MultiCAD: Contrastive Representation Learning for Multi-modal 3D Computer-Aided Design Models

Weijian Ma*
mawj22@m.fudan.edu.cn
School of Computer Science
Fudan University
Shanghai, China

Minyang Xu*
16110240027@fudan.edu.cn
School of Computer Science
Fudan University
Shanghai, China

Xueyang Li
xueyangli21@m.fudan.edu.cn
School of Computer Science
Fudan University
Shanghai, China

Xiangdong Zhou†
xdzhou@fudan.edu.cn
School of Computer Science
Fudan University
Shanghai, China

ABSTRACT

CAD models are multimodal data where information and knowledge contained in construction sequences and shapes are complementary to each other and representation learning methods should consider both of them. Such traits have been neglected in previous methods learning unimodal representations. To leverage the information from both modalities, we develop a multimodal contrastive learning strategy where features from different modalities interact via contrastive learning paradigm, driven by a novel multimodal contrastive loss. Two pretext tasks on both geometry and sequence domains are designed along with a two-stage training strategy to make the representation focus on encoding geometric details and decoding representations into construction sequences, thus being more applicable to downstream tasks such as multimodal retrieval and CAD sequence reconstruction. Experimental results show that the performance of our multimodal representation learning scheme has surpassed the baselines and unimodal methods significantly.

CCS CONCEPTS

• **Computing methodologies** → **Machine learning**; **Unsupervised learning**.

KEYWORDS

Multimodal Machine Learning; Representation Learning; Contrastive Learning; Computer Aided Design

ACM Reference Format:
Weijian Ma, Minyang Xu, Xueyang Li, and Xiangdong Zhou. 2023. MultiCAD: Contrastive Representation Learning for Multi-modal 3D Computer-Aided Design Models. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management (CIKM '23), October 21–25, 2023, Birmingham, United Kingdom.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3583780.3614982

*Equal contribution.
†Corresponding author.

CIKM '23, October 21–25, 2023, Birmingham, United Kingdom

ACM ISBN 979-8-4007-0124-5/23/10…$15.00
https://doi.org/10.1145/3583780.3614982

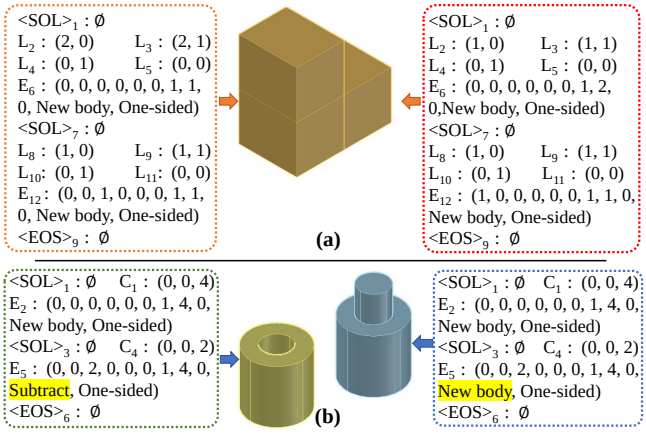

Figure 1: CAD models and their corresponding parametric construction sequences do not have one to one correspondence. (a) Different CAD sequences may correspond to the same shape. (b) Very similar CAD sequences may correspond to very different shapes.

1 INTRODUCTION

Computer-Aided Design (CAD) models encompass a broad spectrum of applications in industry scenarios. They contain a compact and editable representation of 3D products in various expression methods such as construction sequences and rendered 3D shapes [3]. The construction sequence contains a compact understanding of the object from human designers while the 3D shape offers a macro understanding of the object geometry. Learning models dealing with such complex data is required to link the representations across both modalities, which also serve as a foundation for downstream applications such as shape classification, CAD model retrieval as well as CAD model generation [29, 30]. Traditionally, CAD representation learning adopts methods in computer vision domains, which regard the rendered shape as 3D geometry and adopts supervised learning methods on manually annotated datasets such as ModelNet and ShapeNet [4, 51]. However, such methods not only neglect information such as shape details and how human designers construct the CAD model, but are also limited on annotated datasets that are too small and too expensive to obtain compared with those without annotations [22, 35, 43, 51].

Recently, representations of parametric CAD construction sequences have been explored by a few researches [13, 50, 53] via language modelling methods [46]. Modelling CAD sequence representation has the following advantages. First, representations of CAD construction sequences not only encode shape details, but also

include knowledge of the components as well as the relationship between them. Second, prevailing language model is a reasonable choice for CAD sequence generation [37, 38, 46]. Because CAD operation sequences can be regarded as written in a structured language. The generated CAD sequences can then be rendered into CAD shapes by using various geometry kernels and toolboxes [9].

Despite the success of CAD representation learning in either visual [1, 35, 43] or sequence domains [13, 23, 48, 50], we believe that CAD models are actually multimodal data and different modalities are complementary to each other. Building a comprehensive representation on both construction sequence and geometry can boost the performance in downstream tasks but it is challenging to do so, due to the huge domain gap between the sequence representation and pointcloud representation. Moreover, the two modalities have no one-to-one correspondence. As Figure 1 shows, a CAD shape may correspond to various different CAD sequences while a slight modification to a CAD sequence may result in a huge change in CAD geometry. Such observation is key feature of the CAD models, which also forms the basis of our model design.

For representing CAD models, multimodal representation methods that can be referred to focused on multimodal understanding and visual generation [24, 25, 31, 34, 42]. CLIP [42] has shown remarkable performance on image understanding. BLIP series [24, 25] generate free-form caption texts from images. In 3D vision domain, [31, 34] have tried to generate 3D models based on text via diffusion. However, it is not feasible to directly copy the tricks to our problem settings as constructing the representation space of previous works requires massive data, nor can these methods generate long and specialized text such as CAD construction sequences. To our best knowledge, MultiCAD is the first attempt to combine knowledge of both CAD construction sequences and geometry to acquire a multimodal representation. It can be used in CAD sequence generation, 3D shape classification and multimodal retrieval, forming the basis of downstream tasks in both geometric and sequence domains.

In order to integrate information from construction sequences and geometry, we adopt a novel two-stage training method where information from two modalities first interact implicitly in a self-supervised manner, then they are aligned under supervision. The first stage is implemented via a novel multimodal contrastive learning scheme in order to bypass the discrepancy between modalities. Meanwhile, a pretext task for CAD sequence decoding is adopted simultaneously to make sure the representation can be decoded into CAD construction sequences. In the second stage, the domain discrepancy issue is solved by a specially designed geometry pretext task where features between modalities are explicitly aligned and the geometric feature extractor is guided to distinguish between shape details. For data augmentation, a novel translation scheme is proposed to synchronize rotation, scaling and translation into corresponding CAD construction sequences, thus expanding shape augmentation techniques to sequence domains for the first time.

We use a variety of downstream tasks to validate the performance of our multimodal contrastive learning scheme. The experimental result shows that the proposed method has surpassed both the baselines and previous unimodal methods by a large margin. Moreover, we also show qualitatively that our learning scheme can lead to better shape reconstruction results. To summarize, our contribution includes:

- We point out that CAD models are multimodal data and propose a novel framework for multimodal representation learning, which is the first of its kind in CAD parametric model learning.
- A two-stage training method is proposed to guide the learning process of the multimodal representation. Two pretext tasks on both sequence and geometric domains are designed to model the generation and transformation between modalities, which is believed to be the core ability of both unimodal and crossmodal downstream tasks.
- A multimodal translation scheme is proposed to synchronize the transformation between CAD shapes and construction sequences in data augmentation.
- Experiments on a variety of downstream tasks show the effecitveness of our proposed multimodal representation learning scheme. A significant performance gain is witnessed in both sequence generation and visual representation tasks.

