TOWARD UNIFIED MULTIMODAL REPRESENTATION LEARNING: CONTRASTIVE ALIGNMENT OF TEXT, IMAGE, AND POINT CLOUD

Anonymous authors
Paper under double-blind review

ABSTRACT

Contrastive Language-Image Pre-training (CLIP) has shown impressive performance in aligning visual and textual representations. Recent works have extended this paradigm to 3D vision, where a common strategy is to align point-cloud representations with text and image features through pairwise contrastive objectives. However, such pairwise formulations optimize modality pairs separately rather than multimodal tuples jointly. In this paper, we propose Contrastive Tensor Pre-training (CTP), a joint contrastive learning framework that extends the pairwise similarity matrix to a multimodal similarity tensor for text–image–point-cloud alignment. We further study two key aspects of joint multimodal learning. First, we show that a simple geometric scorer can effectively align the three modality representations. Second, we identify partial-match negatives introduced by joint tuple construction and show that masking them improves joint alignment. Experiments on real-world automotive LiDAR from nuScenes and dense point clouds from ULIP-ShapeNet show that CTP outperforms a matched pairwise baseline in both zero-shot classification and instance-level retrieval. The learned representations also transfer to KITTI and Waymo without target-domain training, further supporting the cross-dataset generalization of CTP.

1 INTRODUCTION

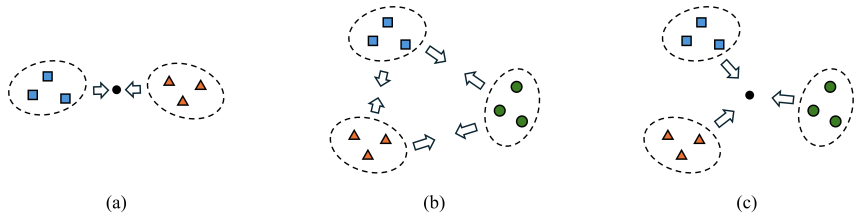


Figure 1: Overview of different contrastive representation learning methods. (a) CLIP: aligns two modalities. (b) Performs pairwise alignment between every modality pair. (c) CTP: aligns all modalities toward one point.

Large Language Models (LLMs) have demonstrated remarkable performance in the textual modality, as well as strong capabilities in scene understanding, reasoning, and decision-making (Brown et al., 2020). Furthermore, Vision–Language Models (VLMs) have achieved the ability to understand visual and textual domains (Liu et al., 2023a; Li et al., 2023; Alayrac et al., 2022), which significantly benefits many downstream tasks such as robotic manipulation (Zitkovich et al., 2023; Kim et al., 2024) and end-to-end autonomous driving (Chen et al., 2024b; Hwang et al., 2024; Ma et al., 2025; Xing et al., 2025; Zhou et al., 2025; Huang et al., 2024). Recently, there has been growing interest in extending the capabilities of LLMs from 2D to 3D domains (Hong et al., 2023; Yang et al., 2025; Chen et al., 2024a; Kerr et al., 2023). Point clouds are a common representation of 3D scenes, providing explicit geometric information and being less sensitive to illumination variations than camera images. These properties make point-cloud perception particularly important for

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

applications such as autonomous driving. However, real-world LiDAR point clouds are often sparse and incomplete (Ma et al., 2024), which makes their alignment with image and text representations particularly challenging.

CLIP (Radford et al., 2021) provides an effective approach for aligning textual and visual modalities (Fig. 1a). The pretrained Vision Transformer (ViT) (Dosovitskiy, 2020) can serve as the image encoder, forming the vision tower for LLMs (Liu et al., 2023a). This shared embedding approach has subsequently motivated extending vision–language representations to 3D modalities. Inspired by the CLIP framework, several studies have explored extending the CLIP embedding space to point clouds (Zhang et al., 2022), showing improved scene understanding (Chen et al., 2023; Peng et al., 2023; Liao et al., 2024), localization (Shubodh et al., 2024), semantic segmentation (Wang et al., 2023c) and object detection (Lu et al., 2023; Pan et al., 2024). A common strategy is to use contrastive learning to align 3D representations with CLIP text embeddings (Zhang et al., 2022) or image embeddings (Hess et al., 2024), or to align them with both modalities through similarity matrices (Zeng et al., 2023; Hegde et al., 2023; Xue et al., 2023; Wang et al., 2023b) (Fig. 1b). Most existing language–image–3D methods optimize alignment through separate modality pairs. More recently, several methods have explored joint multimodal alignment by scoring complete multimodal tuples using statistical or geometric compatibility functions (Saporta et al., 2024; Cicchetti et al., 2025a;b). These developments raise an important question for 3D multimodal representation learning: how should text, image, and point-cloud representations be jointly contrasted and scored within a shared embedding space?

To address this question, we propose Contrastive Tensor Pre-training (CTP), a joint contrastive learning framework for text, image, and point-cloud representations (Fig. 1c). Instead of optimizing each modality pair independently, CTP organizes all text–image–point-cloud combinations in a similarity tensor and contrasts complete multimodal tuples within a unified candidate space. We further study two key aspects of this joint formulation. First, we investigate how multimodal tuples should be scored and show that a simple cosine-based geometric scorer remains effective in the three-modality setting. Although compressing the three pairwise relations into a scalar score can map different geometric configurations to the same value, the CTP scorer has a unique optimal angle configuration at multimodal co-location, where all three normalized representations coincide. This provides a simple and direct geometric constraint for three-modality alignment. Second, joint cross-entropy normalization introduces partial-match negatives along the anchor-associated rows and columns of the similarity tensor. Although these tuples are negative as a whole, they still contain a matched pair involving the anchor, so suppressing their scores can introduce competing alignment signals. Masking these candidates consistently improves the geometric joint methods considered in this work.

We evaluate CTP on real-world automotive LiDAR from nuScenes (Caesar et al., 2019) and dense point clouds from ULIP-ShapeNet (Xue et al., 2024), and further assess cross-dataset transfer to KITTI (Geiger et al., 2012) and Waymo (Sun et al., 2020). Our experiments show that CTP outperforms a matched pairwise baseline in both zero-shot classification and instance-level retrieval.

2 Related Work

Multimodal Alignment. Multimodal alignment has gained increasing attention for its ability to enable models to handle diverse multimodal tasks. Vision–language pre-training (Li et al., 2022; Yao et al., 2021) has recently achieved remarkable success across various downstream applications. Pioneering works such as CLIP (Radford et al., 2021) and ALIGN (Jia et al., 2021) focus on visual and textual modalities, demonstrating effective alignment between the two representations. Various methods have adapted CLIP to extract video (Xu et al., 2021; Luo et al., 2022), audio (Guzhov et al., 2022; Elizalde et al., 2023), and 3D representations (Zhu et al., 2023b). Beyond alignment between two modalities, ImageBind (Girdhar et al., 2023) introduced a multimodal pre-training framework that leverages the image modality as a bridge to bind all modalities together. LanguageBind (Zhu et al., 2023a) instead employs the text modality as the alignment anchor to further enhance multimodal representation learning. These approaches primarily establish multimodal alignment through a common anchor or separate modality pairs rather than directly scoring complete multimodal tuples. More recently, joint multimodal learning has been explored by directly scoring complete multimodal tuples. Symile (Saporta et al., 2024) uses a multilinear inner product to capture statistical dependence among multiple modalities, while TRIANGLE (Cicchetti et al., 2025a) and

GRAM (Cicchetti et al., 2025b) introduce geometric similarity measures based on triangle area and Gram volume, respectively. In contrast to these general multimodal formulations, we study joint contrastive learning specifically for text–image–point-cloud representations, with a focus on how multimodal tuples are scored and how contrastive candidates are constructed.

3D Representation Learning. Recently, there has been growing interest in connecting 3D point clouds with natural language through vision–language pre-training frameworks. Early works such as PointCLIP (Zhang et al., 2022), PointCLIP v2 (Zhu et al., 2023b), and CLIP2Point (Huang et al., 2023) transfer knowledge from pretrained vision–language models to 3D representation learning. LidarCLIP (Hess et al., 2024) extends this direction to real-world LiDAR by aligning LiDAR features with pretrained CLIP image representations. Subsequent methods, including CLIP^2 (Zeng et al., 2023), TriCLIP-3D (Li et al., 2025), OpenShape (Liu et al., 2023b), and Uni3D (Zhou et al., 2024), further explore multimodal supervision for 3D representation learning. ULIP (Xue et al., 2023), ULIP-2 (Xue et al., 2024), and CLIP goes 3D (Hegde et al., 2023) leverage both image and language supervision to align point-cloud representations with pretrained vision–language spaces. JM3D (Wang et al., 2023a) further combines image and language cues into a joint vision–language representation before aligning it with 3D features. Despite incorporating multiple modalities, these approaches primarily formulate contrastive alignment between two representation sets rather than directly contrasting complete text–image–point-cloud tuples. In contrast, CTP treats complete text–image–point-cloud tuples as contrastive candidates and jointly optimizes their similarity within a similarity tensor. We further investigate how tuple scoring and partial-match negatives influence three-modality contrastive alignment.

3 Methodology

In this section, we introduce Contrastive Tensor Pre-training (CTP), a similarity-tensor-based contrastive learning framework for jointly aligning text, image, and point-cloud representations. An overview of the framework is shown in Fig. 2. We first construct a joint multimodal candidate space using a similarity tensor, then define the tuple similarity scorer and tensor contrastive objective, and finally describe how partial-match negatives are handled.

