STATISTICAL CONVOLUTION ON UNORDERED POINT SET

Anh-Duc Nguyen[1] *Seonghwa Choi*[1] *Woojae Kim*[1] *Sanghoon Lee*[1] *Weisi Lin*[2]

[1]Yonsei University [2]Nanyang Technological University

ABSTRACT

In this paper, we propose a new convolutional layer for neural networks on unordered and irregular point set. Most research advanced to date usually face multiple problem related to point cloud density and may require ad-hoc neural network architectures, which overlooks the huge treasure of architectures from computer vision or language processing. To mitigate these shortcomings, we process a point set at its distribution level by introducing statistical convolution (StatsConv). The spotlight feature of StatsConv is that it extracts various statistics to characterize the distribution of the input point set, which makes it highly scalable compared to existing point convolution operators. StatsConv is fundamentally simple, and can be used as a drop-in in any contemporary neural network architecture with negligible changes. Thorough experiments on point cloud classification and segmentation demonstrate the competence of StatsConv compared to the state of the art.

***Index Terms*—** Point Cloud, Convolution, Geometry, Deep Learning

1. INTRODUCTION

Convolutional neural networks (CNNs) [1] have easily been the most impactful factor to complex intelligent systems recently [2–7]. Thanks to them, the states of the art of various tasks in computer vision and natural language processing have been heightened rapidly. However, most of the achievements are in domain where signals are regularly distributed over temporal or spatial grids. In irregular domains, their performance is nowhere near that in the regular counterparts.

3D point processing is important as there is no overhead of converting point cloud to other regular representations like voxels, which also reduces the risk of losing information [8], but an effective and common way to process a point cloud has not yet been established. In our point of view, there are three main difficulties mounting to this shortcoming. Firstly, to apply common operators (for *e.g.*, convolution), a locality of points must be predefined, but it is not obvious how to effectively do this. Secondly, the semantics of a point cloud is independent of the point order, which implies the learned function should be permutation-invariant. Lastly, a point cloud can be of arbitrary size, which makes a learning system struggle to learn and scale.

Recently, there have been several studies investigating the problem of deep learning on unstructured point set [9–17]. PointNet [9] and its succession, PointNet++ [10], are the two flagships to tackle this problem. In general, they learn multilayer perceptrons (MLPs) to process input point clouds, and resort to max pooling to achieve permutation invariance. However, PointNet and PointNet++ are engineered network architectures, so they are not versatile and are difficult to blend into other networks. Also, max pooling throws away information which might be helpful to make inference about the input [18].

Zaheer *et al.* [13] proposed a set convolution layer which learns a permutation-invariant and/or -equivariant by means of max and sum operators, but these are very simple operators and may not capture and propagate all the necessary information about the point cloud to the next layer.

PointCNN [14] learns a linear transformation per neighborhood to weight and permute a subset of points, and then performs the usual matrix multiplication to map the point cloud to another space. However, learning that way does not guarantee permutation-equivariance/invariance.

In this paper, we propose a novel convolution operator based on the statistics of the point cloud. Our formulation consists of a global feature, which is based on different moments extracted from the point cloud, and a per-point feature, which is a simple non-linear transformation of the input point. The proposed convolution operator is a simple and primitive LEGO cell, which allows us to construct suitable architectures suited to any point set problem. We demonstrate the competitive performance of the proposed operator over existing states of the art in two tasks: point cloud classification and segmentation. For brevity, we dub our layer as StatsConv. Code to reproduce the paper is available at `https://github.com/justanhduc/StatsConv`.

2. PRELIMINARIES

2.1. Sufficient Statistics

In statistics, *sufficiency principle* concerns about finding a *sufficient statistic* for the distribution parameters such that any inference about the parameters of the distribution given the sufficient statistic should be the same no matter what value of

This work was supported by Samsung Research Funding Center of Samsung Electronics under Project Number SRFC-IT1702-08.

the random variable is observed. Following [19], the formal definition of a sufficient statistic is recalled as follows:

Definition 1 (Sufficient statistics)**.** *A statistic $T(X)$ is sufficient for the distribution parameter θ if the distribution of the sample X conditioning on the value of $T(X)$ is independent of θ.*

The sufficiency principle is attractive in two aspects: (1) the information about the "shape" of the distribution is fully preserved, and (2) sufficient statistics is relatively robust to the size of the sample thanks to the reduction. Thus, the sufficiency principle offers a way to efficiently process unordered point sets whose sizes are potentially large.

