LidarCLIP or: How I Learned to Talk to Point Clouds

Georg Hess[†,1,2] Adam Tonderski[†,1,3]
Christoffer Petersson[1,2] Kalle Åström[3] Lennart Svensson[2]
[1]Zenseact [2]Chalmers University of Technology [3]Lund University
{first.last}@zenseact.com lennart.svensson@chalmers.se karl.astrom@math.lth.se

Abstract

Research connecting text and images has recently seen several breakthroughs, with models like CLIP, DALL·E 2, and Stable Diffusion. However, the connection between text and other visual modalities, such as lidar data, has received less attention, prohibited by the lack of text-lidar datasets. In this work, we propose LidarCLIP, a mapping from automotive point clouds to a pre-existing CLIP embedding space. Using image-lidar pairs, we supervise a point cloud encoder with the image CLIP embeddings, effectively relating text and lidar data with the image domain as an intermediary. We show the effectiveness of LidarCLIP by demonstrating that lidar-based retrieval is generally on par with image-based retrieval, but with complementary strengths and weaknesses. By combining image and lidar features, we improve upon both single-modality methods and enable a targeted search for challenging detection scenarios under adverse sensor conditions. We also explore zero-shot classification and show that LidarCLIP outperforms existing attempts to use CLIP for point clouds by a large margin. Finally, we leverage our compatibility with CLIP to explore a range of applications, such as point cloud captioning and lidar-to-image generation, without any additional training. Code and pre-trained models at github.com/atonderski/lidarclip.

1. Introduction

Connecting natural language processing (NLP) and computer vision (CV) has been a long-standing challenge in the research community. Recently, OpenAI released CLIP [30], a model trained on 400 million web-scraped text-image pairs, that produces powerful text and image representations. Beside impressive zero-shot classification performance, CLIP enables interaction with the image domain in a diverse and intuitive way by using human language. These capabilities have resulted in a surge of work building upon CLIP embeddings within multiple applications, such

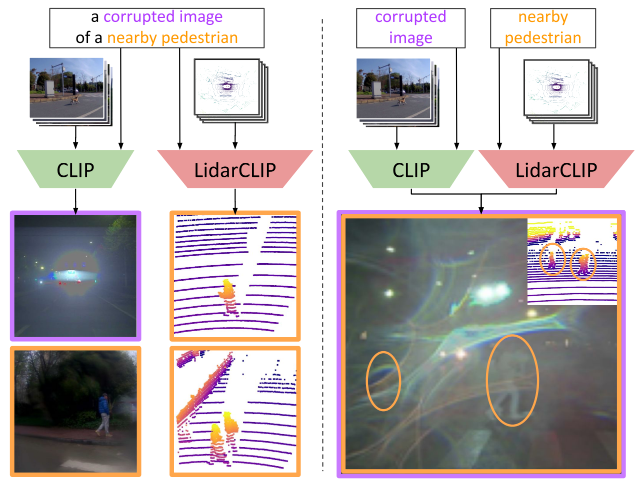

Figure 1. Like CLIP, LidarCLIP has many applications, including retrieval for data curation. We demonstrate that the two can be combined through different queries to retrieve potentially safety-critical scenes that a camera-based system may handle poorly (right). Such scenes are nearly impossible to retrieve with a single modality (left).

as image captioning [28], image retrieval [2, 16], semantic segmentation [45], text-to-image generation [31, 33], and referring image segmentation [21, 39].

While most works trying to bridge the gap between NLP and CV have focused on a single visual modality, namely images, other visual modalities, such as lidar point clouds, have received far less attention. Existing attempts to connect NLP and point clouds are often limited to a single application [5,34,44] or designed for synthetic data [43]. This is a natural consequence due to the lack of large-scale text-lidar datasets required for training flexible models such as CLIP in a new domain. However, it has been shown that the CLIP embedding space can be extended to new languages [4] and new modalities, such as audio [40], without the need for huge datasets and extensive computational resources. This raises the question if such techniques can be applied to point clouds as well, and consequently open up a body of research on point cloud understanding, similar to what has emerged for images [39, 45].

[†]These authors contributed equally to this work.

We propose LidarCLIP, a method to connect the CLIP embedding space to the lidar point cloud domain. While combined text and point cloud datasets are not easily accessible, many robotics applications capture images and point clouds simultaneously. One example is autonomous driving, where data is both openly available and large scale. To this end, we supervise a lidar encoder with a frozen CLIP image encoder using pairs of images and point clouds from the large-scale automotive dataset ONCE [25]. This way, the image encoder's rich and diverse semantic understanding is transferred to the point cloud domain. At inference, we can compare LidarCLIP's embedding of a point cloud with the embeddings from either CLIP's text encoder, image encoder, or both, enabling various applications.

While conceptually simple, we demonstrate LidarCLIP's fine-grained semantic understanding for a wide range of applications. LidarCLIP outperforms prior works applying CLIP in the point cloud domain [17, 43] on both zero-shot classification and retrieval. Furthermore, we demonstrate that LidarCLIP can be combined with regular CLIP to perform targeted searches for rare and difficult traffic scenarios, *e.g.*, a person crossing the road while hidden by water drops on the camera lens, see Fig. 1. Finally, LidarCLIP's capabilities are extended to point cloud captioning and lidar-to-image generation using established CLIP-based methods [9, 28].