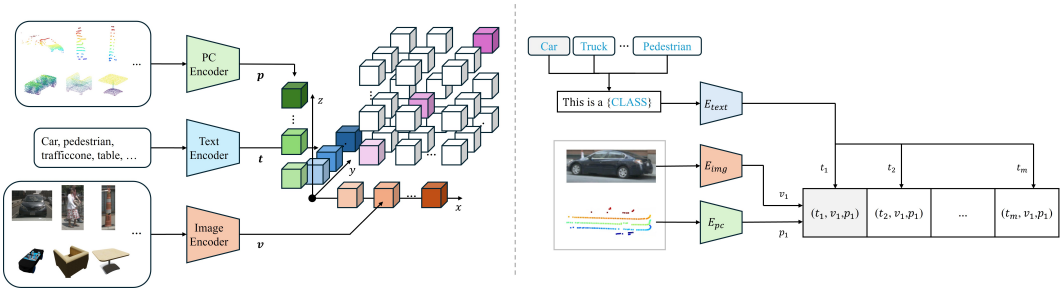


Figure 2: Overview of the CTP framework. Given matched text, image, and point-cloud inputs, modality-specific encoders map them into a shared embedding space. The resulting embeddings are arranged along three axes to construct a similarity tensor, where each entry represents the similarity of a text–image–point-cloud tuple. During zero-shot classification, class prompts are encoded as text candidates and jointly scored against the image and point-cloud representations to predict the class label.

3.1 Similarity Tensor

Contrastive alignment between two modalities is commonly formulated using a pairwise similarity matrix (Radford et al., 2021). For more than two modalities, a straightforward extension is to optimize separate pairwise contrastive objectives for each modality pair (Xue et al., 2023; Zeng et al., 2023; Hegde et al., 2023). For q modalities and a mini-batch of size b, this formulation produces $\frac{q(q-1)}{2}$ similarity matrices, each containing b^2 pairwise scores. In contrast, the Cartesian product of the same mini-batch contains b^q possible q-way tuples, each jointly indexed by one sample

from every modality. Pairwise and joint formulations therefore differ in the structure of their contrastive candidate spaces: pairwise objectives normalize candidates within individual modality pairs, whereas joint objectives normalize complete multimodal tuples as contrastive candidates (Saporta et al., 2024).

We focus on three modalities in this work: text, image, and point cloud, denoted by T, I, and P, respectively. Given a mini-batch of matched triplets,

$$\mathcal{B} = \{(T_i, I_i, P_i)\}_{i=1}^{b}, \tag{1}$$

the three modality-specific encoders produce unit-normalized representations

$$\mathbf{t}_i = \frac{E_{\text{text}}(T_i)}{\|E_{\text{text}}(T_i)\|_2}, \qquad \mathbf{v}_i = \frac{E_{\text{img}}(I_i)}{\|E_{\text{img}}(I_i)\|_2}, \qquad \mathbf{p}_i = \frac{E_{\text{pc}}(P_i)}{\|E_{\text{pc}}(P_i)\|_2}, \tag{2}$$

where $\mathbf{t}_i, \mathbf{v}_i, \mathbf{p}_i \in \mathbb{R}^d$, with d denoting the dimensionality of the shared embedding space.

Pairwise contrastive learning. The conventional pairwise formulation optimizes three independent bidirectional contrastive objectives:

$$\mathcal{L}_{\text{pair}} = \frac{1}{3}\left(\mathcal{L}^{\text{pair}}_{T\leftrightarrow I} + \mathcal{L}^{\text{pair}}_{T\leftrightarrow P} + \mathcal{L}^{\text{pair}}_{I\leftrightarrow P}\right). \tag{3}$$

For example, the text–image objective is

$$\mathcal{L}^{\text{pair}}_{T\leftrightarrow I} = \frac{1}{2}\left(\mathcal{L}^{\text{pair}}_{T\to I} + \mathcal{L}^{\text{pair}}_{I\to T}\right), \tag{4}$$

where the text-to-image term follows the standard CLIP objective (Radford et al., 2021):

$$\mathcal{L}^{\text{pair}}_{T\to I} = -\frac{1}{b}\sum_{i=1}^{b}\log\frac{\exp\left(\mathbf{t}_i^\top \mathbf{v}_i/\tau\right)}{\sum_{j=1}^{b}\exp\left(\mathbf{t}_i^\top \mathbf{v}_j/\tau\right)}. \tag{5}$$

Here, τ denotes the temperature parameter. For a fixed text anchor $\mathbf{t}_i$, Eq. 5 contrasts the matched image $\mathbf{v}_i$ against $b-1$ mismatched image candidates, yielding b candidates in total. The reverse direction and the other modality pairs are defined analogously. We use this formulation as the matched pairwise baseline in our experiments.

Tensor contrastive learning. CTP instead constructs a joint candidate space over complete text–image–point-cloud tuples. We define a similarity tensor

$$\mathbf{S} \in \mathbb{R}^{b\times b\times b}, \qquad S_{ijk} = s(\mathbf{t}_i, \mathbf{v}_j, \mathbf{p}_k), \tag{6}$$

where $s(\cdot,\cdot,\cdot)$ denotes a tuple similarity scorer. The indices i, j, and k independently index the text, image, and point-cloud samples in the mini-batch, respectively. Each entry S_{ijk} therefore represents the similarity of one text–image–point-cloud tuple. The diagonal entries S_{iii} correspond to matched triplets, while the remaining entries form tuple-level negative candidates with different degrees of cross-modal matching.

Analogous to pairwise contrastive learning, we define a directional tensor loss by fixing one modality sample as the anchor. Fixing the text sample $\mathbf{t}_i$ leaves the image and point-cloud indices free, forming a two-dimensional candidate plane $\mathbf{S}_{i,:,:} \in \mathbb{R}^{b\times b}$. The text-anchored tensor loss over the complete candidate plane is

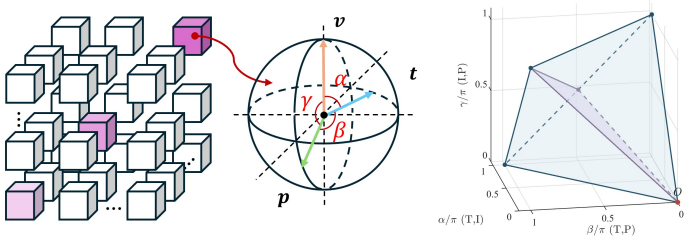
$$\mathcal{L}^{\text{tensor}}_{T\to(I,P)} = -\frac{1}{b}\sum_{i=1}^{b}\log\frac{\exp\left(S_{iii}/\tau\right)}{\sum_{j=1}^{b}\sum_{k=1}^{b}\exp\left(S_{ijk}/\tau\right)}. \tag{7}$$

For each text anchor, Eq. 7 contrasts the matched tuple against all b^2 image–point-cloud combinations in the corresponding plane. In comparison, the pairwise text-to-image objective in Eq. 5 contains only b image candidates for each text anchor.

The other two anchor directions are defined analogously by fixing the image or point-cloud sample. The complete tensor contrastive objective averages the three directional losses:

$$\mathcal{L}_{\text{tensor}} = \frac{1}{3}\left(\mathcal{L}^{\text{tensor}}_{T\to(I,P)} + \mathcal{L}^{\text{tensor}}_{I\to(T,P)} + \mathcal{L}^{\text{tensor}}_{P\to(T,I)}\right). \tag{8}$$

The key distinction between Eqs. 3 and 8 lies in their contrastive candidate spaces. Fixing one modality in a pairwise objective leaves one sample index to enumerate, resulting in b candidates. In the tensor objective, fixing one modality leaves two sample indices to vary jointly, resulting in b^2 multimodal tuple candidates. Pairwise contrastive learning therefore normalizes candidates from one modality at a time, whereas tensor contrastive learning jointly normalizes combinations of the other two modalities.



(a) Feature space (b) Angle space

Figure 3: Geometric interpretation of the triplet angle space. (a) Feature space: three modality embeddings in a similarity-tensor tuple define three pairwise angles on the unit hypersphere. (b) Angle space: the three angles form a tetrahedral feasible region. The light purple plane corresponds to $\beta = \gamma = \theta$, and the point O denotes $\alpha = \beta = \gamma = 0$, where all three modality embeddings coincide.

Importantly, this distinction is independent of the specific tuple scorer $s(\cdot, \cdot, \cdot)$. The similarity tensor determines *which multimodal combinations are contrasted*, whereas the scorer determines *how each tuple is scored*. We discuss the design of the tuple scorer in the next section, followed by the treatment of partial-match negatives in the tensor candidate space.

3.2 Similarity Scorer

Each entry of the similarity tensor corresponds to three unit-normalized modality representations, $\mathbf{t}_i$, $\mathbf{v}_j$, and $\mathbf{p}_k$. Their relative geometry can be characterized by three pairwise angles,

$$\alpha_{ij} = \arccos(\mathbf{t}_i^\top \mathbf{v}_j), \qquad \beta_{ik} = \arccos(\mathbf{t}_i^\top \mathbf{p}_k), \qquad \gamma_{jk} = \arccos(\mathbf{v}_j^\top \mathbf{p}_k). \tag{9}$$

As shown in Fig. 3, each multimodal tuple can therefore be represented by an angle configuration (α, β, γ) in a three-dimensional angle space. The geometrically feasible configurations form a tetrahedral domain.

Joint contrastive learning requires a scalar similarity score for each multimodal tuple. For a geometric scorer, this can be expressed as a mapping from the triplet angle configuration to a scalar,

$$S_{ijk} = f(\alpha_{ij}, \beta_{ik}, \gamma_{jk}). \tag{10}$$

Since this mapping compresses a three-dimensional geometric configuration into a single scalar, different angle configurations may receive the same similarity score. The choice of f therefore determines how tuple geometries are ranked and which configurations are favored during joint contrastive learning.

