Contrastive Point Cloud Pretraining for Enhanced Transformers

Divyashree Shivalingappa Koti, Joshua L. Phillips, Frederick S. Cottle
Middle Tennessee State University
Murfreesboro, TN, USA
dsk2v@mtmail.mtsu.edu, joshua.phillips@mtsu.edu, rick.cottle@mtsu.edu

2024 IEEE 36th International Conference on Tools with Artificial Intelligence (ICTAI) | 979-8-3315-2723-5/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICTAI62512.2024.00057

***Abstract*—Transformers, although first designed for sequence processing, can also handle unordered sets like point cloud data. Additionally, contrastive pretraining has emerged as a successful technique in image processing but remains unexplored for point cloud data. We develop and integrate a new point cloud pretraining technique inspired by the Simple Framework for Contrastive Learning (SimCLR) into the Set Transformer (ST) and Point Cloud Transformer (PCT) architectures and explore model performance using a novel 3D body scan dataset and the canonical datasets ShapeNet and ModelNet. For the 3D body scan dataset, this integration boosts initial training performance and maintains overall higher performance for classification tasks, and demonstrates better stability/convergence for regression tasks in comparison to non-pretrained (Naïve) counterparts. Furthermore, experiments examining strong generalization (relative performance on previously unseen classes) show improvement for pretrained models compared to Naïve models. Consistent benefits across tasks and data sets are observed based on additional experiments performed on the ShapeNet core dataset. Overall, we show how contrastive pretraining for point cloud data is a viable strategy for improving the performance of Transformers on downstream tasks and accelerating the training process.**

***Index Terms*—Point Cloud, Contrastive, Set Transformer, Point Cloud Transformer (PCT), ShapeNet, ModelNet**

I. Introduction

Set Transformer (ST) [9] and Point Cloud Transformer (PCT) [5] are transformer [15] variants designed to process unordered point cloud data, leveraging permutation invariance. Unlike traditional grid-structured objects, such as Vision Transformer (ViT) [3], PCT enhances input embedding with farthest point sampling and nearest neighbor search, capturing the local context, whereas ST processes unordered sets using the formal attention mechanism. Contrastive pretraining has also emerged as a technique to improve downstream fine-tuning on models for processing grid-structured data [2] thereby helping to speed up the process and improving accuracy in classification tasks via extracting better features based on similarity and dissimilarity in instances. To the best of our knowledge, the contrastive pretraining technique with a transformer on point cloud data remains unexplored.

We conduct experiments using 3D point cloud body scan data, a novel dataset initially collected using KX-16 Body Scanner [14] for a psychology study under IRB protocol[1]. The dataset is used for classification tasks such as self-identification, gender classification, and age categorization, and regression tasks predicting height, weight, and age. These tasks are beneficial for applications in the fashion and design industry, where precise personalization and real-time adjustments are essential. Additionally, we classify the ShapeNet [18] and ModelNet [5] datasets comprising 55 and 40 object classes, respectively.

Overall, our experiments aim to explore the suitability of the Set Transformer and Point Cloud Transformer for the 3D body scan dataset and the synthetic datasets ShapeNet and ModelNet, analyzing the potential advantages and challenges of Contrastive Pretrained Set Transformer (CPST) and Contrastive Point Cloud Transformer (CPCT). Our primary focus in this work is a comparative study between the Naive and the corresponding Contrastive pretrained models. We also assess the generalization capabilities of both models in their Naive and contrastive forms by comparing weak generalization (new examples within the trained class distribution) and strong generalization (examples outside the trained distribution) [16] across both datasets. Generalization is evaluated using *Kullback–Leibler (KL) divergence* as a distance metric between weak and strong generalization across tasks.

[1]No data was individually identifiable by the researchers.