In summary, our contributions are the following:

- We propose LidarCLIP, a new method for embedding lidar point clouds into an existing CLIP space.
- We demonstrate the effectiveness of LidarCLIP for retrieval and zero-shot classification in automotive data, where it outperforms existing CLIP-based methods.
- We show that LidarCLIP is complementary to its CLIP teacher and even outperforms it in certain retrieval categories. By combining both methods, we further improve performance and enable retrieval of safety-critical scenes in challenging sensing conditions.
- Finally, we show that our approach enables a multitude of applications off-the-shelf, such as point cloud captioning and lidar-to-image generation.

2. Related work

CLIP and its applications. CLIP [30] is a model with a joint embedding space for images and text. The model consists of two encoders, a text encoder $\mathcal{F}_T$ and an image encoder $\mathcal{F}_I$, both of which produce a single feature vector describing their input. Using contrastive learning, these feature vectors have been supervised to map to a common language-visual space where images and text are similar if they describe the same scene. By training on 400 million text-image pairs collected from the internet, the model has a diverse textual understanding.

The shared text-image space can be used for many tasks. For instance, to do zero-shot classification with K classes, one constructs K text prompts, *e.g.*, "a photo of a $\langle$class name$\rangle$". These are individually embedded by the text encoder, producing a feature map $Z_T \in \mathbb{R}^{K \times d}$. The logits for an image I are calculated by comparing the image embedding, $\mathbf{z}_I \in \mathbb{R}^d$, with the feature map for the text prompts, Z_T, and class probabilities p are found using the softmax function, softmax$(Z_T \mathbf{z}_I)$. In theory, any concept encountered in the millions of text-image pairs could be classified with this approach. Further, by comparing a single prompt to multiple images, CLIP can also be used for retrieving images from a database.

Multiple works have built upon the CLIP embeddings for various applications. DALL-E 2 [31] and Stable Diffusion [33] are two methods that use the CLIP space to condition diffusion models for text-to-image generation. Other works have recently shown how to use text-image embeddings to generate single 3D objects [35] and neural radiance fields [38] from text. In [45], CLIP is used for zero-shot semantic segmentation without any labels. Similarly, [39] extracts pixel-level information for referring semantic segmentation, *i.e.*, segmenting the part of an image referred to via a natural linguistic expression. We hope that LidarCLIP can spur similar applications for 3D data.

CLIP outside the language-image domain. Beside new applications, multiple works have aimed to extend CLIP to new domains, and achieved impressive performance in their respective domains. For videos, CLIP has been used for tasks like video clip retrieval [22, 24] and video question answering [42]. In contrast to our work, these methods rely on large amounts of text-video pairs for training. Meanwhile, WAV2CLIP [40] and AudioCLIP [14] extend CLIP to audio data for audio classification, tagging, and retrieval. Both methods use contrastive learning, which typically requires large batch sizes for convergence [6]. The scale of automotive point clouds would require extensive computational resources for contrastive learning, hence we supervise LidarCLIP with a simple mean squared error, which works well for smaller batch sizes and has been shown to promote the learning of richer features [4].

Point clouds and natural language. Recently, there has been increasing interest in connecting point clouds and natural language, as it enables an intuitive interface for the 3D domain and opens possibilities for open-vocabulary zero-shot learning. In [8] and [26], classifiers are supervised with pre-trained word embeddings to enable zero-shot learning. Parts2words [36] explores 3D shape retrieval by mapping scans of single objects and descriptive texts to a joint embedding space. However, a key limitation of these approaches is their need for dense annotations [8, 26] or

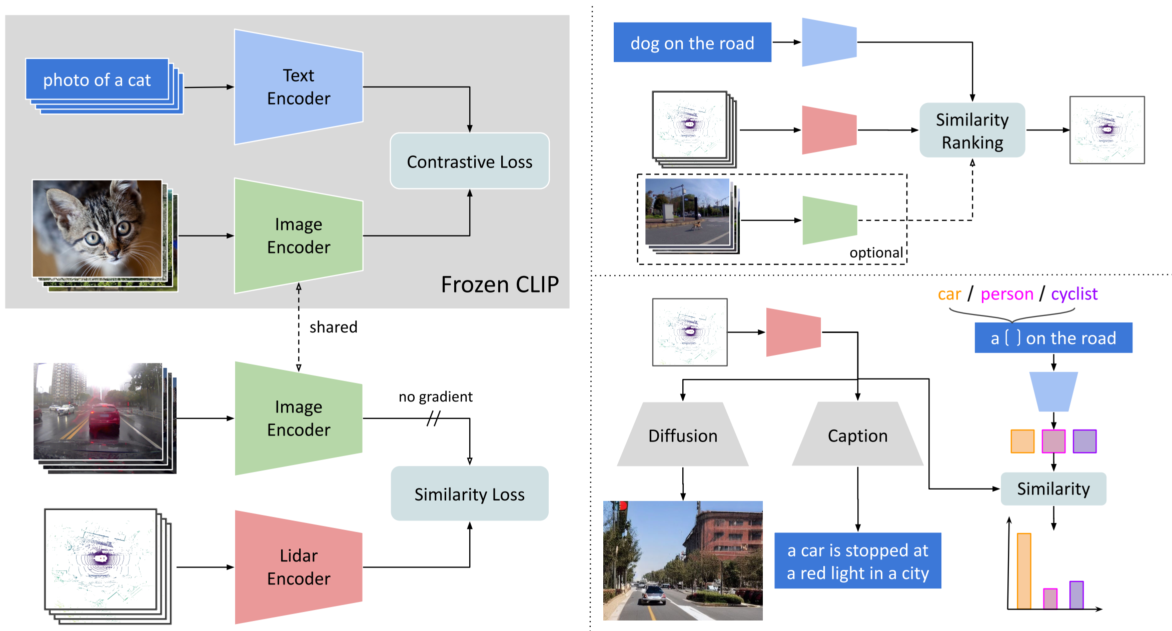

Figure 2. Overview of LidarCLIP. We use existing CLIP image and text encoders (top left) and learn to embed point clouds into the same feature space (bottom left). To that end, we train a lidar encoder to match the features of the frozen image encoder on a large automotive dataset with image-lidar pairs. This enables a wide range of applications, such as scenario retrieval (top right), zero-shot classification, as well as lidar-to-text and lidar-to-image generation (bottom right).

detailed textual descriptions [36], making them unable to leverage the vast amount of raw automotive data considered in this paper.