2 RELATED WORK

Representation learning for 3D point clouds Early learning-based methods for 3D object focus on fully or semi-supervised learning [7, 16, 57]. Supervised representation learning requires large-scale, manually annotated datasets, which is extremely time-consuming owing to the irregular structure of point clouds [35, 43]. Therefore, recent studies on point clouds have moved to self-supervised representation learning. Yang et al.[55] propose a self-supervised framework based on autoencoder for cuboid shape abstraction via mapping point clouds into compact cuboid representation. Chen et al. [6] embed 3D point clouds with local features and fed to a point integration model to produce a set of 3D structure points under chamfer distance. Rao et al.[40] propose to learn point cloud representation via bidirectional reasoning between local structures and global shape without manual supervision. However, such methods neglect information from another modality, which is often complementary to point clouds.

Representation learning of CAD models The success of language models have inspired researches on learning representations of parametric CAD construction sequences [15]. DeepCAD [50] is a transformer-based generative model of CAD sequence [17, 18, 46] where the latent space constructed by transformer encoder-decoder pair can be used for unconditional generation. SketchGen [33] designs a language with simple syntax as the extra perception to generate CAD sketches automatically, solving the heterogeneous problem in graph-based CAD sketch encoding. JoinABLe [48] uses boundary representations for weakly supervised learning without help of additional object category labels or other manual guidance, assembling entities to a complete model. SkexGen [53] separately models drawing commands and parameters of sketch and extrusions. It succeeds in CAD sequence generation but fails to model relationship between sketch and extrusion pairs, nor does it obtain any shape information. However, these methods neglect geometric information, which is vital to CAD model understanding and downstream tasks.

CAD Sequence Reconstruction Several methods have been proposed to reconstruct CAD sequences from other modalities. FaceFormer [47] reconstructs a 3D CAD model from a single 2D drawing line through the face identification results. Free2CAD [23]

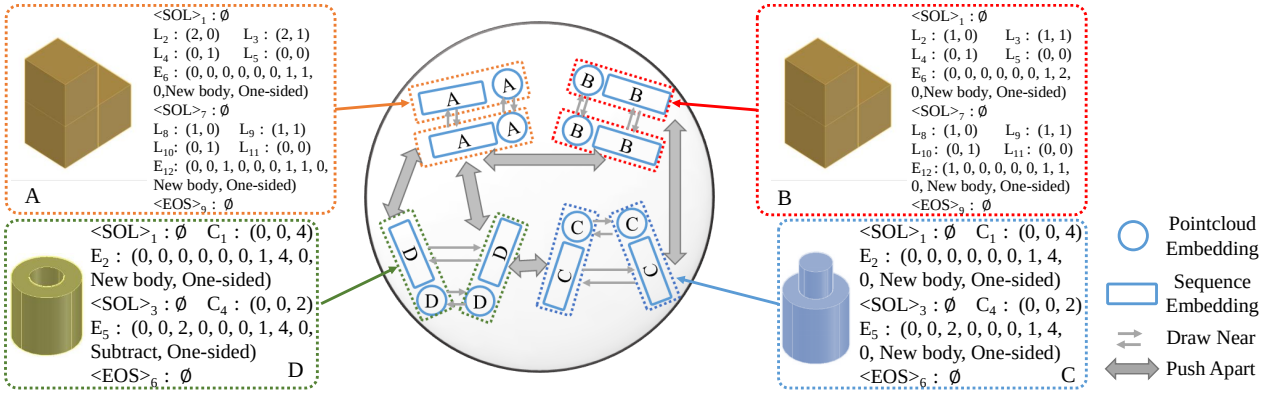

Figure 2: The illustration of our Multi-Modal Contrastive Learning (MMCL) scheme. The rectangles stand for sequence embeddings and circles stand for pointcloud embeddings. The embeddings of two CAD models in each modality will be pushed apart because either their sequence expressions or their geometry are not similar. The embeddings will be drawn near only when they are two positive examples of a CAD model.

regards sketch-based CAD modeling as seq2seq translation problem, disassembling the drawn sketches and translating them into individual CAD command. Point2Cyl [45] translates CAD pointclouds into sketch-and-extrusion pairs by base-barrel segmentation and implicit geometric rendering. However, it can only handle CAD models with a few extrusions and does not output CAD sketch commands and parameters.

Multimodal representation learning on 3D objects 3D objects often have representations in various modalities, such as point clouds, meshes, multi-view images, text description and CAD construction sequences. Liu et al. [28] adopt late fusion on point clouds and image features via contrastive learning and transfers multimodal representation into an image feature extractor to estimate 3D object pose. CMCV [20] exploits correlation between views and modalities by a lightweight late fusion method, guiding network to obtain features of 2D images and 3D point clouds jointly without manual annotation. CAT-Det [56] adopts Point Transformer and Vision Transformer to extract point cloud and image features and fuse them via another transformer. CrossPoint [1] maintains alignment between 3D and 2D objects by maximizing consistency between point cloud and 2D image in invariant space. However, such multimodal interactions happen only at geometry domain.

Different from previous works, our goal is to build a comprehensive representation of CAD models from point clouds and construction sequences, which are significantly different modalities but complementary to each other. To the best of our knowledge, this is the first work bridging the gap between parameterized CAD sequences and 3D point clouds to build a comprehensive representation for downstream tasks, which is achieved by our proposed multimodal contrastive learning scheme.

3 THE PROPOSED METHOD

3.1 Representation Learning of CAD models

For learning comprehensive representation of CAD models, we argue that information from both geometry and construction sequences should be focused on. Point clouds are scattered, non-uniform data while construction sequence is token-based data with various length. The construction sequences can provide geometric representations with shape details like edges and corners, as well as design flow of human architects. On the other hand, CAD geometries, such as point clouds, grants construction sequences a macro understanding of corresponding shape. As shown in Figure 1, information from another modality helps when one modality fails to distinguish between different data.

A CAD model M can be interpreted as point clouds, construction sequences and so on. Each interpretation corresponds to a modality that can be referred to when learning the representation of M. Let $\mathcal{R}(M)$ denote the representation of a CAD model M, $\mathcal{R}(M)$ can be regarded as the fusion $\mathcal{F}$ between representations from different modalities, as shown in Equation 1

$$\mathcal{R}(M) = \mathcal{F}(h_1(M_1), h_2(M_2), ..., h_n(M_n)), \tag{1}$$

where M_i and $h_i(M_i)$ represent the raw data and the embedding of representations in modality i. In MultiCAD, we adopt two modalities, namely pointcloud and CAD construction sequences. Point cloud features h_p are extracted from a pointnet++ [36] encoder while a transformer encoder [46, 50] is employed to extract features h_s from the CAD construction sequences.

A two-stage training pipeline is adopted to obtain the multimodal representation of CAD models. In the first stage, a representation space is built by considering both feature from its own modality and hint from the other. For representations in each modality, two embeddings are nearby only when their sequences and geometric representations are both alike. In the second stage, representations of two modalities are aligned for multimodal translation purposes. The knowledge of generating sequence representations are granted to pointcloud feature extractor via a supervised training method.

3.2 Multimodal Representation Learning

For CAD models, the geometric representation and sequence representation are significantly different and have no one-to-one correspondence, as is shown in Figure 1. Point clouds are unstructured data while the construction sequences are semi-structured, token-based data. It is difficult to put them into a unified feature extractor or to explicitly align the two representations at the start of the training procedure. In this sense, MultiCAD separately extracts features of the two modalities and adopts implicit crossmodal interaction via contrastive learning [8, 10, 44, 52, 58] in the first stage of training process. Such strategy is illustrated in Figure 2.