II. Related work

Transformer for point clouds: Representing data in the form of 3D point clouds enables precise and comprehensive analysis across fields. Transformers, with their inherent permutation invariance from the self-attention mechanism, are well-suited for point cloud processing. Variants like Point Cloud Transformer (PCT) [5] along with a few other models [4], [7], [10], [19], and [13] (convolutional based), leverage self-attention mechanism in different forms while maximizing the task performance. PCT introduces optimized offset-attention and neighbor embeddings, while ST uses permutation-invariant attention with *Induced Set Attention Blocks* (ISAB) layers and *Pooling by Multihead-Attention* (PMA) block for efficient feature extraction. Despite the availability of other models, ST's time and space efficiency and PCT's optimized techniques make them the most viable options for integration with contrastive models.

Contrastive method: Contrastive learning, introduced by [6], allows a model to cluster positive (neighboring) pairs while separating negative (non-neighboring) pairs in latent space without an external distance metric. The Simple framework for Contrastive Learning (SimCLR) [2] extended this to

2375-0197/24/$31.00 ©2024 IEEE
DOI 10.1109/ICTAI62512.2024.00057

the image domain by creating batches of positive (a sample and its augmented version) and negative pairs, facilitating unsupervised contrastive learning. Inspired by the contrastive benefits, several works explore unsupervised learning [17], self-supervised learning [1], [12], or improvised contrastive loss function as in [8]. However, no models have explored transformers for contrastive learning on point clouds.

III. Methods

This section outlines the model implementation and experimental setup for ShapeNet and ModelNet classification, as well as self-identification, gender classification, binned-age classification, and regression tasks (weight, height, and age prediction). The Naive Set Transformer, designed using ISAB and PMA layers, is employed both as a stand-alone model and as a base for the contrastive pre-trained approach. We design the Set Transformer for both classification and regression tasks using ISAB and PMA layers, as discussed in Section II. For the PCT implementation, we adopt the neighbor embedding technique from [5], utilizing Linear, BatchNorm, and ReLU (LBR) layers, followed by Sampling and Grouping (SG) layers to extract local features.

Extending the Naive Set Transformer and Naive PCT, we introduce a contrastive pretraining technique inspired by the SimCLR approach to both models to utilize the potential advantages offered in improving the performance. As a result, we expect a high performance by the contrastive models with consistent success across the tasks and the capability to identify an unseen object more precisely due to improved organization of the latent space.

The 3D point cloud body scan has 96 individual samples, each consisting of at least 40,000 points. We normalize these point clouds by scaling to ensure consistency across samples, facilitating numerical stability during learning. Of the available metadata (age, height, weight, ethnicity, and gender), missing values are handled by calculating the Chamfer distance from the missing data to other samples and assigning the value from the closest match. The dataset is split into training and validation sets in an 80:20 ratio, and training batches are created through random subsampling of points.

The ShapeNet dataset [18] has approximately 35,000 training and 5,100 validation samples, each consisting of 15,000 points, and normalized using standard scalar. The ModelNet dataset [5] has 9,800 training and 2,400 test samples (based on the official splits), each consisting of 2,000 points. Since the datasets are imbalanced, we form balanced batches at each epoch by randomly subsampling from all classes to match the count of the least-represented class (undersampling).

We implement the contrastive learning technique in a similar technique as in [2], leveraging positive and negative pairings to pretrain the model. This establishes an unsupervised objective for pretraining the model. Random subsamples of data points are paired with non-overlapping subsamples from the corresponding scans. The pretrained model is then fine-tuned for downstream classification and regression tasks[2].

[2]No class balancing strategy is applied during contrastive pretraining.

As part of tracking the overall efficiency of the Naive and contrastive pretrained models (CPST and CPCT) in classifying seen (weak generalization or WG) and unseen (strong generalization or SG) category data, we use KL divergence on normalized probabilities to measure the distance for self-identification and multi-class classification tasks. For binary gender classification, the accuracy metric directly compares strong generalization in the Naive vs Contrastive approaches. Similarly, for regression, we calculate the Mean-Squared Error (MSE) for each left-out instance and then average the results to compare model performance.

