this paper, we propose Geometry Enhanced Masked Image Modeling (GeoMIM) to transfer the knowledge of the LiDAR model in a pretrain-finetune paradigm for improving the multi-view camera-based 3D detection. GeoMIM is a multi-camera vision transformer with Cross-View Attention (CVA) blocks that uses LiDAR BEV features encoded by the pretrained BEV model as learning targets. During pretraining, GeoMIM’s decoder has a semantic branch completing dense perspective-view features and the other geometry branch reconstructing dense perspective-view depth maps. The depth branch is designed to be camera-aware by inputting the camera’s parameters for better transfer capability. Extensive results demonstrate that GeoMIM outperforms existing methods on nuScenes benchmark, achieving state-of-the-art performance for camera-based 3D object detection and 3D segmentation.

1. Introduction

Multi-view camera-based 3D detection is an emerging critical problem in computer vision [20, 46, 47, 19, 28, 29, 36, 26, 31, 32, 40, 41]. To improve the detection performance, recent works [9, 27, 21] often choose to use a pretrained LiDAR model as the teacher and transfer its knowledge to a camera-based student network. Various techniques, such as LIGA-Stereo [13], CMKD [17], and BEVDistill [8], have been proposed to leverage the rich geometry information of the LiDAR model’s BEV (bird’s eye view) features.

Utilizing a pretrained LiDAR model to provide auxiliary supervision has become a widely adopted design that can enhance the performance of camera-based models. However, we contend that this design is not optimal due to a significant domain gap between the BEV features of the LiDAR model and those of the camera-based model. This domain gap arises from the 3D and sparse characteristics of LiDAR point clouds compared to the dense 2D images captured by the camera. Additionally, the LiDAR model’s BEV features are grounded in ground truth depth, while those of the camera-based model are typically inferred from 2D images, a problem that is often ill-posed. We empirically demonstrate their domain gap with a pilot study as shown in Tab. 1. We find that utilizing a LiDAR teacher to provide auxiliary supervision can indeed improve an ImageNet-pretrained [33] camera-based model, but is unable to improve a stronger camera-based model initialized by recent powerful self-supervised pretraining. In other words, directly utilizing the pretrained LiDAR model to distill the final camera-based model might not be an optimal design and does not necessarily lead to performance gain.

To better take advantage of the LiDAR model, in this paper, we propose Geometry Enhanced Masked Image Modeling (GeoMIM) to transfer the knowledge of the LiDAR model in a pretrain-finetune paradigm for improving the multi-view camera-based 3D detection. It is built upon a multi-camera vision transformer with Cross-View Attention (CVA) blocks and enables perspective-view (PV) representation pretraining via BEV feature reconstruction from masked images. Specifically, during pretraining, we parti-
tion the training images into patches and feed a portion of them into the encoder following Masked Autoencoder [14]. Our GeoMIM decoder then uses these encoded visible tokens to reconstruct the pretrained LiDAR model’s BEV feature in the BEV space instead of commonly used RGB pixels [49, 14, 30] or depth points [3] as in existing MAE frameworks. To achieve this PV to BEV reconstruction, we first devise two branches to decouple the semantic and geometric parts, with one branch completing dense PV features and the other reconstructing the depth map. The dense PV features can then be projected into the BEV space with the depth distribution following Lift-Splat-Shoot (LSS) [37]. We further equip the two branches with the proposed CVA blocks in their intermediate layers to allow each patch to attend to tokens in other views. It enhances the decoder’s capability of joint multi-view inference which is especially critical for BEV feature reconstruction. Finally, the depth branch is designed to be camera-aware with the additional encoding of cameras’ parameters as input, making the pretrained GeoMIM better adapt to downstream tasks with different cameras.

To demonstrate the effectiveness of GeoMIM, we fine-tune the pretrained backbone to conduct multi-view camera-based 3D detection and 3D segmentation on the nuScenes [7] dataset. We achieve state-of-the-art results of 64.4 NDS (NuScenes Detection Score) and 70.5 mIoU (mean intersection over union) for 3D detection and segmentation on the NuScenes test set, which are 2.5% and 1.1% better than previously reported best results [36, 22]. Additionally, we verify that the backbone pretrained on nuScenes dataset can be successfully transferred to Waymo Open dataset [42], improving the mAP (mean average precision) of the ImageNet-initialized 3D detector by 6.9%.