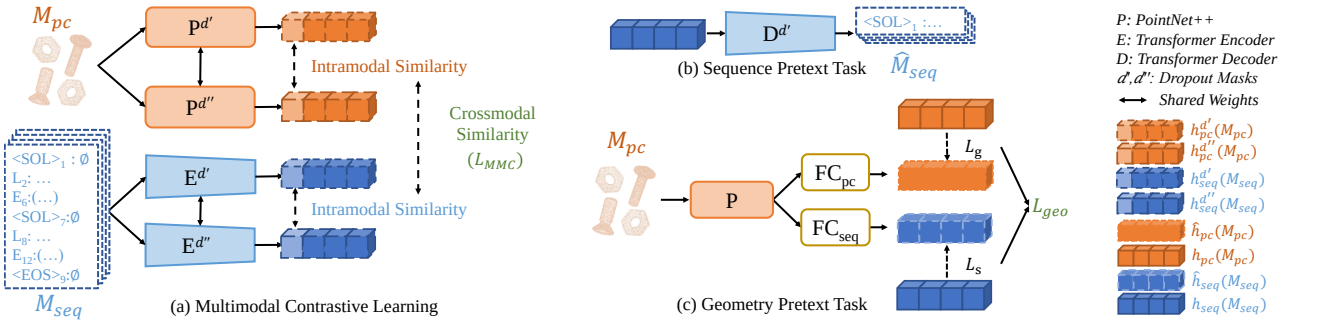

Figure 3: Network details of multimodal contrastive learning and the pretext tasks. (a) The architecture of MMCL. Positive pairs are generated via dropout of feature extractors. (b) Sequence pretext task reconstructs input sequence from sequence representation via transformer decoder. (c) The geometry pretext task. In this task, the PointNet++ is trained from both point cloud embeddings and the sequence embeddings. $\mathcal{L}_g$ is used to preserve the ability to acquire geometric representations. $\mathcal{L}_s$ is used to fit the CAD sequence representations.

To integrate both sequential and geometric information into the contrastive learning scheme, we propose Multi-Modal Contrastive Loss (MMCL, L_{MMC}), which can be regarded as a multimodal extension of InfoNCE [32]. MMCL calculates the logsoftmax of the similarities between the representation $\mathcal{R}_i^{d'}(M)$ of sample i and the representation of its positive example $\mathcal{R}_i^{d''}(M)$ in a batch of size N_m. The formal definition of MMCL is shown in Equation 2 :

$$\mathcal{L}_{MMC} = -\alpha * (1 - \text{sim}_{\text{MM}}(\mathcal{R}_i^{d'}(M), \mathcal{R}_i^{d''}(M)))^{\gamma} * \log \frac{e^{\text{sim}_{\text{MM}}(\mathcal{R}_i^{d'}(M), \mathcal{R}_i^{d''}(M))/\tau}}{\sum_{j=1}^{N_m} e^{\text{sim}_{\text{MM}}(\mathcal{R}_i^{d'}(M), \mathcal{R}_j^{d''}(M))/\tau}}, \tag{2}$$

where $\alpha * (1 - \text{sim}_{\text{MM}}(\mathcal{R}_i^{d'}(M), \mathcal{R}_i^{d''}(M)))^{\gamma}$ constitutes a focal factor inspired by focal loss [27]. α is a hyperparameter for scaling the loss and γ is for adjusting the punishment of well-contrasted examples. τ is the temperature hyperparameter.

In MMCL, for calculating multimodal similarity sim_{MM}, a straightforward measurement scheme is proposed, as shown in Equation 3

$$\text{sim}_{\text{MM}}(\mathcal{R}_i^{d'}(M), \mathcal{R}_i^{d''}(M)) = (\text{sim}(h_{seq_i}^{d'}, h_{seq_i}^{d''})^{\frac{1}{2}} * \text{sim}(h_{pc_i}^{d'}, h_{pc_i}^{d''})^{\frac{1}{2}})^{\sigma}, \tag{3}$$

where the similarity of sequence and point cloud embeddings between two CAD models are defined as $\text{sim}(h_{seq_i}^{d'}, h_{seq_i}^{d''})$ and $\text{sim}(h_{pc_i}^{d'}, h_{pc_i}^{d''})$ respectively. σ is a hyperparameter adjusting mean and deviation of sim_{MM}. The similarity within each modality is defined as the cosine similarity between embeddings $h_{i_j}^{d'}$ and $h_{i_j}^{d''}$, as Equation 4 shows.

$$\text{sim}(h_{i_j}^{d'}, h_{i_j}^{d''}) = \frac{h_{i_j}^{d'\top} h_{i_j}^{d''}}{\left\|h_{i_j}^{d'}\right\| \cdot \left\|h_{i_j}^{d''}\right\|} \; i \in \{pc, seq\}. \tag{4}$$

MMCL extends InfoNCE to multimodal domain by defining the similarity calculation strategy between modalities and introducing focal factor as a pairwise loss regulator.

For similarity calculation strategy, as the similarity between two CAD models lies between that of the sequence representations and shape representations, adopting the mean of the two is reasonable. We define sim_{MM} as the geometric mean of similarities between sequence and point cloud embeddings. This is because some CAD models are alike in one modality and quite different in another, as shown in Figure 1. We should pay more attention to the modality showing less confidence.

For the focal factor design, we have the following observations. During the training process, when the representation of different models are alike in both modalities, sim_{MM} will become significantly larger than models similar in one modality. This is especially the case for a model and its positive example where sim_{MM} of positive pairs quickly grows to ~ 1 at the beginning of the training process. Therefore, as shown in Equation 2, we adopt the focal factor $\alpha * (1 - \text{sim}_{\text{MM}}(\mathcal{R}_i^{d'}(M), \mathcal{R}_i^{d''}(M)))^{\gamma}$ as a regularizer to suppress these cases and make MMCL focus on models similar in one modality, namely the *hard negatives*.

In this sense, the embeddings of two modalites will be drawn near only when they are alike in both sequence and point cloud representations, as illustrated in Figure 2. Since multimodal interaction happens completely at similarity calculation between representations from different CAD models, MultiCAD bypasses the discrepancy issues between modalities and successfully integrates multimodal interaction into contrastive learning process. As a result, the representation space aligned and uniform and serving as a good initialization for downstream tasks in each modality.

3.3 Training method

In this section, we first introduce our two-stage training pipeline. A sequence pretext task is trained along with multimodal contrastive learning first, as introduced in section 3.3.2. A geometry pretext task is trained in the second stage, details refer to section 3.3.3.

3.3.1 Training Pipeline. The training strategy of MultiCAD can be divided into two parts. First, representations of different modalities interact via Multimodal Contrastive Learning, as illustrated in Figure 2. During this stage, the sequence pretext task is trained simultaneously because we pay special attention to the decoding ability of the sequence representation. The geometric pretext task is conducted afterwards so that representation between two modalities are aligned. Training details of the two stages are as follows.

In the first stage, multimodal contrastive learning and the sequence pretext task are conducted simultaneously. Network architecture details in this stage are shown in part (a) and part (b) of Figure 3. The input to the network is the point cloud M_{pc_i} and the

construction sequence M_{seq_i} of a CAD model i. Training strategy is similar to simcse [14] where positive pairs are pointcloud embeddings $h^{d'}_{pc_i}, h^{d''}_{pc_i}$ and sequence embeddings $h^{d'}_{seq_i}, h^{d''}_{seq_i}$ generated by applying different dropout masks $\{d', d''\}$ to the pointcloud encoder P and sequence encoder E. $\mathcal{L}_{MMC}$ is calculated over generated positive pairs, as shown in Equation 2. All sequence representations h_{seq} are then fed into sequence decoder D to conduct sequence pretext tasks. Details of sequence pretext task refer to Section 3.3.2.

The loss of the first stage is a weighted sum over $\mathcal{L}_{MMC}$ from multimodal contrastive learning and $\mathcal{L}_{seq}$ from sequence pretext task, as is shown in Equation 5 where λ is a factor balancing the losses.

$$\mathcal{L}_{CAD} = \mathcal{L}_{MMC} + \lambda \mathcal{L}_{seq}. \tag{5}$$

After the first stage of training completes, a sequence representation considering the geometric condition is obtained and such representation can be decoded to valid construction sequences. The sequence representation is then fixed and will be used as a supervised training target in the second stage.

In the second stage, only pointnet++ is trained under the geometry pretext task. The pointnet++ is trained under a supervised manner where both geometric embeddings and sequence embeddings are targets. Detail of geometry pretext task and the corresponding loss refers to Section 3.3.3 and Figure 3 (c).

In this sense, sequence representation is built under the guidance of CAD geometry in the first phase. Geometry representation is also aligned with sequence representation in the second phase.