We train and validate both Naive and Contrastive pretrained models (ST & PCT) on 3D body scans using a batch size of 4, with 8K points per sample for both models and 2K for validation. For ShapeNet, we use a batch size of 32 for PCT and 16 for ST, with 2K points per subsample to train and validate. For ModelNet, we maintain a batch size of 32 for both models. During contrastive pretraining on ModelNet data, we subsample 256 to train and 128 points to validate (unlike the 1024 points used in [5]), as the smaller point clouds could lead to overfitting, and smaller subsamples facilitate better augmentation. These parameters were fine-tuned through extensive experimentation for optimal performance.

We use the Adam optimizer for both models, subjected to 250 epochs of 10 independent runs for ShapeNet and ModelNet and 10 independent Monte Carlo samples for 3D body scans. We calculate standard errors using 1.96 standard errors above and below the mean to estimate 95% confidence intervals. Then we compare Contrastive pretrained (CPST/CPCT) to Naive models based on observed stability, learning speed, accuracy, and loss across classification and regression tasks.

A. Naive models

In this section, we define the experimental setup of the Set Transformer and Point Cloud Transformer. Some of the shared hyperparameters like learning rate and dropout, are tuned based on the task at hand with both models but are maintained constant with their Contrastive counterparts.

1) Set Transformer (ST): The ST model architecture follows the design by [11], with hyperparameters such as *embedding dimension, inducing dimension, number of attention heads, and stacks of ISAB layers* tuned for optimal performance. The Gelu activation function and cross-entropy loss function are used across the board. The number of ISAB layers varied between 2 to 6, with 3 layers proving optimal for most tasks while maintaining other hyperparameters depending on resource availability. Embedding dimension (32, 64), attention heads (4, 16, 32), and inducing dimension (32, 64, 128, 256) are the major factors influencing stability and accuracy. The best results were achieved with an embedding dimension of 64, 16 attention heads, and inducing dimensions of 128/256.

2) Point Cloud Transformer (PCT): The architectural design of the PCT model is utilized as designed by [5]. Point Cloud Transformer with Neighbor embedding is constructed with 2 LBR layers and 2 SG layers, applying the zoom augment effect from SG layers to output reduced point cloud

to half of the original subsample. Reducing these further results in degraded performance. For ShapeNet and ModelNet classification we use the loss crafted with the regularization technique to smooth out the labels, as this performs better than regular cross-entropy, while 3D body scan does well with standard cross entropy.

B. Contrastive learning

Contrastive Pretrained Set transformer(CPST) and Contrastive Point Cloud Transformer (CPCT) represent novel approaches presented in this work experimented on the point cloud datasets. Contrastive pretrained models come in two phases: contrastive pretraining of the base models and the fine-tuning process for the downstream tasks. An overview of the contrastive learning, with an ST/PCT used as a base model extracting the *feature embeddings* and learning contrastively, is demonstrated in pseudocode 1. The contrastively pretrained model learns in an unsupervised fashion and is then fine-tuned for both classification and regression tasks using the same experimental setup as the Naive models for both generalizations.

Set Transformer or PCT as the base model is similar to the Naive models defined in section III-A with most of the hyperparameter maintained as-described. We use a temperature of 2.0 that yields better results in terms of faster convergence and stabilizes the pretraining and fine-tuning phases. Both models are trained up to 200 epochs, making sure the model stabilizes with no further active learning.

We develop independent CPST and CPCT models for 3D body scans, ShapeNet, and ModelNet point cloud datasets. Data is augmented by randomly choosing subsets of points, and batches are constructed with replacement to achieve sizes of 64 & 32 (ST & PCT), 80 & 112 (ST & PCT), and 160 (ST & PCT) for the 3D body scans, ShapeNet, and ModelNet datasets, respectively. Larger batch sizes are crucial for pretrained models to learn underlying patterns, aligning with with similar observations stated in [2]. We set the number of batches per epoch to 150 for the 3D body scans and ModelNet and 200 for ShapeNet.