Other methods, such as PointCLIP [43] and CLIP2Point [17], use CLIP to bypass the need for text-lidar pairs entirely. Given a point cloud, they render it from multiple viewpoints and apply CLIP's image encoder to these renderings. While this works well with dense point clouds of a single object, the approach is not feasible for sparse and large-scale automotive data with heavy occlusions. In contrast, our method relies on an encoder specifically designed for the point cloud domain, avoiding the overhead introduced by multiple renderings and allowing for more flexibility in the model choice.

3. LidarCLIP

In this work, we encode lidar point clouds into the existing CLIP embedding space. As there are no datasets with text-lidar pairs, we cannot rely on the same contrastive learning strategy as the original CLIP model to directly relate point clouds to text. Instead, we leverage that automotive datasets contain millions of image-lidar pairs. By training a point cloud encoder to mimic the features of a frozen CLIP image encoder, the images act as intermediaries to connect text and point clouds; see Fig. 2.

Each training pair consists of an image $\mathbf{x}_I$ and the corresponding point cloud $\mathbf{x}_L$. Regular CLIP does not perform alignment between pairs, but some preprocessing is needed for point clouds. To align the contents of both modalities, we transform the point cloud to the camera coordinate system and drop all points that are not visible in the image. As a consequence, we only perform inference on frustums of the point cloud, corresponding to a typical camera field of view. We note that this preprocessing is susceptible to errors in sensor calibration and time synchronization, especially for objects along the edge of the field of view. Furthermore, the preprocessing does not handle differences in visibility due to sensor mounting positions, *e.g.*, lidars are typically mounted higher than cameras in data collection vehicles, thus seeing over some vehicles or static objects. However, using millions of training pairs reduces the impact of such noise sources.

The training itself is straightforward. An image is passed through the frozen image encoder $\mathcal{F}_I$ to produce the target embedding $\mathbf{z}_I$ whereas the lidar encoder $\mathcal{F}_L$ embeds a point cloud creating embedding $\mathbf{z}_L$

$$\mathbf{z}_I = \mathcal{F}_I(\mathbf{x}_I), \quad \mathbf{z}_L = \mathcal{F}_L(\mathbf{x}_L). \tag{1}$$

We train $\mathcal{F}_L$ to maximize the similarity between features $\mathbf{z}_I, \mathbf{z}_L \in \mathbb{R}^d$, using the mean squared error (MSE)

$$\mathcal{L}_{\text{MSE}}(\mathbf{z}_L, \mathbf{z}_I) = \frac{1}{d}(\mathbf{z}_I - \mathbf{z}_L)^T(\mathbf{z}_I - \mathbf{z}_L). \tag{2}$$

We also run ablations using the cosine similarity loss,

$$\mathcal{L}_{\text{cos}}(\mathbf{z}_L, \mathbf{z}_I) = -\frac{\mathbf{z}_I^T \mathbf{z}_L}{\|\mathbf{z}_I\|\|\mathbf{z}_L\|}, \tag{3}$$

By using a similarity loss that only considers positive pairs, as opposed to using a contrastive loss, we avoid the need for large batch sizes [6, 7] and the accompanying computational requirements. Furthermore, the benefits of contrastive learning are reduced in our setting, as we aim to map a new modality into an existing feature space, rather than learning an expressive feature space from scratch.

3.1. Joint retrieval

Retrieval is one of the most successful applications of CLIP and is highly relevant for the automotive industry. By retrieval, we mean the process of finding samples that best match a given natural language prompt out of all the samples in a large database. In an automotive setting, it is used to sift through the abundant raw data for valuable samples. Although CLIP works well for retrieval out of the box, it inherits the fundamental limitations of the camera modality, such as poor performance in darkness, glare, or water spray. LidarCLIP can increase robustness by leveraging the complementary properties of lidar.

Relevant samples are retrieved by computing the similarity between a text query and each sample in the database, in the CLIP embedding space, and identifying the samples with the highest similarity. These calculations may seem expensive, but the embeddings only need to be computed once per sample, after which they can be cached and reused for every text query. Following prior work [30], we compute the retrieval score using cosine similarity for both image and lidar

$$s_I = \frac{\mathbf{z}_T^T \mathbf{z}_I}{\|\mathbf{z}_T\| \|\mathbf{z}_I\|}, \qquad s_L = \frac{\mathbf{z}_T^T \mathbf{z}_L}{\|\mathbf{z}_T\| \|\mathbf{z}_L\|}, \tag{4}$$

where $\mathbf{z}_T$ is the text embedding. If a database only contains images or point clouds, we use the corresponding score (s_I or s_L) for retrieval. However, if we have access to both images and point clouds, we can jointly consider the lidar and image embeddings to leverage their respective strengths.

