Self-Supervised Feature Learning from Partial Point Clouds via Pose Disentanglement

Meng-Shiun Tsai[1]* Pei-Ze Chiang[1]* Yi-Hsuan Tsai[2] Wei-Chen Chiu[1]
s11021098@gmail.com ztex080104518@gmail.com wasidennis@gmail.com walon@cs.nctu.edu.tw

***Abstract*— Self-supervised learning on point clouds has gained a lot of attention recently, since it addresses the label-efficiency and domain-gap problems on point cloud tasks. In this paper, we propose a novel self-supervised framework to learn informative features from partial point clouds. We leverage partial point clouds scanned by LiDAR that contain both content and pose attributes, and we show that disentangling such two factors from partial point clouds enhances feature learning. To this end, our framework consists of three main parts: 1) a completion network to capture holistic semantics of point clouds; 2) a pose regression network to understand the viewing angle where partial data is scanned from; 3) a partial reconstruction network to encourage the model to learn content and pose features. To demonstrate the robustness of the learnt feature representations, we conduct several downstream tasks including classification, part segmentation, and registration, with comparisons against state-of-the-art methods. Our method not only outperforms existing self-supervised methods, but also shows a better generalizability across synthetic and real-world datasets.**

I. Introduction

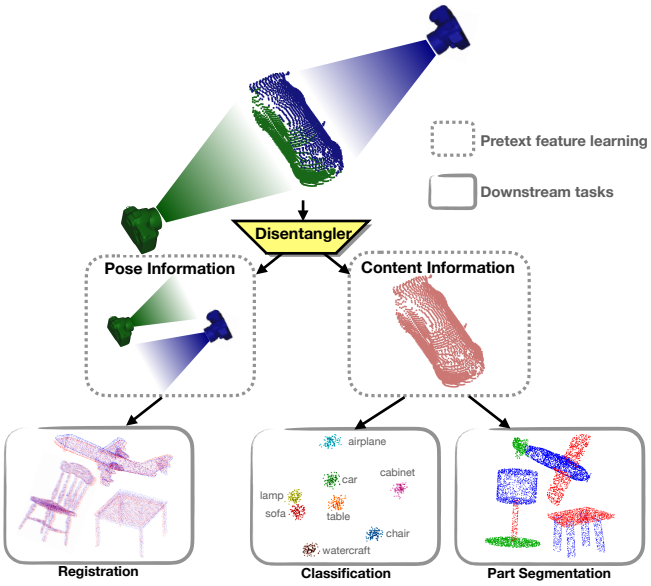


Fig. 1. We propose to learn the representation of 3D point cloud data in a self-supervised manner via point cloud completion. Our model learns to disentangle the feature into the content and pose parts, where the former is beneficial for the downstream tasks of classification and part segmentation, while the latter helps to improve the task of point cloud registration.

Point clouds provide one of the most intuitive representations for 3D object models, and they are extensively adopted as the data format for the recognition tasks in different application scenarios, such as autonomous vehicles, robotics, and architectonics. Recently, with the rapid development of deep learning techniques, various network architectures are proposed for learning effective features of the point cloud data, e.g., PointNet [1], PointNet++ [2], and DGCNN [3]. However, the success of learning the recognition models based on these feature backbones typically relies on the large-scale supervised dataset, in which it could be quite expensive to manually collect the ground truth labels and the learnt feature representations of point cloud data are less generalizable across different tasks.

These potential issues motivate various research development in learning effective feature representations from point clouds via unsupervised or self-supervised manner, such as solving jigsaw puzzles [4], [5], contrastive learning [6], [7], and local structure modeling [8], [9]. In this paper, we aim to seek for a different solution that is also intuitive and suitable for the point cloud data. Inspired by the success of feature learning in the image [10] and the natural language [11] domains, which have been developed for years, we adopt an analogous strategy of image inpainting [10], but in a 3D point cloud version via *learning to complete* the partial point cloud of a 3D object. As such, similar benefits for feature learning as shown in image inpainting [10] can be achieved by point cloud completion, which not only needs the semantic understanding of the point cloud data, but also learns better holistic feature representations via reconstructing the plausible missing parts.

However, simply completing the partial point cloud, as in the case of image domain, may not suffice the need for learning effective feature representations, as the point cloud data has different characteristics from images. That is, given a 3D object point cloud, there could be multiple angles to view this data, such that each angle produces a different partial point cloud (in the top of Figure 1). Therefore, although the final completed point cloud is the same for all the partial point clouds obtained from the same object, the completion network may require to learn a different feature representation for each view-angle during the completion process. To take view-angle into consideration, instead of using existing point cloud completion frameworks [12], [13], which only produce one feature representation that *entangles* both the view-angle information and the content cue for the point cloud data, we propose to *disentangle* these two factors and learn more effective feature representations in a self-supervised manner (see Figure 1).

[1]Department of Computer Science, National Chiao Tung University
[2]Phiar Technologies
∗ Both authors contributed equally to the pape
· https://ms-tsai.github.io/Partial-Point-Clouds-Disentangler-Project-Page/

Specifically, we utilize two encoders to extract the content and pose (i.e., view-angle) features individually from each partial point cloud input. Then, in addition to performing completion using the content feature, the pose feature should be able to predict the pose of the input data and guide the reconstruction in specific view-angle. Thus, we introduce another two modules: 1) a pose regression network to predict the view-angle of the partial point cloud using the pose feature; and 2) a partial reconstruction network to recover a specific partial point cloud, through the combination of a content feature from a view-angle i and a pose feature from another view-angle j. As a result, no matter which view-angle the content feature is extracted from, the partial reconstruction should be mainly guided by the pose feature. Therefore, our framework encourages the content and pose features to be more compact on themselves while being more disentangled to each other, which better facilitates the feature representation learning process.

We conduct extensive experiments to show the effectiveness of our self-supervised framework: 1) our learned content feature is able to provide favorable performance against state-of-the-art methods on the downstream tasks of classification and part segmentation, and 2) our learned pose features contribute to the pose-relevant downstream task of point cloud registration. The main contributions of our work are summarized as follows:

- We propose a self-supervised framework based on point cloud completion for learning the feature representations from the partial point cloud data.
- We develop a pipeline to disentangle point cloud feature representations into the content and pose factors, which enables the model to learn effective features.
- We show that the learnt content and pose features improve several downstream tasks (i.e., classification, part segmentation, and registration), while showing the generalizability across synthetic and real-world datasets.