In CTP, we adopt a simple additive cosine scorer,

$$\begin{aligned} S_{ijk}^{\mathrm{CTP}} &= \mathbf{t}_i^\top \mathbf{v}_j + \mathbf{t}_i^\top \mathbf{p}_k + \mathbf{v}_j^\top \mathbf{p}_k \\ &= \cos\alpha_{ij} + \cos\beta_{ik} + \cos\gamma_{jk}. \end{aligned} \tag{11}$$

Substituting Eq. 11 into Eq. 7, the loss for a fixed text anchor $\mathbf{t}_i$ can be written as

$$\ell_i = \log \sum_{j=1}^{b} \sum_{k=1}^{b} \exp\left(\frac{S_{ijk}^{\mathrm{CTP}} - S_{iii}^{\mathrm{CTP}}}{\tau} \right). \tag{12}$$

This loss encourages the matched text–image–point-cloud tuple to obtain a higher joint similarity than competing tuples in the same anchor plane. Although mapping the three-dimensional angle configuration (α, β, γ) to a scalar similarity score can assign the same value to different geometric configurations, the optimum of the CTP scorer is unique in the angle space. Since each pairwise cosine similarity is upper bounded by 1, S_{ijk}^{CTP} reaches its maximum value 3 only when $\alpha_{ij} = \beta_{ik} = \gamma_{jk} = 0$, corresponding to the point O in Fig. 3, where $\mathbf{t}_i = \mathbf{v}_j = \mathbf{p}_k$. Thus, the cosine-sum scorer provides a simple geometric bias toward multimodal co-location despite the many-to-one mapping from angle configurations to scalar scores.

Alternative tuple scorers impose different notions of multimodal similarity. Symile (Saporta et al., 2024) uses a multilinear inner product to model joint statistical dependence and is not determined solely by the three pairwise angles. TRIANGLE (Cicchetti et al., 2025a) and GRAM (Cicchetti et al., 2025b) instead use geometric measures based on triangle area and Gram volume, respectively, which admit additional non-coincident optimum configurations. Detailed formulations and geometric comparisons of these scorers are provided in Appendix C.1.

3.3 Tensor Contrastive Loss

As shown in Eq. 7, fixing one modality as the anchor leaves two sample indices free, forming a two-dimensional candidate plane in the similarity tensor. The cross-entropy loss compares the matched tuple against all b^2 tuples in this plane through the softmax denominator. Among these candidates are *partial-match negatives*: tuples that are negative as a whole but still contain a matched modality pair involving the anchor.

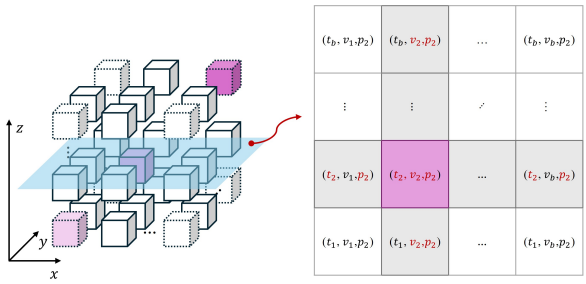


Figure 4: Masking partial-match negatives in an anchor plane. For a fixed point-cloud anchor $\mathbf{p}_2$, the center cell represents the matched tuple $(\mathbf{t}_2, \mathbf{v}_2, \mathbf{p}_2)$. The highlighted row and column contain partial-match negatives that preserve one matched pair involving the anchor and are removed by CTP. The remaining candidates are retained for contrastive normalization.

Consider a candidate plane with a fixed point-cloud anchor $\mathbf{p}_k$, where each entry S_{ijk} corresponds to the tuple $(\mathbf{t}_i, \mathbf{v}_j, \mathbf{p}_k)$. The matched tuple is given by $i = j = k$. When $i = k \neq j$, the candidate is a tuple-level negative but still contains the matched text–point pair $(\mathbf{t}_k, \mathbf{p}_k)$. Likewise, when $j = k \neq i$, the candidate contains the matched image–point pair $(\mathbf{v}_k, \mathbf{p}_k)$. These candidates form the anchor-associated row and column shown in Fig. 4, and we refer to them as *partial-match negatives*.

To analyze how partial-match negatives contribute to the tensor loss, consider the matched text–point similarity $a_k = \mathbf{t}_k^\top \mathbf{p}_k$ for a fixed point-cloud anchor $\mathbf{p}_k$. We define the softmax probability of candidate $(\mathbf{t}_i, \mathbf{v}_j, \mathbf{p}_k)$ within the corresponding anchor plane as

$$\pi_{ij}^P = \frac{\exp\left(S_{ijk}^{\mathrm{CTP}}/\tau\right)}{\sum_{m=1}^{b}\sum_{n=1}^{b}\exp\left(S_{mnk}^{\mathrm{CTP}}/\tau\right)}. \tag{13}$$

Viewing the pairwise similarities in the CTP scorer as intermediate scalar variables, let ℓ_k^P denote the loss for the fixed point-cloud anchor $\mathbf{p}_k$. The matched text–point similarity a_k appears in both the matched score S_{kkk}^{CTP} and every row-wise partial-match score S_{kjk}^{CTP} with $j \neq k$. Holding the remaining similarity terms fixed, the partial derivative of the loss with respect to a_k is

$$\frac{\partial \ell_k^P}{\partial a_k} = \frac{1}{\tau}\left(\pi_{kk}^P - 1 + \sum_{j\neq k}\pi_{kj}^P\right). \tag{14}$$

Here, $\pi_{kk}^P - 1$ is the contribution associated with the matched tuple, whereas $\sum_{j\neq k}\pi_{kj}^P$ arises from the row-wise partial-match negatives that share the same matched text–point similarity. Since the former is non-positive and the latter is non-negative, the partial-match terms partially offset the alignment signal associated with the matched tuple. The same structure arises for the matched image–point similarity and the corresponding column. A detailed derivation is provided in Appendix C.2.

Motivated by this analysis, CTP removes the anchor-sharing partial-match negatives while retaining the matched tuple and all remaining candidates.

4 EXPERIMENTS

4.1 SETUP

4.1.1 DATASETS

We conduct our main experiments on object-level text–image–point-cloud triplets from nuScenes (Caesar et al., 2019) and ULIP-ShapeNet (Xue et al., 2024), representing real-world automotive LiDAR and dense synthetic 3D shapes, respectively.

For nuScenes, we construct triplets from the official training and validation splits. For each reference object, LiDAR observations from up to 10 sweeps are aggregated into an object-level point cloud and paired with the corresponding image crop. We use Qwen3-VL-8B-Instruct (Bai et al., 2025) to generate pseudo-captions from the cropped images as textual inputs for alignment training. The aggregated point-cloud cohort requires at least 64 unique points per object, resulting in 29,917 training and 10,711 evaluation triplets. To assess cross-dataset generalization in real-world LiDAR, we additionally evaluate nuScenes-trained models on KITTI (Geiger et al., 2012) and the Waymo Open Dataset (Sun et al., 2020), without target-domain training. Additional details are provided in Appendix A.

ULIP-ShapeNet contains 52,460 prepared objects across 55 categories, each associated with an 8,192-point cloud, rendered images, and captions. We construct a custom category-stratified 80/20 split, yielding 41,968 training and 10,492 evaluation objects. Both training and evaluation use the first stored image view, its first associated caption, and all 8,192 points.

4.1.2 IMPLEMENTATION DETAILS

We use pretrained CLIP ViT-B/32 text and image encoders, where the text encoder is a Transformer-based language model (Vaswani et al., 2017) and the image encoder is a ViT (Dosovitskiy, 2020). For point clouds, we use a randomly initialized LitePT-S (Yue et al., 2026) encoder. A linear projection maps the point-cloud representation into the shared 512-dimensional embedding space, and all modality embeddings are ℓ_2-normalized before similarity computation. All three encoders are jointly optimized during training. We train all methods for 50 epochs with a batch size of 128 using AdamW (Loshchilov & Hutter, 2017) with a weight decay of 0.05. The initial learning rates are 10^{-5} for the pretrained text and image encoders and 5×10^{-4} for the point encoder and projection layer, followed by a 10% linear warm-up and cosine decay. Further implementation details are provided in Appendix B.

4.1.3 ZERO-SHOT CLASSIFICATION AND RETRIEVAL

As shown in Fig. 2, zero-shot classification uses class prompts as text candidates and jointly compares them with the image–point-cloud representation. For each dataset, we predefine the class vocabulary and encode one text prompt per class using the template `"This is a {CLASS}"`. Given an evaluation image–point-cloud pair $(\mathbf{v}_q, \mathbf{p}_q)$, all class-text representations $\mathbf{t}_c$ are scored and the class with the highest score is selected. For CTP and the pairwise baseline, joint classification uses

$$s_c = \frac{1}{2}\mathbf{v}_q^\top \mathbf{t}_c + \frac{1}{2}\mathbf{p}_q^\top \mathbf{t}_c. \tag{15}$$

For CTP, this produces the same class ranking as the full cosine-sum tuple scorer because $\mathbf{v}_q^\top \mathbf{p}_q$ is constant across candidate classes. Symile, GRAM, and TRIANGLE are evaluated using their corresponding native scorers. We report overall accuracy and mean per-class accuracy (mAcc).

For joint retrieval, each evaluation sample provides a query consisting of its caption and image, $(\mathbf{t}_q, \mathbf{v}_q)$, and all point-cloud embeddings $\{\mathbf{p}_k\}$ in the fixed evaluation pool are ranked. For CTP and the pairwise baseline, the retrieval score is

$$r_{qk} = (\mathbf{t}_q + \mathbf{v}_q)^\top \mathbf{p}_k, \tag{16}$$

which is equivalent in ranking to the full cosine-sum tuple scorer because $\mathbf{t}_q^\top \mathbf{v}_q$ is constant with respect to candidate k. The other methods use their corresponding native scorers. A retrieval is considered correct if at least one point cloud with the same object identity as the query appears within the top K ranked candidates. We report Recall@K for $K \in \{1, 5, 10\}$.

Table 1: Zero-shot joint classification results (%) on the nuScenes evaluation set. Car, Truck, and Bus are merged into Vehicle. Ped., C.V., Motor., and T.C. denote Pedestrian, Construction Vehicle, Motorcycle, and Traffic Cone, respectively.