We consider various methods of performing joint retrieval. When providing both modalities with the same text prompt, we find simply adding the features, $\mathbf{z}_{I+L} = \mathbf{z}_I + \mathbf{z}_L$, to give the best performance. For separate prompts per modality, we instead add their similarity scores $s_{I+L} = s_L + s_I$. We also explore methods to aggregate independent rankings for each modality. One such approach is to consider the joint rank to be the mean rank across the modalities. Inspired by [27] we also consider a two-step re-ranking process, where one modality selects a set of candidates which are then ranked by the other modality.

One of the most exciting aspects of joint retrieval is the possibility of using different queries for each modality. For example, imagine trying to find scenes where a large white truck is almost invisible in an image due to extreme sun glare. In this case, one can search for scenes where the image embedding matches "an image with extreme sun glare" while considering the lidar embeddings' similarity to "a scene containing a large truck". This kind of scene would be almost impossible to retrieve using a single modality.

4. Experiments

Datasets. Training, and most of the evaluation, is done on the large-scale ONCE dataset [25], with roughly 1 million scenes. Each scene consists of a 360° lidar sweep and seven camera images, resulting in ~7 million unique training pairs. We withhold the validation and test sets and use these for the results presented below.

Implementation details. We use the official CLIP package and models, specifically the most capable vision encoder, ViT-L/14, which has a feature dimension $d = 768$. As our lidar encoder, we use the Single-stride Sparse Transformer (SST) [11] (randomly initialized). Due to computational constraints, our version of SST is down-scaled and contains about 8.5M parameters, which can be compared to the ~85M and ~300M parameters of the text and vision encoders of CLIP. The specific choice of backbone is not key to our approach; similarly to the variety of CLIP image encoders, one could use a variety of different lidar encoders. However, we choose a transformer-based encoder, inspired by the findings that CLIP transformers perform better than CLIP ResNets [30]. SST is trained for 3 epochs, corresponding to ~20 million training examples, using the Adam optimizer and the one-cycle learning rate policy. For full details, we refer to our code.

Retrieval ground truth & prompts. One difficulty in quantitatively evaluating the retrieval capabilities of LidarCLIP is the lack of direct ground truth for the task. Instead, automotive datasets typically have fine-grained annotations for each scene, such as object bounding boxes, segmentation masks, etc. This is also true for ONCE, which contains annotations in terms of 2D and 3D bounding boxes for five classes, and metadata for the time of day and weather. We leverage these detailed annotations and available metadata to create as many retrieval categories as possible. For object retrieval, we consider a scene positive if it contains one or more instances of that object. To probe the spatial understanding of the model, we also propose a "nearby" category, searching specifically for objects closer than $15\,\mathrm{m}$. We verify that the conclusions hold for thresholds between $10\,\mathrm{m}$ and $25\,\mathrm{m}$. Finally, to minimize the effect of prompt engineering, we follow [13] and average multiple text embeddings to improve results and reduce variability. For object retrieval, we use the same 85 prompt templates as in [13], and for the other retrieval categories, we use similar patterns to generate numerous relevant prompts templates. The exact prompts are provided in the source code.

	Fine-tuned on	Cls.	Obj.
PointCLIP [43]	-	29.1%	25.0%
CLIP2Point [17]	-	31.1%	26.2%
CLIP2Point [17]	ShapeNet	29.8%	28.2%
CLIP2Point [17]	ONCE	21.4%	3.2%
Image	-	58.6%	67.1%
LidarCLIP (ours)	ONCE	43.6%	62.1%
Joint (ours)	(see above)	**60.8%**	**73.3%**

Table 1. Zero-shot classification on ONCE *val*, top-1 accuracy averaged over classes/object instances.

4.1. Zero-shot classification

We would like to study zero-shot classification on ONCE, where scenes may contain many objects and classes. We construct this task by treating each annotated object in ONCE as a separate classification sample. Typically, LidarCLIP outputs a set of voxel features that are pooled into a single, global CLIP feature. For object classification, we instead generate object embeddings by only pooling features for voxels inside the corresponding bounding box, without any object-specific training/fine-tuning.

We compare our performance with PointCLIP [43] and CLIP2Point [17]. To transfer CLIP to 3D, both methods render point clouds from multiple "virtual" viewpoints, apply the CLIP image encoder, and pool the features from different views. Although these methods work without any additional training of the CLIP model, CLIP2Point proposes to fine-tune the image encoder on rendered point clouds with supervision from image embeddings of the same scene, similar to our training. We evaluate both their provided ShapeNet weights and a version we train on ONCE[1] using the same scene-level data as LidarCLIP. To evaluate, we follow their proposed protocol and render only points within each annotated bounding box. Although this differs from LidarCLIP's global processing, it is analogous to the methods' original single-object setting and greatly improves their performance. Further, we use the prompts proposed in [43] instead of our prompt ensembling, as we find them to work better.

We report top-1 accuracy in Tab. 1, both averaged equally over all instances and classwise, as the data contains a few majority classes. LidarCLIP convincingly outperforms its lidar counterparts [17, 43], demonstrating the gain of training a modality-specific encoder rather than transferring point clouds to the image domain. For completeness, we also extract image crops corresponding to each object bounding box and classify them using CLIP. While performing very well, CLIP, as well as PointCLIP and CLIP2Point, are given clear advantages over LidarCLIP, which processes the entire scene without 'zooming in' on each object separately. However, the 'zooming in' also creates a discrepancy between CLIP2Point's fine-tuning on ONCE, which contains large scenes with multiple objects, and the object-centric evaluation. This explains why CLIP2Point performs better using ShapeNet weights in this setting. Finally, we also propose a joint classifier by averaging image and lidar features. This performs the best of all approaches, showing that LidarCLIP extracts features complementary to CLIP. We emphasize that LidarCLIP's instance-level classification performance is achieved without any dense annotations in 3D or 2D.

