Successful Contrastive Pretraining of Set Transformer

Divyashree Shivalingappa Koti
Computational and Data Science
Middle Tennessee State University
Murfreesboro, USA
dsk2v@mtmail.mtsu.edu

Joshua Lee Phillips
Computational and Data Science
Middle Tennessee State University
Murfreesboro, USA
Joshua.Phillips@mtsu.edu

***Abstract*—A transformer initially designed to deal with a sequence of data, is now adopted to deal with unordered sets, which is well known as Set Transformer. This project explores Set Transformer in studying point cloud 3D Body Scans for self-identification task. A standard simple Set Transformer is implemented using Induced Set Attention Blocks and Pooling layers to take advantage of computational speed, and improved precision [4] in the classification. Further Simple framework Contrastive Learning technique [5] is combined with Set Transformer as a reform technique to boost the pros offered by the standard Set Transformer and overcome the limitations, by speeding up the process, higher accuracy, greater stability, and improved generalization. Results from the experiments clearly demonstrate a Contrastive pre-trained model can achieve an accuracy of up to 100% in less than $\frac{2}{5}^{th}$ of time required by a standard Set Transformer while being able to achieve up to only up to 80% accuracy. The contrastive pre-trained model also indicates an improved generalization to unseen class data in comparison with the standard Set Transformer. Overall Contrastive pre-trained model exhibits more excelling performance than a standard Set Transformer. Which can be used as a good foundation for further explorations like Generative Adversarial Set Transformer (GAST) models in generating point cloud data.**

I. Introduction

A transformer is an initial model that has exhibited large sequence processing using a multi-head attention mechanism that has the capacity to learn the contextual information from different positions [1]. Which then further extended to image processing called as Vision Transformer(ViT), to utilize the efficiency offered by the transformer, where each image is made into patches and linearly embedded to create a 1D sequence as input to the transformer [2]. Although ViT has shown significant performance with the grid-like structured data it highly depends on the positional embedding. Also, ViT depends on the pixel values which becomes expensive with higher pixel values or loose precision with smaller values leading to an alternative Set Transformer to address these issues [4].

The set transformer is the extended version of a Transformer that is gaining significance in applications where positional embedding is insignificant. A set transformer takes in a set of unordered input to process and learn the underlying pattern. Unlike traditional grid-structured objects that are necessarily associated with a position, point cloud data can take advantage of a Set Transformer to be precisely learned as they are represented by the coordinates.

One potential application is in the field of fashion and design industry to virtually visualize, precisely personalized clothing and stitch them on the go with fine refinements. To facilitate this idea, this project explores Set Transformer and Contrastive learning techniques on point cloud 3D body scans. Where both methods are subjected to a classification task on Self-identification, in order to get a good grasp of how well the methodology aid in achieving the end goal.

To begin with, 75 people participated in getting the body scans that were collected by Dr. Frederick Steven Cottle using KX-16 Body Scanner [3][1]. Figure 1 shows an example of the point cloud 3D body scan[2]. All samples' point cloud data ranges from 51K to a maximum of 63K points.

In order to accomplish an efficient model we incorporate a standard Set Transformer, a pre-trained contrastive learning model fine-tuned on classification, and a systematic comparison study between weak generalization and strong generalization using both the techniques on the point cloud 3D body scan data.

II. Background

The set transformer was initially discussed by Lee J. et al. [4], citing solutions to permutation-invariant datasets of any size. The paper discusses the attention-based set transformer, using inducing points for a scalable approach. *Induced Set Attention Block(ISAB)* speeds up the computation and helps to capture the good features. *Pooling by MAB* using attention on the aggregated feature vectors to determine the influential instance before the softmax layer helps in performance improvement. Where the paper experiments with different

[1]Originally 3D body scans were collected for a psychology project.
[2]This is a subsampled representation of point clouds.

Fig. 1: 3D Body scan

datasets solving for classification on image datasets, maximum value regression, set anomaly detection, and Point cloud classification to exhibit the explored advantages. With respect to the point cloud classification experiment, conducted on ModelNet40 3D objects dataset, they were able to record the highest accuracy of **90%** using Induced Set Attention Blocks(ISAB) with pooling. However, the computational complexity of MAB is $\mathcal{O}(n^2)$ on large sets is time-consuming. Also, the highest accuracy is achieved comparatively on larger sets by ISAB+Pooling. A contrastive pre-trained model has the potential to address these issues, in speeding up the process, and may enable to achieve higher accuracy.

