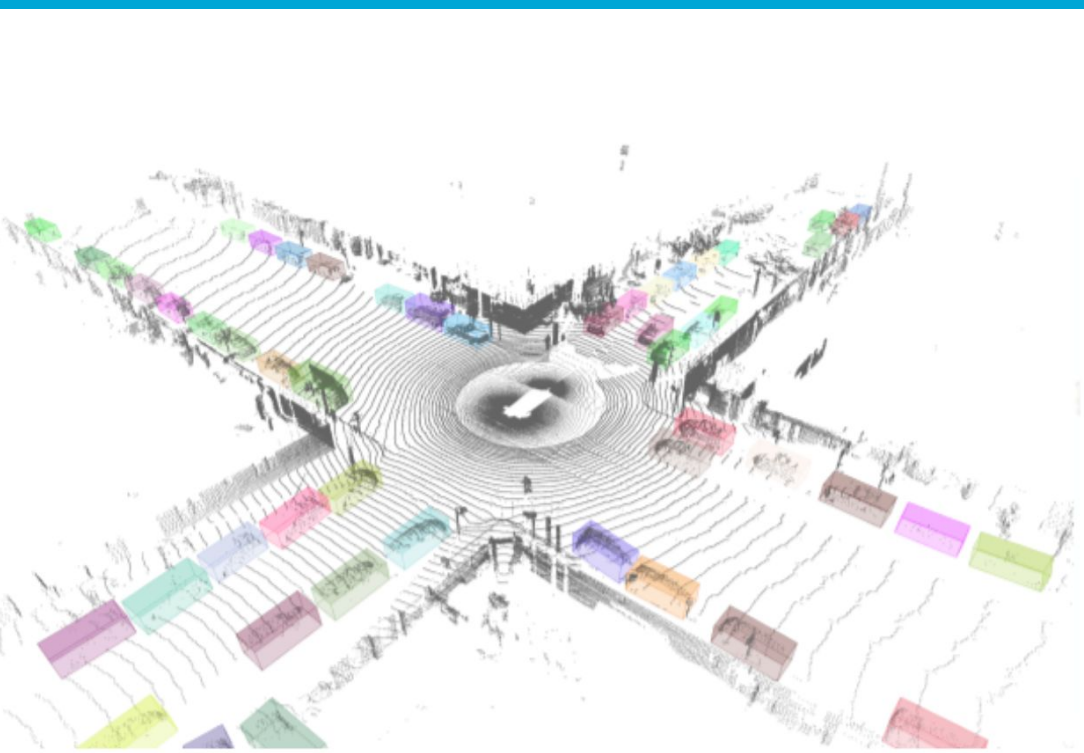
Zero-shot and Few-shot Learning for 3D Object Detection Using Language Model

Tishar Sinha



TU Delft Delft University of Technology

KOGNIC

Zero-shot and Few-shot Learning for 3D Object Detection Using Language Model

MSc Thesis

by

Tishar Sinha

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on November 30, 2023.

Student number: 5277876
Project duration: December, 2022 – November, 2023
Thesis Supervisor: Dr. Holger Caesar, TU Delft, supervisor
Andreas Lindholm, Kognic, supervisor

TU Delft

Zero-shot and few-shot learning for 3D Object Detection Using Language Model

Tishar Sinha
Delft University of Technology

Abstract

This research introduces a novel approach for 3D object detection utilizing language models, with a particular focus on addressing the challenges that have been encountered in the autonomous vehicle domain. The primary objective revolves around addressing the constraints associated with object detection models that rely heavily on labeled data and are resource-intensive, while also showing limited proficiency in recognizing new, unseen objects. The thesis presents the PointGLIP model, a novel integration of zero-shot and few-shot learning methodologies, that utilizes language models to augment object detection in 3D point cloud data. PointGLIP utilizes GLIP encoders, which are known for their capability to combine textual and visual data. This methodology facilitates the transfer of pre-trained information from a 2D domain to a 3D domain by converting point clouds into depth maps. This conversion process enables object detection without the need for extensive or any prior 3D training. The model's ability to generalize and transfer learning from 2D to 3D environments is demonstrated by experiments conducted utilizing the nuScenes dataset. The results revealed that the model shows poor performance in the context of both zero-shot and few-shot learning. This shows that the model struggles to perform detections with depth maps which indicates a struggle in the transfer of knowledge from 2D to 3D contexts. In 2D, the integration of descriptive text through language models provides a unique approach to contextual understanding, though the outcomes demonstrated that further refinements are necessary for consistent reliability.

1. Introduction

The automobile industry is currently undergoing a shift towards the use of autonomous vehicles, which have demonstrated substantial advancements in recent years. The progress in 2D and 3D computer vision problems has been propelled by the emergence of deep learning techniques. Autonomous vehicles are outfitted with sensors that enable them to perceive their environment, facilitating navigation and collision avoidance. With the ongoing transformation of the automobile industry towards autonomous driving, there is a growing demand for advanced perception tasks, including but not limited to object detection, image segmentation, and pose estimation. At present, the commercial autonomous vehicle has successfully attained level 3 of the SAE International standards, thus allowing for automated navigation under predefined circumstances. According to the Society of Automotive Engineers (SAE), the vehicle at this stage exhibits a degree of autonomous functionality, allowing for limited self-driving capabilities. At this level, the vehicle is capable of limited self-driving and will drive if the necessary conditions are met [29]. Although the present autonomous driving system mandates that individuals remain ready to assume control of the vehicle when required and maintain awareness of their surroundings, the forthcoming advancements in autonomous driving technology will progressively lessen the reliance on human intervention. Object detection plays a crucial role in increasing the autonomy of these vehicles by identifying the location of nearby objects.