3.3.2 The sequence pretext task. The understanding of CAD sequences and the decoding ability of a CAD representation are of vital importance since sequence representation is aimed at generating CAD sequences. In this sense, we follow DeepCAD [50] and add sequence reconstruction as a special pretext task along with the training process. For a CAD model M with command sequence $M_{seq} = [C_1; ...; C_N]$ including both the command type c_t and parameters c_p of C_i, the sequence representation $h_{seq}(M_{seq})$ will be fed into a transformer decoder and a reconstructed CAD sequence $\hat{M}_{seq}$ is obtained. A sequence reconstruction loss $\mathcal{L}_{seq}$ is added to the first stage of training process, shown in Equation 6.

$$\mathcal{L}_{seq} = \sum_{i=1}^{N_m} \ell\left(\hat{c}_t^i, c_t^i,\right) + \beta \sum_{i=1}^{N_m} \sum_{j=1}^{K_m} \ell\left(\hat{c}_p^{i,j}, c_p^{i,j}\right). \tag{6}$$

In Equation 6, $\hat{c}_t^i$ denotes predicted command type and c_t^i is the ground truth. $\hat{c}_p^{i,j}$ and $c_p^{i,j}$ represent predicted and ground truth parameters respectively. ℓ is the standard cross entropy loss same as DeepCAD [50]. K_m is the number of parameters and β is a factor balancing loss of command and argument. For more details about CAD sequences and their modelling methods, please refer to [50].

3.3.3 The geometry pretext task. To further correlate the point cloud representation and sequence representation, we adopt a geometry pretext task operating on pointcloud feature extractor in the hope of multimodal alignment. We use both pointcloud and sequence representation in the multimodal representation space to help integrate the knowledge of sequence generation into pointnet++ feature extractor and preserve the ability of geometric representation. The architecture is shown in part (c) of Figure 3 where

Algorithm 1 Sinkhorn Algorithm

Input: $\{\hat{h}_{seq}(M_{seq})\}_{i=1}^{n}, \{h_{seq}(M_{seq})\}_{j=1}^{n}, \epsilon$, probability vectors μ, ν
$\sigma = \frac{1}{n}1_n, \pi^{(1)} = 11^T$
$C_{ij} = c(\hat{h}_{seq_i}, h_{seq_j}), A_{ij} = e^{-\frac{C_{ij}}{\epsilon}}$
for $t = 1, 2, 3, ...$ **do**
  $Q = A \odot \pi^{(t)}$ ▷ $\odot$ is Hadamard Product
  for $k = 1, 2, 3, ..., K$ **do**
    $\delta = \frac{\mu}{nQ\sigma}, \sigma = \frac{\nu}{nQ^T\delta}$
  end for
end for
$OT = \langle \pi, C \rangle$ ▷ $\langle \cdot, \cdot \rangle$ is the Frobenius dot-product
return OT

FC_{pc} and FC_{seq} are used to generate geometric representations $\hat{h}_{pc}(M_{pc})$ and sequence representations $\hat{h}_{seq}(M_{seq})$ respectively.

As pointnet++ naturally generates logits used for classification, MMD loss [19] is utilized to measure the geometric discrepancy $\mathcal{L}_g$ between the output of FC_{pc} and the geometric representation so as to preserve the ability to acquire geometric features.

For transferring CAD sequence representations, as similar CAD geometries often correspond to significantly different CAD construction sequences, losses such as MSE will guide $\hat{h}_{seq}(M_{seq_i})$ to the midpoint of corresponding sequence embeddings when the network converges. In this sense, a specially designed sequence discrepancy $\mathcal{L}_s$ is adopted to fit the sequence representations generated by FC_{seq} for downstream sequence generation tasks. $\mathcal{L}_s$ is a combination of MSE loss and Optimal Transport (OT) [2, 5, 12] where OT helps pulling $\hat{h}_{seq}(M_{seq_i})$ out of the midpoint and the cosine similarity based transportation plan helps guide $\hat{h}_{seq}(M_{seq})$ to the nearby cluster centers of $h_{seq}(M_{seq})$ within a batch, namely the neighborhood of the corresponding sequence embedding[5]. Such batchwise transport plan also helps the pointcloud feature extractor gain the knowledge about the distinctions between geometrically similar CAD models, namely the shape details. Formally, OT is calculated via Equation 7

$$\begin{aligned} &OT(\hat{h}_{seq}(M_{seq_i}), h_{seq}(M_{seq_j})) \\ &= \min_{T \in \Pi(u,v)} \sum_{i=1}^{N} \sum_{j=1}^{K} T_{ij} \cdot c(\hat{h}_{seq}(M_{seq_i}), h_{seq}(M_{seq_j})) \\ &= \min_{T \in \Pi(u,v)} \langle T, C \rangle, \end{aligned} \tag{7}$$

where $\Pi(u, v) = \{T \in R_+^{N \times K} | \; T1_K = \frac{1}{N}1_N, T1_N = \frac{1}{K}1_K\}$. 1_K denotes 1-dimensional one vector. C is the cost matrix where $C_{ij} = c(\hat{h}_{seq}(M_{seq_i}), h_{seq}(M_{seq_j}))$ ($c(\cdot, \cdot) = 1 - \cos(\cdot, \cdot)$). Implementation of OT follows Sinkhorn algorithm [11] shown in Algorithm 1.

The loss of the geometry pretext task is shown in Equation 8 where μ goes from 1 to 0 and ϕ goes from 0 to 1.

$$\begin{aligned} \mathcal{L}_{geo} =& \mu \mathcal{L}_g + \phi \mathcal{L}_s \\ \mathcal{L}_g =& MMD(\hat{h}_{pc}(M_{pc_i}) \\ \mathcal{L}_s =& MSE(\hat{h}_{seq}(M_{seq_i}), h_{seq}(M_{seq_i})) + \\ & OT(\hat{h}_{seq}(M_{seq_i}), h_{seq}(M_{seq_i})). \end{aligned} \tag{8}$$

3.4 Data Augmentation

As CAD models are sophisticated and multimodal, data augmentation is difficult since single modal augmentation tends to cause semantic misalignment [1]. For a good multimodal CAD data augmentation scheme, it is required to add diversity of both modalities and follow the genre of human designs. Previous attempt [50] augments CAD models in sequence domain, but often brings about invalid topology, nor does it preserve human design styles.

To alleviate this problem, we conduct augmentation in geometric domain and propose a translation strategy to convert the augmented geometric shapes into corresponding CAD sequences. Specifically, the geometric shape of a CAD model is randomly scaled, translated and rotated. Then the corresponding parameters in the CAD construction sequence, namely sketch size, location and orientation, are altered according to the adjustment of geometric shapes.

For the corresponding CAD sequence of a rotated, scaled and translated CAD geometry, the formal definition is as follows. For an augmented CAD geometry with scaling S, translation T and rotation R, the scale is multiplied by a scale factor S_c. The orientation is perturbed by R_c. The sketch plane location is adjusted by R_c and further jittered by T_c. S_c, T_c and R_c obey normal distribution. The augmentation strategy $\mathrm{S}(M)$ for a model M is shown in Equation 9

$$
\begin{aligned}
&S(M) = [C_1^*; ...; C_N^*],\\
&C_i^* = [c_t^{i^*}, c_p^{i^*}],\\
&c_t^{i^*} = c_t^i,\\
&c_p^{i^*} = \begin{cases} c_p * S_c & c_p \in \{e_1, e_2, s\}, \quad S_c \sim N(0.8, 1.2) \\ c_p * R_c & c_p \in \{\theta, \phi, \gamma\}, \quad R_c \sim N(0.8, 1.2) \\ T(c_p) & c_p \in \{p_x, p_y, p_z\} \\ c_p & \text{otherwise} \end{cases},\\
&T(c_p) = \begin{cases} r * \cos(\theta * R_c) + T_c & c_p = p_x \\ r * \cos(\phi * R_c) + T_c & c_p = p_y \\ r * \cos(\gamma * R_c) + T_c & c_p = p_z \end{cases},\\
&r = (p_x^2 + p_y^2 + p_z^2)^{1/2}, \quad T_c \sim N(-0.05, 0.05),
\end{aligned}
\tag{9}
$$

where e_1 and e_2 refers to extrusion height in both directions. s refers to scale of the corresponding sketch profile. p_x, p_y, p_z represent the location while θ, ϕ, γ refers to orientation of a sketch plane.