II. Related Work

Supervised Learning for Point Cloud Data. Due to the popularity of 3D scanning technologies and the advancement of deep learning techniques, learning to extract features from the point cloud data for recognition tasks is one of the active research topics. One early attempt via deep networks is PointNet [1] which uses the max-pooling operation to aggregate the point-wise features into the global one, thus achieving the permutation invariant property. While PointNet neglects the local structure between neighboring points, its successor, PointNet++ [2], adopts the hierarchical neural network to progressively extract the features of a point cloud from multiple resolutions. Furthermore, DGCNN [3] leverages the graph neural network [14] to process the nearby points and their edges via the k-nearest neighbor algorithm in each of the intermediate feature spaces. Other methods that also try to capture the relationships among the local regions in point clouds are developed in [15], [16], [17].

However, all the aforementioned works are based on the complete point cloud with supervised information, which is costly to collect in terms of time and expense. Even when we are able to use the synthetic dataset where the supervision is easier to obtain, the feature learnt from the synthetic data could be less generalizable to the real-world data due to the domain gap [18]. In this work, we thereby focus on the self-supervised learning scheme to learn effective representations that can generalize better across datasets.

Unsupervised / Self-Supervised Feature Learning for Point Cloud Data. Recently, various works have been proposed to explore the unsupervised and self-supervised schemes for learning feature representations of 3D point clouds, where the training objectives or the labels are produced from the data itself. An intuitive objective is derived via performing the self-reconstruction [19], [20], in which the semantic and compact feature representations are learnt in the bottleneck between the encoding and decoding processes. For instance, the encoder of 3DCapsuleNet [21] is designed to process 3D point clouds and aggregate the final latent capsules based on the dynamic routing, followed by the decoder to reconstruct the original point cloud composed of multiple point patches. [22] proposes an autoencoder architecture where the rich point-wise features in multiple stages are progressively produced through the seed generation module. Eckart *et al.* [23] further extend the traditional autoencoder paradigm to form the bottleneck layer being modeled by a discrete generative model. In addition to the self-reconstruction, MAP-VAE [8] proposes the multi-angle analysis to introduce the local self-reconstruction, leading a better modeling on the local geometry of point clouds.

Another effective technique is stemmed from contrastive learning. For instance, PointContrast [7] proposes to use a pretext task, where the mapping between two point clouds taken by viewing from different perspectives should be consistent even after performing the random geometric transformations on them. [24] relies on the similar consistency but extends to include the spatial augmentations on the data, and adopts the self-ensembling mechanism to drive the learning. Other methods are inspired by the techniques originally applied on 2D images. For instance, Sauder *et al.* [5] adopts the jigsaw puzzle idea on 3D point cloud, in which the point cloud is decomposed into multiple parts followed by random permutation, and then the network is trained by putting these parts back to their original positions. Similarly, Alliegro *et al.* [4] further integrate multi-task learning into the same framework as [5]. In addition, [25] destroys shape parts and learns the feature representation via the process of distinguishing the destroyed parts and restoring them.

Different from the above methods, our approach is inspired by the idea of 2D image inpainting [10] but in a 3D version, in which we propose to self-learn the completion process from partial point clouds as the pretext task. Furthermore, beyond point cloud completion, we extend our framework to consider the pose information, so that features can be disentangled during the completion process to learn effective feature representations. We note that, a recent work [26] also adopts the idea of point cloud completion to drive feature

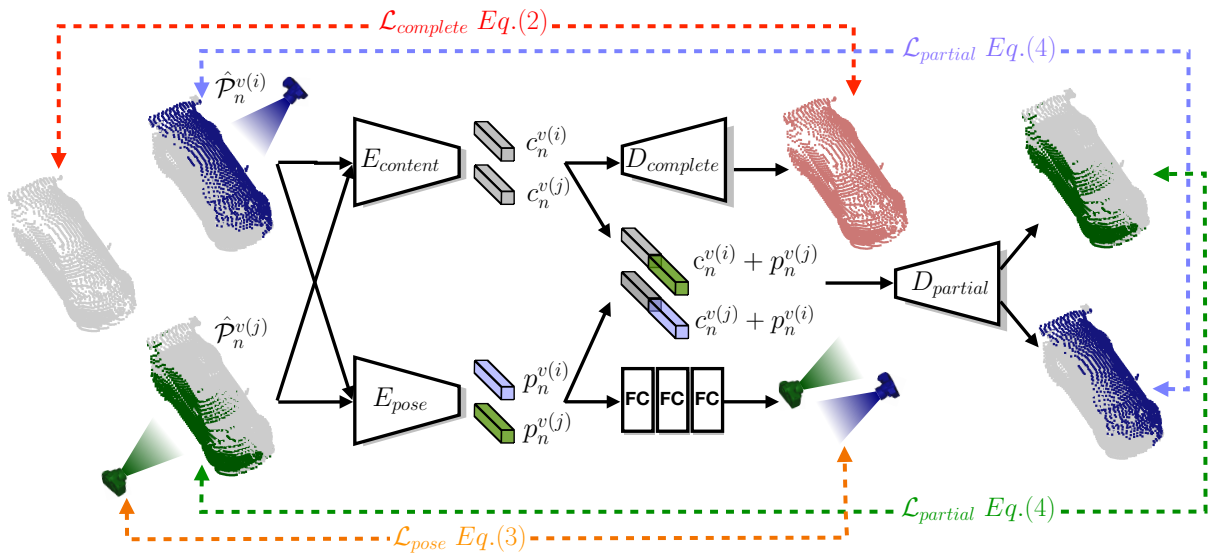


Fig. 2. Our framework aims to learn two distinct features (c_n, p_n) via disentanglement that consists of three main branches: 1) a completion branch containing a content encoder $E_{content}$ and a decoder $D_{complete}$ that learns the content features c_n, through completing partial point clouds of the same 3D object. 2) a pose regression branch with a pose encoder E_{pose} that learns the pose features p_n, through predicting view-angles where a partial point cloud is scanned from. 3) a partial reconstruction branch $D_{partial}$, which leverages combined content features with exchanged pose features to reconstruct the partial inputs. Note that "+" denotes the concatenation operation and dashed lines in colors are the loss functions we employ for three branches.

learning but does not fully leverage the pose information, in which this work can be treated as an ablated variant of our model. Later in experiments we provide the ablation study to verify the benefit of our model design with respect to this work.