Current fully supervised object detection algorithms [20, 42] perform well in detecting common categories such as pedestrians, vehicles, and traffic signs, among others. Data is critical for the training of these object detection algorithms. These models require a large amount of data to train on. The process of annotating autonomous driving data is time- and resource-intensive, especially for lidar point cloud and radar data. Additionally, there is a disparity between typical data and more informative data that reveals corner cases such as the presence of an ambulance, a fire truck, or a person resting on the side of the road. These models lean towards overfitting when the training data is less. Due to the lack of availability of ample samples of these uncommon objects while training these model still has difficulty recognizing uncommon objects. Identifying these objects will increase the detection model's robustness and the vehicle's reliability. In addition, if the model is to detect a new object, it must be trained on a large number of samples of that object, which requires a significant amount of time and resources.

In order to mitigate the issue of limited data availability for the model, research has been carried out in the domains of zero-shot and few-shot learning. Zero-shot learning is a type of learning in which a model can learn from the data without being trained explicitly. In contrast to few-shot learning, in which a model can only learn from a limited number of examples [36]. These models have undergone pre-training using a substantial amount of data, allowing them to effectively apply their acquired knowledge to other downstream tasks, including object detection and image classification. In recent years, several language models have emerged as important tools for zero-shot learning, leveraging their semantic understanding and transfer learning capabilities to discern and process visual information in scenarios with scarce or no labeled data. Availability of data, self-supervised learning, development of transformer model architecture, and improvement of computer hardware have led to the rise of large language models (LLM) in the last few years, like with Dall-e, ChatGPT, CLIP [30], BERT [9], Bard, GPT-3 [3], etc. In computer vision, there are several tasks utilizing language models, such as classification [30], object tracking [27], image captioning [28], text-to-image generation [31,33], and image retrieval [1,17]. By combining text comprehension with visual perception, language models can provide a more comprehensive analysis of scenes and objects. Language models can aid in object detection by providing a contextual understanding of objects through textual description comprehension. This textual comprehension complements the language model by identifying semantic connections that can aid the object detection process, especially new objects. Language models can facilitate zero-shot or few-shot learning, whereas traditional object detection models require lots of labeled data. The language model can generate textual descriptors for classes of objects that have not been explicitly trained in the vision model, enabling it to generalize better. The use of language models for computer vision has risen in the last couple of years because of the massive availability of image and caption pairs. Due to the unavailability of textual descriptions of 3D data, however, language models for 3D perception remain underdeveloped. In addition, research has been conducted in the field of 3D image classification, but it has been limited to classification [5, 18]. PointGLIP, an object detection model based on the GLIP language model, is proposed to address the issue of the lack of 3D data availability and the identification of new, unseen objects.

We propose PointGLIP which transfers 2D pre-trained knowledge of visual and textual encoders to a 3D understanding of point clouds. Similar to other prevalent language models in computer vision [5, 18], this model allows for zero-shot and few-shot learning, addressing the lack of 3D data for uncommon objects. To solve the problem of identifying new class categories, the model can be trained on both text and image data, and texts contain a broader set of visual concepts than any predefined concept in images. This facilitates the model's applicability to new visual concepts and domains. The model converts the point cloud into 2D depth maps to resolve the lack of textual description data for 3D point clouds in language models.

In summary, our contributions are as follows:

- We propose a 3D object detector incorporating a language model for zero-shot and few-shot applications. This model extends GLIP's capabilities, bringing contextual and semantic insights to point cloud analysis.
- The model also showcases how the use of text description can help identify objects and can also help in detecting new unseen objects.
- The experimental results conducted on the widely used nuScenes dataset indicate that the utilization of raw depth maps for transferring pre-trained 2D knowledge into 3D on GLIP encoders is insufficient for achieving 3D object detection.

2. Related works

2.1. Zero-shot and few-shot learning

The paradigms of few-shot learning and zero-shot learning aim to tackle the difficulty of learning when there is a shortage of labeled data. In the few-shot learning paradigm, the model undergoes training on a diverse range of tasks to obtain a suitable initialization. The process of initialization can then be refined on a limited dataset for a specific purpose to produce precise predictions. In contrast, the zero-shot approach aimed to leverage semantic connections between familiar and unfamiliar categories, frequently utilizing embeddings or qualities to recognize novel categories. Zero-shot learning has been the subject of research in the context of 2D classification tasks, as evidenced by studies conducted by researchers [25,30,40]. Similarly, the application of zero-shot learning to 3D classification tasks has also been explored, as indicated by investigations conducted by researchers [6–8, 43]. However, limited progress has been made in the domain of object detection, particularly in the realm of three-dimensional (3D) object detection. The reason for this is that object detection also entails a challenging localization task. The existing approaches for few-shot learning can be broadly classified into two categories: meta-learning methods and transfer-learning methods. Meta-learning methods employ a learning-to-learn mechanism, wherein the model is trained on many few-shot tasks derived from a base dataset. This approach facilitates rapid adaptation of the model to real few-shot tasks. In addition, transfer-learning techniques involve the first pre-training of

the detector on the base dataset, followed by fine-tuning on the novel dataset.

2.2. Transfer Learning

Object detection models can acquire general features from a large dataset by utilizing a network that has been pre-trained on the large dataset. With less data, these detectors can subsequently be fine-tuned for the particular object detection task. As an example, a CNN trained on ImageNet [11], an extensive collection comprising millions of images across numerous categories will be equipped with filters capable of distinguishing a wide array of objects, textures, and shapes. Subsequently, these can be tailored by fine-tuning for a specific downstream task, such as object detection or segmentation. The increasing amount of textual data accessible via the internet has significantly expanded the utility of language models in the domain of transfer learning for perception tasks. By combining language and vision, these method facilitates the interpretation and analysis of 2D and 3D scenes, enabling a deeper understanding of objects and their relationships in complex environments.