2. Related Works

Masked Image Modeling Inspired by BERT [11] for Masked Language Modeling, Masked Image Modeling (MIM) becomes a popular pretext task for visual representation learning [6, 14, 2, 48, 1, 4, 53, 3, 51]. MIM aims to reconstruct the masked tokens from a corrupted input. SimMIM [49] points out that raw pixel values of the randomly masked patches are a good reconstruction target and a lightweight prediction head is sufficient for pretraining. Different from SimMIM, MAE [14] only takes the visible patches as the input of the encoder. Mask tokens are added in the middle of the encoder and the decoder. BEiT [6] utilizes a pretrained discrete VAE (dVAE) [39, 38] as the tokenizer. PeCo [12] proposed to apply perceptual similarity loss on the training of dVAE can drive the tokenizer to generate better semantic visual tokens, which helps pretraining. In contrast to those works, our GeoMIM utilizes a geometry-rich LiDAR model and transfers its knowledge via MIM pretraining, aiming to improve the multi-view camera-based 3D models.

Multi-view camera-based 3D detection The field of camera-based 3D object detection has seen significant progress in recent years [46, 47, 29, 20, 28, 26, 36, 52]. FCOS3D [46] proposed a fully convolutional single-stage detector for monocular 3D object detection. DETR3D [47] extends the DETR framework to the 3D domain, and proposes a framework for end-to-end 3D object detection. BEVFormer [29] combines BEV (bird’s eye view) representation and transformer networks for 3D object detection. BEVDepth [28] focuses on accurately estimating the depth of objects in the BEV representation. Additionally, considering the promising performance of the LiDAR-based detectors, there are several papers that use a pretrained LiDAR detector for knowledge distillation [16]. LIGA-Stereo [13] proposes to mimic the LiDAR BEV features for training a camera-based detector. UVTR [27] represents different modalities in a unified manner and supports knowledge transfer with the voxel representations. More recent BEVDistill [8] and CMKD [17] not only use the LiDAR BEV features for knowledge distillation but also transfer the teacher’s knowledge through sparse instance distillation and response-based distillation respectively. In comparison, we utilized the pretrained LiDAR model in a pretraining-finetuning paradigm to avoid the LiDAR-camera BEV domain gap.

3. Method

Employing a pretrained LiDAR-based detection model to provide auxiliary learning guidance to train camera-based 3D understanding models has shown promising results in recent years [13, 8, 9, 27, 17]. However, because of the domain gap between the LiDAR and camera modalities, we observe that when a camera-based model is already strong, directly supervising it with the LiDAR teacher fails to improve the camera-based model as shown in Tab. 1.

To address this problem, we propose GeoMIM to better transfer the LiDAR model’s knowledge to the camera-based model in a pretrain-finetune paradigm. GeoMIM pretrains a multi-view camera-based model via Masked Image Modeling (MIM) [49]. Unlike existing 2D MAE works [14, 30], we project the semantic features to the BEV (bird’s eye view) space and use the LiDAR BEV features in the 3D space as the reconstruction targets for pretraining. The pretrained LiDAR model is only used in the pretraining stage, and is discarded in the finetuning stage to avoid introducing the LiDAR-camera BEV domain gap. We illustrate the proposed GeoMIM in Fig. 1.

Masking and Encoder Given the multi-view input images \( X = \{ x_i \in \mathbb{R}^{3 \times H \times W}, i = 1, 2, \ldots, N \} \) where \( N, H, W \) are the number of views, image height, and width, we randomly mask a proportion of input image patches (tokens) and use a Swin Transformer [33] as the encoder to encode the visible tokens. The encoded representations, \( F^e \in \mathbb{R}^{N \times C \times L} \) where \( C \) and \( L \) denote the number of di-
dimensions and the number of visible tokens, are then filled with a shared mask token \([M]\) \(\in \mathbb{R}^{C}\) at the masked locations and further processed by the decoder for reconstruction.