Feature Disentanglement. Disentanglement aims to decompose the feature representation into multiple parts for better explaining the factors behind data variation. Numerous methods are developed in 2D images, such as the conditional GAN, auxiliary classifier GAN [27], InfoGAN [28] and conditional VAE [29], all of which are used to disentangle the latent space of GAN [30] and VAE [31]. Moreover, introducing disentanglement into representation learning benefits various computer vision applications. For instance, [32], [33] disentangle the pose information from person images to obtain the purified feature of person, which improves the performance in the task of person re-identification. Zhou *et al.* [34] factors out the shape and pose features from the 3D mesh data to boost the performance in the tasks of pose transfer and shape retrieval. In our proposed framework, we disentangle the content and pose information of the 3D point cloud representation, in which the content feature is beneficial for the downstream tasks of classification and part segmentation, while the pose feature boosts the task of point cloud registration.

III. Proposed Method

As motivated in the introduction, the objective of our proposed framework is to perform self-supervised learning via partial point cloud completion and learn the 3D point cloud feature representations, which is decomposed into the content and pose parts. The overall architecture of our proposed framework is illustrated in Figure 2, where there are three main branches: completion branch, pose regression branch, and partial reconstruction branch. We now detail our proposed framework in the following subsections.

A. Completion Branch

The purpose of this branch is to perform 3D partial point cloud completion, which is proposed in this paper as an analogous strategy to 2D image inpainting [10] for self-learning feature representations. To this end, given a dataset composed of N complete point clouds of 3D objects $\mathbb{P} = \{\mathcal{P}_n\}_{n=1}^{N}$, for each of the complete point cloud $\mathcal{P}_n$, we are able to generate many of its corresponding partial point clouds $\{\hat{\mathcal{P}}_n^{v(k)}\}_{k=1}^{K}$ as being scanned from K different pre-defined viewpoints $\{v(k)\}_{k=1}^{K}$. Then, for an input partial point cloud $\hat{\mathcal{P}}_n^{v(k)}$, we use the content encoder $E_{content}$ to extract its content feature $c_n^{v(k)} = E_{content}(\hat{\mathcal{P}}_n^{v(k)})$, followed by the completion decoder $D_{complete}$ to reconstruct the corresponding complete point cloud $\mathcal{P}_n$ from $c_n^{v(k)}$.

Here, we adopt the standard point cloud feature extractor (e.g., PointNet [1] or DGCNN [3]) as the content encoder $E_{content}$. For the completion decoder $D_{complete}$, we utilize a morphing-based decoder in MSN [12] to predict $\mathcal{P}_n$. To measure the completion quality, we use Earth Mover's Distance (EMD) [35] loss (which is typically approximated via the auction algorithm [36] in practical implementation). Therefore, given two point clouds $\mathcal{P}_i, \mathcal{P}_j \in \mathbb{R}^3$ with an equal size $|\mathcal{P}_i| = |\mathcal{P}_j|$, the EMD loss function is:

$$d_{EMD}(\mathcal{P}_i, \mathcal{P}_j) = \min_{\phi(\mathcal{P}_i)\to\mathcal{P}_j} \sum_{x\in\mathcal{P}_i} \|x - \phi(x)\|_2 , \quad (1)$$

where ϕ is an optimal bijection function that allows each point in $\mathcal{P}_i$ to find a nearest corresponding point in $\mathcal{P}_j$.

Moreover, in addition to the EMD loss that aims at guiding the entire generated point cloud to maximally cover the ground truth one, we also adopt the expansion loss $\mathcal{L}_{expan}$ as proposed in [12] to better handle local regions in the point

cloud. As a result, our loss function in the completion branch for a point cloud $\mathcal{P}_n$ in a view-angle $v(k)$ can be written as:

$$\begin{aligned}\mathcal{L}_{complete} &= d_{EMD}(\mathcal{P}_n, D_{complete}(c_n^{v(k)})) \\ &+ \lambda_{ex} * \mathcal{L}_{expan}(D_{complete}(c_n^{v(k)})).\end{aligned} \tag{2}$$

where we follow [12] to set $\lambda_{ex} = 0.1$.

B. Pose Regression Branch

Although the self-supervised objective in Section III-A enables our completion model to learn feature representations as an initial step, it is still insufficient to fully exploit the rich information contained in partial point cloud captured under various view-angles. Thus, in the pose regression branch, we aim to learn the pose feature that can predict the pose of the partial data and assist in the feature learning process. We first use the standard point cloud feature extractor (e.g., PointNet [1] or DGCNN [3]) as the pose encoder E_{pose} to extract the pose feature $p_n^{v(k)} = E_{pose}(\hat{\mathcal{P}}_n^{v(k)})$, which represents the feature for the view-angle $v(k)$ of the point cloud $\mathcal{P}_n$. Then, three fully-connected layers are constructed as the pose regressor for predicting the view-angle.

Specifically, our pose regression branch is learnt to predict the camera position where the partial data is scanned from. We define our camera position in the spherical coordinate system (γ, θ, ϕ), where we only rotate the data along the polar angle θ and the azimuthal angle ϕ. Note that we fix the radial distance γ to have a consistent point cloud density and use the degree as unit of θ and ϕ. Although we pre-define the poses and can consider pose prediction as a classification problem, we find that retaining it as the regression task serves better for feature learning To optimize the pose regression branch, we use the mean square error (MSE). Given a partial point cloud $\hat{\mathcal{P}}_n^{v(k)}$, our pose regression branch should predict the camera position $\hat{v}(k)$. The loss function thus can be written as:

$$\mathcal{L}_{pose} = \|v(k) - \hat{v}(k)\|_2. \tag{3}$$

C. Partial Reconstruction Branch

Based on Section III-A and III-B, we have constructed two self-supervised objectives, individually for the point cloud completion and the pose regression tasks. However, there is still a lack of how to make connections between content and pose features, such that the learnt feature representation in each branch is more compact and meaningful. To achieve it, we propose to disentangle these two factors via the partial reconstruction branch.