[1]As in the original manuscript, we find that ViT-B/32 performs best. Training setup and ViT-L/14 results are in the supplementary material.

4.2. Retrieval

To evaluate retrieval, we report the commonly used Precision at rank K (P@K) [12, 23, 32], for $K = 10, 100$, which measures the fraction of positive samples within the top K predictions. Recall at K is another commonly used metric [12, 32], however, it is hard to interpret when the number of positives is in the thousands, as is the case here. We evaluate the performance of three approaches: lidar-only, camera-only, and the joint approach proposed in Sec. 3.1. We compare our performance to PointCLIP [43] and CLIP2Point [17], for which we render scene-level point clouds (details in the supplementary). Tab. 2 shows that PointCLIP and CLIP2Point are poorly suited for the large-scale point clouds considered here, even though fine-tuning on ONCE greatly improves the CLIP2Point performance. We also include a version of LidarCLIP supervised by ViT-B/32 for direct comparison to existing methods.

Object-level. Interestingly, LidarCLIP performs slightly better than image CLIP for object retrieval despite being trained to mimic the image features. We hypothesize some classes' features to be more similar across instances in the point cloud than in the image. A class breakdown (see supplementary material), for instance, shows large gains for LidarCLIP in the cyclist class, where we believe the lidar encoder generalizes to cyclists that go undetected by the image encoder. Simultaneously, upon qualitative inspection, we find that the lidar encoder confuses trucks with buses, as these appear more similar in lidar data than in images. We also attempt to retrieve scenes where objects of a given class are close to the ego vehicle. Here, we can see that joint retrieval truly shines. One interpretation is that the lidar is more reliable at determining distance, while the image can be leveraged to distinguish between classes (such as trucks and buses) based on textures and other fine details only visible in the image.

Scene-level. Object-level retrieval focuses on *local* details of a scene and should trigger even for a single occluded pedestrian on the side of the road. Therefore, we run another set of experiments focusing on *global* properties such as weather, time of day, and general 'crowdedness' of the

P@K	10	100	10	100	10	100	10	100	10	100	10	100
Category	Objects		Nearby objects		Time of Day		Weather		Crowdedness		Overall	
PointCLIP	0.30	0.29	0.06	0.09	0.30	0.51	0.40	0.54	0.35	0.34	0.244	0.293
CLIP2Point	0.24	0.24	0.08	0.05	0.50	0.49	0.45	0.48	0.55	0.40	0.288	0.262
CLIP2Point†	0.58	0.60	0.46	0.42	0.85	0.76	0.75	0.75	0.80	0.64	0.625	0.586
Image-B	0.84	0.82	0.76	0.67	0.95	0.96	**1.00**	**1.00**	0.75	0.80	0.834	0.810
LidarCLIP-B	0.92	0.84	0.80	0.75	0.60	0.95	0.90	0.77	0.80	0.78	0.869	0.810
Joint-B	0.94	**0.85**	0.84	0.76	**1.00**	0.89	**1.00**	**1.00**	**0.95**	0.84	0.881	0.843
Image-L	0.84	0.81	0.76	0.67	0.95	0.96	**1.00**	**1.00**	0.90	0.82	0.856	0.810
LidarCLIP-L	0.92	0.82	0.88	0.78	0.60	0.76	0.65	0.86	0.75	0.81	0.813	0.803
Joint-L	**0.96**	0.84	**0.90**	**0.81**	**1.00**	**1.00**	**1.00**	**1.00**	0.90	**0.89**	**0.944**	**0.876**

Table 2. Retrieval for scenes corresponding to various categories. We report precision at ranks 10 and 100. B and L correspond to the CLIP version (ViT-B/32 or ViT-L/14). †ONCE fine-tuning. Interestingly, Lidar-B outperforms Image-B and Lidar-L, but Joint-L strongly outperforms all other approaches. Detailed results are available in the supplementary material.

Loss function	P@10	P@100
Mean squared error	**0.869**	**0.810**
Cosine similarity	0.781	0.748

Table 3. Ablation of the LidarCLIP-B training loss. We report precision at ranks 10 and 100, averaged over all prompts. Training with MSE leads to better retrieval performance.

scene. In Tab. 2, we see that the lidar is clearly outperformed by the camera for determining time of day. This seems expected, and, if anything, it is somewhat surprising that lidar can do significantly better than random guessing. Again, we see that joint retrieval consistently gets the best of both worlds and, in some cases, such as finding crowded scenes, clearly outperforms both single-modality methods.

Separate prompts. Inspired by the success of joint retrieval and the complementary sensing of lidar and camera, we present some qualitative examples where different prompts are used for each modality. Thus, we can find scenes that are difficult to identify with a single modality. Fig. 3 shows retrieval examples where the image was prompted for glare, extreme blur, water on the lens, corruption, and lack of objects in the scene. At the same time, the lidar was prompted for nearby objects such as cars, trucks, and pedestrians. As seen in Fig. 3, the examples indicate that we can retrieve scenes where these objects are almost completely invisible in the image. Such samples are very valuable for both the training and validation of autonomous driving systems.