2.3. Language models in vision

Language models like ALIGN [23] and CLIP [30] model which is pre-trained on text-image pair show good performance in downstream tasks like zero-shot image classification. The CLIP [30] consists of a text and image encoder. Both encoders generate a joint embedding which is then used to calculate the similarity between images and text using contrastive learning. Similarly, CO-OP [47] further automates the textual prompt in the CLIP [30] model to improve CLIP performance. CLIP-Adapter [14] further proposes fine-tuning with feature adapter on either the visual or language branch of the CLIP [30] model to improve its performance. Other works that include CLIP are MaskCLIP [13] for semantic segmentation, DenseCLIP [32] for image pixel prediction, DetCLIP [41], Tip-Adapter [44], etc. For a task like object detection, a more object-level representation is required which the CLIP model lacks. For object detection, GLIP [27] proposes a model that combines object detection with phrase grounding by redefining detection as grounding. This allows it to learn from both data and find objects in pictures and connect them with the right words. Phrase grounding is the process of linking natural language phrases with actual world items or regions. It includes comprehending the meaning of a natural language phrase and recognizing the objects or regions of the world to which it refers. In a sentence such as "the cat is sitting on the mat," phrase grounding would require recognizing the cat and the mat as physical things, as well as the location of the mat where the cat is sitting. To convert an object detection model into a grounding model, they replace the part of the program that identifies objects with a new part that matches language phrases to regions in the image. They do this by calculating a score that measures how well the language in the phrase matches the visual features of the region in the image. Then they use this score to associate the phrase with the region in the image.

All these language models conduct knowledge transfer within the same modality i.e. images. For lidar modality LidarCLIP [18] proposes a technique for linking text to lidar data via CLIP embedding. The model supervises the lidar encoder using a frozen CLIP encoder. The objective is to teach the lidar encoder to resemble the CLIP embedding. CLIP's extensive and diverse semantic understanding is transferred to the lidar encoder using both picture and lidar point cloud. Mean squared error is used to maximize the similarity between the two embeddings. PointCLIP [45] circumvents the requirement for a lidar encoder and instead encodes 3D point clouds into multi-view depth maps, aligning them with 3D category texts using CLIP, a vision-language model, for efficient, zero-shot classification.

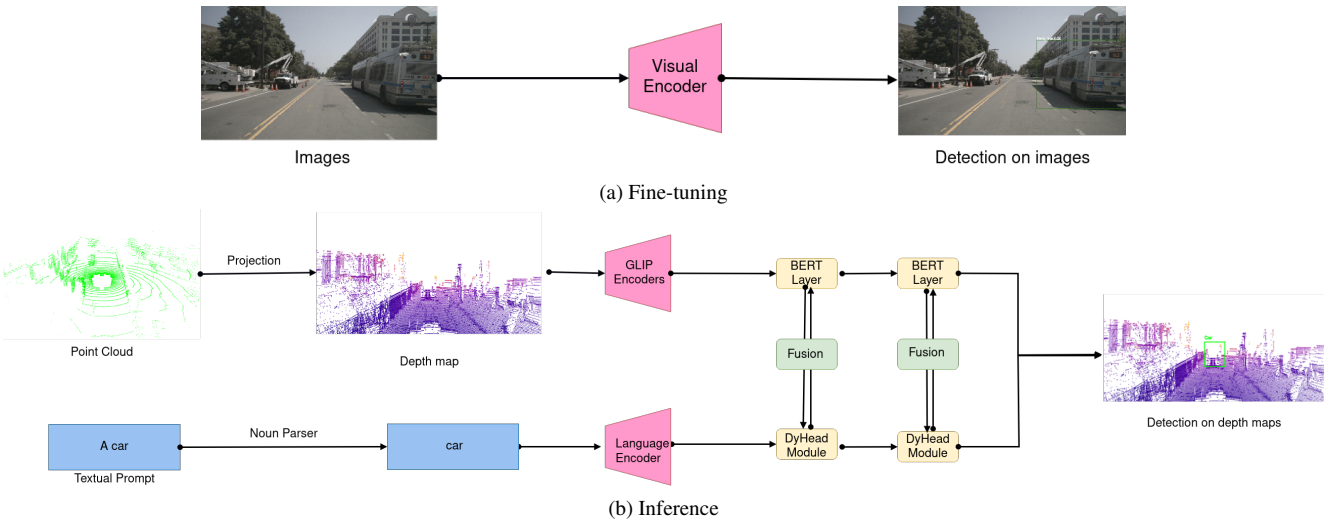
3. Method

Our framework aims to address the two shortcomings that are present in supervised 3D object detection models. 1) The training process for these models requires a substantial quantity of annotated data, thereby taking considerable time and resources. 2) The generalizability of the model is inadequate, posing a challenge in the detection of an unseen object. In this section, we introduce the proposed PointGLIP. In Section 3.1, we first revisit PointCLIP as we adopted a similar approach to PointCLIP to transfer 2D pre-trained knowledge to into 3D. In Section 3.2 we will discuss the changes made to adapt the PointCLIP model.

3.1. PointCLIP

PointCLIP [45] is a 3D classification model that aims to extend the capabilities of CLIP (Contrastive Vision-Language Pre-training) from 2D visual recognition to 3D point cloud understanding by using the CLIP's visual and textual encoders. The visual encoder is ResNet-50 [16] and the textual encoder is transformer [38]. Initially, it simply projects all the points from the point cloud onto a pre-defined image plane to generate multi-view depth maps. They adopted a perspective projection [15] without any post-rendering [35]. These depth maps do not contain any color information and are from raw points which results in low time and computation cost.

Each view generated from the projection is then processed independently through a CLIP pre-trained visual encoder to obtain view-wise features. For text, they used a template for each category as a "point cloud depth map of a [CLASS]". The textual descriptions of 3D categories are encoded using CLIP's textual encoder to create a zero-shot



(a) Fine-tuning

(b) Inference

Figure 1. The pipeline of the proposed model. The visual encoder is fine-tuned on nuScenes images in the upper part of the image. Then for the point cloud, this fine-tuned encoder is used on depth maps generated by the projection of the point cloud on 6 camera plane. Simultaneously, the textual description of the image is passed through a noun parser and then the nouns are passed on to the language encoder. The output is the object detected in depth maps based on the text provided in the input. This whole process is zero-shot as the model is not trained or fine-tuned on point clouds or depth maps.

classifier. The final prediction for the point cloud is obtained by aggregating the predictions across different views in a weighted manner, acknowledging that different views contribute differently to the recognition of the entire scene. The correlation between vision and language representations is used for zero-shot classification, i.e., recognizing categories that were not seen during training. The evaluation is carried out through zero-shot classification, where the model predicts 3D categories based on the alignment between CLIP-encoded point cloud and 3D category texts without having seen examples of these categories during training. The model is suited for denser point clouds like in indoor applications that's why they have tested it on ModelNet10 [39], ModelNet40 [39], and ScanObjectNN [37].