We conducted experiments starting with smaller batch sizes, incrementing in multiples of 8 until a sufficiently large enough batch size was attained. At this point, we observed enhanced learning in the contrastive models with increasing batch sizes; however, smaller batch sizes had slower learning rates, and the model tended to get stuck in local minima during the fine-tuning process. Additionally, the model benefits even more from larger batch sizes with increased sample sizes.

To train and validate, data is augmented by subsampling point clouds at each training step. For the body scan dataset, we use 1K points; ShapeNet, with its medium-sized point clouds, requires a minimum of 2K points for ST and 1K for PCT; and ModelNet, with the smallest point cloud availability, benefits from 512 points, indicating that higher-density point clouds enable more effective augmentation, leading to improved learning outcomes.

For our models, fine-tuning involves additional layers, including a linear projection of embeddings onto higher dimensional space, followed by a non-linear Leaky-RELU activation and a dropout layer. For a thorough investigation of the linear projection, we tune it to dimensions of [2048, 1024, 256, 128] during the fine-tuning process. For CPST, a 128-dimensional projection with a dropout of 0.1 (0.05 for ModelNet) was optimal across all tasks. In contrast, CPCT does better with 2048 dimensions (dropout of 0.4) for the ShapeNet classification and 1024 with a dropout of 0.1 (0.2 for ModelNet) for the rest of the tasks. The same set of hyperparameters (optimizer, learning rate, and epochs) are used as Navie models for the finetuning process.

Algorithm 1 Contrastive Learning architecture

$(y1\_embed) = \text{base\_model}(batch1)$
$(y2\_embed) = \text{base\_model}(batch2)$
$y1\_embed = \text{Linear}(e\_dim, p\_dim)(y1\_embed)$
$y2\_embed = \text{Linear}(e\_dim, p\_dim)(y2\_embed)$
$y1\_embed = \text{Norm}(y1\_embed)$
$y2\_embed = \text{Norm}(y2\_embed)$
$y = \text{Mul}(y1\_embed, y2\_embed.T) * temp$

C. Tasks

1) Classification: As part of the classification task, we conducted experiments on self-identification, gender classification, and binned-age classification on 3D body scans and object classification on the ShapeNet and ModelNet datasets using the architectural setup discussed earlier.

- **Self-identification :** In this task, each sample is its own class, meaning the number of samples equals the number of targets. A model is asked to identify a sample by learning from 80% of the point cloud data, and then the remaining 20% of the unseen data is used for validation.
- **Gender classification :** Gender metadata serves as the target (male or female) for binary classification. As the female population exceeds the male population, it necessitates the class balancing strategy, for which we used an undersampling technique. Here, the minority(male) class is completely considered, while the majority(female) class population is randomly sampled to match the size of the minority class at each epoch.
- **Binned-age Classification:** As age can be both discrete and continuous, we perform both classification and regression tasks using the Naive and Contrastive models. For classification, the age data is divided into four evenly distributed bins using quantile binning. The experimental setup is identical to the above-mentioned design with an exception to PCT, which showed a tendency to overfit. To address this, a step learning rate with a gamma of 0.1 was applied at the 60th epoch.
- **ShapeNet classification:** This is a highly imbalanced multi-class classification task. We balance it using an undersampling strategy by considering the least count class and sampling the rest of the class samples at random, constructing a fresh batch at every training step. Architectural design needs a little bit of tweaking with

PCT to include L2 normalization which avoids overfitting at the very early stage.

- **ModelNet classification:** This is also a multi-class classification with imbalanced classes and the least available point clouds. Similar to ShapeNet classification we use the undersampling technique, constructing a fresh batch at every train step.