Method	mAcc	Vehicle	Ped.	Bicycle	Trailer	C.V.	Motor.	Barrier	T.C.
ULIP-style (Xue et al., 2023)	71.07	15.55	57.92	95.69	84.51	76.34	90.60	69.18	78.79
Symile (Saporta et al., 2024)	43.11	21.31	30.07	1.72	77.10	55.91	97.44	50.75	10.61
TRIANGLE (Cicchetti et al., 2025a)	65.75	17.29	49.27	93.10	86.20	78.49	94.87	48.40	58.33
GRAM (Cicchetti et al., 2025b)	67.19	42.52	29.07	95.69	88.55	72.04	81.20	55.49	72.98
CTP (Ours)	**74.26**	34.19	61.36	97.41	85.19	78.49	97.44	71.53	68.43

Table 2: Text–image-to-point-cloud instance retrieval (%) on nuScenes and ULIP-ShapeNet.

Method	nuScenes			ULIP-ShapeNet		
	R@1	R@5	R@10	R@1	R@5	R@10
ULIP-style (Xue et al., 2023)	3.82	13.52	20.72	15.65	40.83	55.06
Symile (Saporta et al., 2024)	0.49	2.66	4.92	3.55	14.74	25.30
TRIANGLE (Cicchetti et al., 2025a)	6.57	20.04	30.24	37.70	68.72	79.61
GRAM (Cicchetti et al., 2025b)	4.82	15.90	24.60	28.85	60.04	73.09
CTP (Ours)	**7.44**	**22.42**	**32.18**	**38.44**	**69.06**	**80.13**

4.2 Results and Analysis

Table 1 reports zero-shot joint classification results on the nuScenes triplet dataset. We use a ULIP-style pairwise baseline under the same training setting as CTP: all three encoders are trainable, and the loss is defined in Eq. 3. Real-world LiDAR point clouds are sparse and often incomplete, making their alignment with image and text representations particularly challenging. The pairwise baseline achieves an mAcc of 71.07%, outperforming Symile, GRAM, and TRIANGLE, while CTP obtains the highest mAcc of 74.26%.

Among the joint methods, Symile achieves the lowest classification mAcc on nuScenes. Its MIP scorer is designed to capture joint statistical dependence rather than geometric co-location of modality representations, leading to a different inductive bias from CTP in the text–image–point-cloud alignment setting. GRAM and TRIANGLE achieve higher mAcc than Symile but remain below CTP in this setting. As discussed in Sec. 3.2, their area- and volume-based geometric measures admit non-coincident optimum configurations, whereas the cosine-sum scorer used by CTP has a unique optimal angle configuration at multimodal co-location. However, the existence of these additional optima does not imply that training necessarily converges to such degenerate configurations. Indeed, the competitive performance of GRAM and TRIANGLE indicates that both methods can still learn meaningful multimodal alignment in the high-dimensional embedding space. We therefore interpret these additional optima as providing a weaker geometric bias toward co-location rather than constituting a failure mode. This interpretation is consistent with the moderate performance differences observed in our experiments.

Table 2 further evaluates instance-level multimodal alignment through text–image-to-point-cloud retrieval. On nuScenes, CTP achieves the highest retrieval performance among the compared methods, reaching 7.44% R@1, 22.42% R@5, and 32.18% R@10.

A larger separation between pairwise and joint geometric methods is observed on ULIP-ShapeNet. The pairwise baseline obtains 15.65% R@1, 40.83% R@5, and 55.06% R@10, whereas GRAM, TRIANGLE, and CTP all substantially improve instance retrieval. CTP achieves the best performance, reaching 38.44% R@1, 69.06% R@5, and 80.13% R@10. In contrast, Symile remains below the pairwise baseline on both datasets, consistent with its different statistical rather than geometric notion of multimodal similarity.

Among the geometric joint methods, the retrieval performance follows the same ordering on both datasets: CTP, TRIANGLE, and GRAM. This trend is consistent with the geometric analysis. The cosine-sum scorer used by CTP uniquely favors co-location of the three modality embeddings, while

the triangle-area scorer admits additional degenerate configurations and the Gram-volume scorer has an even broader set of zero-volume configurations corresponding to linearly dependent embeddings.

The relative gains of the joint geometric methods are also larger on ULIP-ShapeNet than on nuScenes. This observation is consistent with the hypothesis that richer 3D point clouds facilitate joint multimodal alignment, whereas sparse and incomplete LiDAR point clouds make the alignment problem more challenging and reduce the performance gap between pairwise and joint methods.

Table 3: Cross-dataset joint classification mAcc (%) on KITTI and Waymo after training on nuScenes.

Method	KITTI	Waymo
ULIP-style (Xue et al., 2023)	50.59	57.76
Symile (Saporta et al., 2024)	34.87	42.67
TRIANGLE (Cicchetti et al., 2025a)	60.47	46.71
GRAM (Cicchetti et al., 2025b)	52.47	49.26
CTP (Ours)	**64.25**	**65.07**

To evaluate cross-dataset generalization, we directly apply the nuScenes-trained models to KITTI and Waymo without target-domain training or adaptation. Both evaluation sets are constructed following the same object-level text–image–point-cloud triplet pipeline as nuScenes.

As shown in Table 3, all compared methods retain zero-shot classification capability under this domain shift. Among them, CTP achieves the highest mAcc on both datasets, reaching 64.25% on KITTI and 65.07% on Waymo. These results show that the multimodal representations learned on nuScenes remain effective across different real-world LiDAR domains. The consistent performance of CTP on both the denser KITTI point clouds and the sparser Waymo point clouds further suggests robustness to variations in 3D point-cloud density and dataset distribution. Overall, the observed cross-dataset transferability supports the potential of the learned point-cloud encoder as a general 3D representation backbone for downstream multimodal applications.

Figure 5 compares joint classification performance with and without masking partial-match negatives. For the geometric joint scorers, TRIANGLE, GRAM, and CTP, masking consistently improves mAcc on both nuScenes and ULIP-ShapeNet. This observation supports the motivation for excluding partial-match negatives, which are negative at the tuple level but still preserve a valid modality pair involving the anchor.

In contrast, the effect of masking on Symile is less consistent across datasets, suggesting that the influence of partial-match negatives depends on the underlying tuple similarity function. Overall, these results indicate that masking is particularly beneficial for the geometric joint alignment objectives considered in this work.

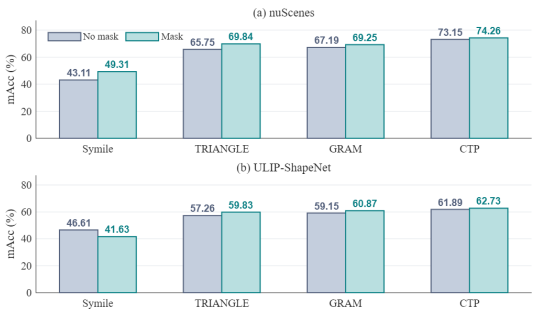


Figure 5: Effect of masking partial-match negatives on zero-shot joint classification mAcc (%) on nuScenes and ULIP-ShapeNet. No mask uses all tuple candidates, whereas Mask removes anchor-sharing partial-match negatives.

5 Conclusion

We presented CTP for joint contrastive alignment of text, image, and point-cloud representations. Our experiments on nuScenes and ULIP-ShapeNet show that CTP achieves strong category-level alignment and instance-level cross-modal retrieval compared with a matched pairwise baseline. We further show that the cosine-sum scorer provides a direct geometric bias toward multimodal co-location, while alternative area- and volume-based scorers admit additional non-coincident optima. Finally, we identify partial-match negatives in joint contrastive learning and find that masking them consistently improves the performance of the geometric joint objectives considered in this work. These results highlight the importance of scorer geometry and candidate construction for joint language–image–3D representation learning.

AI USE STATEMENT

We used generative AI tools during the preparation of this work. ChatGPT was used to assist with language editing, improving the clarity and organization of the manuscript, and generating or modifying portions of the experimental code. All AI-assisted manuscript revisions and code modifications were independently reviewed and verified by the authors. In addition, we used Qwen3-VL-8B-Instruct to generate pseudo-captions from cropped object images and category labels for the nuScenes, KITTI, and Waymo triplet datasets, as described in Appendix A.1. The pseudo-captions were generated using a fixed pipeline. The authors take full responsibility for the final manuscript, experimental methodology, code, results, and all AI-assisted content.

REPRODUCIBILITY STATEMENT

We provide the information necessary to reproduce our experiments in the Appendix and supplementary material. Appendix A details the construction of text–image–point-cloud triplets from the autonomous-driving datasets and the corresponding data processing procedures. Appendix B provides the complete training configurations and hyperparameters, while Appendix C contains additional theoretical derivations and analyses supporting the proposed method. The supplementary material includes the implementation code for triplet construction, pseudo-caption generation, model training, and evaluation, together with the corresponding configuration files. These materials are intended to facilitate reproduction of the reported results.

REFERENCES

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems*, 35:23716–23736, 2022.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

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. *arXiv preprint arXiv:1903.11027*, 2019.

Boyuan Chen, Zhuo Xu, Sean Kirmani, Brain Ichter, Dorsa Sadigh, Leonidas Guibas, and Fei Xia. Spatialvlm: Endowing vision-language models with spatial reasoning capabilities. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14455–14465, 2024a.

Long Chen, Oleg Sinavski, Jan Hünermann, Alice Karnsund, Andrew James Willmott, Danny Birch, Daniel Maund, and Jamie Shotton. Driving with llms: Fusing object-level vector modality for explainable autonomous driving. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 14093–14100. IEEE, 2024b.

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*, pp. 7020–7030, 2023.

Giordano Cicchetti, Eleonora Grassucci, and Danilo Comminiello. A triangle enables multimodal alignment beyond cosine similarity, 2025a. URL https://arxiv.org/abs/2509.24734.