3.2. Proposed Model

Due to the fact that the PointCLIP model is well-suited for classification tasks and dense point clouds, we modified it to accommodate object detection tasks and autonomous vehicle dataset. Implementing a different dataset and modifying the model to function with a larger, sparser dataset are among these modifications. Then, to convert the model from a classification model into an object detection model, the visual and textual encoders were modified. Finally, the perspective projection was applied to the new dataset and its ground truth. These modifications will be discussed in this section.

3.2.1 Visual and textual encoders

To bridge the modal gap between 2D images and 3D point clouds so that we can use the pre-trained 2D model in the 3D object detection model, we adopted both the visual and textual encoders from GLIP [27] to replace the CLIP's encoders as they were suited for the classification task. The visual encoder used is DyHead [10] and BERT [12] is the textual encoder. The GLIP encoders [27] are employed in this research as they integrate object detection with phrase grounding, enabling the model to undergo pre-training using both image and text data. The integration of a text or language model within the GLIP framework serves as a repository of knowledge that provides contextual and semantic comprehension. When it comes to identifying new objects inside an image, the language model shows the ability to interpret textual labels or descriptions that correspond to the visual elements present. The GLIP model is capable of concluding the characteristics of unfamiliar things by establishing correlations with associated entities that it has acquired knowledge from textual input. Through this process, GLIP exhibits the ability to extrapolate beyond the constraints of its training data and effectively identify and comprehend items, much like the method by which humans recognize intangible objects they have encountered just through textual information.

For a good trade-off between computation time and performance, we chose the GLIP-T encoder, which is based on the Swin-Tiny backbone for the visual encoder, the BERT model for the textual encoder, and was pre-trained on the

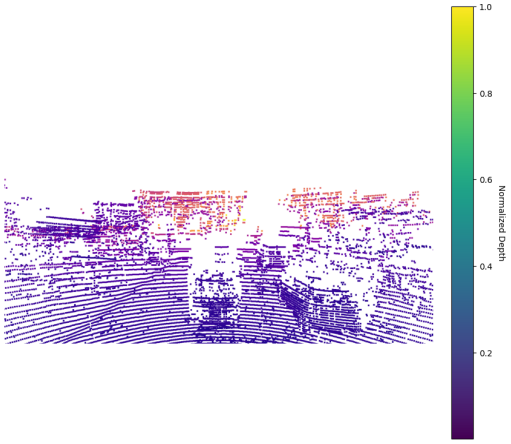


Figure 2. A normalized depth map of a point cloud

following data: 1) O365 [34], 2) GoldG, 0.8M human-annotated gold grounding data curated by MDETR [24], including Flickr30K, VG Caption [26], and GQA [22], and 3) Cap4M, 4M image-text pairs collected from the web with boxes generated by GLIP-T [27]. We tried few-shot learning with nuScenes images dataset. However, due to the low average precision score in the few-shot setting, we adopted fine-tuning the encoder with the images by varying the number of images used for fine-tuning. Furthermore, in order to adapt the GLIP to work on nuScenes dataset, all the annotations are converted into MS COCO annotation format, which is required in GLIP. For the language part in GLIP detection, it takes a textual prompt as input along with the image. In the text, the nouns are extracted by passing through the NLP parser [2] which identifies all the nouns and then performs the grounding task along with the vision encoder. Given object classes [construction_vehicle, barrier, bus,..., animal], for detecting all classes in an image, we use the following prompt as the input, where we put the names of all the classes joined by ", ".

Prompt= "construction_vehicle, barrier, bus, ... , animal"

In this each class name is a candidate phrase to be grounded.

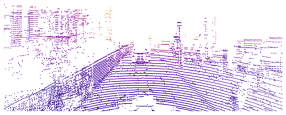
3.2.2 Point cloud to 2D depth map conversion

In order to transform a point cloud into representations that can be accessed by the GLIP framework, the generation of 2D depth maps from the point cloud are performed. The initial step involves the selection of a projection method. Two often used methods for representing objects in two-dimensional space are perspective projection and orthographic projection. The choice between these methods depends on the specific application and the desired outcomes. In the context of perspective projection, it may be observed that points located at greater distances from the camera viewpoint will tend to seem closer together in the two-dimensional array, simulating the perceptual experience of human vision. On the other hand, orthographic projection is characterized by its ability to preserve the relative distances between points, regardless of their depth. We chose a perspective projection method, inspired by PointCLIP [45], due to its lower computational requirements and ability to replicate human visual perception. This approach generates realistic images while preserving the depth information. The depth maps are derived from six distinct viewpoints in an attempt to replicate the camera perspectives seen in the nuScenes dataset, which the model has been fine-tuned on. The views are precisely characterized as a composite of Euler angles and translation vectors, representing the orientation and position, respectively, for each viewpoint. These parameters are derived from the camera extrinsic parameters of the nuScenes dataset. This defines the coordinate system and a virtual camera viewpoint, or "viewing frustum," from which the point cloud will be observed. The method initially calculates the 2D projection coordinates on the image plane for each point, using the 3D coordinates provided in the camera's coordinate system. Subsequently, a grid-based approach is utilized to allocate the depth value of every individual point across the pixels of the image. The basic approach involves defining a region surrounding every projected point using the parameters size_x and size_y. The depth value is distributed across the pixels inside this region, and this distribution is affected by the specific sub-pixel position of the point. The given depth value of a pixel is directly proportional to its proximity to the exact projected point, as the allocation is weighted. By modifying the weights, a smoother transition between depth values is created, replicating the natural gradients seen in real-world images. In order to account for scenarios when points extend outside the limits of the image, this approach incorporates a masking technique. By using this approach, any points that fall beyond the valid range of the image are deleted, guaranteeing that only relevant pixels are modified. After the distribution of the depth values, the method follows to normalize the results. This normalization process takes into consideration scenarios where numerous depth values may have been assigned to the same pixel. The mean depth for each pixel is calculated using a procedure that involves adding the weights and the weighted depth values, followed by an element-wise division. This process ensures the completeness of the depth map. It is essential to recognize that in the process of conversion, there is a potential for the loss or distortion of information about the original spatial connections between points. The extent of this loss or distortion is contingent upon the specific projection and discretization methods utilized. In a manner similar to the point cloud, the ground truth values of bounding boxes are similarly transformed into two-dimensional points. After-