Given two partial point clouds $\hat{\mathcal{P}}_n^{v(i)}$ and $\hat{\mathcal{P}}_n^{v(j)}$ derived from the same complete point cloud $\mathcal{P}_n$ but with different viewpoints (i.e., $i \neq j$), we hypothesize that the content features from two views should be similar to each other (i.e., $c_n^{v(i)} \sim c_n^{v(j)}$), as they are from the same 3D object and are used to obtain the same completion output. As such, if we combine one of the content features with a specific pose feature, this combined feature should follow the pose information to reconstruct the partial point cloud, regardless of which content feature is selected.

To realize this objective, we use a partial decoder $D_{partial}$, which takes the concatenated content and pose features as the input, but with different viewpoints, e.g., concatenation of $c_n^{v(i)}$ and $p_n^{v(j)}$, to reconstruct a partial point cloud $\hat{\mathcal{P}}_n^{v(j)}$. The loss of this partial point cloud reconstruction is based on the EMD loss (1):

$$\begin{aligned}\mathcal{L}_{partial} &= d_{EMD}(\hat{\mathcal{P}}_n^{v(i)}, D_{partial}(c_n^{v(j)}, p_n^{v(i)})) \\ &+ d_{EMD}(\hat{\mathcal{P}}_n^{v(j)}, D_{partial}(c_n^{v(i)}, p_n^{v(j)})),\end{aligned} \tag{4}$$

where we consider two terms by exchanging the content and pose features in two viewpoints.

D. Model Pre-training as Pretext

Overall Objective. Without the use of any annotations, we combine all the aforementioned loss functions from the three branches as our final self-supervised objectives:

$$\mathcal{L}_{all} = \lambda_c \mathcal{L}_{complete} + \lambda_{pa} \mathcal{L}_{partial} + \lambda_{po} \mathcal{L}_{pose}, \tag{5}$$

where the hyperparameters λ balance between losses, for all the experiments, we set $\lambda_c = \lambda_{pa} = 0.5$ to equally balance two reconstruction-based objectives and set $\lambda_{po} = 0.01$.

Dataset, Data Generation, Implementation Details. We follow the standard self-supervised setting in [5], [37] and use the ShapeNetCore.v1 dataset [38] for the pretext task in model pre-training, where our learnt content and pose encoders (i.e., $E_{content}$ and E_{pose}) will be used to extract the content and pose features from the point cloud data respectively to perform various downstream tasks. In the pretext task, we follow the similar experimental settings as in MSN [12] for point cloud completion, where we choose a total of $35,827$ 3D models of 8 classes (i.e., table, chair, car, airplane, sofa, lamp, watercraft, and cabinet) from the ShapeNet. We use Blender software [39] to render the partial point clouds for each of the 3D models as being scanned from 26 pre-defined viewpoints. The 3D point coordinates in the point cloud are normalized into a unit sphere (i.e., within $[-1, 1]$). For each point cloud, we uniformly sample 1024 points. The pre-training of our proposed model on the ShapeNet runs for 50 epochs with the Adam optimizer. The learning rate is initialized to 0.001 and decreased by 90% every 20 epochs, and the batch size is set to 32.

Model Architecture. In our experiments, we use the standard point cloud feature extractors (e.g., PointNet [1] or DGCNN [3]) as our encoders, and adopt the morphing-based decoder proposed by MSN [12] for our decoders. In particular, as the decoder generates the point cloud via composing multiple local patches, we utilize 16 local patches for the completion decoder $D_{complete}$, while the partial decoder $D_{partial}$ only uses a single local patch for partial point cloud reconstruction. For both the content encoder $E_{content}$ and pose encoder E_{pose}, their architectures are identical to each other but without sharing weights. For the pose regressor, we use three fully-connected layers with BatchNorm and ReLU operations between every adjacent layer. Source code and models will be released to the public for reproducibility.

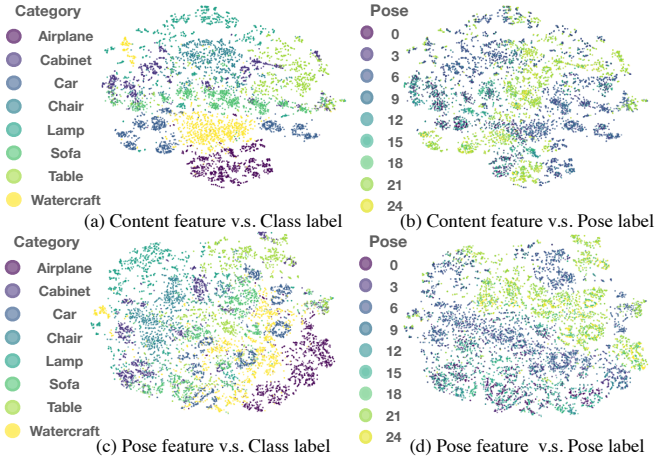
IV. Experimental Results and Analysis

Following the setting in self-supervised point cloud methods [5], [37], we conduct extensive experiments to show the effectiveness and robustness of the point cloud features learnt by our proposed pretext task pre-trained on ShapeNet [38]. Specifically, during training the classifiers of downstream tasks, the pretrained content and pose encoders via our pretext task remain fixed, in which we use the content features for point cloud classification and part segmentation, and the pose features for the registration. In the following, we first present the analysis of our pretext task via point cloud completion then provide results of each downstream task.

A. Pretext Analysis

Features Distribution. We assess the quality of our learnt content and pose features by visualizing the feature embedding using t-SNE in Figure 3. We extract both content and pose features from the fixed $E_{content}$ and E_{pose} respectively, which is pre-trained on ShapeNet using the DGCNN backbone at 50 epoch. In Figure 3.a and 3.c, we plot feature distributions with respect to the class label in different colors (note that class labels are not used in training). It shows that data with the same class is grouped together for the content feature, while the pose feature is more invariant to the class label. Similar observations are also found with respect to the pose label in Figure 3.b and 3.d, where the content feature is more invariant to the pose label. This evidence verifies our proposed feature disentanglement process for learning the distinct characteristics of the content and pose features.