To make sure the augmented CAD models of $\mathrm{S}(M)$ have legal geometric representation, we further examine them by attempting to render its point cloud and discard all failure cases.

In this sense, both the geometric shape and the sequence representation of CAD models are augmented. The topological validity and the human design style are also preserved.

4 EXPERIMENTS

4.1 Datasets

DeepCAD. DeepCAD is a multimodal CAD dataset composed of parametric construction sequences which can be converted into pointclouds. MultiCAD uses both parametric sequences and rendered pointclouds for training and testing.

Fushion 360. Fushion 360 [49] is similar to DeepCAD but only contains 8,625 CAD construction sequences. There exists a domain gap between construction sequences in DeepCAD and Fushion360.

Mechanical Components Benchmark The Mechanical Components Benchmark (MCB) [21] is a dataset consisting of 58,696 3D mechanical components with 68 classes. MCB is a challenging benchmark for classification as the distinction between classes lies in shape details. We use all data in MCB to test pointcloud classification ability of the geometric representation.

4.2 Implementation details

Runtime Environments. All experiments are conducted on two RTX 3090 GPUs along with Pytorch 1.11.0 running on CUDA 11.3.

Experiment Settings The reported results are trained under DeepCAD dataset with batch size 512 for 1000 epochs. Dropout rate is 0.3 for transformer encoder while 0.1 for pointnet++ encoder. Learning rate is set to 0.001 with 2000 steps linear warmup and gradient clipping of 1.0. For reconstruction tasks, we follow [39] and select 32 candidates via top-K sampling. For hyperparameters of MMCL, we fix α at 2, γ increases from 0 to 2 and σ decreases from 2 to 1 linearly. For training stability, we pretrain pointnet++ encoder in a simcse-like self-supervised manner first.

4.3 Baseline Methods

As MultiCAD is the first work of multimodal CAD representation learning, no previous work can make direct comparisons. We make slight modifications to several multimodal interaction methods. The feature extractors of all baseline methods are the same as MultiCAD.

- **CLIP:** The first baseline is adopting CLIP [42], a multimodal contrastive learning method operating on image-text pairs. We replace the image encoder with our pointcloud encoder and use the sequence encoder to serve as text encoder in our problem setting. For CAD sequence decoding, the sequence deocder proposed in Section 3.3.2 is trained afterwards based on embeddings generated by sequence encoder.
- **CrossPoint:** The second baseline is a modification of CrossPoint [1], a recent work on 3D multimodal representation learning. In CrossPoint, two views of a 3D model are first aligned via a contrastive loss. Another contrastive loss is further added above the feature extractors where crossmodal representations are further aligned. In our problem setting, we replace pointcloud feature extractor as sequence encoder. Image feature extractor is replaced as pointcloud feature extractor. A CAD sequence decoder is trained on sequence embedding after multimodal pretraining completes.

4.4 Experimental Results

4.4.1 Measurements of the Representation Space. As is mentioned above, multi-modal contrastive learning can make representation space of CAD models more aligned and uniform. We use k-means clustering and calculate Sum Squared Error (SSE) of each cluster and Silhoutte Coefficient (SC) [41] for measurement. SSE can be regarded as alignment metric while SC for alignment and uniformity. Formal definition of SSE and SC are shown in Equation 10 and Equation 11 where x_i is a data point in set X of size n, a_i is the average distance between x_i and other points in its cluster and b_i is the distance between x_i and other clusters.

$$\mathrm{SSE} = \Sigma_{i=1}^{n}(x_i - \bar{x})^2. \tag{10}$$

$$\mathrm{SC} = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}. \tag{11}$$

Method	DeepCAD (Test Set)						Fushion 360 (Test Set)					
	Pointcloud → Sequence			Sequence → Pointcloud			Pointcloud → Sequence			Sequence → Pointcloud		
	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10
Unimodal Methods												
DeepCAD	36.48%	61.87%	68.19%	37.66%	52.30%	59.37%	8.92%	16.78%	22.10%	7.08%	35.41%	70.82%
Self-supervised Pointnet++	21.47%	45.39%	51.35%	20.79%	38.47%	44.53%	6.34%	14.75%	18.67%	9.13%	46.38%	64.75%
Baseline Methods												
CLIP [42]	53.85%	67.81%	74.41%	60.87%	76.44%	82.38%	30.64%	52.88%	61.99%	34.23%	62.01%	71.08%
CrossPoint [1]	64.07%	77.12%	81.66%	66.17%	80.88%	85.26%	15.43%	39.16%	50.84%	22.35%	46.27%	70.57%
Our Method												
MultiCAD	72.44%	80.19%	84.84%	80.82%	92.18%	94.76%	57.91%	72.01%	**76.91%**	61.97%	81.37%	87.27%
MultiCAD w/ Aug.	**77.23%**	**85.96%**	**89.35%**	**84.29%**	**93.79%**	**95.82%**	**58.54%**	**72.65%**	76.89%	**63.06%**	**83.40%**	**88.63%**

Table 1: Qualitative results for multimodal CAD retrieval. Accuracy stands for the average of command and parameter accuracy for CAD sequences and IoU for CAD geometries. For unimodal methods, we train an encoder form another modality to fit the embeddings. Baseline methods are implemented as shown in section 4.3.

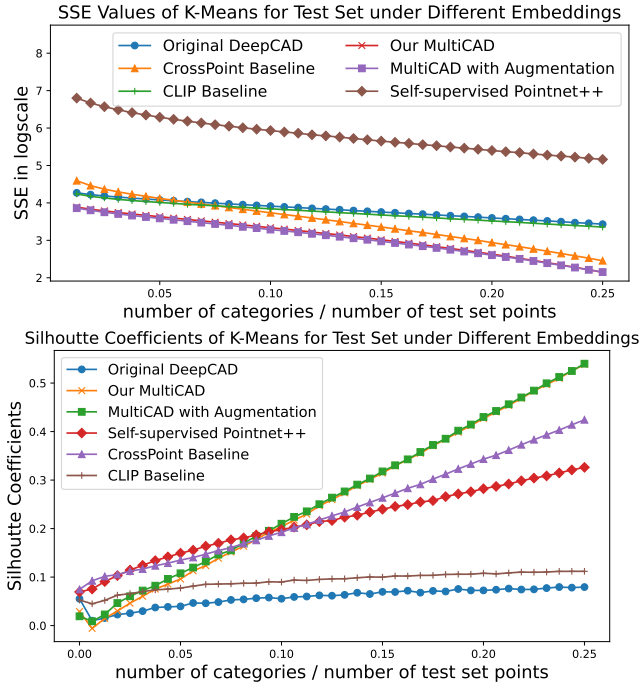

Figure 4: Illustration of alignment and uniformity in the latent space. Each line refers to Sum of Squared Errors (SSE) and Silhoutte Coefficients (SC) of test set embeddings in different numbers of clustering centers under different training strategies. The X axis refers to the proportion between number of cluster centers and the size of the test set.

We evaluate all methods on the unlabelled DeepCAD test set and report SSE and SC under different cluster numbers in different methods. Results are plotted in Figure 4. From the plots we can observe that our model's SSE and SC is better than both unimodal methods and baselines. Table 2 shows quantitative results where our model has reduced mean SSE and has increased mean SC by a large margin, illustrating that our model has combined knowledge of both CAD sequences and geometry, and the latent space of our model is far more aligned and uniform than baselines and unimodal methods.

Method	Mean SSE ↓	Mean SC ↑
Unimodal Methods		
Pointnet + + w/ simcse‡	64.11	0.213
Original DeepCAD	14.72	0.058
Baseline Methods		
CLIP* [42]	14.35	0.196
CrossPoint* [1]	13.90	0.237
Our Method		
MultiCAD	9.43	0.263
MultiCAD w/ Aug.	**9.23**	**0.269**

Table 2: Average Sum of Square Errors and Silhoutte Coefficients of DeepCAD test set embeddings under different numbers of clustering centers. Each row refers to a different training strategy. ∗ indicates applying the modality interaction methods to the feature extractors of CAD sequences and pointclouds.