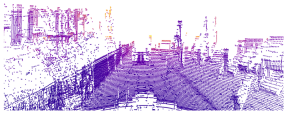
ward, any spots that are subject to occlusion or lie beyond the range of view of the camera, as determined by the six pairs of Euler angles and translation, are eliminated. As a consequence, the number of bounding box parameters is reduced from the original 11 to only 5. The parameters consist of the minimum values for x and y, the maximum values for x and y, and the class ID of the object.

3.2.3 Other Changes

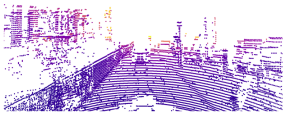
(a) nuScenes back camera image



(b) Depth maps with 4 as size x and size y along with 3 interpolated points.



(c) Depth maps with 4 as size x and size y along with 5 interpolated points.



(d) Depth maps with 6 as size x and size y along with 3 interpolated points.

Figure 3. Depth map comparison with image.

The point cloud contains variations in the number of points inside point clouds from different samples, a maximum limit of 50,000 points per sample was established. Additionally, ghost-bounding boxes were added to represent the ground truth. This approach was implemented because of the disparity in the number of bounding boxes in each sample. By incorporating ghost boxes with a value of 0 for all bounding box parameters, we were able to attain a more uniform distribution of bounding boxes throughout each sample. The ground truth consists of 11 parameters that define a bounding box. These parameters include the x, y, and z coordinates of the bounding box's center, as well as the length and width of the box. Additionally, the rotation quaternion of the bounding box and the class ID of the object are included. These ghost-bounding boxes were removed in the evaluation process. After the point cloud to depth maps conversion process, it was seen that the depth maps exhibited sparsity. To address this issue, cubic interpolation was used to introduce more points in the depth maps. Moreover, the size_x and size_y parameters were increased from 1 to 4 to improve the density of depth maps. Figure 3 shows a comparison of depth maps with an image. Figure 3a is a camera RGB image from nuScenes dataset. Figures 3b, 3c, 3d are depth maps with different size_x and size_y values and different numbers of added points by cubic interpolation.

To summarize the model Figure 1 showcases the model architecture. Initially, the visual encoder is fine-tuned on image data. This fine-tuning made the encoder detect objects in images with image and textual descriptions as input. Now for 3D object detection, the point cloud is transformed into 6 different 2D depth maps, which are then fed to the encoder. Similarly, the caption data is first repeated for the 6 depth maps and then fed to the textual encoder. This depth map and caption pair then output the detection in the depth maps. The encoder is not trained on these depth maps and the 3D data, which makes the model run in a zero-shot manner.

4. Experiment

4.1. Dataset and Evaluation Matrix

Dataset. nuScenes [4] a big, publicly accessible dataset for autonomous driving recorded in Boston and Singapore is chosen because it provides detection of 23 objects, which covers not just conventional classes such as automobiles, trucks, pedestrians, etc., but also includes rare classes such as police officers, traffic cones, child pedestrians, and police vehicles, among others. It contains one thousand unique driving sequences from six cameras, a 32-beam lidar, five radars, GPS, and an inertial measurement unit.

Evaluation Metrics. The algorithm computes the Intersection over Union (IoU) metric to evaluate the overlap between the identified object and the ground truth objects. The objects are classified as true positives, false positives, or false negatives based on the Intersection over Union (IoU) metric. Subsequently, the detections are organized in ascending order based on their respective ratings. After performing the calculation of precision and recall curves, the next step involves the interpolation of the precision curve. Ultimately, the algorithm calculates the Average Precision (AP) for each class and then determines the Mean Average Precision (mAP) by aggregating the AP values across all classes.

4.2. Results

4.2.1 GLIP Results

The GLIP model is initially trained on the 3,5,10 shot configuration for the visual encoder. Then instead of a few-shot, the model was fine-tuned using 1%, 2%, and 5% images from the nuScenes dataset. The outcomes of fine-tuning and few-shot are shown in table 1. The fine-tuning was performed for two distinct reasons. One reason was the model average precision score being so low between 1% and 5% when few-shot was utilized. The other reason is that the few-shot method was not completely few-shot, as one image contained multiple objects of different

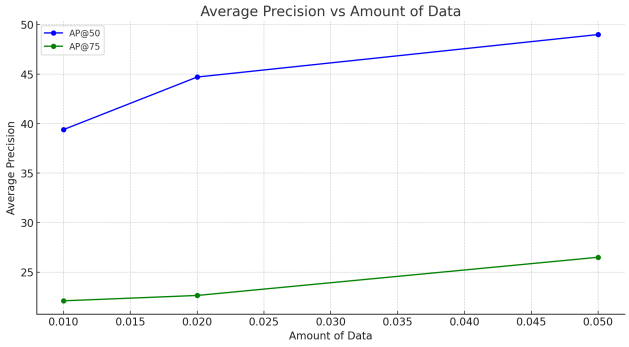


Figure 4. Graph to show amount of data used to fine-tune GLIP vs average precision.