2) Regression: For regression tasks, experiments were conducted to predict the height, weight, and age of a person using the 3D body scan point cloud data with Mean Squared Error (MSE) used as the evaluation metric.

D. Generalization

1) *Weak Generalization* is a technique of model evaluation where the model has seen a part of the data features (subsampled point cloud data) corresponding to a target value, and during the validation, a model encounters the unseen data features and is asked to assign to its corresponding value. In our case, we do 80:20 mutually exclusive point cloud data for the 3D body scans and utilize the already separated train and validation data for ShapeNet dataset. To train the model, point cloud data is randomly subsampled on each epoch for both the training and validation process.
2) *Strong Generalization* is a technique where a model's ability to handle unseen samples or classes during validation, even though it has encountered similar samples or target values during training. We implement the leave-one-out strategy training on all but one sample, resulting in as many models as there are classes. For example, consider the self-identification task to identify 96 individuals; 96 independent models will be trained.

For classification or regression tasks, we design the model similar to its corresponding weak generalization, as discussed in sections III-A and III-B, and repeat the experiment using both Naive and Contrastive models. In each run, the model is trained as in the weak generalization process, and at the end of training, we validate on left-out data sample. For self-identification, ShapeNet, and ModelNet classification tasks, we build a confusion matrix of size $number\_class * number\_class$ constructed to analyze how confused the model is in recognizing the unseen sample. We then compare the confusion probability distribution with the weak generalization probability distribution (ignoring the darker diagonal in the weak generalization matrix).

Similarly, experiments are repeated using the CPST and CPCT models for both weak and strong generalizations, constructing confusion matrices for the classification tasks (Note that the Contrastively Pretrained model is aware of all samples during pretraining). To analyze model generalization, we use the weak generalization as a standard matrix to calculate the KL divergence distance to the corresponding strong generalization matrix. For each run, we take the final probability matrices from both generalizations, average them across the simulations, and then normalize. Resultant normalized matrices are then used to compute the KL divergence. To compare probability distributions, we nullify the diagonal elements of the weak generalization matrices (by assigning a small value of $1e^{-12}$), allowing a fair comparison with the leave-one-out strategy and expecting to observe similar distributions between both generalizations.

Gender classification is a binary classification problem where leaving out an entire class will not suffice for a model to learn underlying distinguishing patterns. Instead, we leave out individual samples, as each can be gender-identified based on its point cloud data. Accuracy is used to evaluate generalization to unseen samples in both Naive and Contrastive models, with higher accuracy indicating better generalization.

Similarly, for regression tasks, the average Mean Squared Error (MSE) is used as an evaluation metric for the generalization. The lower the average loss becomes, the better the model is generalizing for the regression task.

All implementation details are on our GitHub repository[3].

IV. Results

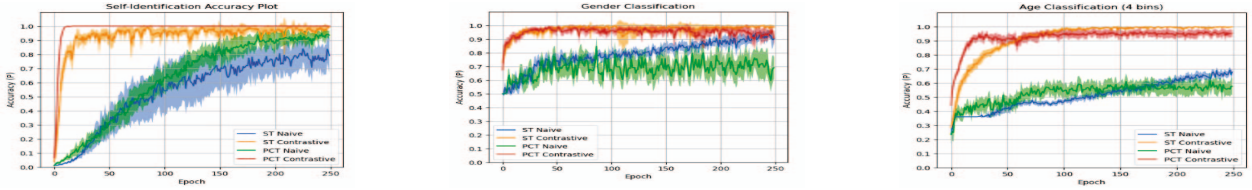
All experiments are subjected to weak and strong generalizations using both Naive and Contrastively pretrained models of ST and PCT. Fig. 1, 2 and 3 depict the weak generalization results on the validation set, exhibiting the model's performance across Naive ST, Naive PCT, CPST, and CPCT models. All plots illustrate the mean accuracy/loss over 250 epochs averaged across 10 Monte Carlo samples with a 95% confidence interval. Generalization results are represented in Tables I, and II.