**GeoMIM Decoder** To transfer the rich geometry knowledge of a pretrained LiDAR detector to our camera-based model, we jointly project the multi-view semantic features according to their estimated depth maps to the BEV space and use the same scene’s LiDAR BEV features as the reconstruction targets. Specifically, our GeoMIM uses two *decoupled* decoders, each of which consists of 8 Transformer [43] blocks. The semantic decoder \(D_{\text{sem}}\) reconstructs the dense camera-view semantic features \(F^w \in \mathbb{R}^{N \times C \times \frac{H}{16} \times \frac{W}{16}}\) of the \(N\) camera views and the other geometry decoder \(D_{\text{geo}}\) predicts dense camera-view depth maps \(D \in \mathbb{R}^{N \times B \times \frac{H}{16} \times \frac{W}{16}}\) of the \(N\) camera views, where \(B\) denotes the number of depth bins. The depth map and semantic feature can be expressed as

\[
D = D_{\text{geo}}(F^w, [M]), \quad F^s = D_{\text{sem}}(F^w, [M]) .\tag{1}
\]

We can then obtain the camera BEV features \(F_{\text{BEV}}^{I}\) by jointly projecting the multi-view semantic features to the BEV space with the Lift-Splat-Shoot (LSS) [37] operation according to the predicted dense depth maps,

\[
F_{\text{BEV}}^{I} \in \mathbb{R}^{C \times N_x \times N_y} = \text{LSS}(F^s, D), \tag{2}
\]

where \(N_x, N_y\) are the numbers of bins in the \(x\) and \(y\) axis of the BEV feature maps respectively. Empirically, the two decoders share the first half of the Transformer blocks for efficiency.

Unlike existing works that separately process the multi-view input images, we propose a novel *Cross-View Attention (CVA)* block to model the interaction across different views to better reconstruct the LiDAR BEV features from input images. Our intuition is that as the multi-view images are naturally overlapped, proper interaction across views is beneficial to align those images and better infer the LiDAR BEV features. Instead of explicitly using the epipolar lines to associate pixels across the multi-view images, we partition the camera-view tokens of the multiple views into groups according to their row indices and only allow the tokens belonging to the same row of the \(\frac{1}{16}\) input resolution to interact with each other. The interaction is modeled by the self-attention operation [43]. Notably, our proposed CVA has linear computation complexity to the input image size and is therefore much more efficient compared to global self-attention. We illustrate the proposed CVA in Fig. 2. We use the CVA block as the 2th and 6th attentions blocks of the decoder. Note that we do not add it to the backbone and no extra computation is introduced when finetuning the encoder.
Accurately reconstructing depth with the geometry decoder implicitly requires the decoder to infer the camera’s intrinsic parameters, which is difficult to generalize to an unseen dataset as the data may be collected with different cameras. To achieve better transferability across different downstream tasks, we encode the camera’s intrinsic and extrinsic parameters using a linear projection layer and use the resulting features to scale the geometry decoder’s feature using the Squeeze-and-Excitation module [18]. Importantly, we do not require the camera’s information when finetuning on downstream tasks since only the decoder uses the camera’s information during pretraining. We demonstrate that the camera-aware depth reconstruction branch leads to better performance when finetuning on tasks that differ from the pretraining dataset.

**Loss**  We use the mean squared error (MSE) loss between the projected camera BEV features and the pretrained LiDAR BEV features for pretraining,

\[
L_{\text{rec}} = \| (F^{I}_{BEV} - F^{L}_{BEV}) \|^2_2,
\]

where \(F^{I}_{BEV} \in \mathbb{R}^{C \times N_x \times N_y} \) denotes the pretrained LiDAR model’s BEV features. In addition, we incorporate a depth prediction task. Following prior arts [28], we use the ground truth discrete depth \(D_{GT} \) derived from the LiDAR point cloud and calculate the binary cross entropy (BCE) loss as the depth loss,

\[
L_{\text{depth}} = \text{BCE}(D, D_{GT}).
\]

The overall loss can be expressed as

\[
L = L_{\text{rec}} + \alpha L_{\text{depth}},
\]

where \(\alpha\) balances the two loss terms, which is set as 0.01 experimentally. Empirically, we observe that the depth loss can enhance the convergence speed, which is crucial for pretraining large models.