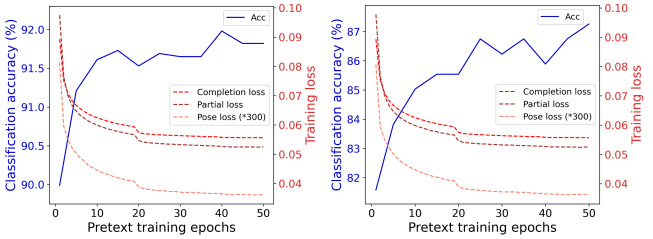
Relevance b/w Pretext Loss & Downstream Accuracy. We plot the training loss with respect to downstream classification accuracy in Figure 4 to verify the effectiveness of our objective function. Both Figure 4.a and Figure 4.b show the gradual accuracy gain on the downstream tasks as the training loss decreases, which indicates that the optimization of our model is effective in feature learning.



(a) Content feature v.s. Class label (b) Content feature v.s. Pose label

(c) Pose feature v.s. Class label (d) Pose feature v.s. Pose label

Fig. 3. t-SNE visualization of content and pose features labeled with class and pose labels respectively on the ShapeNet dataset. The class label indicates the 8 categories data that we used for pre-training, while the pose label indicates the 26 viewpoints that we use for rendering partial data. Note that the pose labels with closer value denote they have similar viewpoints.



(a) ModelNet40 classification task (b) ScanObjectNN classification task

Fig. 4. Pretext training loss versus downstream classification accuracy. Note that our pretext model is trained on the ShapeNet dataset, while two classification tasks are conducted on the ModelNet40 (a) and ScanObjectNN (b) benchmarks respectively.

B. Downstream Task 1: Classification

Dataset. ModelNet40 [40] and ScanObjectNN [18] benchmarks are adopted for the downstream task of classification, and make comparisons against several state-of-the-art supervised and unsupervised methods. ModelNet40 is composed of 12,311 CAD models from 40 categories, in which it is split into 9,843 and 2,468 models for training and testing. For the ScanObjectNN dataset, which is a real-world point cloud dataset based on scanned indoor scene, we use its two variants, "OBJ_ONLY" and "PB_T50_RS", that are widely adopted for evaluating pre-trained features in the classification downstream tasks [4], [6]. "OBJ_ONLY" contains 2,890 models with 15 categories, and "PB_T50_RS" is stemmed from "OBJ_ONLY" and augmented with various perturbations (e.g., translation, rotation, scaling, and cropping with background, resulting in total 14,298 models), and hence is considered to be the hardest case in the ScanObjectNN. We follow the default setting to the train/test split for "OBJ_ONLY" and "PB_T50_RS" by 2,309/581 and 11,416/2,882 respectively.

Experimental Setting. We follow the standard experimental procedure as in [8], [5], where a linear Support Vector Machine (SVM) is trained on the 3D point cloud features extracted from each of the methods, to evaluate the effectiveness of the feature representations in classification (note that our model uses the content feature here).

Our proposed model is pre-trained on ShapeNet, which uses the same pre-trained dataset as the unsupervised learning methods [41], [21], [42], [9], [37], [5] presented in our experiments. Please note that for both our proposed model and the unsupervised learning methods, the training set of ModelNet40 (or ScanObjectNN) is only used to train the linear SVM classifier. For fair comparisons with GraphTER [43] and PointGLR [6], we reproduce their performance by training their pretext model on ShapeNet, where the same reproduction of [43], [6] is also applied later on other downstream tasks of part segmentation and point cloud registration.

Results. Table I and Table II show the results on the ModelNet40 and ScanObjectNN datasets respectively. Please note that, for all tables in this paper, we denote our full model of using PointNet [1] and DGCNN [3] as the backbone of encoders as "PN_ours" and "DGCNN_ours".

In Table I, we show that our proposed model performs favorably against other unsupervised methods, thus verifying the robustness of our self-supervised model in learning more

TABLE I

CLASSIFICATION ACCURACY (%) ON MODELNET40. WE COMPARE OUR MODEL WITH VARIOUS STATE-OF-THE-ART SELF-SUPERVISED METHODS (DENOTED AS "SSL"). IN THE BOTTOM TWO GROUPS, WE COMPARES METHODS EITHER USING THE POINTNET-BASED OR DGCNN-BASED BACKBONE. THE FIRST TWO SUPERVISED METHODS ARE SERVED AS REFERENCES AS THEY FULLY TRAIN THE ENTIRE NETWORK. (†: REPRODUCED BY TRAINING THE PRETEXT MODEL ON SHAPENET.)

Methods	SSL	Accuracy
PointNet [1]	✗	89.2
DGCNN [3]	✗	92.9
GraphTER† [43]	✓	87.8
FoldingNet [41]	✓	88.4
PointCapsNet [21]	✓	88.9
Multi-Tasks [9]	✓	89.1
Yang *et al.* [22]	✓	90.9
PointGLR† [6]	✓	91.7
Sauder *et al.* [5]	✓	87.3
ACD [37] (PointNet++)	✓	89.8
Chen *et al.* [25]	✓	89.9
PN_ours	✓	**90.1**
Jing *et al.* [42]	✓	89.8
Sauder *et al.* [5]	✓	90.6
STRL [24]	✓	90.9
ParAE [23]	✓	91.6
DGCNN_ours	✓	**92.0**

TABLE II

CLASSIFICATION ACCURACY (%) ON SCANOBJECTNN DATASET. WE COMPARE OUR MODEL WITH STATE-OF-THE-ART SELF-SUPERVISED METHODS (DENOTED AS "SSL"), WHERE SUPERVISED METHODS ARE SERVED AS REFERENCES AS THEY FULLY TRAIN THE ENTIRE NETWORK. (†: REPRODUCED BY TRAINING THE PRETEXT MODEL ON SHAPENET.)

Methods	SSL	OBJ_ONLY	PB_T50_RS
Pointnet [1]	✗	79.2	68.2
DGCNN [3]	✗	86.2	78.1
GraphTER† [43]	✓	72.8	60.3
PointGLR† [6]	✓	85.2	73.4
PN_ours	✓	84.0	70.6
DGCNN_ours	✓	**87.3**	**74.8**

holistic and effective feature representations. Furthermore, the competitive performance (or with a small gap) of our model compared to the supervised baselines shows the practical potential of our 3D point cloud features. In Table II, we show that our model even outperforms all the supervised and self-supervised learning baselines in "OBJ_ONLY", while providing comparable performance in "PB_T50_RS" with respect to the supervised methods. Particularly, such results verify the generalizability of the feature representation learnt by our method across synthetic and real-world datasets.