2.2. Sample Central Moments

In statistics, a moment is a quantitative measure that characterizes the shape of a distribution. Sometimes, it is more common to find the moments about the mean of a distribution, which is called *central moments*. Mathematically, the k^{th} central moment is defined as:

$$\mu_k = E[(X - E[X])^k], \tag{1}$$

where $E[X]$ denotes the expectation of the random variable X. In practice, the integral is usually intractable or the probability density function is not available for interesting data, so one has to resort to their estimations based on a random sample drawn from the distribution, which is so-called *sample central moments*. Given a random sample $\mathcal{X}$ containing observations X, the sample mean (first moment) is calculated as:

$$\bar{X} = \frac{1}{|\mathcal{X}|} \sum_{X \in \mathcal{X}} X, \tag{2}$$

and the k^{th} sample central moment is given by:

$$\bar{\mu}_k = \frac{1}{|\mathcal{X}|} \sum_{X \in \mathcal{X}} (X - \bar{X})^k. \tag{3}$$

The sample central moments are fairly simple and consistent estimators. However, they are usually biased, and more critically, they often are not truly sufficient statistics. Nevertheless, they can still be a good alternative when the parametric form of the distribution is unknown.

3. STATISTICAL CONVOLUTION ON UNORDERED POINT SET

3.1. Statistical Convolution

Figure 1 lays out a top view of the proposed method. Let $\mathcal{X}$ be an unordered input point cloud and $X_i \in \mathcal{X}$ be a point in $\mathcal{X}$. A point X_i may contain any kind of features, which can be 3D coordinates, color information, or abstract features from intermediate network layers. To reduce and fix the cardinality of $\mathcal{X}$ without losing its "shape" information significantly, we leverage the properties of the sample moments and extract k different moments $\bar{\mu}_1, \bar{\mu}_2, ..., \bar{\mu}_k$ from the set $\mathcal{X}$ following (3). To make the features more descriminative, we project these moments into a new space by applying a non-linear fully-connected (FC) layer:

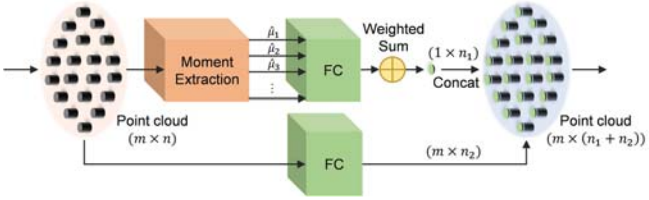

Fig. 1: A schematic of StatsConv. StatsConv has two branches: moment extraction and FC branches. In the moment extraction branch, different moments are extracted and passed through an FC layer before being linearly combined together. In the second branch, the point cloud is mapped point-wise to another space. The outputs of the two branches are concatenated.

$$\bar{\mu}'_i = fc_\sigma(\bar{\mu}_i; W_{\bar{\mu}}, b_{\bar{\mu}}), \tag{4}$$

where the trainable parameters $W_{\bar{\mu}}$ and $b_{\bar{\mu}}$ are shared for all the moments. Next, we reduce these moments to a single feature vector by:

$$\bar{\mu} = \sum_{i=1}^{k} w_i \bar{\mu}'_i, \tag{5}$$

where each w_i is a trainable scalar. This feature vector acts as not only a global feature that encapsulates the semantics but also a top-down local feature which captures most of the interesting curvatures and twists of the point cloud distribution. We refer to (5) as an *all-reduce* StatsConv.