GLIP Model	Mode	AP@50	AP@75
GLIP-T	1-shot	1.67	0.32
GLIP-T	3-shot	3.42	1.23
GLIP-T	10-shot	4.62	1.62
GLIP-T	Fine-tune 1%	39.4	22.1
GLIP-T	Fine-tune 2%	44.7	22.6
GLIP-T	Fine-tune 5%	49	26.5

Table 1. Table to show both the few-shot and fine-tune performance of GLIP model.

classes, resulting in more images for certain classes (e.g., cars, trucks, etc.) for the same number of images used for the other classes. The dimensions of the image remain unchanged from nuScenes camera images at 1600×900. To perform validation, one thousand images were utilized. The outcomes of the fine-tuning process are illustrated in the figure 4. The graph showcases the remarkable generalization and transfer learning capabilities of the GLIP model as it reaches 49 AP with only 5% of nuScenes images, which was not previously trained on autonomous vehicle datasets such as nuScenes. This score is still lower as compared to fully supervised 3D object detection methods such as CIA-SSD [46], and MDRNet [19] which achieve above 85% average precision. The advantage of using GLIP comes in identifying an uncommon object category or an object that the model has not seen as it requires less data to fine-tune and also the model can be pre-trained using textual data.

4.2.2 Zeroshot Object Detection

Setting We utilized the visual encoder which was fine-tuned on 5% of the nuScenes data for zero-shot object detection, as it exhibited an average precision score of 49%. BERT [12] is employed as the textual encoder and DyHead [10] is utilized as the visual encoder. Since zero-shot does not necessitate training data, the complete test set was utilized for evaluation with a batch size of 100 point cloud. We evaluated a 20% portion of the nuscenes point cloud. These point clouds are first projected onto 6 camera perspectives in order to emulate the camera views found in the nuScenes dataset. The input for the textual prompt consists of the names of all classes that are joined by ", ". By applying cubic interpolation between every two points, the point cloud is densified by 5 points. Additionally, the pixel size for each point in the depth is increased to 4 along the x and y axes to enhance the density of the depth map. Both the intersection over union threshold and the confidence threshold of GLIP visual encoders are maintained at 0.5 during evaluation.

Performance In the zero-shot configuration, the model exhibited poor performance, failing to accurately identify an individual object. 259 were false positives and 461883 were false negatives. As a consequence, the mean average precision and average precision class-wise are both equal to 0.0. This demonstrates that the model lacks the ability to transfer knowledge effectively from 2D to 3D.

4.2.3 Few-shot Object Detection

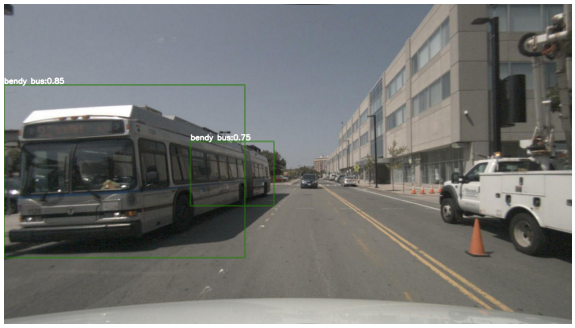
Setting We utilized the same encoders as before for the few-shot configuration; however, these encoders were trained using depth maps derived from the point cloud. The depth maps utilized in this experiment are identical to those produced in the zero-shot experiment. 2% of the total number of images in the nuScenes dataset were utilized to generate these depth maps. The model structure and all other parameter values remained unchanged from the zero-shot object detection experiment with a batch size of 10. Further to utilize the depth maps in GLIP training the ground truth was also transformed for the depth maps and then converted into COCO annotation format. For depth maps also we used 1000 images for validation similar to images fine-tuning of GLIP.

Performance In a few-shot setting, the GLIP model stopped because of early stopping after no improvement was observed after 8 epochs. The model shows 6.99% AP@50 and 4.89% AP@75 before early stopping.

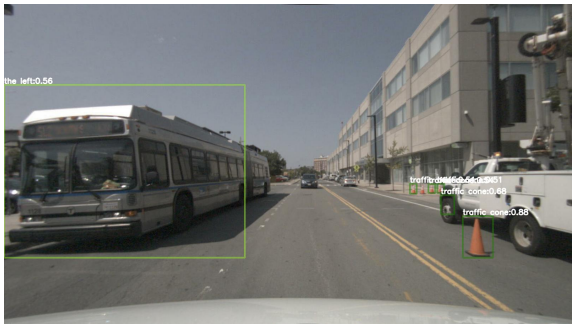
4.2.4 Object detection with more descriptive text

This experiment shows the use of language in object detection tasks. The language model adds contextual understanding of objects which enables the model to detect objects that have not been fully trained on in the vision model. Figure 5 shows some output of the GLIP model when given different prompt as input along with the image. For figure 5a and figure 5b the prompt given was classes from nuScenes dataset and the output was correct. In figure 5c the output was wrong as the model detected left in the image. The GLIP model contains the capability to comprehend and incorporate detailed spatial and relational language cues, such as directional terms like "left," "right," "top," and others,

(a) Prompt contain all classes of nuscenes dataset



(b) Prompt was "Bendy bus"



(c) Prompt was "Traffic cones on the left of the vehicle"

Figure 5. GLIP image output with different prompts.

into its visual recognition mechanism. The model's ability to perceive and comprehend spatial relationships enables it to accurately determine the location and identity of objects inside an image, utilizing their respective positions and orientations, as explained in the provided text. The mentioned descriptions are transformed into feature representations, which are subsequently combined with visual features. This procedure guarantees that the model possesses the ability to not only identify items but also comprehend their spatial relationships as defined in the text. In this particular instance, the detection yielded inaccurate results due to the model's lack of training on textual descriptions containing directional phrases in the nuScenes dataset. In this experiment, we used the same visual encoder which was fine-tuned on 5% of the nuScenes dataset, and the same language encoder as other experiments.