Ablations. As described in Sec. 3, we have two primary candidates for the training loss function. MSE encourages the model to embed the point cloud in the same position as the image, whereas cosine similarity only cares about matching the directions of the two embeddings. We compare the retrieval performance of two models trained using these losses in Tab. 3. To reduce training time, we use the

Method	P@10	P@100
Image only	0.856	0.810
LidarCLIP only	0.813	0.803
Mean feature	**0.944**	**0.876**
Mean norm. feature	**0.944**	0.875
Mean score	0.919	0.874
Mean ranking	0.888	0.854
Reranking - img first	0.925	0.867
Reranking - lidar first	0.875	0.860

Table 4. Ablation of joint retrieval methods. We report precision at ranks 10 and 100, averaged over all prompts. All methods improve upon the single-modality models, but averaging lidar and image features before normalization achieves the best performance.

Train set	ONCE		nuScenes	
P@K	10	100	10	100
Image	0.69	0.65	0.69	0.65
LidarCLIP	0.46	0.40	0.79	0.64
Joint	**0.74**	**0.69**	**0.81**	**0.70**

Table 5. nuScenes *val* retrieval with different *train* sets. Performance is averaged over classes. LidarCLIP supports the joint retrieval, even when trained and evaluated on separate datasets.

ViT-B/32 CLIP version, rather than the heavier ViT-L/14. The results show that using MSE leads to significantly better retrieval, even though retrieval uses cosine similarity as the scoring function. We also perform ablations on the different approaches for joint retrieval described in Sec. 3.1. As shown in Tab. 4, the simple approach of averaging the camera and lidar features gives the best performance, and it is thus the approach used throughout the paper.

Domain transfer. For studying the robustness of LidarCLIP under domain shift, we evaluate its retrieval per-

Figure 3. Example of retrieval using separate prompts for image and lidar. We query for images with blur, water spray, corruption, and lack of objects, and for point clouds with nearby objects. We combine the scores of these separate queries to find edge cases that are extremely valuable during the training/validation of camera-based perception systems. Hidden bus (left) and pedestrian (right) are highlighted in red, both in the image and point cloud.

formance on a different dataset than it was trained on, namely, the nuScenes dataset [3]. Compared to ONCE, the nuScenes lidar sensor has fewer beams (32 vs 40), lower horizontal resolution, and different intensity characteristics. Further, nuScenes is collected in Boston and Singapore, while ONCE is collected in Chinese cities. The challenge of transferring between these datasets has been shown in unsupervised domain adaptation [25]. Similar to ONCE, the retrieval ground truth is generated from object annotations.

We compare the model trained on ONCE with a reference model trained on nuScenes in Tab. 5. As expected, the differences in sensor characteristics hamper the ability to perform lidar-only retrieval on the target dataset. Notably, we find that the joint method is robust to this effect, showing almost no domain transfer gap, and outperforming camera-only retrieval even with the ONCE-trained lidar encoder.

Investigating lidar sensing capabilities. Besides its usefulness for retrieval, LidarCLIP can offer more understanding of what concepts can be captured with a lidar sensor. While lidar data is often used in tasks such as object detection [41], panoptic/semantic segmentation [1, 19], and localization [10], research into capturing more abstract concepts with lidar data is limited and focused mainly on weather classification [15, 37]. However, we show that LidarCLIP can indeed capture complex scene concepts, as already demonstrated in Tab. 2.

Motivated by this, we explore the ability of LidarCLIP to extract color information, by retrieving scenes with "a ⟨color⟩ car". As illustrated in Figure 4, while LidarCLIP struggles to capture specific colors accurately, it consistently differentiates between bright and dark colors. Such partial color information may be valuable for systems fusing lidar and camera information. Additionally, LidarCLIP learns meaningful features for overall scene lighting conditions, as illustrated in Figure 5. It can retrieve scenes based on the time of day, and is even able to distinguish scenes

Figure 4. Top-5 retrieved examples from LidarCLIP for different colors. Note that images are only for visualization, point clouds were used for retrieval. LidarCLIP consistently differentiates black and white but struggles with specific colors.

Figure 5. Top-5 retrieved examples from LidarCLIP for different lighting conditions (image only for visualization). LidarCLIP is surprisingly good at understanding the lighting of the scene, to the point of picking up on oncoming headlights with great accuracy.

with many headlights from regular night scenes. Notably, all retrieved scenes are sparsely populated, indicating that LidarCLIP does not rely on biases associated with street congestion at different times of the day.

4.3. Generative applications

To demonstrate the flexibility of LidarCLIP, we integrate it with two existing CLIP-based generative models.

	C (L)	C (I)	L	I	L+C	I+C	[18]
FID ↓	83.0	81.7	68.7	58.7	53.7	46.9	114.2
CLIP-FID ↓	33.4	31.2	20.1	15.4	15.1	11.4	25.0

Table 6. FID and CLIP-FID (ViT-B/32) for ≈6k generated images from the ONCE *val*. L=lidar, I=image, C+(L/I)=caption only, from L/I.

For lidar-to-text generation, we utilize an image captioning model called ClipCap [28], and for lidar-to-image generation, we use CLIP-guided Stable Diffusion [33]. In both cases, we replace the expected text or image embeddings with our point cloud embedding.

We evaluate image generation with the widely used Fréchet Inception Distance (FID) [29]. For this, we randomly select ≈6000 images from ONCE *val* and generate images using CLIP-generated captions, CLIP features, or a combination of both. Notably, this setting not only evaluates the image generation performance but also serves as a proxy for assessing the captioning quality. While FID is widely used, it has been shown to sometimes align poorly with human judgment [20]. To complement this evaluation, we use CLIP-FID, with a different CLIP model to avoid any bias. We also implement pix2pix [18] as a baseline for lidar-to-image generation; details are in the supplementary material. Our results, presented in Tab. 6, demonstrate that incorporating captions significantly improves the photo-realism of the generated images. Interestingly, LidarCLIP with captions even outperforms image CLIP without captions, underscoring the effectiveness of our approach in generating high-quality images from point cloud data.