In problems such as point cloud segmentation where we are required to label each point, it is crucial to have a per-point feature for every point in the input set. To do so, we simply lift each point in the set to a new dimension via an FC layer:

$$X' = fc_\sigma(X; W, b). \tag{6}$$

To propagate the per-point and global features to the next layer, we concatenate the two together and obtain the output as:

$$Y = X' \oplus \bar{\mu}. \tag{7}$$

Inspired by DeepSets [13], we propose another version which propagates the sum of the per-point and global features:

$$Y = X' + \bar{\mu}. \tag{8}$$

We note that for the same output dimension, the concatenation version is "thinner" as the dimension of each of the outputs of fc_σ is only half of the output dimension.

Our formulation of StatsConv bears some resemblance to PointNet [9] and DeepSets [13]. In PointNet, the authors proposed to extract point-specific feature, which promotes the equivariance property, and global feature extracted by max pooling, which makes the network invariant to point order. Similarly, DeepSets also extracts per-point feature and leverages max and sum operators to achieve both invariance and equivariance. We note that max and sum are also statistics of a random sample. In that sense, StatsConv can be seen as a generalization of PointNet and DeepSets. This implies that even though permutation-invariance and equivariance are not our original goal, StatsConv actually possesses these properties, which we show in the next section.

3.2. Properties

Permutation Invariance. A function $\mathbf{f}$ is permutation invariant if for any permutation π, we have:

$$\mathbf{f}(\mathcal{X}) = \mathbf{f}(\mathcal{X}_\pi), \tag{9}$$

where $\mathcal{X}_\pi = \pi(\mathcal{X})$ is the same set $\mathcal{X}$ but having a different ordering of elements. In other words, the function value is independent of the order of X_i. Clearly, all the moments characterize the distribution of the points, so they do not change when the points are ordered differently. All the subsequent transformations also do not concern about the set order, which makes StatsConv invariant to permutation.

Permutation Equivariance. If a function is permutation-equivariant, a permutation of the inputs results in the same permutation of the outputs. In mathematical terms, a function $\mathbf{f}$ is permutation-equivariant if:

$$\begin{aligned}\mathbf{f}(\mathcal{X}) &= \mathbf{f}([X_{\pi(1)}, X_{\pi(2)}, ..., X_{\pi(n)}]) \\ &= [f_{\pi(1)}(\mathcal{X}), f_{\pi(2)}(\mathcal{X}), ..., f_{\pi(n)}(\mathcal{X})],\end{aligned} \tag{10}$$

where π is any permutation. Since FC transforms every point in a point cloud individually and independently, it is obviously equivariant to permutation. Moreover, the global feature derived from moments is permutation-invariance, and hence permutation-equivariant. Thus, the formulations in (7) and (8) are also independent of the point order.

Weak Set-Density Invariance. To demonstrate this, we first show that the all-reduce version of StatsConv converges to a fixed point as the point cloud density increases, which enables the general StatsConv to be weakly density-invariant, by proving the following theorem:

Theorem 1. *The all-reduce StatsConv with an activation function continuous almost everywhere converges in probability to a fixed constant defined by the population moments as the sample size increases.*

Proof. Suppose the activation function in (4) is continuous almost everywhere. Let $\hat{X}^{(n)} := \bar{X}$ and $\mu_i^{(n)} := \bar{\mu}_i$ be the sample mean and i^{th} moment calculated from a random sample of size n, respectively. By the weak *Law of Large Number*, the sequence $\hat{X}^{(n)} \xrightarrow{P} \mu$ as $n \to \infty$, *i.e.*, the sample mean converges to the true mean in probability when the sample size increases to infinity. *Continuous Mapping Theorem* says that if $\hat{X}^{(n)} \xrightarrow{P} \mu$, then it follows that $\mu_i^{(n)} \xrightarrow{P} \mu_i$ because the derivation of $\mu_i^{(n)}$ from $\hat{X}^{(n)}$ in (3) is continuous. Moreover, $\mu_i'^{(n)} := fc_\sigma(\mu_i^{(n)}; W_{\bar{\mu}}, b_{\bar{\mu}})$ also tends to $\mu_i' := fc_\sigma(\mu_i; W_{\bar{\mu}}, b_{\bar{\mu}})$ thanks to the continuity assumption of σ. Finally, by *Slutsky's Theorem*, the linear combination $\mu'^{(n)} := \sum_{i=1}^k w_i \mu_i'^{(n)}$ converges in probability to $\mu' := \sum_{i=1}^k w_i \mu_i'$ given the fact that each $\mu_i'^{(n)}$ converges in probability to μ_i', and each w_i is a constant. Thus, the all-reduce StatsConv converges to a fixed point which is a weighted combination of the population moments. □