5. Discussion

In the zero-shot setting, the reason the above model fails to detect any object shows that the model generalization and transfer learning capabilities of the GLIP model are not powerful enough to detect objects in depth maps when it is pre-trained and fine-tuned on images. To resolve this we further tried to fine-tune the model with a depth map we call it a few-shot as now the model is fine-tuned on depth maps. This didn´t improve the results which indicate that directly using raw depth maps in GLIP is not the right direction to incorporate language models in 3D object detection. The model is limited by GLIP capability. GLIP performed well around 50% when we used it on images with only 5% data. One way is to improve the depth map quality by not just projecting the points onto the image plane but by using a deep learning model to learn how to create depth maps from point clouds for a particular image plane. Improving the depth map could improve the detection. The language model in the vision domain like in PointCLIP works well with the depth maps because it was a classification task. Object detection tasks involve object-level representation which the PointCLIP [45] model lacks. Another reason is that in classification tasks there was only one object per point cloud but in the case of our approach we chose nuScenes dataset which has multiple objects in a point cloud and occlusion is also present in these point clouds which makes it harder for the encoder to detect objects. Moreover, models, like PointCLIP was tested on synthetic datasets like ModelNet10 citewu20153d, ModelNet40 [39], and ScanObjectNN [37] which were dense. On the other hand nuscenes [4] is a sparse dataset.

6. Future Work

In improving the current model, there are several promising avenues for future research. Firstly, as discussed above improving the quality of the depth maps presents a significant opportunity for enhancement. Another potential direction involves integrating a lidar encoder to process lidar point clouds effectively. This approach would be complemented by employing the GLIP image encoder on images, facilitating a supervisory mechanism for the lidar encoder specifically for 3D object detection. Such an architecture has already been explored with promising results in the context of classification tasks, as evidenced by the LidarCLIP [18] study. Moreover, the introduction of an adapter to the model represents a further area for exploration. This adapter, once incorporated, could potentially elevate the model's performance. This adapter can be fine-tuned with a few images and point clouds while freezing the other modules. A similar approach is presented by CLIP2Point [21].

7. Conclusion

In conclusion, this project delved into the integration of language models with 3D object detection. Our research introduced an approach with our zero-shot and few-shot object detection model, blending textual inputs with point cloud data to enhance object detection capabilities. This method addresses two critical challenges in fully supervised object detection models: the extensive need for labeled data, which is both time-consuming and resource-intensive, and the models' limited generalization ability to recognize new objects. By implementing zero-shot and few-shot learning techniques, our model tries to demonstrate the ability to detect objects with minimal or no prior 3D training, which is good for the generalization ability of the object detection model and can be used in identifying new objects. Additionally, the incorporation of language models plays a pivotal role in this advancement. Their proven excellence in zero-shot and few-shot performance across various vision tasks, such as classification and object detection, brings an essential contextual and semantic understanding to the framework. Due to the scarcity of textual description data for point clouds, we used a 2D model called GLIP and used transfer learning to transfer the knowledge of the 2D encoder into the point cloud. To bridge this gap, we employed GLIP, a 2D language model adept at merging textual and visual data. By leveraging GLIP's capability to transfer 2D pre-trained knowledge into a 3D context through the conversion of point clouds into depth maps. For few-shot, we trained the GLIP visual encoder with the depth maps generated by the projection of the point cloud onto 6 cameras view. The integration of language models, despite offering intriguing contextual insights, requires further refinement for consistent application in real-world scenarios. The study underscores the necessity for continued research and development in this domain, especially concerning the quality of depth map generation and the potential integration of a lidar encoder. Future advancements in these areas could significantly enhance the model's performance, paving the way for more robust and efficient autonomous vehicle technologies.

References

[1] Alberto Baldrati, Marco Bertini, Tiberio Uricchio, and Alberto Del Bimbo. Effective conditioned and composed image retrieval combining clip-based features. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21466–21474, 2022. 2

[2] Steven Bird, Ewan Klein, and Edward Loper. *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.", 2009. 5

[3] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. *CoRR*, abs/2005.14165, 2020. 2

[4] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631, 2020. 6, 8

[5] Runnan Chen, Youquan Liu, Lingdong Kong, Xinge Zhu, Yuexin Ma, Yikang Li, Yuenan Hou, Yu Qiao, and Wenping Wang. Clip2scene: Towards label-efficient 3d scene understanding by clip. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7020–7030, 2023. 2

[6] Ali Cheraghian, Shafin Rahman, Dylan Campbell, and Lars Petersson. Mitigating the hubness problem for zero-shot learning of 3d objects. *arXiv preprint arXiv:1907.06371*, 2019. 2

[7] Ali Cheraghian, Shafin Rahman, Townim F Chowdhury, Dylan Campbell, and Lars Petersson. Zero-shot learning on 3d point cloud objects and beyond. *International Journal of Computer Vision*, 130(10):2364–2384, 2022. 2

[8] Ali Cheraghian, Shafin Rahman, and Lars Petersson. Zero-shot learning of 3d point cloud objects. In *2019 16th International Conference on Machine Vision Applications (MVA)*, pages 1–6. IEEE, 2019. 2

[9] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 764–773, 2017. 2

[10] Xiyang Dai, Yinpeng Chen, Bin Xiao, Dongdong Chen, Mengchen Liu, Lu Yuan, and Lei Zhang. Dynamic head: Unifying object detection heads with attentions. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7373–7382, 2021. 4, 7

[11] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 3

[12] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 4, 7

[13] Zheng Ding, Jieke Wang, and Zhuowen Tu. Open-vocabulary universal image segmentation with maskclip, 2023. 3

[14] Peng Gao, Shijie Geng, Renrui Zhang, Teli Ma, Rongyao Fang, Yongfeng Zhang, Hongsheng Li, and Yu Qiao. Clip-adapter: Better vision-language models with feature adapters. *International Journal of Computer Vision*, pages 1–15, 2023. 3

[15] Ankit Goyal, Hei Law, Bowei Liu, Alejandro Newell, and Jia Deng. Revisiting point cloud shape classification with a

simple and effective baseline. In *International Conference on Machine Learning*, pages 3809–3820. PMLR, 2021. 3