To further illustrate the advantages of our proposed model on representation space, we conduct an experiment on multimodal CAD retrieval, which is challenging since similar CAD geometries may correspond to significantly different CAD shapes and vice versa. The experiment is conducted on DeepCAD and Fushion 360 test set where both construction sequence and point cloud of each CAD model are converted into a sequence embedding and a point-cloud embedding via the corresponding encoders. Each embedding is used as a query while embeddings from the opposite modality serve as candidates. We find K Nearest Neighbors in representation space under Euclidean Distance between embeddings and report the results. The accuracy of sequence retrieval is calculated via averaging command and parameter accuracy of CAD construction sequences and IoU between pointclouds are utilized to measure accuracy of retrieved CAD shapes. As is shown in Table 1, MultiCAD has largely surpassed both baselines and unimodal results.

In the following sections, we will exhibit our model's performance on downstream tasks: converting CAD point clouds to CAD sequences and downstream vision recognition tasks on geometric feature extractors.

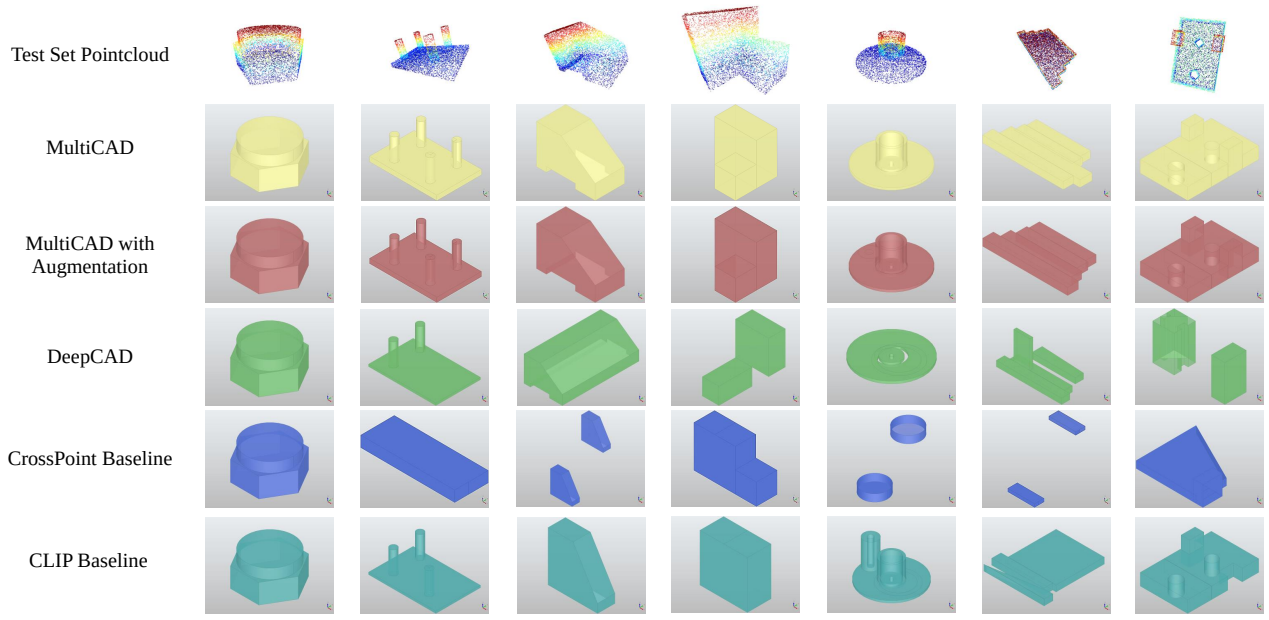

Figure 5: Qualitative results of converting pointclouds to CAD construction sequences. Pointclouds are from DeepCAD test set.

Method	$\text{Acc}_{(ct)}$ ↑	$\text{Acc}_{(cp)}$ ↑	Median CD↓	Invalid Rate↓
Methods tested on DeepCAD dataset follow:				
Unimodal Method				
DeepCAD†	80.39	69.60	0.919	15.44
Baseline Methods				
CrossPoint* [1]	71.67	58.04	2.873	24.83
CLIP* [1]	69.3	57.48	2.576	23.96
Our Method				
MultiCAD	88.21	79.52	0.875	13.67
MultiCAD w/ Aug.	**89.43**	**79.81**	**0.809**	**11.46**
Methods tested on Fushion360 dataset follow:				
Unimodal Method				
DeepCAD†	67.09	57.65	8.92	25.17
Baseline Methods				
CrossPoint* [1]	61.85	55.76	26.11	39.55
CLIP* [42]	67.03	56.39	14.32	21.24
Our Method				
MultiCAD	77.84	72.44	4.33	17.95
MultiCAD w/ Aug.	**79.27**	**71.73**	**4.22**	**16.52**

Table 3: Quantitative Results for CAD Sequence Reconstruction from Pointclouds.† indicates results achieved by re-implementing the methods proposed in DeepCAD[50] along its source code. $*$ indicates applying the modality interaction methods to the feature extractors of CAD sequences and pointclouds. $\text{Acc}_{(cp)}$**,** $\text{Acc}_{(ct)}$**, and Invalid Rate are all multiplied by 100%; Median CD is multiplied by** 10^2**.**

4.4.2 Performance on Sequence Reconstruction. In this part we test the performance of converting CAD point clouds into CAD sequences, which is the starting point of user editing. The experiment is conducted on DeepCAD and Fushion360 test set. We test the likeliness between the reconstructed CAD sequence and the original one using sequence modelling metrics from DeepCAD. The experimental results of different methods are shown in Table 3. $\text{Acc}_{(ct)}$ and $\text{Acc}_{(cp)}$ stand for accuracy of commands and parameters between predicted sequence and ground truth. Median CD refers to the median Chamfer Distance between the point clouds generated by the reconstructed sequence and the ground truth. Invalid Rate represents the proportion of the decoded sequences which cannot be constructed by a CAD kernel or converted into pointclouds, serving as a validity metric for reconstruction [50]. A qualitative result of the reconstructed CAD sequences is shown in Figure 5.

From the experiment results, we can discover that MultiCAD and its augmented version has surpassed both baseline and unimodal methods and by a large margin in all metrics. This means that, for CAD models, representation containing information from both modalities is vital for multimodal translation tasks such as converting CAD point clouds into CAD sequences. Moreover, the result of the baseline method shows the necessity of training the sequence decoder simultaneously and not aligning representations from different modalities at the beginning of the training procedure.

4.4.3 Performance on Geometric Representation. In this section, we test the geometric representation ability of our multimodal contrastive learning scheme via pointcloud classification.

We first evaluate the transfer performance of unimodal methods, multimodal baselines and our method. As DeepCAD is not labelled, we pretrain all methods on DeepCAD and compare the classification performance on MCB via linear probing, namely fixing the feature extractor and tuning the linear layer only. The results are shown in Table 4. From the table we can conclude that information from CAD sequence is beneficial to CAD geometric representation. Moreover, our multimodal interaction scheme can better grasp the information from CAD construction sequences.

We also compare the geometric representation ability of our multimodal contrastive learning scheme against different models trained via supervised learning on MCB dataset. Results are listed in Table 5. In this experiment, our network is first pretrained on DeepCAD dataset via our multimodal contrastive learning scheme.

Method	Backbone	Acc_{object} ↑	Acc_{class} ↑
Unimodal			
Pointnet + + w/ simcse[#]	Pointnet++	88.62%	75.88%
Original DeepCAD	Pointnet++	91.72%	81.24%
Mulitmodal			
Crosspoint* [1]	Pointnet++	89.59%	80.35%
CLIP* [42]	Pointnet++	91.80%	81.26%
MultiCAD	Pointnet++	92.30%	82.02%
MultiCAD w/ Aug	Pointnet++	**93.06**%	**82.87**%

Table 4: Classification performance of transfer learning on MCB dataset. All methods are pretrained on DeepCAD dataset and tested on MCB via linear probing. # indicates the pretrained Pointnet++ illustrated in Section 4.2. ∗ indicates applying the modality interaction methods to the feature extractors of CAD sequences and pointclouds.