Some qualitative results are shown in Fig. 6. We find that both generative tasks work fairly well out of the box. The generated images are not entirely realistic, partly due to a lack of tuning on our side, but there are clear similarities with the reference images. This demonstrates that our lidar embeddings can capture a surprising amount of detail. We hypothesize that guiding the diffusion process locally, by projecting regions of the point cloud into the image plane, would result in more realistic images. We hope that future work can investigate this avenue. Similarly, the captions can pick up the specifics of the scene. However, we notice that more 'generic' images result in captions with very low diversity, such as "several cars driving down a street next to tall buildings". This is likely an artifact of the fact that the captioning model was trained on COCO, which only contains a few automotive images and has a limited vocabulary.

5. Limitations

For the training of LidarCLIP, a single automotive dataset was used. While ONCE [25] contains millions of image-lidar pairs, they originate from about 1,000 densely sampled sequences, meaning that the dataset lacks diversity compared to the 400M text-image pairs used to train CLIP [30]. Effectively, LidarCLIP has mainly transferred CLIP's knowledge within an automotive setting and is not expected to work in a more general setting, such as indoors. An interesting future direction would be to train LidarCLIP on multiple datasets, with a variety of lidar sensors, scene conditions, and geographic locations.

Figure 6. Example of generative application of LidarCLIP. A point cloud is embedded into the CLIP space (left, image only for reference) and used to generate text (top) and images. The image generation can be guided with only the lidar embedding (middle) or with both the lidar embedding and the generated caption (right).

6. Conclusions

We propose LidarCLIP, which encodes lidar data into an existing text-image embedding space. Our method is trained using image-lidar pairs and enables multi-modal reasoning, connecting lidar to both text and images. While conceptually simple, LidarCLIP performs well over a range of tasks. For retrieval, we present a method for combining lidar and image features, outperforming their single-modality equivalents. Moreover, we use the joint retrieval method for finding challenging scenes under adverse sensor conditions. We also demonstrate that LidarCLIP enables several interesting applications off-the-shelf, including point cloud captioning and lidar-to-image generation. We hope LidarCLIP can inspire future work to dive deeper into connections between text and point cloud understanding, and explore tasks such as referring object detection and open-set semantic segmentation.

Acknowledgements. This work was partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation. Computational resources were provided by the Swedish National Infrastructure for Computing at C3SE and NSC, partially funded by the Swedish Research Council, grant agreement no. 2018-05973.

References

[1] Mehmet Aygun, Aljosa Osep, Mark Weber, Maxim Maximov, Cyrill Stachniss, Jens Behley, and Laura Leal-Taixé. 4d panoptic lidar segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5527–5537, 2021. 7

[2] 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. 1

[3] 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. 7

[4] Fredrik Carlsson, Philipp Eisen, Faton Rekathati, and Magnus Sahlgren. Cross-lingual and multilingual clip. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 6848–6854, 2022. 1, 2

[5] Dave Zhenyu Chen, Angel X Chang, and Matthias Nießner. Scanrefer: 3d object localization in rgb-d scans using natural language. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XX*, pages 202–221. Springer, 2020. 1

[6] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020. 2, 4

[7] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15750–15758, 2021. 4

[8] 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

[9] Prafulla Dhariwal and Alexander Quinn Nichol. Diffusion models beat GANs on image synthesis. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021. 2

[10] Mahdi Elhousni and Xinming Huang. A survey on 3d lidar localization for autonomous vehicles. In *2020 IEEE Intelligent Vehicles Symposium (IV)*, pages 1879–1884, 2020. 7

[11] Lue Fan, Ziqi Pang, Tianyuan Zhang, Yu-Xiong Wang, Hang Zhao, Feng Wang, Naiyan Wang, and Zhaoxiang Zhang. Embracing single stride 3d object detector with sparse transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8458–8468, June 2022. 4

[12] M Rami Ghorab, Dong Zhou, Alexander O'connor, and Vincent Wade. Personalised information retrieval: survey and classification. *User Modeling and User-Adapted Interaction*, 23(4):381–443, 2013. 5

[13] Xiuye Gu, Tsung-Yi Lin, Weicheng Kuo, and Yin Cui. Open-vocabulary object detection via vision and language knowledge distillation. In *International Conference on Learning Representations*, 2022. 4

[14] Andrey Guzhov, Federico Raue, Jörn Hees, and Andreas Dengel. Audioclip: Extending clip to image, text and audio. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 976–980, 2022. 2

[15] Robin Heinzler, Philipp Schindler, Jürgen Seekircher, Werner Ritter, and Wilhelm Stork. Weather influence and classification with automotive lidar sensors. In *2019 IEEE intelligent vehicles symposium (IV)*, pages 1527–1534. IEEE, 2019. 7

[16] 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. 1

[17] 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*, 2023. 2, 3, 5

[18] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017. 8

[19] Alok Jhaldiyal and Navendu Chaudhary. Semantic segmentation of 3d lidar data using deep learning: a review of projection-based methods. *Applied Intelligence*, pages 1–12, 2022. 7