Contrastive learning is initially discussed in Dimensionality reduction by Learning an Invariant Mapping (DrLIM) [6] stating with prior knowledge a model is capable to draw all the neighbors (positive pairs) together while moving apart the non-neighboring (negative pairs) pairs in a latent space without the use of an external distance metric. Based on contrastive learning a simple framework(SimCLR) with ResNet-50 as a base model, presented in [5], experiments with the ImageNet dataset where for every derived minibatch a paired sample to its own augmented version is considered a positive pair, whereas the rest of the samples in the batch are considered to be their own class. This unlabelled learning by the contrastive pre-trained model can serve as a better starting point for a fine-tuning model helping to speed up the process and better accuracy in classification tasks, by extracting better features based on similarity and dissimilarity in instances. SimCLR study also focuses on how the larger batch size gives more negative pairs helping convergence while more training steps with large batch size fill in the gap between randomly sampled batches resulting in improved performance.

On exploring these methods, it becomes necessary to evaluate a model's robustness in classifying unseen categorical data. *Generalization* is a concept where a model's capacity is said to be weakly generalized if the model successfully recognizes a class or category of data that it has already encountered during training but differs in its representation. Whereas strong generalization is the model's capacity to identify a data class that it has never come across during the training process. This is similar to the human process of learning by variable-binding based on the pattern that already knows and draws a connection to the new unseen instance [8]. When a simple model is given this task it may perform poorly as its capacity is limited, whereas the contrastive model which has the capacity to bring the similar one near and pushes the dissimilar ones apart may comparatively have better performance. As a way to quantify the generalization *Kullback–Leibler divergence(KL divergence)* can be used to calculate the distance between weak and strong generalizations given by the equation 1 [9].

$$D_{KL}(P||Q) = \sum_{x \epsilon X} P(x) log\left(\frac{P(x)}{Q(x)}\right)$$

III. Methods

In this section, we discuss briefly model implementation and in detail experimental setup. We begin by implementing the standard set transformer using ISAB and pooling layer, which is used as a stand-alone model and as a base model for the Contrastive pre-trained model. To utilize the advantage offered by ISAB and the pooling layer as we discussed in the background we design the Set Transformer using these layers for the classification task. Sparse categorical cross entropy is used as a loss function during training in both standard Set Transformer, Contrastive pertaining, and fine-tuning model.

Expanding to the standard Set Transformer, we introduce SimCLR with a Set Transformer method to utilize the advantages offered by both methods in improving the performance of the classification and speed.

As our next step we implement the Simple framework Contrastive learning(SimCLR) technique, as specified in [5], where a positive pair is considered to be its own augmented self in the minibatch and the rest of the data points form negative pair. This model will be used as a pre-trained model to further finetune the downstream classification task.

As part of tracking the overall efficiency of the standard Set transformer Model and using the Contrastive Learned pre-trained model in classifying seen(Weak generalization) and unseen(Strong generalization) category data we use KL divergence on the normalized probabilities to calculate the distance. From which lower the value the model works better with strong generalization.

To conduct all the experiments 3D Body scan point cloud data is loaded by randomly choosing 51K point cloud data to keep all samples uniform. The experimental setup to train and validate the model goes with 80:20 on the subsampling of the point cloud data. For each batch of iteration, the subsample of point cloud data is selected randomly without repetition at the given point. For the standard and fine-tuning process batch size is kept constant to 4 and a subsample of size 8000 for the training process. For each of the reasonable experiments Monte Carlo simulation of 10 rounds was applied and accuracy was plotted as the evaluation metric. Based on the observed stability, variation, learning pace, and accuracy best model was chosen from each method for the final comparative study.

A. Set Transformer

The architectural design of the base model is set up as designed by D. Ludwig in [7]. Further, the hyperparameters *embedded dimension(embed_dim), inducing dimension(induce_dim), number of attention heads(num_heads), Stacks of ISAB layers(stack), regularization(dropout)*, learning rate, and activation function are tuned to obtain an optimal solution. For all of the experiments conducted Adam optimizer is kept constant, varying the learning rate of 1e-02 to 1e-04, and was observed with a very small value model would be in a local minima for a longer period of time whereas a higher rate, loss tends to decrease for the starting few epochs and ends up with exploding loss. Stack was varied between values 2 to 6, maintaining other hyperparameters depending on the resource availability, and was observed the stack with **3** gives the optimal solution. Regularization was experimented with 0.0, 0.05, and 0.2, and **0.05** was chosen as the best value. The remaining hyperparameters embed_dim(32,64), num_heads(4,16, 32), and induce_points(32,64,128) which are the major contributors, were observed to perform better in terms of stability and accuracy at **embed_dim: 64, num_heads:16 and induce_points:128**.
Overall a smaller architecture model learns at a slower pace, and an increasing variation. Whereas a moderate architecture with lower regularization performs better. This moderate architecture is run for 250 epochs and followed the 10 rounds of Monte Carlo simulation.