Method	Backbone	Acc_{object} ↑	Acc_{class} ↑
Pointnet + +[†]	Pointnet++	87.45%	73.68%
PointCNN[†] [26]	PointCNN	93.89%	81.85%
SpiderCNN[†] [54]	SpiderCNN	93.59%	79.70%
MultiCAD w/ Aug[§]	Pointnet++	**94.55**%	**85.13**%

Table 5: Classification results of methods with different backbones on MCB dataset. † indicates supervised training on MCB dataset. § indicates pretraining on DeepCAD dataset and finetune the entire network on MCB dataset.

The entire network is then finetuned on MCB. The table shows that the result on both Acc_{class} and Acc_{object} have even surpassed benchmark results with far stronger backbones such as PointCNN and SpiderCNN significantly. Such result is aspiring since the models in MCB is far more complicated than most of the models in DeepCAD dataset. Moreover, the great advantages stronger backbones bring to classification performance can be wiped out when information from the sequence modality is introduced. The results in Table 4 and Table 5 further show that introducing sequence modality as well as batchwise Optimal Transport plan have granted feature extractor more semantic information such as shape details, which is the core distinction between categories in MCB [21].

4.5 Ablation study

In this part we show the effectiveness of our model design as well as the training strategy adopted by MultiCAD. For different training strategies, we report multimodal retrieval results, which is a direct measurement of the multimodal representation space. The results are shown in Table 6 where each − represents removing a component from model shown in the previous entry in the table.

From Table 6 we have several observations. First, contrastive learning creates an aligned and uniform representation space that improves multimodal retrieval by separating embeddings, even without cross-modal information. This is evident from the ∼ 10% R@1 performance gain between methods E and F.

Moreover, our proposed MMCL introduces geometric information to the CAD representation, which draws near CAD models in the representation space only when they are both alike in CAD geometry and CAD sequences. The introduction of geometric information makes multimodal retrieval possible, thus yielding ∼ 20% performance gains at R@1, as shown between method D and E, which is also the biggest improvement. As for training strategies, our proposed two-stage training strategy requires no explicit alignment between multimodal representations, which improves the training stability of multimodal contrastive learning. This brings about 4 ∼ 8% performance gains at R@1, as shown between method C and D. The introduction of focal loss makes the model focus on models that are similar in one modality but not alike in the other during MMCL, bringing ∼ 2% performance gains at R@1.

ID	Methods	DeepCAD (Test Set) Pointcloud → Sequence R@1	Pointcloud → Sequence R@5	Sequence → Pointcloud R@1	Sequence → Pointcloud R@5
A.	MultiCAD	72.44%	80.19%	80.82%	92.18%
B.	*Training Strategy* − OT at geometry pretext task	70.37%	78.98%	77.85%	88.27%
C.	− Focal Loss	68.03%	76.94%	75.78%	86.23%
D.	− Two Stage Training Strategy	64.54%	74.78%	67.73%	81.13%
E.	*Modality Interaction* − Pointcloud Introduction*	45.62%	66.12%	46.01%	67.35%
F.	*Contrastive Learning* − Contrastive Learning*	36.48%	61.87%	37.66%	52.30%

Table 6: Ablation study on multimodal retrieval. - means removing a component from the model exhibited in the previous entry of the table. ∗ represents training a pointnet++ to fit the representation space when training completes.

Finally, adopting optimal transport at geometry pretext task makes pointcloud representation approach neighborhood of corresponding sequence representation and encodes more geometric details in pointcloud representation, which further brings forth 3 ∼ 4% performance gains at R@1 and R@5 in pointcloud retrieval.

5 CONCLUSION AND FUTURE WORK

This paper unveils the multi-modality of CAD models and shows the potential of considering both CAD construction sequence and its geometry when learning CAD representations for effective information and knowledge management. We define a novel similarity measurement strategy and propose a multimodal contrastive loss named MMCL. A representation learning model named MultiCAD is designed based on MMCL. Two pretext tasks are built along with multimodal contrastive learning to make the representation applicable to downstream tasks. Extensive experiments on both CAD sequences and point clouds show that MultiCAD has remarkable performance on both sequence and geometric downstream tasks. Future work includes adopting more CAD data formats such as B-rep as well as modelling the geometry of high complexity parts.

ACKNOWLEDGMENTS

This work was supported by National Key Research and Development Program of China, No.2018YFB1402600.

REFERENCES

[1] Mohamed Afham, Isuru Dissanayake, Dinithi Dissanayake, Amaya Dharmasiri, Kanchana Thilakarathna, and Ranga Rodrigo. 2022. Crosspoint: Self-supervised cross-modal contrastive learning for 3d point cloud understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 9902–9912.

[2] Rishabh Bhardwaj, Tushar Vaidya, and Soujanya Poria. 2022. KNOT: Knowledge Distillation Using Optimal Transport for Solving NLP Tasks. In *Proceedings of the 29th International Conference on Computational Linguistics*. International Committee on Computational Linguistics, Gyeongju, Republic of Korea, 4801–4820. https://aclanthology.org/2022.coling-1.425

[3] Jorge D Camba, Manuel Contero, and Pedro Company. 2016. Parametric CAD modeling: An analysis of strategies for design reusability. *Computer-Aided Design* 74 (2016), 18–31.

[4] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. 2015. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012* (2015).

[5] Liqun Chen, Dong Wang, Zhe Gan, Jingjing Liu, Ricardo Henao, and Lawrence Carin. 2021. Wasserstein contrastive representation distillation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 16296–16305.

[6] Nenglun Chen, Lingjie Liu, Zhiming Cui, Runnan Chen, Duygu Ceylan, Changhe Tu, and Wenping Wang. 2020. Unsupervised learning of intrinsic structural representation points. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 9121–9130.

[7] Ruihang Chu, Xiaoqing Ye, Zhengzhe Liu, Xiao Tan, Xiaojuan Qi, Chi-Wing Fu, and Jiaya Jia. 2022. TWIST: Two-Way Inter-label Self-Training for Semi-supervised 3D Instance Segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 1100–1109.

[8] Ching-Yao Chuang, Joshua Robinson, Yen-Chen Lin, Antonio Torralba, and Stefanie Jegelka. 2020. Debiased contrastive learning. *Advances in neural information processing systems* 33 (2020), 8765–8775.

[9] Blender Online Community. 2018. *Blender - a 3D modelling and rendering package*. Blender Foundation, Stichting Blender Foundation, Amsterdam. http://www.blender.org

[10] Jiequan Cui, Zhisheng Zhong, Shu Liu, Bei Yu, and Jiaya Jia. 2021. Parametric contrastive learning. In *Proceedings of the IEEE/CVF international conference on computer vision*. 715–724.

[11] Marco Cuturi. 2013. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems* 26 (2013).

[12] Jiali Duan, Liqun Chen, Son Tran, Jinyu Yang, Yi Xu, Belinda Zeng, and Trishul Chilimbi. 2022. Multi-modal alignment using representation codebook. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 15651–15660.

[13] Yaroslav Ganin, Sergey Bartunov, Yujia Li, Ethan Keller, and Stefano Saliceti. 2021. Computer-aided design as language. *Advances in Neural Information Processing Systems* 34 (2021), 5885–5897.

[14] Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings. *arXiv preprint arXiv:2104.08821* (2021).

[15] Wen Gao, Xuanming Zhang, Qiushi He, Borong Lin, and Weixin Huang. 2022. Command prediction based on early 3D modeling design logs by deep neural networks. *Automation in Construction* 133 (2022), 104026.

[16] Rohit Girdhar, David F Fouhey, Mikel Rodriguez, and Abhinav Gupta. 2016. Learning a predictable and generative vector representation for objects. In *European Conference on Computer Vision*. Springer, 484–499.

[17] Kai Han, Yunhe Wang, Hanting Chen, Xinghao Chen, Jianyuan Guo, Zhenhua Liu, Yehui Tang, An Xiao, Chunjing Xu, Yixing Xu, et al. 2022. A survey on vision transformer. *IEEE transactions on pattern analysis and machine intelligence* (2022).