[20] Tuomas Kynkäänniemi, Tero Karras, Miika Aittala, Timo Aila, and Jaakko Lehtinen. The role of imagenet classes in fréchet inception distance. In *Proceedings of International Conference on Learning Representations, ICLR*, 2023. 8

[21] Timo Lüddecke and Alexander Ecker. Image segmentation using text and image prompts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7086–7096, 2022. 1

[22] Huaishao Luo, Lei Ji, Ming Zhong, Yang Chen, Wen Lei, Nan Duan, and Tianrui Li. Clip4clip: An empirical study of clip for end to end video clip retrieval and captioning. *Neurocomputing*, 508:293–304, 2022. 2

[23] Haoyu Ma, Handong Zhao, Zhe Lin, Ajinkya Kale, Zhangyang Wang, Tong Yu, Jiuxiang Gu, Sunav Choudhary, and Xiaohui Xie. Ei-clip: Entity-aware interventional contrastive learning for e-commerce cross-modal retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18051–18061, 2022. 5

[24] Yiwei Ma, Guohai Xu, Xiaoshuai Sun, Ming Yan, Ji Zhang, and Rongrong Ji. X-clip: End-to-end multi-grained contrastive learning for video-text retrieval. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 638–647, 2022. 2

[25] Jiageng Mao, Minzhe Niu, Chenhan Jiang, hanxue liang, Jingheng Chen, Xiaodan Liang, Yamin Li, Chaoqiang Ye, Wei Zhang, Zhenguo Li, Jie Yu, Hang Xu, and Chunjing Xu.

One million scenes for autonomous driving: ONCE dataset. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*, 2021. 2, 4, 7, 8

[26] Björn Michele, Alexandre Boulch, Gilles Puy, Maxime Bucher, and Renaud Marlet. Generative zero-shot learning for semantic segmentation of 3d point clouds. In *2021 International Conference on 3D Vision (3DV)*, pages 992–1002. IEEE, 2021. 2

[27] Sewon Min, Kenton Lee, Ming-Wei Chang, Kristina Toutanova, and Hannaneh Hajishirzi. Joint passage ranking for diverse multi-answer retrieval. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6997–7008, Online and Punta Cana, Dominican Republic, Nov. 2021. Association for Computational Linguistics. 4

[28] Ron Mokady, Amir Hertz, and Amit H Bermano. Clipcap: Clip prefix for image captioning. *arXiv preprint arXiv:2111.09734*, 2021. 1, 2, 8

[29] Gaurav Parmar, Richard Zhang, and Jun-Yan Zhu. On aliased resizing and surprising subtleties in GAN evaluation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11410–11420, 2022. 8

[30] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR, 2021. 1, 2, 4, 8

[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*, 2022. 1, 2

[32] Mehwish Rehman, Muhammad Iqbal, Muhammad Sharif, and Mudassar Raza. Content based image retrieval: survey. *World Applied Sciences Journal*, 19(3):404–412, 2012. 5

[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. 1, 2, 8

[34] David Rozenberszki, Or Litany, and Angela Dai. Language-grounded indoor 3d semantic segmentation in the wild. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXIII*, pages 125–141. Springer, 2022. 1

[35] Aditya Sanghi, Hang Chu, Joseph G. Lambourne, Ye Wang, Chin-Yi Cheng, Marco Fumero, and Kamal Rahimi Malekshan. Clip-forge: Towards zero-shot text-to-shape generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18603–18613, June 2022. 2

[36] Chuan Tang, Xi Yang, Bojian Wu, Zhizhong Han, and Yi Chang. Parts2words: Learning joint embedding of point clouds and texts by bidirectional matching between parts and words. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6884–6893, June 2023. 2, 3

[37] Jose Roberto Vargas Rivero, Thiemo Gerbich, Valentina Teiluf, Boris Buschardt, and Jia Chen. Weather classification using an automotive lidar sensor based on detections on asphalt and atmosphere. *Sensors*, 20(15), 2020. 7

[38] Can Wang, Menglei Chai, Mingming He, Dongdong Chen, and Jing Liao. Clip-nerf: Text-and-image driven manipulation of neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3835–3844, June 2022. 2

[39] Zhaoqing Wang, Yu Lu, Qiang Li, Xunqiang Tao, Yandong Guo, Mingming Gong, and Tongliang Liu. Cris: Clip-driven referring image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11686–11695, 2022. 1, 2

[40] Ho-Hsiang Wu, Prem Seetharaman, Kundan Kumar, and Juan Pablo Bello. Wav2clip: Learning robust audio representations from clip. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 4563–4567. IEEE, 2022. 1, 2

[41] Yutian Wu, Yueyu Wang, Shuwei Zhang, and Harutoshi Ogai. Deep 3d object detection networks using lidar data: A review. *IEEE Sensors Journal*, 21(2):1152–1171, 2021. 7

[42] Hu Xu, Gargi Ghosh, Po-Yao Huang, Dmytro Okhonko, Armen Aghajanyan, Florian Metze, Luke Zettlemoyer, and Christoph Feichtenhofer. VideoCLIP: Contrastive pre-training for zero-shot video-text understanding. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Online, Nov. 2021. Association for Computational Linguistics. 2

[43] 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. 1, 2, 3, 5

[44] Lichen Zhao, Daigang Cai, Lu Sheng, and Dong Xu. 3dvg-transformer: Relation modeling for visual grounding on point clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2928–2937, 2021. 1

[45] Chong Zhou, Chen Change Loy, and Bo Dai. Extract free dense labels from clip. In *Proceedings of the European Conference on Computer Vision*, pages 696–712. Springer, 2022. 1, 2