C. Downstream Task 2: Part Segmentation

Dataset. In addition to the downstream task of classification for showcasing the capacity of our content features in modeling the holistic point cloud, here we experiment on another task, part segmentation, to investigate how the fine-grained information of local 3D points is maintained in our learnt content features. We adopt ShapeNet-Part dataset [44] for this evaluation, which is commonly used on the part segmentation task. ShapeNet-Part dataset consists of 16,872 models with 16 categories, with being split into 13,998 and 2,874 for training and testing. Depending on the object category, 3D points are annotated by 2 to 6 part labels, where there are in total 50 distinct labels for the whole dataset.

Experimental Setting. We follow the same architecture for the part classifier designed for part segmentation task in the PointNet [1] framework. Similar to the classification task, our proposed self-supervised model is first pre-trained on ShapeNet, and the model is fixed as a feature extractor while training a part segmentation classifier. Following [1], we extract point-wise features of the final convolutional layer (before max-pooling) in the content encoder $E_{content}$ as local features, while using the content features as global features for learning the segmentation part classifier. The evaluation metric is the average of intersection-over-union (i.e., mIoU).

Results. Table III shows the results and the comparison of our proposed model against several methods on ShapeNet-Part dataset. Our model provides better performance than the self-supervised methods, in which it demonstrates that the content features learnt via disentangling from partial point clouds with different viewpoints are able to benefit the extraction of more discriminative point-wise features. Moreover, our proposed model is competitive in comparison to the supervised methods that train the entire network on ShapeNet-Part, which shows the robustness of our learnt 3D point cloud features that are used to only train the part classifier.

D. Downstream Task 3: Point Cloud Registration

Dataset. Other than studying the content features in classification and part segmentation, here we adopt a pose-relevant downstream task, i.e., registration, to evaluate the pose features learnt by our proposed model. We conduct experiments for point cloud registration based on ModelNet40. We follow the same procedure of data preparation as in DCP [45], in which the whole ModelNet40 dataset is randomly split into training and testing sets regardless of object categories. For each point cloud $\mathcal{P}$, a randomly-drawn rigid transformation $\mathcal{T}$ is applied on $\mathcal{P}$ to obtain the transformed point cloud $\mathcal{T}(\mathcal{P})$, where $\{\mathcal{P}, \mathcal{T}(\mathcal{P})\}$ together with $\mathcal{T}$ becomes an input-output pair for learning the registration model.

Experimental Setting. We adopt the same architecture for the registration model and follow the training procedure proposed by DCP [45] which uses DGCNN [3] as their feature extractor. Our proposed model is first pre-trained on ShapeNet, and then we replace the feature extractor in DCP with our pre-trained pose encoder E_{pose} to train the registration task. Note that E_{pose} is fixed during the training process to verify the effectiveness of our learnt features. For evaluation, we focus on estimating the orientation/rotation between point clouds in this experiments and adopt the root mean squared error (RMSE) as the evaluation metric.

Results. As shown in Table IV, the pose features learnt from our model pre-trained on the ShapeNet dataset outperform other self-supervised methods on RMSE(R). Moreover, our proposed model is competitive against the state-of-the-art

TABLE III

PART SEGMENTATION RESULTS ON SHAPENET-PART DATASET. WE COMPARE OUR MODEL WITH STATE-OF-THE-ART SELF-SUPERVISED METHODS (DENOTED AS "SSL"), WHERE SUPERVISED METHODS ARE SERVED AS REFERENCES AS THEY FULLY TRAIN THE ENTIRE NETWORK. (ψ: PRETEXT MODEL IS TRAINED ON MODELNET40; $*$: REPORTED BY MAP [8]; †: REPRODUCED BY TRAINING THE PRETEXT MODEL ON SHAPENET DATASET.)

Methods	SSL	mIoU	Aero	Bag	Cap	Car	Chair	Ear ph.	Guitar	Knife	Lamp	Laptop	Motor	Mug	Pistol	Rocket	Skate	Table
PointNet [1]	✗	83.7	83.4	78.7	82.5	74.9	89.6	73.0	91.5	85.9	80.8	95.3	65.2	93.0	81.2	57.9	72.8	80.6
DGCNN [3]	✗	85.2	84.0	83.4	86.7	77.8	90.6	74.7	91.2	87.5	82.8	95.7	66.3	94.9	81.1	63.5	74.5	82.6
Latent-GAN* [19]	✓	57.0	54.1	48.7	62.6	43.2	68.4	58.3	74.3	68.4	53.4	82.6	18.6	75.1	54.7	37.2	46.7	66.4
MAP$^{\psi}$ [8]	✓	68.0	62.7	67.1	73.0	58.5	77.1	67.3	84.8	77.1	60.9	90.8	35.8	87.7	64.2	45.0	60.4	74.8
GraphTER† [43]	✓	82.3	81.7	76.0	83.2	74.9	84.8	64.6	90.9	87.1	82.5	**95.6**	56.3	**93.2**	81.6	**56.2**	71.0	67.8
PointGLR† [6]	✓	84.6	81.5	**84.7**	81.7	75.4	**89.8**	76.1	89.8	84.9	83.3	95.2	65.4	92.8	79.9	55.7	73.7	**83.5**
PN_ours	✓	83.8	81.6	73.2	83.5	74.2	89.0	70.9	89.9	85.5	80.6	95.0	66.4	92.6	82.0	53.7	72.7	82.9
DGCNN_ours	✓	**85.1**	**82.3**	83.5	**84.5**	**77.3**	**89.8**	**76.3**	**91.0**	**87.3**	**84.2**	95.5	**67.8**	92.5	**82.8**	52.1	**73.9**	**83.5**

TABLE IV

REGISTRATION RESULTS ON THE MODELNET40 DATASET. WE COMPARE OUR PROPOSED MODEL WITH SUPERVISED METHODS REPORTED BY DCP [45] AND SELF-SUPERVISED METHODS REPRODUCED BY USING THE OFFICIAL IMPLEMENTATIONS. (†: REPRODUCED BY TRAINING THE PRETEXT MODEL ON THE SHAPENET DATASET.)