Giordano Cicchetti, Eleonora Grassucci, Luigi Sigillo, and Danilo Comminiello. Gramian multimodal representation learning and alignment, 2025b. URL https://arxiv.org/abs/2412.11959.

Alexey Dosovitskiy. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Benjamin Elizalde, Soham Deshmukh, Mahmoud Al Ismail, and Huaming Wang. Clap learning audio concepts from natural language supervision. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5. IEEE, 2023.

Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

Rohit Girdhar, Alaaeldin El-Nouby, Zhuang Liu, Mannat Singh, Kalyan Vasudev Alwala, Armand Joulin, and Ishan Misra. Imagebind: One embedding space to bind them all. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 15180–15190, 2023.

Andrey Guzhov, Federico Raue, Jörn Hees, and Andreas Dengel. Audioclip: Extending clip to image, text and audio. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 976–980. IEEE, 2022.

Deepti Hegde, Jeya Maria Jose Valanarasu, and Vishal Patel. Clip goes 3d: Leveraging prompt tuning for language grounded 3d recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2028–2038, 2023.

Georg Hess, Adam Tonderski, Christoffer Petersson, Kalle Åström, and Lennart Svensson. Lidarclip or: How i learned to talk to point clouds. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 7438–7447, 2024.

Yining Hong, Haoyu Zhen, Peihao Chen, Shuhong Zheng, Yilun Du, Zhenfang Chen, and Chuang Gan. 3d-llm: Injecting the 3d world into large language models. *Advances in Neural Information Processing Systems*, 36:20482–20494, 2023.

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*, pp. 22157–22167, 2023.

Xin Huang, Eric M Wolff, Paul Vernaza, Tung Phan-Minh, Hongge Chen, David S Hayden, Mark Edmonds, Brian Pierce, Xinxin Chen, Pratik Elias Jacob, et al. Drivegpt: Scaling autoregressive behavior models for driving. *arXiv preprint arXiv:2412.14415*, 2024.

Jyh-Jing Hwang, Runsheng Xu, Hubert Lin, Wei-Chih Hung, Jingwei Ji, Kristy Choi, Di Huang, Tong He, Paul Covington, Benjamin Sapp, et al. Emma: End-to-end multimodal model for autonomous driving. *arXiv preprint arXiv:2410.23262*, 2024.

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 *International conference on machine learning*, pp. 4904–4916. PMLR, 2021.

Justin Kerr, Chung Min Kim, Ken Goldberg, Angjoo Kanazawa, and Matthew Tancik. Lerf: Language embedded radiance fields. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 19729–19739, 2023.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.

Fan Li, Zanyi Wang, Zeyi Huang, Guang Dai, Jingdong Wang, and Mengmeng Wang. Triclip-3d: A unified parameter-efficient framework for tri-modal 3d visual grounding based on clip. *arXiv preprint arXiv:2507.14904*, 2025.

Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International conference on machine learning*, pp. 12888–12900. PMLR, 2022.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pp. 19730–19742. PMLR, 2023.

Guibiao Liao, Jiankun Li, and Xiaoqing Ye. Vlm2scene: Self-supervised image-text-lidar learning with foundation models for autonomous driving scene understanding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 3351–3359, 2024.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916, 2023a.

Minghua Liu, Ruoxi Shi, Kaiming Kuang, Yinhao Zhu, Xuanlin Li, Shizhong Han, Hong Cai, Fatih Porikli, and Hao Su. Openshape: Scaling up 3d shape representation towards open-world understanding. *Advances in neural information processing systems*, 36:44860–44879, 2023b.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Yuheng Lu, Chenfeng Xu, Xiaobao Wei, Xiaodong Xie, Masayoshi Tomizuka, Kurt Keutzer, and Shanghang Zhang. Open-vocabulary point-cloud object detection without 3d annotation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 1190–1199, 2023.

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.

Xianzheng Ma, Yash Bhalgat, Brandon Smart, Shuai Chen, Xinghui Li, Jian Ding, Jindong Gu, Dave Zhenyu Chen, Songyou Peng, Jia-Wang Bian, et al. When llms step into the 3d world: A survey and meta-analysis of 3d tasks via multi-modal large language models. *arXiv preprint arXiv:2405.10255*, 2024.

Yukai Ma, Tiantian Wei, Naiting Zhong, Jianbiao Mei, Tao Hu, Licheng Wen, Xuemeng Yang, Botian Shi, and Yong Liu. Leapvad: A leap in autonomous driving via cognitive perception and dual-process thinking. *arXiv preprint arXiv:2501.08168*, 2025.

Chenbin Pan, Burhaneddin Yaman, Senem Velipasalar, and Liu Ren. Clip-bevformer: Enhancing multi-view image-based bev detector with ground truth flow. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15216–15225, 2024.

Songyou Peng, Kyle Genova, Chiyu Jiang, Andrea Tagliasacchi, Marc Pollefeys, Thomas Funkhouser, et al. Openscene: 3d scene understanding with open vocabularies. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 815–824, 2023.

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*, pp. 8748–8763. PmLR, 2021.

Adriel Saporta, Aahlad Puli, Mark Goldstein, and Rajesh Ranganath. Contrasting with symile: Simple model-agnostic representation learning for unlimited modalities, 2024. URL `https://arxiv.org/abs/2411.01053`.

Sai Shubodh, Mohammad Omama, Husain Zaidi, Udit Singh Parihar, and Madhava Krishna. Lip-loc: Lidar image pretraining for cross-modal localization. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 948–957, 2024.

Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, Vijay Vasudevan, Wei Han, Jiquan Ngiam, Hang Zhao, Aleksei Timofeev, Scott Ettinger, Maxim Krivokon, Amy Gao, Aditya Joshi, Yu Zhang, Jonathon Shlens, Zhifeng Chen, and Dragomir Anguelov. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

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.

Haowei Wang, Jiji Tang, Jiayi Ji, Xiaoshuai Sun, Rongsheng Zhang, Yiwei Ma, Minda Zhao, Lincheng Li, Zeng Zhao, Tangjie Lv, et al. Beyond first impressions: Integrating joint multi-modal cues for comprehensive 3d representation. In *Proceedings of the 31st ACM International Conference on Multimedia*, pp. 3403–3414, 2023a.

Xiao Wang, Guangyao Chen, Guangwu Qian, Pengcheng Gao, Xiao-Yong Wei, Yaowei Wang, Yonghong Tian, and Wen Gao. Large-scale multi-modal pre-trained models: A comprehensive survey. *Machine Intelligence Research*, 20(4):447–482, 2023b.

Yuanbin Wang, Shaofei Huang, Yulu Gao, Zhen Wang, Rui Wang, Kehua Sheng, Bo Zhang, and Si Liu. Transferring clip's knowledge into zero-shot point cloud semantic segmentation. In *Proceedings of the 31st ACM International Conference on Multimedia*, pp. 3745–3754, 2023c.

Shuo Xing, Chengyuan Qian, Yuping Wang, Hongyuan Hua, Kexin Tian, Yang Zhou, and Zhengzhong Tu. Openemma: Open-source multimodal model for end-to-end autonomous driving. In *Proceedings of the Winter Conference on Applications of Computer Vision*, pp. 1001–1009, 2025.

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. *arXiv preprint arXiv:2109.14084*, 2021.

Le Xue, Mingfei Gao, Chen Xing, Roberto Martín-Martín, Jiajun Wu, Caiming Xiong, Ran Xu, Juan Carlos Niebles, and Silvio Savarese. Ulip: Learning a unified representation of language, images, and point clouds for 3d understanding. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 1179–1189, 2023.

Le Xue, Ning Yu, Shu Zhang, Artemis Panagopoulou, Junnan Li, Roberto Martín-Martín, Jiajun Wu, Caiming Xiong, Ran Xu, Juan Carlos Niebles, and Silvio Savarese. Ulip-2: Towards scalable multimodal pre-training for 3d understanding, 2024. URL https://arxiv.org/abs/2305.08275.

Senqiao Yang, Jiaming Liu, Renrui Zhang, Mingjie Pan, Ziyu Guo, Xiaoqi Li, Zehui Chen, Peng Gao, Hongsheng Li, Yandong Guo, et al. Lidar-llm: Exploring the potential of large language models for 3d lidar understanding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 9247–9255, 2025.

Lewei Yao, Runhui Huang, Lu Hou, Guansong Lu, Minzhe Niu, Hang Xu, Xiaodan Liang, Zhenguo Li, Xin Jiang, and Chunjing Xu. Filip: Fine-grained interactive language-image pre-training. *arXiv preprint arXiv:2111.07783*, 2021.

Yuanwen Yue, Damien Robert, Jianyuan Wang, Sunghwan Hong, Jan Dirk Wegner, Christian Rupprecht, and Konrad Schindler. Litept: Lighter yet stronger point transformer, 2026. URL https://arxiv.org/abs/2512.13689.

Yihan Zeng, Chenhan Jiang, Jiageng Mao, Jianhua Han, Chaoqiang Ye, Qingqiu Huang, Dit-Yan Yeung, Zhen Yang, Xiaodan Liang, and Hang Xu. Clip2: Contrastive language-image-point pretraining from real-world point cloud data. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 15244–15253, 2023.

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*, pp. 8552–8562, 2022.

Junsheng Zhou, Jinsheng Wang, Baorui Ma, Yu-Shen Liu, Tiejun Huang, and Xinlong Wang. Uni3d: Exploring unified 3d representation at scale. In *International Conference on Learning Representations*, volume 2024, pp. 46766–46782, 2024.

Xingcheng Zhou, Xuyuan Han, Feng Yang, Yunpu Ma, and Alois C Knoll. Opendrivevla: Towards end-to-end autonomous driving with large vision language action model. *arXiv preprint arXiv:2503.23463*, 2025.