B. Contrastive learning

As a unique part of this work, *Contrastive learning with a Set transformer* is a novel approach to have experimented on the point cloud dataset. A set transformer is used as a base model to extract the feature embeddings required by the Contrastive Learning model to construct the logits. Set transformers used for the purpose are similar to the standard set transformer as we mentioned in the earlier section that includes the hyperparameter settings. Contrastive learning hyperparameter temperature is tried out with values [0.5, 1.0, 2.0]. Where 2.0 gave better results in terms of faster convergence, and stability during both the pre-train and fine-tuning phases. To obtain the pre-trained model we run the experiment for 200 epochs.
To conduct the experiment on the given dataset, 3D point cloud data, data is augmented by randomly choosing subsets of point cloud for the given subset sample. Further batches are constructed with replacement to form a longer batch size (typically in our case 32) which aligns as quoted in [5] and the number of batches per epoch is set to 100. Unlike training with a small batch size of 4, as we did with a standard set transformer model, the contrastive learning model learns well with a constant decrease in a loss, but during the fine-tuning process, it begins to learn at a faster pace but then it is observed to be stuck in the local minima. On repeating the experiment with an increase in batch size to 16 and 32, batch size of 32 was observed to be converging faster than the earlier batch sizes. Subsamples for these batch sizes were adjusted based on resource availability for 16 we could go up to 4k, and with **32** we could only get up to 1024-point cloud data points. Thus, Contrastive learning indeed benefits from a comparatively large batch size for the point cloud dataset but does not benefit much from a longer training period as after a certain loss value model is observed to be mostly constant. [5] also suggests the contrastive learning model benefits from a wide and deeper network, but with a set transformer an optimal value combination of embedded dimension, number of heads, and induce points, the pretraining and fine-tuning both become unstable especially as the convergence nears to zero.
On setting the pre-trained contrastive learned model, we define a linear layer with a non-linear GELU activation function and a regularization term set to 0.1 for the fine-tuning part. The fine-tuning part was tried just using the linear layer projection which still does good, but adding nonlinearity with regularization gives a slight bump in the learning process faster. For a thorough investigation on the linear projection during fine-tuning effect, we tried a projection set to [1024, 256, 128]. On careful observation projection dimension of 128 archives has good stability and lesser variation on the Monte Carlo simulation of 10 rounds. Finetuning is conducted with the same set of hyperparameters as in the optimizer, learning rate, and number of epochs as that of the Set transformer.

C. Generalization

To evaluate the model's ability to recognize an unknown class, we construct a model to train on all the category data except for one(*strong generalization*), each time of the experiment. So on every run, we train the model standard set transformer just like we did in the basic process(*weak normalization*) for the same amount of time and at the end of the learning, we validate on the left-out data class, and see how confused is the model and evaluate based on if this confusion matches with the distribution of the weak normalization.
Similarly, we try to experiment with having a pre-trained model as a base and further fine-tune, it to form a strong generalization matrix just like we did in the first set of experiments. This experiment is in the notion of a model being aware of the data in the contrastive learning pre-training process and to see if it is able to assign to its own class or give a better distribution similar to that of the weak generalization.
In both cases for comparison we use weak generalization as a standard matrix to calculate the KL divergence. Weak generalization aligns with strong generalization implementation.
Using Monte Carlo simulation, we take the final probability matrix from each run, which is basically a 75*75 confusion matrix for both weak and strong generalizations. These are averaged across the simulations, and then subjected to normalization, which is further given to the KL divergence to calculate the distance. As a way of probability distribution comparison, we do nullify(we use a small value of 1e-9) all the diagonal elements of the standard set transformer

matrices, so that we can compare with leave out one strategy, expecting to observe the similar distribution between both generalizations.