Methods	SSL	RMSE(R) ↓
PointNetLK [46]	✗	15.095
FGR [47]	✗	9.363
DCP-v2 [45] (PN)	✗	7.061
DCP-v2 [45] (DGCNN)	✗	1.143
PointGLR† [6]	✓	3.900
GraphTER† [43]	✓	2.927
PN_ours	✓	6.719
DGCNN_ours	✓	**1.557**

TABLE V

CONTENT V.S. POSE FEATURES. COMPARISONS BETWEEN CONTENT AND POSE FEATURES IN THE CLASSIFICATION, PART SEGMENTATION AND REGISTRATION TASKS. (MN40: MODELNET40; SON: WE USE "OBJ_ONLY" ON SCANOBJECTNN; REGIS.: WE USE RMSE(R) AS EVALUATION METRIC.)

Backbone	Feature	Classification MN40	Classification SON	Part Seg.	Regis.
PN	content	**90.11**	**83.99**	**83.83**	6.875
	pose	88.98	78.49	83.54	**6.719**
DGCNN	content	**91.98**	**87.26**	**85.05**	2.920
	pose	90.36	82.79	84.46	**1.557**

supervised methods that fully train the entire network on ModelNet40, which verifies the robustness of our pose features on the point cloud registration task.

E. Ablation Study

Comparisons between Content and Pose Features. In the previous experimental results, we adopt the content feature for the tasks of classification and part segmentation, while the pose feature is utilized for the registration. Here we provide results of using different features for these downstream tasks. Table V provides the results of applying either the content or pose feature on all the downstream tasks.

We observe that, in comparison to the pose feature, the content feature works best for the classification and part segmentation tasks, where the former needs more semantic and holistic understanding of the point clouds while the latter needs fine-grained modeling on the local geometric structures. This verifies that the content feature learnt by our proposed method is able to compactly model both the global and local structural information of the input point cloud. When it comes to the downstream task of registration, the pose feature instead contributes better than the content feature. Such results are also aligned with the findings from several prior works [48], [46], where the camera pose estimation task could provide benefits for the registration problem.

Comparisons among Different Model Designs. We study different model designs here. In addition to our full model, we experiment other designs: 1) the framework of only using the completion branch as proposed by the work of [26] (denoted as "Comp-only"); 2) the framework of only using the pose regression branch (denoted as "PR-only"); 3) the joint learning framework (denoted as "JL"), where the completion and pose regression branches share one encoder trained in a multi-tasking manner. The quantitative comparison among these designs is provided in Table VI.

As we expected, Comp-only can learn more semantic information and obtain better results than PR-only in the classification and part segmentation tasks, while PR-only learns more camera related information and obtains better results than Comp-only in the registration task. Furthermore, our full model can extract more robust and effective features than the other designs, especially in the real-world data classification task. Interestingly, JL obtains worse results than PR-only in the classification task. We hypothesize that, as completion and pose regression have quite different characteristics, having one encoder to jointly learn both tasks may result in worse feature learning, thus verifying again the contribution of our proposed disentanglement method.

V. CONCLUSIONS

We propose a self-supervised framework for learning representations of 3D point clouds. Based on the objectives composed of completion, reconstruction, and pose regression for the partial point cloud data, our model learns to disentangle the content and pose factors. Our learnt content and pose feature representations of 3D point clouds experimentally demonstrate the superior performance in comparison to other self-supervised methods in various downstream tasks

TABLE VI

COMPARISONS WITH DIFFERENT MODEL DESIGNS. NOTE THAT "COMP-ONLY" ONLY USES THE COMPLETION BRANCH, "PR-ONLY" ONLY ADOPTS THE POSE REGRESSION BRANCH, AND "JL" INDICATES THE JOINT LEARNING MANNER WITH THE SHARED ENCODER. (MN40: MODELNET40; SON: WE USE "OBJ_ONLY" ON SCANOBJECTNN.)

Backbone	Methods	Classification		Part Seg.	Regis.
		MN40	SON		
PN	Comp-only	89.71	83.13	83.72	7.858
	PR-only	89.42	79.00	83.60	6.865
	JL	89.34	78.49	83.77	6.843
	Ours full	**90.11**	**83.99**	**83.83**	**6.719**
DGCNN	Comp-only	91.37	86.06	84.93	3.231
	PR-only	90.64	82.62	84.47	1.644
	JL	90.52	81.93	84.49	1.741
	Ours full	**91.98**	**87.26**	**85.05**	**1.557**

such as classification, part segmentation, and registration.

Acknowledgement. This project is supported by MOST 111-2636-E-A49-003 and MOST 111-2628-E-A49-018-MY4.

REFERENCES

[1] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *CVPR*, 2017.

[2] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," in *NeurIPS*, 2017.

[3] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic graph cnn for learning on point clouds," *ACM Transactions on Graphics (TOG)*, 2019.

[4] A. Alliegro, D. Boscaini, and T. Tommasi, "Joint supervised and self-supervised learning for 3d real-world challenges," *ArXiv:2004.07392*, 2020.

[5] J. Sauder and B. Sievers, "Self-supervised deep learning on point clouds by reconstructing space," in *NeurIPS*, 2019.

[6] Y. Rao, J. Lu, and J. Zhou, "Global-local bidirectional reasoning for unsupervised representation learning of 3d point clouds," in *CVPR*, 2020.

[7] S. Xie, J. Gu, D. Guo, C. R. Qi, L. J. Guibas, and O. Litany, "Pointcontrast: Unsupervised pre-training for 3d point cloud understanding," in *ECCV*, 2020.

[8] Z. Han, X. Wang, Y.-S. Liu, and M. Zwicker, "Multi-angle point cloud-vae: unsupervised feature learning for 3d point clouds from multiple angles by joint self-reconstruction and half-to-half prediction," in *ICCV*, 2019.

[9] K. Hassani and M. Haley, "Unsupervised multi-task feature learning on point clouds," in *ICCV*, 2019.

[10] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros, "Context encoders: Feature learning by inpainting," in *CVPR*, 2016.