Bin Zhu, Bin Lin, Munan Ning, Yang Yan, Jiaxi Cui, HongFa Wang, Yatian Pang, Wenhao Jiang, Junwu Zhang, Zongwei Li, et al. Languagebind: Extending video-language pretraining to n-modality by language-based semantic alignment. *arXiv preprint arXiv:2310.01852*, 2023a.

Xiangyang Zhu, Renrui Zhang, Bowei He, Ziyu Guo, Ziyao Zeng, Zipeng Qin, Shanghang Zhang, and Peng Gao. Pointclip v2: Prompting clip and gpt for powerful 3d open-world learning. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 2639–2650, 2023b.

Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In *Conference on Robot Learning*, pp. 2165–2183. PMLR, 2023.

A Triplet Dataset

A.1 Triplet Construction

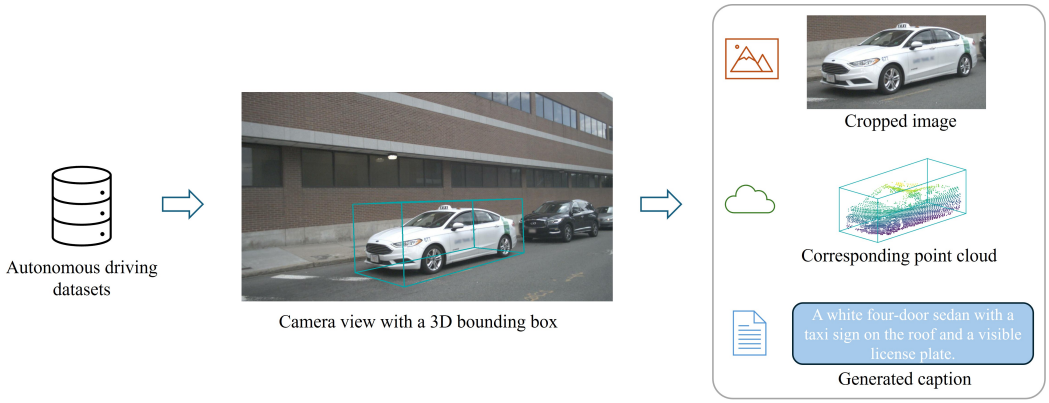


Figure 6: Overview of object-level triplet construction using a nuScenes example. For each annotated object, we extract the corresponding image crop and LiDAR point cloud and generate a pseudo-caption from the cropped image and object category. The point cloud shown here aggregates up to 10 LiDAR sweeps.

We construct object-level text–image–point-cloud triplets from nuScenes (Caesar et al., 2019), KITTI (Geiger et al., 2012), and the Waymo Open Dataset (Sun et al., 2020). As shown in Fig. 6, each annotated object is associated with a point-cloud segment, a corresponding image crop, and a textual description. The 3D annotations and camera–LiDAR correspondences establish the association between the image and point-cloud observations, while image-conditioned pseudo-captioning provides the textual modality.

For nuScenes, we construct the training and evaluation sets from the official training and validation scenes, respectively. Reference frames are selected at 2-second intervals. For each reference frame, we aggregate the current LiDAR scan and nine preceding scans. Using sensor calibration and ego poses, all scans are transformed into the reference ego coordinate system. We then extract the points inside each annotated 3D bounding box at the reference frame.

To reduce points near the bottom of an object box, we transform the extracted points into the box-local coordinate system using the full box rotation and retain points satisfying

$$z_{\text{local}} \geq -\frac{h}{2} + \delta, \qquad \delta = 0.1\,\text{m}, \tag{17}$$

where h denotes the box height. This operation removes a thin region at the box bottom. The final nuScenes cohort retains objects for which all ten scans are available and that contain at least 64 unique XYZ points after filtering.

For the image modality, we retain nuScenes annotations with visibility tokens of at least 2 and project each object's 3D bounding box into the available camera views. A view is considered valid when at least 80% of its projected 2D bounding rectangle lies inside the image. Among valid views, we select the one with the largest in-image projected area, using proximity to the image center to break ties. We extract the corresponding image crop with a 5-pixel margin, clipped to the image boundaries. The annotation visibility criterion and the image-boundary criterion are applied separately.

KITTI follows the same general triplet-construction procedure as nuScenes, but uses annotated track identities and frame-specific 3D boxes to align the same object across frames before aggregation. Figure 7 shows the effect of this temporal aggregation. We apply the same box-bottom exclusion and image-cropping settings as for nuScenes, and retain aggregated point clouds containing at least 50 points.

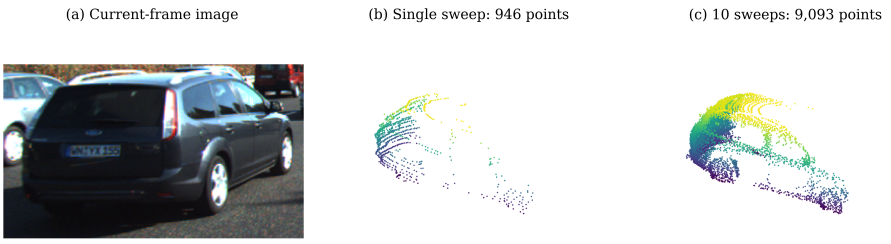


Figure 7: Temporal aggregation in KITTI. The single-frame object point cloud is substantially densified by aggregating up to 10 frames of the same tracked object.

Waymo differs in that we use single-frame object point clouds rather than temporal aggregation. LiDAR boxes are associated with projected camera boxes using the segment identifier, frame timestamp, and object identifier. We retain object point clouds containing at least 15 points. If an object is visible in multiple cameras within the same frame, each valid camera crop forms a separate triplet.

We use Qwen3-VL-8B-Instruct (Bai et al., 2025) to generate descriptive pseudo-captions for the nuScenes, KITTI, and Waymo triplet datasets. For each object, the captioning model is provided with the cropped RGB image and its object category. The same caption-generation procedure is applied to all three datasets.

For reproducibility, we provide the system and user prompts used for caption generation below. In the user prompt, {`category`} is replaced by the corresponding object label.

> **System prompt:**
>
> You generate one concise factual sentence about the specified road object. Follow every constraint exactly and do not add scene context.
>
> **User prompt:**
>
> Target object category: {`category`}.
>
> Describe only the target object in one factual sentence. Describe its visible shape, approximate scale, orientation, and distinctive structure. Mention color only when it is clearly visible. Do not mention the image, photo, scene, background, camera, or crop. Do not infer motion, speed, intent, or hidden attributes. If an attribute is uncertain, omit it.

A.2 Dataset Statistics

Figure 8 summarizes the category distributions of the autonomous-driving triplet datasets. The nuScenes dataset contains 29,917 training triplets and 10,711 evaluation triplets. For cross-dataset evaluation, the KITTI subset contains 764 triplets across eight categories, and the Waymo subset contains 34,287 triplets across four categories.

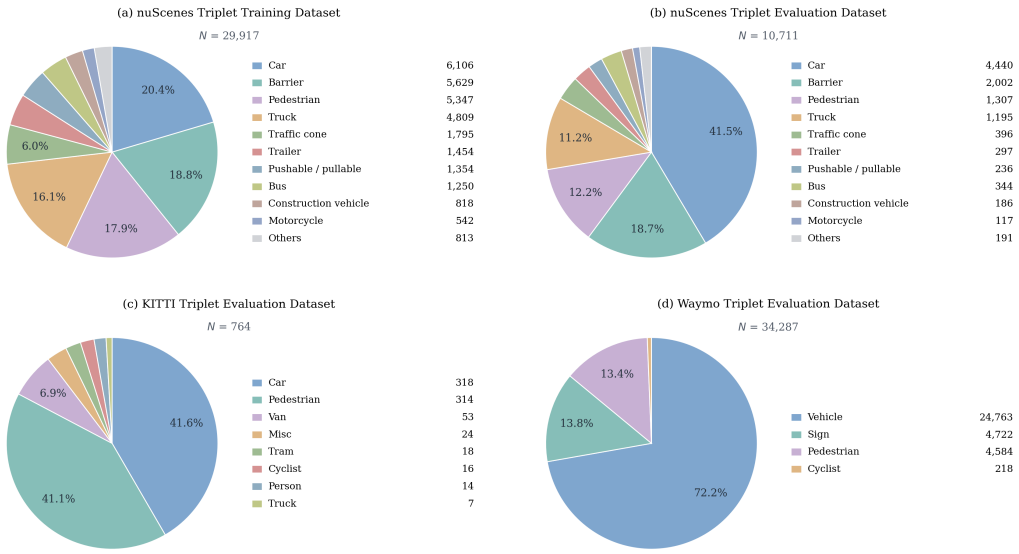


Figure 8: Category distributions of the constructed triplet datasets: (a) nuScenes training, (b) nuScenes evaluation, (c) KITTI evaluation, and (d) Waymo evaluation. N denotes the number of triplets, and legend values indicate category counts. For nuScenes, "Others" groups the Bicycle, Bicycle Rack, Debris, Emergency, and Animal categories for visualization.

A.3 Representative Triplets

Figures 9, 10, and 11 present selected triplets from the nuScenes, KITTI, and Waymo evaluation sets, respectively. These examples qualitatively show differences in point-cloud density and geometric coverage across the datasets. The selected nuScenes point clouds remain relatively sparse and incomplete despite temporal aggregation, whereas the KITTI examples exhibit denser and more complete geometric coverage. Waymo uses single-frame point clouds and, in the selected examples, shows sparser coverage than KITTI.

B Implementation Details

Each cropped RGB image is resized with bilinear interpolation so that its longer side is 224 pixels while preserving the aspect ratio, and is then centered on a black 224×224 canvas. Images are normalized using the CLIP channel statistics. The same deterministic preprocessing is used for training and evaluation. Text is tokenized using the CLIP tokenizer with a context length of 77 tokens, with longer descriptions truncated.