D. Visualization

On successful implementation of the standard Set Transformer and Contrastive Pre-trained model, we embarked on visualizing the extracted features by both methods, to see if either of the models was able to get those features that are human detectable. To accomplish this we sampled 10 feature vectors(taken before the softmax layer) for each sample and applied 2D Principle Component Analysis(PCA) and 2D t-Distributed Stochastic Neighbor Embedding(t-SNE) methods. PCA plot exhibited ambiguity between samples indicating the sample are not linearly separable for either of the methodologies. However, t-SNE gave pretty good clear results in distinguishing each result. With Standard Set Transformer we observed tiny group formation across the plot but did not quite give out useful feature information along either axis. Whereas with the Contrastive Pre-trained model the plot looked uniform, where some features could be observed to be on opposite sides(thin vs fat), and have an average body to lie along the 0 horizontal axis, which was unlike with Standard Set Transformer. Although we could observe some human visible features as expected it is not uniform, as the model captures added intrinsic features that are clearly separated in higher dimensions but tend to lie together in lower dimensions as 2D[3].

IV. Results

A. Standard Set transformer

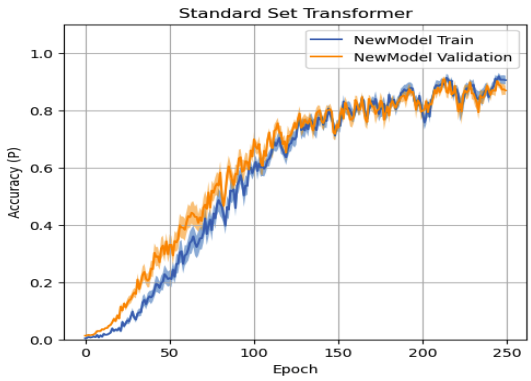

Fig. 2: Standard Set Transformer Accuracy Plot

The standard set transformer model is evaluated using the accuracy metric on the self-identification task by the validation data as shown in Figure 2. This figure is for the best results achieved by the moderate architecture of the model with a linear projection of 128 using the batch size of **4** and subsample size of **2048**. From observation, the growth of the accuracy seems to be stochastic in nature but overall it increases in a linear fashion portraying the learning ability of the model. Maximum accuracy achieved over **250** epochs is around **92%**.

B. Contrastive learning

The contrastive learned model comes in two parts one with the pre-trained contrastive model and the finetuning part for the self-identification classification task.

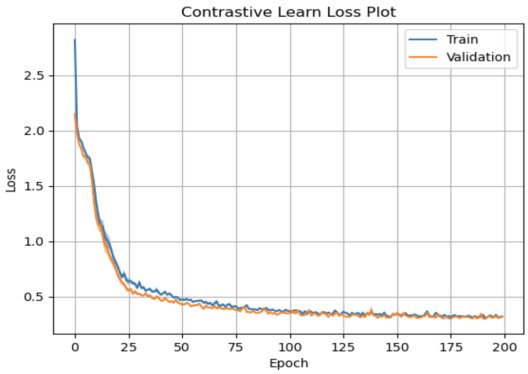

Fig. 3: Contrastive Learning Loss Plot

1) Pretrained model: The pre-trained contrastive model is a learning process represented by the loss of the model as shown in Figure3. This figure is for the final architecture chosen as discussed in the Methods section. The loss curvature starts off with a higher value but within the initial few epochs it accelerates and a sudden dip in the loss can be observed and falls below 0.5, this behavior remains the same across the architectures differing with the time period(epochs). Even with a batch size of 4 to 16 during the contrastive learning process, the loss is observed to be following the trend but it fails to present us with a Contrastive learning advantage during fine-tuning process. Figure 3 also represents the *Monte Carlo simulated loss curvature* for 10 rounds and observed that there is not much variation each time the model is trying to learn the underlying pattern[4].

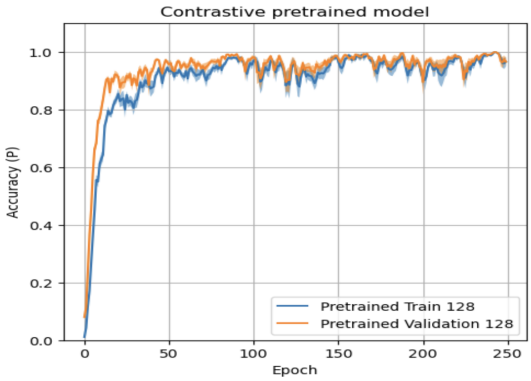

Fig. 4: Contrastive Learned Finetuning accuracy Plot

[3]Plot is not included here, can be cross-referenced on Github.