The proof above holds only when all the estimators of the true central moments are unbiased. In practice, this might not be the case. For instance, the unbiased estimator of the variance is obtained by dividing by $|\mathcal{X}| - 1$, not $|\mathcal{X}|$ as in (3). Nevertheless, they are asymptotically consistent estimators of the true moments, which means when $|\mathcal{X}|$ is large, the bias can be negligible, so the theorem still holds up to some tolerance.

3.3. Implementation Details

In our implementation, if not mentioned otherwise, we set the number of sample central moments to be six. Besides the central moments, we also resorted to three order statistics which includes max, min, and median. We utilized the same training/testing split and augmentation pipeline for both ModelNet datasets and ShapeNet-part as PointNet [9].

We implemented ResNet18 [20] and UNet [21] for classification and segmentation, respectively. For both tasks, we minimized the cost functions using SGD with a learning rate 3e-3 together with a momentum term of 0.9. For more details, we refer readers to the Supplementary Materials.

4. EXPERIMENTAL RESULTS

4.1. Point cloud classification

The mean classification accuracies of all the benchmarking methods on ModelNet10 is tabulated in Table 1. It can be seen that for this classification task, our method is very competitive against recent models, and even outperforms the state of the art in ModelNet10. Interestingly, we failed to train PointNet on ModelNet10 using the same settings for ModelNet40.

Table 2 delivers the image classification accuracies on MNIST and CIFAR10. On MNIST, our StatsConv is highly competitive to other models even though in our experiment, we used only 160 points per cloud, which is far fewer than the other competing models. On CIFAR10, while PointNet++ totally fails the task, our model still provides a meaningful

Table 1: Classification accuracy (%) on ModelNet10 and ModelNet40. The accuracy inside brackets is obtained in the case of using only 128 points per cloud. Red, blue, and green indicate the first, second, and third best results, respectively.

Method	ModelNet10	ModelNet40
ShapeNets [22]	83.5	77.3
PointNet [9]	55.5	89.2 (87.1)
PointNet++ [10]	-	90.7 (86.0)
DeepSets [13]	-	87.1 (82.2)
ECC [15]	90.0	87.4
VSL [16]	91.0	84.5
StatsConv-cat	91.2 (90.2)	89.6 (88.7)
StatsConv-sum	29.7	89,3
StatsConv (max)	90.7	-
StatsConv (4 moments)	91.0	-

Table 2: Classification accuracy (%) on MNIST and CIFAR10.

Method	MNIST	CIFAR10
ECC [15]	99.1	-
PointNet++ [10]	99.5	10.0
StatsConv	99.0	64.4

Table 3: Mean IoU scores (%) on ShapeNet.

Method	mIoU
3DCNN [9]	79.4
Yi *et al.* [17]	81.4
PointNet [9]	83.7
StatsConv	80.5

result. This reinforces our claim that a wholly-engineered architecture like PointNet/PointNet++ might not be as versatile as a building block like StatsConv as StatsConv is able to work on many tasks and datasets with reasonable performance.

4.2. Segmentation

In this task, we validated the proposed UNet on the ShapeNet part segmentation dataset [17]. As can be seen from Table 3, our result is not quite close to existing work, but the performance is still reasonable. Nevertheless, the UNet model here was adapted with only minimal changes and not much effort in parameter tuning. By a better setting of hyperparameters and/or specific architecture tuning, it is expected that the model achieves higher performance, but this is not our original goal.

4.3. Ablation study

Number of statistics. We demonstrated the influence of the number of statistics on the performance in Table 1, which shows the best results when using nine moments. Also, from Figure 2, we can see that instances from semantically similar classes (for *e.g.*, table and desk or night stand and dresser) are better separated in feature space when using the nine moments.