The point-cloud encoder uses only XYZ coordinates. For the driving datasets, each object point cloud is centered by subtracting its centroid while retaining its metric scale. ULIP-ShapeNet uses the normalized coordinates provided by the prepared assets. All stored points are retained without fixed-size sampling. Before entering LitePT-S, points within each voxel are averaged using a grid size of 0.05 in metric coordinates for the driving datasets and normalized coordinates for ULIP-ShapeNet. The encoded point features are mean-pooled before the linear projection into the shared embedding space.

Table 4 summarizes the main training configuration. All compared methods use the same input processing, encoder architectures, initialization scheme, and optimization settings. Unless otherwise stated, reported results use the final checkpoint after 50 epochs.

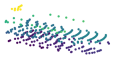
Category	Image	Point cloud	Caption
Car nuScenes		 N = 313	A silver car with a front-facing orientation, featuring a grille, headlights, and side mirrors.
Pedestrian nuScenes		 N = 156	A pedestrian with short hair, wearing a light-colored sleeveless top and dark shorts, stands upright with feet apart.
Bicycle nuScenes		 N = 137	A yellow bicycle with a step-through frame, two wheels, handlebars, and a seat is upright and positioned parallel to a curb.
Traffic cone nuScenes		 N = 104	A cylindrical traffic cone with two horizontal white reflective bands and an orange body stands upright.

Figure 9: Selected triplets from the nuScenes evaluation set. Each row shows the object category, corresponding image crop, point cloud, and pseudo-caption. N denotes the number of saved points.

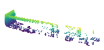
Category	Image	Point cloud	Caption
Car KITTI		 N = 3,679	A white station wagon with a rear hatch, four doors, and a visible roof rack.
Pedestrian KITTI		 N = 3,458	A standing human figure with a light-colored shirt and dark pants, viewed from the side, has a rectangular object on their back.
Cyclist KITTI		 N = 3,184	A cyclist wearing a white tank top and dark pants is seated on a bicycle with a black saddle and handlebars, positioned upright with the bike's frame and wheels visible.
Truck KITTI		 N = 12,047	A black truck with a rectangular cargo container and visible rear lights, oriented horizontally with a logo and text on its back.

Figure 10: Selected triplets from the KITTI evaluation set. Each row shows the object category, corresponding image crop, point cloud, and pseudo-caption. N denotes the number of saved points.

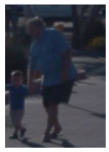
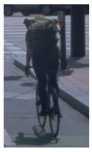
Category	Image	Point cloud	Caption
Vehicle Waymo		 N = 560	A dark-colored vehicle with a visible rear taillight is captured in a low-light setting, appearing as a silhouette with discernible geometric contours and wheels.
Pedestrian Waymo		 N = 130	A blurred adult and child, both wearing blue tops, are walking side-by-side on a paved surface under bright sunlight, with the adult appearing larger due to proximity.
Sign Waymo		 N = 68	A tall, rectangular blue sign with white text and symbols, mounted vertically, appears slightly blurred and is partially obscured by foliage in the background.
Cyclist Waymo		 N = 208	A cyclist wearing a dark jacket and carrying a large backpack rides a bicycle with thin tires and a visible chain, casting a distinct shadow on the paved road.

Figure 11: Selected triplets from the Waymo evaluation set. Each row shows the object category, corresponding image crop, point cloud, and pseudo-caption. N denotes the number of saved points.

Table 4: Implementation and training configuration used in the main experiments.

Setting	Configuration
Text encoder	CLIP ViT-B/32, pretrained and trainable
Image encoder	CLIP ViT-B/32, pretrained and trainable
Point encoder	LitePT-S, randomly initialized and trainable
Point projection	Linear, $504 \rightarrow 512$, with bias
Embedding dimension	512
Embedding normalization	ℓ_2 normalization
Training epochs	50
Global batch size	128
Optimizer	AdamW
Weight decay	0.05
CLIP learning rate	1×10^{-5}
Point/projector learning rate	5×10^{-4}
LR schedule	10% linear warm-up + cosine decay
Gradient clipping	Global norm 1.0
Mixed precision	BF16 encoder forward, FP32 scoring/loss
Logit-scale initialization	$1/0.07$
Logit-scale cap	1000
Random seed	42

C Additional Theoretical Analysis

C.1 Alternative Tuple Scorers

Different multimodal tuple scorers impose different inductive biases. Symile (Saporta et al., 2024), for example, uses the multilinear inner product (MIP),

$$S_{\text{MIP}}(\mathbf{t}, \mathbf{v}, \mathbf{p}) = \sum_{r=1}^{d} t_r v_r p_r. \tag{18}$$

Unlike the CTP scorer, MIP is not determined solely by the three pairwise angles (α, β, γ). A common rotation of the three embeddings preserves all three pairwise angles but can change the MIP value. It therefore captures a different notion of multimodal similarity from the angle-based geometric co-location considered by CTP.

We further consider the geometric properties of CTP and two alternatives inspired by TRIANGLE (Cicchetti et al., 2025a) and GRAM (Cicchetti et al., 2025b). Let

$$a = \cos\alpha, \qquad b = \cos\beta, \qquad c = \cos\gamma. \tag{19}$$

For unit-normalized embeddings, the CTP scorer satisfies $S_{\text{CTP}} = a + b + c$. The total squared pairwise distance is therefore

$$D^2 = \|\mathbf{t} - \mathbf{v}\|_2^2 + \|\mathbf{t} - \mathbf{p}\|_2^2 + \|\mathbf{v} - \mathbf{p}\|_2^2 = 6 - 2(a + b + c) = 6 - 2S_{\text{CTP}}. \tag{20}$$

Thus, maximizing the CTP scorer is equivalent to minimizing the total squared pairwise distance.

TRIANGLE measures the area of the triangle formed by the three embedding endpoints. Its squared area can be written as

$$A^2 = \frac{1}{4}\left[3 - 2(a + b + c) + 2(ab + ac + bc) - (a^2 + b^2 + c^2)\right]. \tag{21}$$

GRAM is based on the volume spanned by the three embedding vectors, whose squared volume is given by the determinant of their Gram matrix,

$$V^2 = 1 + 2abc - a^2 - b^2 - c^2. \tag{22}$$

In training, TRIANGLE and GRAM use the negative area and negative volume, respectively, as tuple scores, so maximizing the contrastive score corresponds to minimizing the geometric measure. Specifically, we use $s_{\text{TRIANGLE}} = -\sqrt{\max(A^2, 0) + \epsilon}$ and $s_{\text{GRAM}} = -\sqrt{\max(V^2, 0) + \epsilon}$ with $\epsilon = 10^{-8}$ for numerical stability.

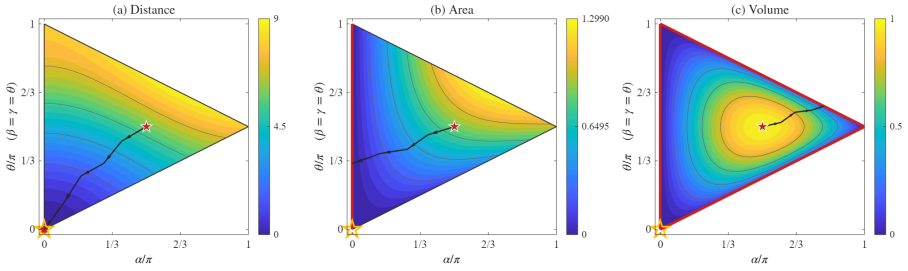


Figure 12: Geometric tuple measures on the $\beta = \gamma = \theta$ slice of the triplet angle space in Fig. 3. The yellow star denotes multimodal co-location, red lines denote additional minima, and black curves illustrate possible directions of decreasing measure. The color bars indicate the value of the corresponding geometric measure at each angle configuration.

Figure 12 provides a two-dimensional view of the triplet angle space by visualizing the light-purple plane in Fig. 3, corresponding to the slice $\beta = \gamma = \theta$. Our desired alignment configuration is the co-location of all three embeddings, corresponding to $(\alpha, \theta) = (0, 0)$ and marked by the yellow star. The red lines indicate additional configurations at which the corresponding geometric measure attains its minimum. Starting from an illustrative configuration marked by the red star, the black

curves show possible directions of decreasing measure. For the total squared pairwise distance, the unique minimum is the co-location configuration, so decreasing the measure leads toward the yellow star. In contrast, area and volume admit additional non-coincident minima, allowing the measure to decrease toward configurations in which the three embeddings are not co-located.

These properties show the different inductive biases of the tuple scorers considered in our experiments. The cosine-sum scorer directly favors pairwise co-location, whereas the area- and volume-based scorers also admit non-coincident optima. Their empirical behavior is compared in Sec. 4.