[4]Monte Carlo simulation is applied just for the plot purpose.

2) Fine-tuning: With a Contrastive pre-trained model and fine-tuning layers for the self-identification task, Figure 4 shows the mean accuracy plot of classification over time. Unlike the standard set transformer, we can clearly visualize that within 1st few epochs, there is an enormous jump, achieving an accuracy of over **80%** within **20** epochs, where it almost took *120* epochs to achieve by the standard set transformer. Later the model proceeds to achieve 100% comparatively at a slow pace. Model learning can be stopped at this point, but we keep it going to have a fair study between the models. This also helps us to observe the behavior of the model post the model's peak performance.

For the final chosen architecture, we observe not much variation and maintain stability for the most part. But as we discussed in the earlier section, where the loss pattern is observed to be the same, but during the fine-tuning process we observed that the stability of the model varies though they recover but the dip in accuracy for some architectures is recorded as low as 60. Another observation is unlike standard Set Transformer unexpected behaviors are also observed in certain cases such as a smaller model with a higher dropout(0.2) or moderate model with a higher number of heads(32) with a low dropout(0.05), where the model behaves almost similar to that of a standard set transformer and also exhibits more stochasticity than the standard model. Training for longer duration on large models with higher projection dimensions in the Fine-tuning layer will eventually drop the accuracy resulting in instability of the model[5].

TABLE I: Results for Generalization

Method	KL Divergence
Standard Set Transformer	6.60
Contrastive Pre-trained model	6.25

3) Generalization: Using KL divergence on normalized probability distribution to calculate the distance between weak generalization and strong generalization produced by standard Set Transformer and Contrastive Pre-trained model is as presented in Table I. Where the *Contrastive Pre-trained model* marginally **excels** the standard Set Transformer, indicating there is a fairly high chance for the Contrastive Pre-trained model to recognize unseen class data than the standard model.

V. Discussion

A. Conclusion

In this work, we classify 3D body scans to self-identify, using a standard Set Transformer, a Contrastive Pre-trained model, and the generalization of each method. Using the unique approach of Contrastive pre-trained with the Set Transformer model is able to perform significantly better in terms of accuracy, speed, and stability than the conventional Set Transformer. The Contrastive Pre-trained model also exhibited its improved performance in generalizing the unseen categorical data compared to the standard Set Transformer.

B. Furture work

As our next step, we would like to use the Contrastive pre-trained model as a base for Generative Adversarial Set Transformer (GAST) approach to utilize the performance improvement shown by the approach, to generate the point cloud data to form a 3D body scan or to create a matching partial body part. This engineered approach is to aid with the precision of forming the missing scan part or to include more point clouds in case of insufficient data.

[5]Data or plot is not included, can be referred on github

REFERENCES

[1] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention Is All You Need. ArXiv. /abs/1706.03762

[2] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., & Houlsby, N. (2020). An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. ArXiv. /abs/2010.11929

[3] [TC]2 Introduces KX-16 Body Scanner. Textile World. (2012, March 20). https://www.textileworld.com/textile-world/new-products/2012/03/tc2-introduces-kx-16-body-scanner-3/

[4] Lee, J., Lee, Y., Kim, J., Kosiorek, A. R., Choi, S., & Teh, Y. W. (2018). Set Transformer: A Framework for Attention-based Permutation-Invariant Neural Networks. ArXiv. /abs/1810.00825

[5] Chen, T., Kornblith, S., Norouzi, M., & Hinton, G. (2020). A Simple Framework for Contrastive Learning of Visual Representations. ArXiv. /abs/2002.05709

[6] R. Hadsell, S. Chopra and Y. LeCun, "Dimensionality Reduction by Learning an Invariant Mapping," 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06), New York, NY, USA, 2006, pp. 1735-1742, doi: 10.1109/CVPR.2006.100.

[7] Ludwig, D. (n.d.). Set transformer MNIST. GitHub. https://github.com/DLii-Research/tf-settransformer/

[8] Webb, T. W., Sinha, I., & Cohen, J. D. (2020). Emergent Symbols through Binding in External Memory. ArXiv. /abs/2012.14601

[9] Wikimedia Foundation. (2023, July 16). Kullback–Leibler divergence. Wikipedia. https://en.wikipedia.org/wiki/Kullback%E2%80%93Leibler_divergence