[11] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *NeurIPS*, 2017.

[12] M. Liu, L. Sheng, S. Yang, J. Shao, and S.-M. Hu, "Morphing and sampling network for dense point cloud completion," in *AAAI*, 2020.

[13] W. Yuan, T. Khot, D. Held, C. Mertz, and M. Hebert, "Pcn: Point completion network," in *3DV*, 2018.

[14] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *ICLR*, 2016.

[15] Y. Li, R. Bu, M. Sun, W. Wu, X. Di, and B. Chen, "Pointcnn: Convolution on x-transformed points," in *NeurIPS*, 2018.

[16] Y. Liu, B. Fan, S. Xiang, and C. Pan, "Relation-shape convolutional neural network for point cloud analysis," in *CVPR*, 2019.

[17] Y. Xu, T. Fan, M. Xu, L. Zeng, and Y. Qiao, "Spidercnn: Deep learning on point sets with parameterized convolutional filters," in *ECCV*, 2018.

[18] M. Angelina Uy, Q.-H. Pham, B.-S. Hua, D. Thanh Nguyen, and S.-K. Yeung, "Revisiting point cloud classification: A new benchmark dataset and classification model on real-world data," in *ICCV*, 2019.

[19] P. Achlioptas, O. Diamanti, I. Mitliagkas, and L. Guibas, "Learning representations and generative models for 3d point clouds," in *ICML*, 2018.

[20] H. Deng, T. Birdal, and S. Ilic, "Ppf-foldnet: Unsupervised learning of rotation invariant 3d local descriptors," in *ECCV*, 2018.

[21] Y. Zhao, T. Birdal, H. Deng, and F. Tombari, "3d point capsule networks," in *CVPR*, 2019.

[22] J. Yang, P. Ahn, D. Kim, H. Lee, and J. Kim, "Progressive seed generation auto-encoder for unsupervised point cloud learning," in *ICCV*, 2021.

[23] B. Eckart, W. Yuan, C. Liu, and J. Kautz, "Self-supervised learning on 3d point clouds by learning discrete generative models," in *CVPR*, 2021.

[24] S. Huang, Y. Xie, S.-C. Zhu, and Y. Zhu, "Spatio-temporal self-supervised representation learning for 3d point clouds," in *ICCV*, 2021.

[25] Y. Chen, J. Liu, B. Ni, H. Wang, J. Yang, N. Liu, T. Li, and Q. Tian, "Shape self-correction for unsupervised point cloud understanding," in *ICCV*, 2021.

[26] H. Wang, Q. Liu, X. Yue, J. Lasenby, and M. J. Kusner, "Unsupervised point cloud pre-training via occlusion completion," in *ICCV*, 2021.

[27] A. Odena, C. Olah, and J. Shlens, "Conditional image synthesis with auxiliary classifier gans," in *ICML*, 2017.

[28] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel, "Infogan: Interpretable representation learning by information maximizing generative adversarial nets," in *NeurIPS*, 2016.

[29] K. Sohn, H. Lee, and X. Yan, "Learning structured output representation using deep conditional generative models," in *NeurIPS*, 2015.

[30] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *NeurIPS*, 2014.

[31] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," in *ICLR*, 2014.

[32] Y. Zou, X. Yang, Z. Yu, B. Kumar, and J. Kautz, "Joint disentangling and adaptation for cross-domain person re-identification," in *ECCV*, 2020.

[33] Y.-J. Li, C.-S. Lin, Y.-B. Lin, and Y.-C. F. Wang, "Cross-dataset person re-identification via unsupervised pose disentanglement and adaptation," in *ICCV*, 2019.

[34] K. Zhou, B. L. Bhatnagar, and G. Pons-Moll, "Unsupervised shape and pose disentanglement for 3d meshes," in *ECCV*, 2020.

[35] H. Fan, H. Su, and L. J. Guibas, "A point set generation network for 3d object reconstruction from a single image," in *CVPR*, 2017.

[36] D. P. Bertsekas, "Auction algorithms for network flow problems: A tutorial introduction," *Computational optimization and applications*, 1992.

[37] M. Gadelha, A. RoyChowdhury, G. Sharma, E. Kalogerakis, L. Cao, E. Learned-Miller, R. Wang, and S. Maji, "Label-efficient learning on point clouds using approximate convex decompositions," in *ECCV*, 2020.

[38] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su *et al.*, "Shapenet: An information-rich 3d model repository," *ArXiv:1512.03012*, 2015.

[39] B. O. Community, *Blender - a 3D modelling and rendering package*, Blender Foundation, 2018. [Online]. Available: http://www.blender.org

[40] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao, "3d shapenets: A deep representation for volumetric shapes," in *CVPR*, 2015.

[41] Y. Yang, C. Feng, Y. Shen, and D. Tian, "Foldingnet: Point cloud auto-encoder via deep grid deformation," in *CVPR*, 2018.

[42] L. Jing, Y. Chen, L. Zhang, M. He, and Y. Tian, "Self-supervised feature learning by cross-modality and cross-view correspondences," *ArXiv:2004.05749*, 2020.

[43] X. Gao, W. Hu, and G.-J. Qi, "Graphter: Unsupervised learning of graph transformation equivariant representations via auto-encoding node-wise transformations," in *CVPR*, 2020.

[44] L. Yi, V. G. Kim, D. Ceylan, I.-C. Shen, M. Yan, H. Su, C. Lu, Q. Huang, A. Sheffer, and L. Guibas, "A scalable active framework for region annotation in 3d shape collections," *SIGGRAPH Asia*, 2016.

[45] Y. Wang and J. M. Solomon, "Deep closest point: Learning representations for point cloud registration," in *ICCV*, 2019.

[46] Y. Aoki, H. Goforth, R. A. Srivatsan, and S. Lucey, "Pointnetlk: Robust & efficient point cloud registration using pointnet," in *CVPR*, 2019.

[47] R. Benjemaa and F. Schmitt, "Fast global registration of 3d sampled surfaces using a multi-z-buffer technique," *Image and Vision Computing*, 1999.

[48] T. Phuc Truong, M. Yamaguchi, S. Mori, V. Nozick, and H. Saito, "Registration of rgb and thermal point clouds generated by structure from motion," in *ICCV Workshops*, 2017.