A. Classification tasks

From the Fig. 1a, 1b and 1c the accuracy plot for self-identification, gender classification, and binned-age classification, CPST and CPCT both have clear wins over Naive counterparts across the tasks. However, for self-identification, CPCT is more stable than the CPST, but with binned-age classification, CPST starts off slow but overtakes CPCT. Whereas with gender classification both models exhibit similar learning behavior. Overall both versions of contrastive models achieve over 90% of accuracy in a short time (facilitated by the initial long step jumps indicating the model's prior knowledge) and maintain stability over longer epochs, unlike Naive models.

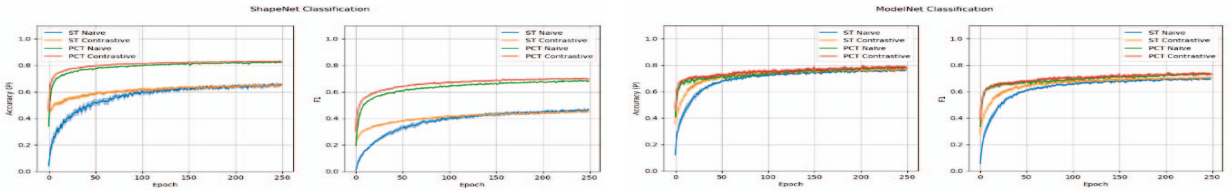
Fig. 2a and 2b, shows plots of an F1 score[4] and accuracy for ShapeNet and ModelNet datasets, depicting the boosted early learning performance with both CPCT and CPST models. While with the ShapeNet dataset, CPCT is observed to maintain high performance throughout the learning phase, CPST eventually merges with its Naive counterpart. Note that though the scores do not look as high as our body scan dataset, note that this is a highly imbalanced set with a large number of samples and suffers due to data quality and low count on point cloud. On the other hand, the ModelNet dataset is observed to have a boosted start with CPST but ends up with just about the same performance on all 4 versions.

[3]GitHub: *CPCP Repository*
[4]To provide balanced evaluation across all classes.



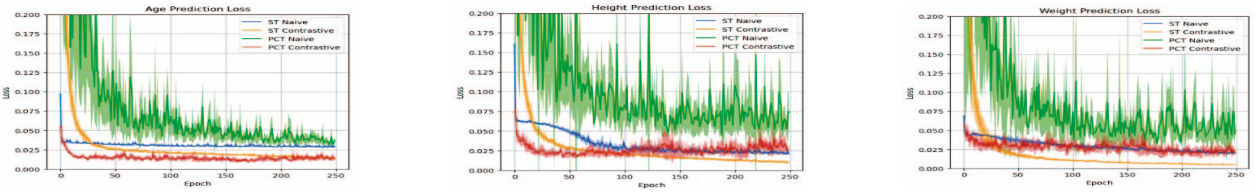
(a) Self-Identification (b) Gender Classification (c) Binned-Age Classification

Fig. 1: Classification accuracy plot for weak generalization obtained using Naive ST, Naive PCT, CPST, and CPCT models. A model is trained and validated using a mutually exclusive point cloud dataset.



(a) Shapenet (b) ModelNet

Fig. 2: Accuracy and F1 result plot for classification using canonical datasets.



(a) Age Prediction (b) Height Prediction (c) Weight Prediction

Fig. 3: Average regression loss for weak generalization plotted using Naive ST, Naive PCT, CPST, and CPCT models across 10 independent Monte Carlo runs. The model is trained solely on point cloud data input as features and corresponding task metadata is used as a target for supervised learning. Plots are zoomed in for better visualization.