After pretraining, we discard the decoders and add a task-specific head on the top of the encoder for downstream tasks finetuning. During finetuning, we only utilize ground-truth supervision and abstain from utilizing the LiDAR model to avoid introducing the aforementioned domain gap.

**Comparison with 2D MAE**  Compared to existing 2D MAE models [14, 49, 30], our proposed GeoMIM’s pretraining has two distinct characteristics: (1) We employ a geometry-rich LiDAR model and transfer its high-level knowledge in the BEV space via MIM pretraining, which can effectively enhance the geometry perception capability of the camera-based model. In contrast, the original MAE [14] reconstructs image pixels and could work well for 2D downstream perception tasks, but is found to be less effective for 3D perception. The reason is that the autonomous driving dataset, e.g., nuScenes [7], is much less diverse than MAE’s pretraining dataset ImageNet-1K [10]. As a result, employing image pixel reconstruction as the pretext task is hard to learn high-quality representations. (2) Contrary to MAE which only calculates the reconstruction loss in the masked tokens, we take all tokens into consideration in our loss. This is because the learning targets we use are from a different modality and in a different geometric space. We can take full advantage of the LiDAR model by using all tokens to calculate the loss. For the masked locations, the objective is a prediction task while for the unmasked locations, it is similar to a distillation task.

**4. Experiment Setups**

To demonstrate the effectiveness of GeoMIM, we conduct experiments by pretraining Swin Transformer [33] backbones with GeoMIM and then finetuning it on various downstream tasks. These tasks include multi-view camera-based 3D detection on nuScenes [7] and Open Waymo [42] datasets, camera-based 3D semantic segmentation on nuScenes dataset, and 2D detection on nuImages dataset.

**Dataset and Evaluation Metrics**  We use the large-scale nuScenes dataset for pretraining and finetuning, which contains 750, 150, and 150 scenes for training, validation, and testing, respectively. Each scene has 6 camera images and LiDAR point cloud covering 360°. Following the official evaluation metrics, we primarily report NuScenes Detection Score (NDS) and mean Average Precision (mAP) for comparison. We also report other five metrics, including ATE, ASE, AOE, AVE, and AAE, to measure translation, scale, orientation, velocity, and attribute errors, respectively, for a more detailed diagnosis.

We also evaluate the transferability of GeoMIM by finetuning the pretrained backbone on the Open Waymo and nuImages datasets. We report LET-3D-APL [23] and LET-3D-AP following the latest official guidelines for comparison. We report Mean Average Precision (mAP) of box and mask on nuImages dataset for 2D object detection and instance segmentation.

**Pretraining**  We pretrain the Swin Transformer backbones on the training split of the nuScenes dataset with multi-view images as input. By default, we pretrain for 50 epochs with an input size of 256 × 704. For ablation studies, we pretrain for 6 epochs unless otherwise specified. We use a pretrained TransFusion-L [5] LiDAR model to provide the reconstruction targets. We randomly mask the multi-view input images with a mask ratio of 50%. We use AdamW [34] optimizer with a learning rate of \(2 \times 10^{-4}\) and weight decay of 0.01. The learning rate is linearly warmed-up for 500 iterations and cosine decayed to 0. We apply the data augmentation strategy in BEVDet [20] to augment the input images and do not use augmentations for the LiDAR inputs. We utilize the Swin-Base and -Large backbones for pretraining, initializing the backbone with self-supervised ImageNet-pretraining [30].

**Finetuning**  We keep the pretrained encoder, abandon the
decoders, and adopt state-of-the-art frameworks for finetuning. We mainly evaluate the performance of the finetuned models on the 3D detection task on the nuScenes dataset. We also assess the transferability of GeoMIM on other downstream tasks.

For the 3D detection on the nuScenes dataset, we utilize the BEVDepth [28] framework with an input size of $512 \times 1408$ for comparison with other state-of-the-art approaches. For ablation studies, we use the BEVDet [20] framework with an input size of $256 \times 704$. For 3D detection on the Open Waymo dataset, the DfM [45, 44] framework is utilized. For 3D segmentation on the nuScenes dataset, we utilize the recent TPVFormer [22] for finetuning. We use Mask-RCNN [15] for object detection and instance segmentation on nuImages. We follow those frameworks’ default settings for finetuning, and include the detailed hyperparameters settings in the supplementary.