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 3

[17] Mariya Hendriksen, Maurits Bleeker, Svitlana Vakulenko, Nanne van Noord, Ernst Kuiper, and Maarten de Rijke. Extending clip for category-to-image retrieval in e-commerce. In *European Conference on Information Retrieval*, pages 289–303. Springer, 2022. 2

[18] Georg Hess, Adam Tonderski, Christoffer Petersson, Lennart Svensson, and Kalle Åström. Lidarclip or: How i learned to talk to point clouds. *arXiv preprint arXiv:2212.06858*, 2022. 2, 3, 8

[19] Dihe Huang, Ying Chen, Yikang Ding, Jinli Liao, Jianlin Liu, Kai Wu, Qiang Nie, Yong Liu, Chengjie Wang, and Zhiheng Li. Rethinking dimensionality reduction in grid-based 3d object detection. *arXiv preprint arXiv:2209.09464*, 2022. 7

[20] Dihe Huang, Ying Chen, Yikang Ding, Jinli Liao, Jianlin Liu, Kai Wu, Qiang Nie, Yong Liu, Chengjie Wang, and Zhiheng Li. Rethinking dimensionality reduction in grid-based 3d object detection, 2023. 1

[21] Tianyu Huang, Bowen Dong, Yunhan Yang, Xiaoshui Huang, Rynson WH Lau, Wanli Ouyang, and Wangmeng Zuo. Clip2point: Transfer clip to point cloud classification with image-depth pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22157–22167, 2023. 8

[22] Drew A Hudson and Christopher D Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6700–6709, 2019. 5

[23] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 4904–4916. PMLR, 18–24 Jul 2021. 3

[24] Aishwarya Kamath, Mannat Singh, Yann LeCun, Gabriel Synnaeve, Ishan Misra, and Nicolas Carion. Mdetr-modulated detection for end-to-end multi-modal understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1780–1790, 2021. 5

[25] Nour Karessli, Zeynep Akata, Bernt Schiele, and Andreas Bulling. Gaze embeddings for zero-shot image classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4525–4534, 2017. 2

[26] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International journal of computer vision*, 123:32–73, 2017. 5

[27] Liunian Harold Li, Pengchuan Zhang, Haotian Zhang, Jianwei Yang, Chunyuan Li, Yiwu Zhong, Lijuan Wang, Lu Yuan, Lei Zhang, Jenq-Neng Hwang, Kai-Wei Chang, and Jianfeng Gao. Grounded language-image pre-training, 2022. 2, 3, 4, 5

[28] Ron Mokady, Amir Hertz, and Amit H Bermano. Clipcap: Clip prefix for image captioning. *arXiv preprint arXiv:2111.09734*, 2021. 2

[29] Society of Automobile Engineers. Sae levels of driving automation™ refined for clarity and international audience, 2021. 1

[30] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 2, 3

[31] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 1(2):3, 2022. 2

[32] Yongming Rao, Wenliang Zhao, Guangyi Chen, Yansong Tang, Zheng Zhu, Guan Huang, Jie Zhou, and Jiwen Lu. Denseclip: Language-guided dense prediction with context-aware prompting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18082–18091, 2022. 3

[33] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 2

[34] Shuai Shao, Zeming Li, Tianyuan Zhang, Chao Peng, Gang Yu, Xiangyu Zhang, Jing Li, and Jian Sun. Objects365: A large-scale, high-quality dataset for object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8430–8439, 2019. 5

[35] Hang Su, Subhransu Maji, Evangelos Kalogerakis, and Erik Learned-Miller. Multi-view convolutional neural networks for 3d shape recognition. In *Proceedings of the IEEE international conference on computer vision*, pages 945–953, 2015. 3

[36] Anil Tilbe. Zero-shot vs few-shot learning: Key insights with 2022 updates, 2022. 2

[37] Mikaela Angelina Uy, Quang-Hieu Pham, Binh-Son Hua, Thanh Nguyen, and Sai-Kit Yeung. Revisiting point cloud classification: A new benchmark dataset and classification model on real-world data. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1588–1597, 2019. 4, 8

[38] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 3

[39] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In

Proceedings of the IEEE conference on computer vision and pattern recognition, pages 1912–1920, 2015. 4, 8

[40] Yongqin Xian, Zeynep Akata, Gaurav Sharma, Quynh Nguyen, Matthias Hein, and Bernt Schiele. Latent embeddings for zero-shot classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 69–77, 2016. 2

[41] Lewei Yao, Jianhua Han, Youpeng Wen, Xiaodan Liang, Dan Xu, Wei Zhang, Zhenguo Li, Chunjing Xu, and Hang Xu. Detclip: Dictionary-enriched visual-concept paralleled pretraining for open-world detection. *Advances in Neural Information Processing Systems*, 35:9125–9138, 2022. 3

[42] Tianwei Yin, Xingyi Zhou, and Philipp Krähenbühl. Center-based 3d object detection and tracking. *CoRR*, abs/2006.11275, 2020. 1

[43] Li Zhang, Tao Xiang, and Shaogang Gong. Learning a deep embedding model for zero-shot learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2021–2030, 2017. 2

[44] Renrui Zhang, Rongyao Fang, Wei Zhang, Peng Gao, Kunchang Li, Jifeng Dai, Yu Qiao, and Hongsheng Li. Tip-adapter: Training-free clip-adapter for better vision-language modeling. *arXiv preprint arXiv:2111.03930*, 2021. 3

[45] Renrui Zhang, Ziyu Guo, Wei Zhang, Kunchang Li, Xupeng Miao, Bin Cui, Yu Qiao, Peng Gao, and Hongsheng Li. Pointclip: Point cloud understanding by clip. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8552–8562, 2022. 3, 5, 8

[46] Wu Zheng, Weiliang Tang, Sijin Chen, Li Jiang, and Chi-Wing Fu. Cia-ssd: Confident iou-aware single-stage object detector from point cloud. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 3555–3562, 2021. 7

[47] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *International Journal of Computer Vision*, 130(9):2337–2348, 2022. 3