B. Regression tasks

Fig. 3a, 3b & 3c depicts the convergence of loss functions in predicting individuals age, height, and weight, respectively. Among the four models across the three tasks, Naive PCT is unstable and struggles to converge. Whereas, its counterpart CPCT is considerably stable and tends to converge at a faster phase, although slight overfitting can be observed with height predictions. Both Naive ST and CPST are pretty stable and have better convergence in all scenarios, however, CPST comparatively surpasses the CPCT and Naive ST models.

C. Generalization

Table I shows the KL divergence results of generalization for classification tasks. Though the results across tasks are not consistent, the CPST model produces promising results for the self-identification and ShapeNet classification tasks, indicating a fair chance for the CPST model to recognize unseen class data. In contrast, for ModelNet, CPCT does better in recognizing unseen data using the Contrastive model, while CPST shows no clear advantage over the Naive approach. This draws a subtle pattern: CPST excels with larger point cloud samples, whereas CPCT does better with smaller point cloud data. Further, for Gender classification with a leave-one-out sample scenario the average accuracy is recorded as **Naive ST: 44%, CPST: 54.6%, Naive PCT: 58.4% & CPCT: 74.99%**, demonstrating that CPST and CPCT significantly outperform random chance.

Table II represents the average loss for the generalization of models over the regression tasks of the unseen data sample. The results indicate that CPCT performs better in predicting unseen continuous values, specifically with height and weight tasks. We believe this to be the case because PCT's inherent augmentation facilitated by the SG layers helps to capture the features better, which directs us to further consider adapting the augmentation technique with ST, which could yield better results.

V. Discussion

A. Conclusion

In this work, we presented a new dataset of 3D point cloud body scans, along with the ShapeNet and ModelNet datasets, to conduct classification and regression experiments. The existing Naive Set Transformer and Naive Point Cloud Transformer are used as base models for the initial set of ex-

TABLE I: KL Divergence Results for Generalization

Model	Set Transformer		Point Cloud Transformer	
Task	*Naive*	*Contrastive*	*Naive*	*Contrastive*
Binned-Age	0.33	2.16	0.84	37.86
Self-Identification	9.98	9.18	1.67	5.64
ShapeNet	1.02	0.84	3.22	3.45
ModelNet	0.801	0.809	1.48	1.28

TABLE II: Regression Loss Results for Generalization

Model	Set Transformer		Point Cloud Transformer	
Task	*Naive*	*Contrastive*	*Naive*	*Contrastive*
Age prediction	0.068	0.087	0.074	0.08
Height prediction	0.033	0.066	0.067	0.047
Weight prediction	0.043	0.053	0.057	0.037

periments. Then, we introduced our novel approach, the CPST and CPCT models, to repeat the experiments for classification and regression tasks. Both models were evaluated for their generalization capability in handling unseen data samples for the tasks.

From the results of both the Naive Transformer and Contrastively pretrained models in classification and regression tasks, our contrastive pretraining boost the learning process by giving a head start. Among the pretrained models, CPST demonstrated superior performance with the 3D body scan dataset, while CPCT performed better with Shapenet data in weak generalization, with both contrastive models quickly reaching high performance on 3D body scans. However, with strong generalization, CPST showed better results in multi-class classification tasks, such as self-identification, ShapeNet, and ModelNet, while CPCT excellgied in regression tasks. Both models do well in gender classification tasks though both models struggled with binned-age generalization.

Overall, we observe a huge benefit with the contrastive pretrained model on 3D body scans compared to the ShapeNet and ModelNet datasets. The large point clouds in 3D body scans aid in effective augmentation, leading to better performance. This decreases with the mid-size point cloud data in ShapeNet and the lowest performance with ModelNet data. Thus, large batch sizes with sufficient space for subsampled data augmentation play a critical role in enhancing the performance with contrastive pretrained models.

B. Furture work

Based on our analysis of where the contrastively pretrained model struggles, we propose applying more complex augmentations to the point cloud data. Since PCT's simple zoom augmentation has shown promising results, adapting this approach for CPST could potentially improve performance. We also putforth idea of randomly choosing varying subset lengths of data and use a distance metric to extract portions of an object, creating a varying density zooming potentially enhancing performance.