### 5. Main Results

In this section, we compare our GeoMIM to prior arts on various benchmarks. We first conduct comparisons between GeoMIM and previous pretraining approaches in Sec. 5.1. We then compare our best results with state-of-the-art results on the nuScenes 3D detection benchmark in Sec. 5.2. To show the transferability of GeoMIM, we present the transfer learning results on other 3 benchmarks in Sec. 5.3. We finally show the quantitative results in Sec. 7.

#### 5.1. Comparison with previous camera-based pretraining methods

We compare our pretraining method, GeoMIM, with previous pretraining approaches to demonstrate its effectiveness in multi-view camera-based 3D detection. Four pretraining approaches for camera-based models are utilized, including the supervised pretraining on ImageNet-1K [33], the contrastive approach EsViT [25], the multi-modal approach UniCL [50], and masked-image-modeling approach MixMAE [30]. Using the BEVDet framework with input size of $256 \times 704$, we finetune the pretrained Swin-B [33] models on nuScenes [7] and compare their performances in Tab. 4. Our approach outperforms other compared approaches in terms of all reported metrics, demonstrating the effectiveness of our pretraining method.

Particularly, our approach achieves 0.472 NDS (NuScenes Detection Score), 2.9% better than the self-supervised pretraining. Notably, our approach is much better
at predicting translation, demonstrating a 3.3% improvement in mATE, which shows that our geometry-enhanced pretraining can help more with localization. Surprisingly, while the contrastive or multi-modal approaches perform much better than the supervised ImageNet pretraining on various 2D visual tasks, they fail to improve the ImageNet-supervised pretraining on the 3D detection task.

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

Tab. 2 shows the comparison of our approach with state-of-the-art methods on nuScenes val set. Our approach achieves state-of-the-art results of 0.605 NDS and 0.523 mAP, demonstrating substantial 2.3% NDS and 4.0% mAP improvements over SOLOFusion [36]. Particularly, the most improvement of NDS comes from the mATE, which improves SOLOFusion by 2.7%. Compared to BEVDensity [28] using the same Swin-B backbone, we improve the NDS and mAP by 5.0% and 5.7% respectively.

On the test set, our single model achieves 64.4% NDS and 56.1% mAP without using extra data and test-time augmentation, which are 2.5% and 2.1% better than the previous state-of-the-art results. Notably, this model performs best among all reported metrics. Compared to BEVStereo [26], the most significant improvement of NDS comes from the mAVE (10.2%), which shows that our geometry-enhanced pretraining is not only better for localization but also improves the velocity estimation. Compared to SOLOFusion, we largely improve the mATE metric (5.3%), showing that our geometry-enhanced pretraining is beneficial for localization.

We also show that our pretraining is scalable in terms of model size. In particular, on the test set, we obtain 1.8% NDS and 1.5% mAP gains by using the larger Swin-L [33] backbone.

5.3. Transfer to various 3D understanding tasks

In this section, we evaluate the transferability of our approach to other datasets and tasks with different frameworks. We use three benchmarks, 3D segmentation on nuScenes dataset [7], 3D detection on Open Waymo dataset [42], and object detection and instance segmentation on nuImages dataset.

As shown in Tab. 3, our approach achieves superior results on all three benchmarks, demonstrating the transferability of our pretraining method. Particularly, on the 3D segmentation task, our approach achieves 68.9% mIoU on the nuScenes val set, surpassing the ImageNet-supervised pretraining [33] results by a large margin. Note that, unlike the 3D detection task, the self-supervised pretraining [30] fails to improve the supervised pretraining because the segmentation task relies more on semantic understanding. In comparison, GeoMIM improves the ImageNet-supervised pretraining for 2.5% mIoU. On the nuScenes test set, we achieve state-of-the-art results, 1.1% mIoU better than the previous best camera-based results in TPVFormer [22]. Moreover, our pretrained backbone can also transfer to datasets that differ from that used in pretraining. On Open Waymo detection benchmark, our pretraining improves the MixMAE [30] self-supervised pretrained model by 3.0%/3.0% on LET-3D APL/AP.