C.2 Gradient Analysis of Partial-Match Negatives

We provide the derivation of Eq. 14. Consider a fixed point-cloud anchor $\mathbf{p}_k$. The corresponding per-anchor tensor loss is

$$\ell_k^P = -\log \frac{\exp\left(S_{kkk}^{\mathrm{CTP}}/\tau\right)}{\sum_{i=1}^{b}\sum_{j=1}^{b}\exp\left(S_{ijk}^{\mathrm{CTP}}/\tau\right)} = -\frac{S_{kkk}^{\mathrm{CTP}}}{\tau} + \log Z_k^P, \tag{23}$$

where

$$Z_k^P = \sum_{i=1}^{b}\sum_{j=1}^{b}\exp\left(S_{ijk}^{\mathrm{CTP}}/\tau\right). \tag{24}$$

Let

$$a_k = \mathbf{t}_k^\top \mathbf{p}_k \tag{25}$$

denote the matched text–point similarity. Under the additive CTP scorer,

$$S_{ijk}^{\mathrm{CTP}} = \mathbf{t}_i^\top \mathbf{v}_j + \mathbf{t}_i^\top \mathbf{p}_k + \mathbf{v}_j^\top \mathbf{p}_k, \tag{26}$$

the term a_k appears in every score with $i = k$, including the matched tuple S_{kkk}^{CTP} and the row-wise partial-match negatives S_{kjk}^{CTP} for $j \neq k$. Treating the pairwise similarities as intermediate scalar variables and holding the remaining terms fixed,

$$\frac{\partial S_{ijk}^{\mathrm{CTP}}}{\partial a_k} = \begin{cases} 1, & i = k, \\ 0, & i \neq k. \end{cases} \tag{27}$$

Differentiating Eq. 23 with respect to a_k gives

$$\frac{\partial \ell_k^P}{\partial a_k} = -\frac{1}{\tau} + \frac{1}{\tau Z_k^P}\sum_{j=1}^{b}\exp\left(S_{kjk}^{\mathrm{CTP}}/\tau\right) \tag{28}$$

$$= \frac{1}{\tau}\left(-1 + \sum_{j=1}^{b}\pi_{kj}^P\right), \tag{29}$$

where

$$\pi_{ij}^P = \frac{\exp\left(S_{ijk}^{\mathrm{CTP}}/\tau\right)}{Z_k^P} \tag{30}$$

is the softmax probability of candidate $(\mathbf{t}_i, \mathbf{v}_j, \mathbf{p}_k)$ within the point-cloud-anchored plane. Separating the matched tuple from the row-wise partial-match negatives yields

$$\frac{\partial \ell_k^P}{\partial a_k} = \frac{1}{\tau}\left(\pi_{kk}^P - 1 + \sum_{j\neq k}\pi_{kj}^P\right), \tag{31}$$

which gives Eq. 14. The term $\pi_{kk}^P - 1$ is associated with the matched tuple, whereas the nonnegative summation arises from partial-match negatives that contain the same matched text–point similarity a_k and therefore partially offsets this contribution.

An analogous derivation applies to the matched image–point similarity $\mathbf{v}_k^\top \mathbf{p}_k$, for which the corresponding partial-match contribution arises from the column $j = k$.

D Additional Results

D.1 Effect of the Number of Sampled Candidates

As discussed in Fig. 4, each anchor is contrasted against b^2 tuple candidates. Row-and-column masking leaves $b^2 - 2b + 2$ candidates, including the positive tuple. We retain the positive tuple and randomly sample $K - 1$ negatives from the remaining set, yielding $K \in \{128, 512, 2048, 8192\}$ candidates per anchor. All runs use a batch size of 128, with other training settings unchanged from the main experiments. Joint text–image-to-point-cloud retrieval is evaluated every five epochs over a fixed pool instances.

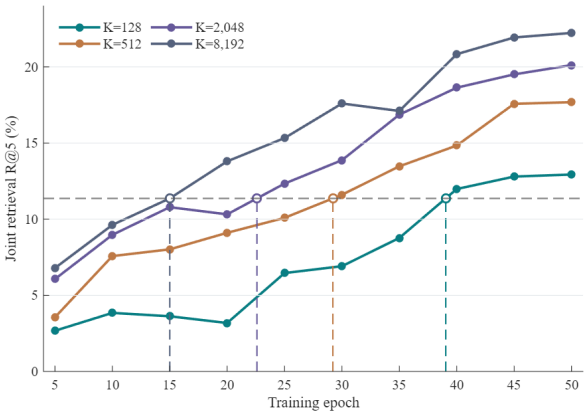


Figure 13: Effect of the training candidate count K on nuScenes joint text–image-to-point-cloud retrieval. Dashed lines mark a shared R@5 level of 11.365%.

Figure 13 shows that larger candidate sets generally yield higher R@5, particularly in later epochs. However, smaller candidate sets can reach the same intermediate retrieval level with additional training epochs. The dashed guides show this at R@5 = 11.365%: $K = 8192$ reaches this level at epoch 15, while $K = 2048$, 512, and 128 reach it at approximately epochs 23, 29, and 39, respectively, using linear interpolation between evaluations. Candidate sampling therefore offers a trade-off between the number of candidates per anchor and the training epochs needed to reach this retrieval level, although smaller candidate sets still achieve lower R@5 at epoch 50.

D.2 Qualitative Results of Joint Retrieval

Figures 14 and 15 show qualitative examples of CTP joint text–image-to-point-cloud retrieval on nuScenes and ULIP-ShapeNet. Each row presents a query image and the top five point clouds retrieved using the paired text and image as the joint query.

On ULIP-ShapeNet, the retrieved point clouds show visually similar object structures, such as the seat and backrest of chairs and the fuselage and wings of airplanes. On nuScenes, the examples illustrate retrieval with sparse and incomplete point clouds, where geometric details are less apparent. These examples show CTP's ability to retrieve relevant point clouds from joint text–image queries across both sparse and dense point clouds.

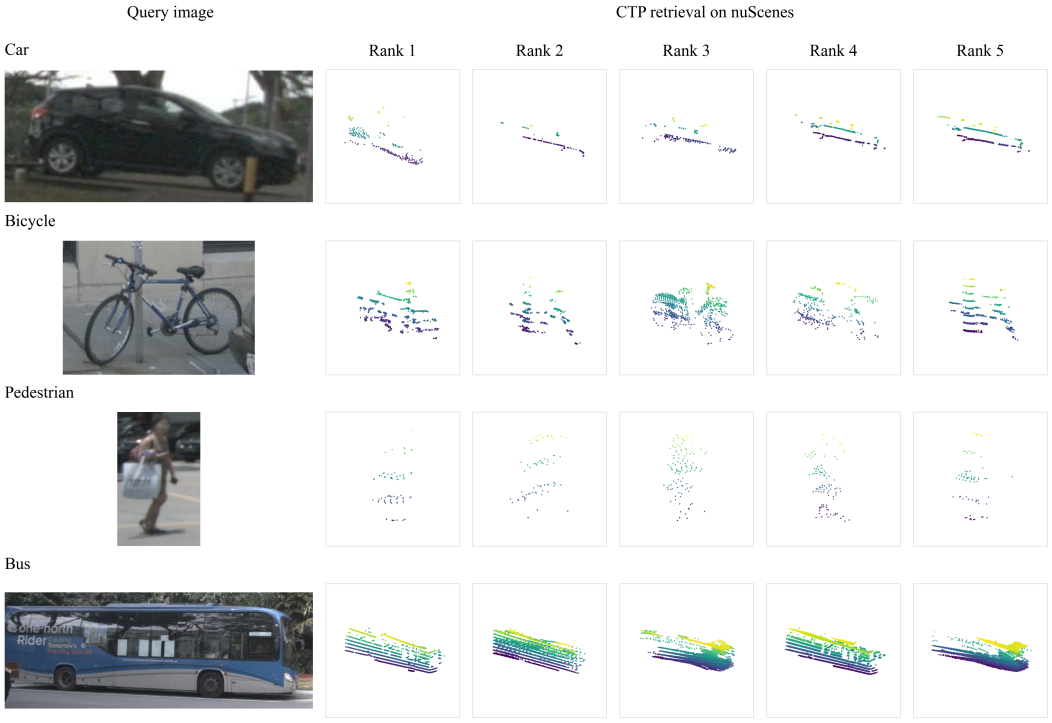


Figure 14: Qualitative examples of CTP joint text–image-to-point-cloud retrieval on nuScenes. Query captions are omitted for brevity.

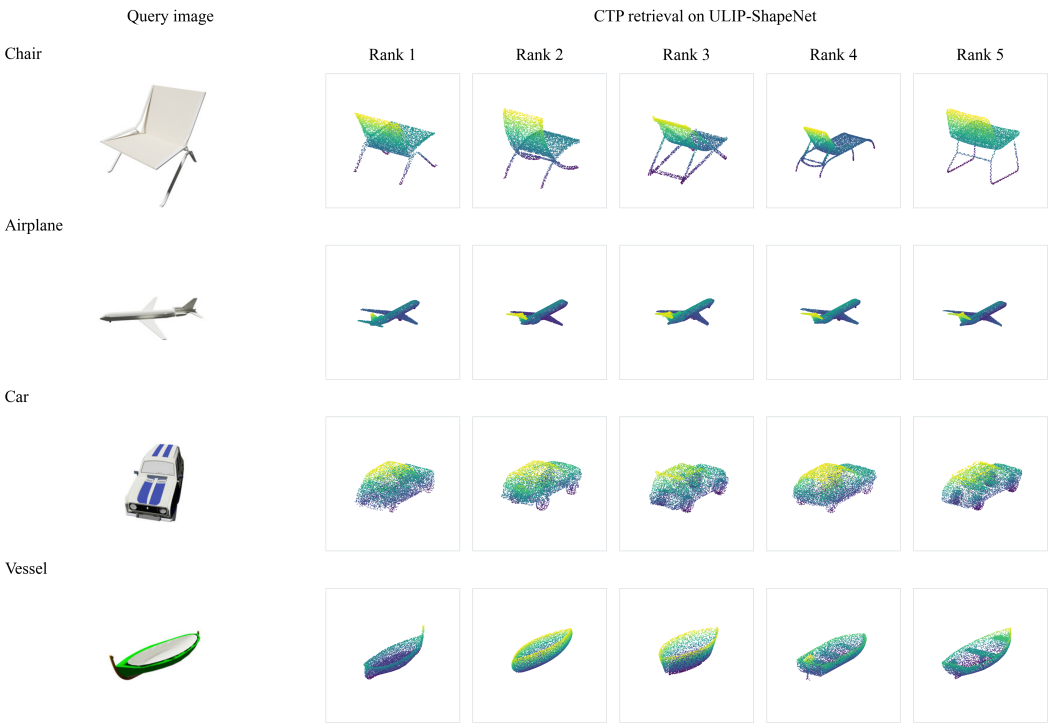


Figure 15: Qualitative examples of CTP joint text–image-to-point-cloud retrieval on ULIP-ShapeNet. Query captions are omitted for brevity.