Moving forward, contrastive-based models could serve as a foundation for generative models, utilizing the performance improvements shown by the approach to generate the point cloud data to form a 3D body scan or to create a matching missing part of an object. This engineered approach is to aid with the precision of forming the missing parts or to include more point clouds in case of insufficient data. This approach can also aid in increasing point clouds for the ShapeNet and ModelNet datasets and observe the resulting behavior.

REFERENCES

[1] Mohamed Afham, Isuru Dissanayake, Dinithi Dissanayake, Amaya Dharmasiri, Kanchana Thilakarathna, and Ranga Rodrigo. 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*, pages 9902–9912, 2022.

[2] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*, 2020.

[3] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[4] Nico Engel, Vasileios Belagiannis, and Klaus Dietmayer. Point transformer. *IEEE Access*, 9:134826–134840, 2021.

[5] Meng-Hao Guo, Jun-Xiong Cai, Zheng-Ning Liu, Tai-Jiang Mu, Ralph R. Martin, and Shi-Min Hu. Pct: Point cloud transformer, 2020.

[6] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1735–1742, 2006.

[7] Xian-Feng Han, Yi-Fei Jin, Hui-Xian Cheng, and Guo-Qiang Xiao. Dual transformer for point cloud analysis. *IEEE Transactions on Multimedia*, 25, 2023.

[8] Li Jiang, Shaoshuai Shi, Zhuotao Tian, Xin Lai, Shu Liu, Chi-Wing Fu, and Jiaya Jia. Guided point contrastive learning for semi-supervised point cloud semantic segmentation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6423–6432, 2021.

[9] J. Lee, Y. Lee, J. Kim, A. R. Kosiorek, S. Choi, and Y. W. Teh. Set transformer: A framework for attention-based permutation-invariant neural networks. arXiv preprint arXiv:1810.00825, 2018.

[10] Dening Lu, Qian Xie, Kyle Gao, Linlin Xu, and Jonathan Li. 3dctn: 3d convolution-transformer network for point cloud classification. *IEEE Transactions on Intelligent Transportation Systems*, 23(12):24854–24865, 2022.

[11] David Ludwig. Set transformer mnist. GitHub, 2022. https://github.com/DLii-Research/tf-settransformer/.

[12] Lucas Nunes, Rodrigo Marcuzzi, Xieyuanli Chen, Jens Behley, and Cyrill Stachniss. Segcontrast: 3d point cloud feature representation learning through self-supervised segment discrimination. *IEEE Robotics and Automation Letters*, 7(2):2116–2123, 2022.

[13] Jinyoung Park, Sanghyeok Lee, Sihyeon Kim, Yunyang Xiong, and Hyunwoo J Kim. Self-positioning point-based transformer for point cloud understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21814–21823, 2023.

[14] [TC]2 Introduces KX-16 Body Scanner. [TC]2 Introduces KX-16 Body Scanner, 3 2012.

[15] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. arXiv preprint arXiv:1706.03762, 2017.

[16] Taylor W. Webb, Ishan Sinha, and Jonathan D. Cohen. Emergent symbols through binding in external memory, 2021.

[17] Saining Xie, Jiatao Gu, Demi Guo, Charles R. Qi, Leonidas Guibas, and Or Litany. Pointcontrast: Unsupervised pre-training for 3d point cloud understanding. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, pages 574–591, Cham, 2020. Springer International Publishing.

[18] Guandao Yang, Xun Huang, Zekun Hao, Ming-Yu Liu, Serge Belongie, and Bharath Hariharan. Pointflow: 3d point cloud generation with continuous normalizing flows. *arXiv*, 2019.

[19] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip HS Torr, and Vladlen Koltun. Point transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 16259–16268, 2021.