Apart from the 3D perception task, we show that our pretrain can also transfer to 2D object detection and instance segmentation tasks. As shown in Tab. 3 (right), GeoMIM improves the self-supervised pretraining by 1.4% APbbox and 2.6% APmask.

6. Ablation Studies

In this section, we conduct ablation studies to evaluate the impact of different design choices on the performance of our proposed GeoMIM on the multi-view camera-based 3D detection task. Unless otherwise specified, we use the Swin-B [33] backbone and pretrain it for 6 epochs. We utilize the BEVDet [20] framework for finetuning the pretrained backbone and report the performance on the nuScenes val set [7]. The gray column indicates the final choice of GeoMIM.

Pretraining epochs and pretraining data. We explore the effect of pretraining epochs and pretraining data on GeoMIM. As shown in Tab. 6, we find that we can improve the mATE

<table>
<thead>
<tr>
<th>Methods</th>
<th>Backbone</th>
<th>Image Size</th>
<th>Extra Data</th>
<th>TTA</th>
<th>mAP↑</th>
<th>NDS↑</th>
<th>mATE↑</th>
<th>mASE↑</th>
<th>mAOE↑</th>
<th>mAVE↑</th>
<th>mAAE↑</th>
</tr>
</thead>
<tbody>
<tr>
<td>FCOS3D [46]</td>
<td>R101-DCN</td>
<td>900 × 1600</td>
<td>×</td>
<td>✓</td>
<td>0.358</td>
<td>0.428</td>
<td>0.690</td>
<td>0.249</td>
<td>0.452</td>
<td>1.434</td>
<td>0.124</td>
</tr>
<tr>
<td>DETR3D [47]</td>
<td>V2-99</td>
<td>900 × 1600</td>
<td>✓</td>
<td>✓</td>
<td>0.412</td>
<td>0.479</td>
<td>0.641</td>
<td>0.255</td>
<td>0.394</td>
<td>0.845</td>
<td>0.133</td>
</tr>
<tr>
<td>UVTR [27]</td>
<td>V2-99</td>
<td>900 × 1600</td>
<td>✓</td>
<td>✓</td>
<td>0.472</td>
<td>0.551</td>
<td>0.577</td>
<td>0.253</td>
<td>0.391</td>
<td>0.508</td>
<td>0.123</td>
</tr>
<tr>
<td>BEVFFormer [29]</td>
<td>V2-99</td>
<td>900 × 1600</td>
<td>✓</td>
<td>✓</td>
<td>0.481</td>
<td>0.569</td>
<td>0.582</td>
<td>0.256</td>
<td>0.375</td>
<td>0.378</td>
<td>0.126</td>
</tr>
<tr>
<td>BEVDet4D [19]</td>
<td>Swin-B</td>
<td>900 × 1600</td>
<td>✓</td>
<td>✓</td>
<td>0.451</td>
<td>0.569</td>
<td>0.511</td>
<td>0.241</td>
<td>0.386</td>
<td>0.301</td>
<td>0.121</td>
</tr>
<tr>
<td>PolarFormer [24]</td>
<td>V2-99</td>
<td>900 × 1600</td>
<td>✓</td>
<td>✓</td>
<td>0.493</td>
<td>0.572</td>
<td>0.556</td>
<td>0.256</td>
<td>0.364</td>
<td>0.439</td>
<td>0.127</td>
</tr>
<tr>
<td>PETRV2 [32]</td>
<td>GLOM-like</td>
<td>640 × 1600</td>
<td>X</td>
<td>✓</td>
<td>0.512</td>
<td>0.592</td>
<td>0.547</td>
<td>0.242</td>
<td>0.360</td>
<td>0.367</td>
<td>0.126</td>
</tr>
<tr>
<td>BEVDepth [28]</td>
<td>ConvNeXt-B</td>
<td>640 × 1600</td>
<td>X</td>
<td>✓</td>
<td>0.520</td>
<td>0.609</td>
<td>0.445</td>
<td>0.243</td>
<td>0.352</td>
<td>0.347</td>
<td>0.127</td>
</tr>
<tr>
<td>BEVStereo [26]</td>
<td>V2-99</td>
<td>640 × 1600</td>
<td>✓</td>
<td>✓</td>
<td>0.525</td>
<td>0.610</td>
<td>0.431</td>
<td>0.246</td>
<td>0.358</td>
<td>0.357</td>
<td>0.138</td>
</tr>
<tr>
<td>SOLOFusion [36]</td>
<td>ConvNeXt-B</td>
<td>640 × 1600</td>
<td>X</td>
<td>✓</td>
<td>0.540</td>
<td>0.619</td>
<td>0.453</td>
<td>0.257</td>
<td>0.376</td>
<td>0.276</td>
<td>0.148</td>
</tr>
<tr>
<td>BEVDistill [8]</td>
<td>ConvNeXt-B</td>
<td>640 × 1600</td>
<td>X</td>
<td>✓</td>
<td>0.498</td>
<td>0.594</td>
<td>0.472</td>
<td>0.247</td>
<td>0.378</td>
<td>0.326</td>
<td>0.125</td>
</tr>
<tr>
<td>GeoMIM</td>
<td>Swin-B</td>
<td>512 × 1408</td>
<td>X</td>
<td>✓</td>
<td>0.547</td>
<td>0.626</td>
<td>0.413</td>
<td>0.241</td>
<td>0.421</td>
<td>0.272</td>
<td>0.127</td>
</tr>
<tr>
<td>GeoMIM</td>
<td>Swin-L</td>
<td>512 × 1408</td>
<td>X</td>
<td>✓</td>
<td>0.561</td>
<td>0.644</td>
<td>0.400</td>
<td>0.238</td>
<td>0.348</td>
<td>0.255</td>
<td>0.120</td>
</tr>
</tbody>
</table>