[18] Kai Han, An Xiao, Enhua Wu, Jianyuan Guo, Chunjing Xu, and Yunhe Wang. 2021. Transformer in transformer. *Advances in Neural Information Processing Systems* 34 (2021), 15908–15919.

[19] Zehao Huang and Naiyan Wang. 2017. Like what you like: Knowledge distill via neuron selectivity transfer. *arXiv preprint arXiv:1707.01219* (2017).

[20] Longlong Jing, Ling Zhang, and Yingli Tian. 2021. Self-supervised feature learning by cross-modality and cross-view correspondences. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 1581–1591.

[21] Sangpil Kim, Hyung-gun Chi, Xiao Hu, Qixing Huang, and Karthik Ramani. 2020. A Large-scale Annotated Mechanical Components Benchmark for Classification and Retrieval Tasks with Deep Neural Networks. In *Proceedings of 16th European Conference on Computer Vision (ECCV)*.

[22] Sebastian Koch, Albert Matveev, Zhongshi Jiang, Francis Williams, Alexey Artemov, Evgeny Burnaev, Marc Alexa, Denis Zorin, and Daniele Panozzo. 2019. Abc: A big cad model dataset for geometric deep learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 9601–9611.

[23] Changjian Li, Hao Pan, Adrien Bousseau, and Niloy J Mitra. 2022. Free2CAD: parsing freehand drawings into CAD commands. *ACM Transactions on Graphics (TOG)* 41, 4 (2022), 1–16.

[24] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv preprint arXiv:2301.12597* (2023).

[25] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. 2022. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International Conference on Machine Learning*. PMLR, 12888–12900.

[26] Yangyan Li, Rui Bu, Mingchao Sun, Wei Wu, Xinhan Di, and Baoquan Chen. 2018. Pointcnn: Convolution on x-transformed points. *Advances in neural information processing systems* 31 (2018).

[27] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2017. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*. 2980–2988.

[28] Zhidan Liu, Zhen Xing, Xiangdong Zhou, Yijiang Chen, and Guichun Zhou. 2022. 3D-Augmented Contrastive Knowledge Distillation for Image-Based Object Pose Estimation. In *Proceedings of the 2022 International Conference on Multimedia Retrieval* (Newark, NJ, USA) *(ICMR '22)*. Association for Computing Machinery, New York, NY, USA, 508–517. https://doi.org/10.1145/3512527.3531359

[29] Sijie Mai, Ya Sun, Ying Zeng, and Haifeng Hu. 2023. Excavating multimodal correlation for representation learning. *Information Fusion* 91 (2023), 542–555. https://doi.org/10.1016/j.inffus.2022.11.003

[30] Muhammad Arslan Manzoor, Sarah Albarri, Ziting Xian, Zaiqiao Meng, Preslav Nakov, and Shangsong Liang. 2023. Multimodality Representation Learning: A Survey on Evolution, Pretraining and Its Applications. *arXiv preprint arXiv:2302.00389* (2023).

[31] Alex Nichol, Heewoo Jun, Prafulla Dhariwal, Pamela Mishkin, and Mark Chen. 2022. Point-E: A System for Generating 3D Point Clouds from Complex Prompts.

[32] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748* (2018).

[33] Wamiq Para, Shariq Bhat, Paul Guerrero, Tom Kelly, Niloy Mitra, Leonidas J Guibas, and Peter Wonka. 2021. Sketchgen: Generating constrained cad sketches. *Advances in Neural Information Processing Systems* 34 (2021), 5077–5088.

[34] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. 2022. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988* (2022).

[35] Omid Poursaeed, Tianxing Jiang, Han Qiao, Nayun Xu, and Vladimir G. Kim. 2020. Self-supervised Learning of Point Clouds via Orientation Estimation. *arXiv: Computer Vision and Pattern Recognition* (2020).

[36] Charles R Qi, Li Yi, Hao Su, and Leonidas J Guibas. 2017. PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. *arXiv preprint arXiv:1706.02413* (2017).

[37] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. Improving language understanding by generative pre-training. (2018).

[38] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI blog* 1, 8 (2019), 9.

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

[40] Yongming Rao, Jiwen Lu, and Jie Zhou. 2020. Global-local bidirectional reasoning for unsupervised representation learning of 3d point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 5376–5385.

[41] Peter J Rousseeuw. 1987. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics* 20 (1987), 53–65.

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

[43] Jonathan Sauder and Bjarne Sievers. 2019. Self-Supervised Deep Learning on Point Clouds by Reconstructing Space. *Neural Information Processing Systems* (2019).

[44] Yonglong Tian, Chen Sun, Ben Poole, Dilip Krishnan, Cordelia Schmid, and Phillip Isola. 2020. What makes for good views for contrastive learning? *Advances in Neural Information Processing Systems* 33 (2020), 6827–6839.

[45] Mikaela Angelina Uy, Yen yu Chang, Minhyuk Sung, Purvi Goel, Joseph Lambourne, Tolga Birdal, and Leonidas Guibas. 2022. Point2Cyl: Reverse Engineering 3D Objects from Point Clouds to Extrusion Cylinders. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.

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

[47] Kehan Wang, Jia Zheng, and Zihan Zhou. 2022. Neural Face Identification in a 2D Wireframe Projection of a Manifold Object. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 1622–1631.

[48] Karl DD Willis, Pradeep Kumar Jayaraman, Hang Chu, Yunsheng Tian, Yifei Li, Daniele Grandi, Aditya Sanghi, Linh Tran, Joseph G Lambourne, Armando Solar-Lezama, et al. 2022. Joinable: Learning bottom-up assembly of parametric cad joints. In *Proceedings of the IEEE/CVF Conference on Computer Vision and*

Pattern Recognition. 15849–15860.

[49] Karl D. D. Willis, Yewen Pu, Jieliang Luo, Hang Chu, Tao Du, Joseph G. Lambourne, Armando Solar-Lezama, and Wojciech Matusik. 2021. Fusion 360 Gallery: A Dataset and Environment for Programmatic CAD Construction from Human Design Sequences. *ACM Transactions on Graphics (TOG)* 40, 4 (2021).

[50] Rundi Wu, Chang Xiao, and Changxi Zheng. 2021. Deepcad: A deep generative network for computer-aided design models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 6772–6782.

[51] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 2015. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 1912–1920.

[52] Xu Xie, Fei Sun, Zhaoyang Liu, Shiwen Wu, Jinyang Gao, Jiandong Zhang, Bolin Ding, and Bin Cui. 2022. Contrastive learning for sequential recommendation. In *2022 IEEE 38th international conference on data engineering (ICDE)*. IEEE, 1259–1273.

[53] Xiang Xu, Karl D.D. Willis, Joseph G Lambourne, Chin-Yi Cheng, Pradeep Kumar Jayaraman, and Yasutaka Furukawa. 2022. SkexGen: Autoregressive Generation of CAD Construction Sequences with Disentangled Codebooks. In *Proceedings of the 39th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 162)*, Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (Eds.). PMLR, 24698–24724. https://proceedings.mlr.press/v162/xu22k.html

[54] Yifan Xu, Tianqi Fan, Mingye Xu, Long Zeng, and Yu Qiao. 2018. Spidercnn: Deep learning on point sets with parameterized convolutional filters. In *Proceedings of the European conference on computer vision (ECCV)*. 87–102.

[55] Kaizhi Yang and Xuejin Chen. 2021. Unsupervised learning for cuboid shape abstraction via joint segmentation from point clouds. *ACM Transactions on Graphics (TOG)* 40, 4 (2021), 1–11.

[56] Y. Zhang, J. Chen, and D. Huang. 2022. CAT-Det: Contrastively Augmented Transformer for Multimodal 3D Object Detection. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, Los Alamitos, CA, USA, 898–907. https://doi.org/10.1109/CVPR52688.2022.00098

[57] Na Zhao, Tat-Seng Chua, and Gim Hee Lee. 2020. Sess: Self-ensembling semi-supervised 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 11079–11087.

[58] Roland S Zimmermann, Yash Sharma, Steffen Schneider, Matthias Bethge, and Wieland Brendel. 2021. Contrastive learning inverts the data generating process. In *International Conference on Machine Learning*. PMLR, 12979–12990.