Table 5: Comparison on nuScenes test set. “Extra data” denotes depth pretraining. “TTA” denotes test-time augmentation.
We also compare the data utilization ability of different pretraining in Tab. 8. If we use self-supervised pretraining MixMAE [30], the performance gain between using 50% and 100% data is 4.2% NDS. If we use GeoMIM pretraining, the improvement increases to 5.1% NDS. The results demonstrate that our GeoMIM can be benefited more by using more data.

**Mask ratio.** We examine the effect of the mask ratio used in the masked image modeling training process on the performance. As shown in Tab. 9, we find that using a mask ratio of 0.5 performs best as a very high mask ratio causes the pretext task too hard while a low mask ratio causes the pretext task too easy.

**Pretraining with distillation or other reconstruction targets.** We compare the performance of GeoMIM with different learning targets, including RGB pixels [14] and the voxelized LiDAR points. Moreover, we use the depth ground truth derived from the LiDAR point cloud for depth pretraining [35]. Following previous works, we also the LiDAR BEV features for conducting distillation pretraining [13, 8, 17]. We initialize the backbone with MixMAE [30] self-supervised pretraining and use its results for comparison. All the pretraining experiments are conducted on the nuScenes dataset.

As shown in Tab. 10, we find the depth or distillation pretraining fails to improve the MixMAE results. Those two pretraining methods are beneficial for object localization to improve the mATE metric, but degenerate mAOE a lot. Using the RGB pixels as the reconstruction targets results in unstable pretraining. In comparison, we use a pretrained LiDAR detection model to extract more meaningful BEV features as the reconstruction targets, which can not only transfer the rich geometry information to the camera-based model but also avoid the noise problem of directly reconstructing LiDAR voxels.

**Ablation on the decoder designs.** We investigate the impact of the proposed decoder designs on the final performance. Apart from reporting results on the nuScenes dataset, we also report the performance on the Open Waymo dataset to show how the design choices affect the transferability of the pretraining.
Figure 3: Example results on nuScenes val images. From top to bottom, the rows are the camera-view image, masked camera-view image, decoded semantic features, and decoded geometry features.

Figure 4: Reconstruction results on nuScenes val images. For each triplet, we show the reconstructed BEV features (left), LiDAR BEV features, and LiDAR point cloud (right) in BEV.

Figure 5: CVA blocks’ attention maps across different cameras. Our CVA enables the cameras at back to attend to the semantic parts in front views.

As shown in Tab. 11, we find that using a decoupled decoder to separately decode the depth and semantic features can improve the NDS for 0.8%, compared to using one decoder to jointly decode them. The decoupled branches force the geometry branch to focus on inferring depth from geometry cues, which maximizes the utilization of the model capacity. Moreover, removing the Cross-View Attention (CVA) blocks results in a 0.5% performance drop because of lacking cross-view interaction for better BEV feature inference. Additionally, we find that further removing the camera-aware design leads to 0.8% LET-3D APL drop on the OpenWaymo dataset. As the pretraining and finetuning might be conducted on different datasets, using camera-aware depth reconstruction is beneficial for the transferability of the pretraining.

We also experiment to consider epipolar geometry in our proposed CVA block. We observed a slight accuracy improvement (0.463 NDS vs. 0.460 NDS). However, including epipolar geometry constraints in CVA block slowed down the training by 1.5 times due to the slower sampling along the epipolar plane on GPUs.

Ablation on backbone size. We investigate the scalability of our pretraining method and show the results in Tab. 12. Our GeoMIM is capable to be scaled up to Swin-L model with 200M parameters. The pretrained Swin-L [33] improves Swin-B on all reported metrics, especially the mAOE performance (7.8%).

7. Qualitative Evaluation

Reconstruction visualization. We show the decoded geometry features and semantic features during pretraining in Fig. 3. Although a high mask ratio is utilized, meaningful patterns can still be observed in the decoded features. In particular, we find the geometry branch focuses on the structure of the driving scenes while the semantic branch can be activated by different semantic regions, including roads, cars, trees, etc. Furthermore, we visualize the reconstructed BEV
features in Fig. 4. The reconstructed BEV features can well restore the LiDAR model’s BEV features, including the road structures and the semantic features.

Cross-view attention maps. We visualize the attention maps of the Cross-View Attention (CVA) blocks in Fig. 5. Through the cross-view interaction, one view is able to attend to the semantic parts of other views.

Convergence curve. We show the convergence curve of different pretraining in Fig. 6. Our pretraining can largely improve the self-supervised [30] and supervised [33] pretraining in terms of the convergence speed, which can match the self-supervised pretraining’s final results with only half of the iterations.

8. Conclusion

In this paper, we proposed a pretraining method, GeoMIM, for multi-view camera-based 3D detection. By leveraging the knowledge of a pretrained LiDAR model in a pretrain-finetune paradigm, GeoMIM aims to transfer its rich geometry knowledge to the camera-based model. Specifically, GeoMIM reconstructs BEV features from masked images via a novel decoder and a cross-view attention mechanism. We demonstrate that GeoMIM significantly outperforms existing state-of-the-art methods on the nuScenes dataset, achieving state-of-the-art results in both camera-based 3D detection and segmentation tasks. Moreover, we verify that the pretrained model can be transferred to the Waymo Open dataset, further showing its effectiveness and generality.

Limitations Despite the promising results, GeoMIM also has some limitations. First, GeoMIM requires a large amount of labeled data for pretraining, which may not be available in some applications. Second, GeoMIM relies on the quality of the LiDAR model’s BEV features, which may not always be accurate or complete. Overall, while GeoMIM shows great potential, further research is needed to address these limitations and improve its applicability in a wider range of applications.

Acknowledgement This project is funded in part by National Key R&D Program of China Project 2022ZD0161100, by the Centre for Perceptual and Interactive Intelligence (CPII) Ltd under the Innovation and Technology Commission (ITC)’s InnoHK, by General Research Fund of Hong Kong RGC Project 14204021. Hongsheng Li is a PI of CPII under the InnoHK.

References

[14] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross B. Girshick. Masked autoencoders are scalable vision learners. In CVPR, 2021. 2, 4, 7
[29] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In ICCV, 2021. 1, 2, 4, 5, 6, 8, 9
[34] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In ICLR, 2017. 4
[38] Jason Tyler Rolfe. Discrete variational autoencoders. In ICLR, 2016. 2