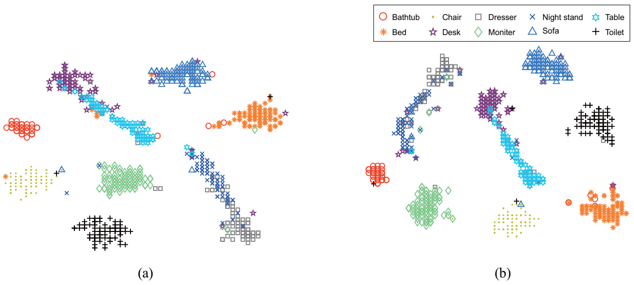

Fig. 2: T-SNE scatter plot of the object features on ModelNet10 test set when using (a) StatsConv with only max and (b) the default StatsConv.

It can be concluded that the number of extracted statistics influences the performance of StatsConv, and that employing barely max/sum as in existing studies is not sufficient.

Weak invariance to set size. In Table 1, we demonstrate the classification results when using only 128 points per cloud in brackets. The results suggest that for StatsConv to work properly, all we need is a representative point cloud so that its moments can be reliably estimated, while its density is weakly relevant.

5. CONCLUSION

We have introduced StatsConv, a general-purpose operator that operates on unordered and irregular signal domains. We have shown that much information about point sets can be retained by modeling the point cloud distribution via means of sample central moments. Our work generalizes some existing works that relying on max or sum to aggregate the information of point cloud as these operators are also statistics. The rigorous experiments showed that our novel formulation surpasses the state of the art in some benchmarks while remaining competitive in some others even though we simply applied existing architectures powered by StatsConv to these tasks.

6. REFERENCES

[1] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural Computation*, vol. 1, no. 4, pp. 541–551, 1989. 1

[2] Woojae Kim, Anh Duc Nguyen, Sanghoon Lee, and Alan Conrad Bovik, "Dynamic Receptive Field Generation for Full-Reference Image Quality Assessment," *IEEE Transactions on Image Processing*, vol. 29, pp. 4219–4231, 2020. 1

[3] Inwoong Lee, Doyoung Kim, and Sanghoon Lee, "3D Human Behavior Understanding using Generalized TS-

LSTM Networks," *IEEE Transactions on Multimedia*, pp. 1–1, mar 2020. 1

[4] Jongyoo Kim, Anh Duc Nguyen, and Sanghoon Lee, "Deep CNN-Based Blind Image Quality Predictor," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 1, pp. 11–24, jan 2019. 1

[5] Woojae Kim, Jongyoo Kim, Sewoong Ahn, Jinwoo Kim, and Sanghoon Lee, "Deep Video Quality Assessor: From Spatio-Temporal Visual Sensitivity to a Convolutional Neural Aggregation Network," in *Proceedings of the European Conference on Computer Vision (ECCV)*, Munich, Germany, 2018, vol. 11205 LNCS, pp. 224–241, Springer. 1

[6] Inwoong Lee, Doyoung Kim, Seoungyoon Kang, and Sanghoon Lee, "Ensemble Deep Learning for Skeleton-Based Action Recognition Using Temporal Sliding LSTM Networks," in *Proceedings of the IEEE International Conference on Computer Vision*. dec 2017, vol. 2017-October, pp. 1012–1020, Institute of Electrical and Electronics Engineers Inc. 1

[7] Anh Duc Nguyen, Seonghwa Choi, Woojae Kim, Sewoong Ahn, Jinwoo Kim, and Sanghoon Lee, "Distribution Padding in Convolutional Neural Networks," in *Proceedings - International Conference on Image Processing, ICIP*. sep 2019, vol. 2019-Septe, pp. 4275–4279, IEEE Computer Society. 1

[8] Yaoqing Yang, Chen Feng, Yiru Shen, and Dong Tian, "Foldingnet: Point cloud auto-encoder via deep grid deformation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 206–215. 1

[9] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas, "Pointnet: Deep learning on point sets for 3D classification and segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 652–660. 1, 3, 4

[10] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," in *Advances in Neural Information Processing Systems*, pp. 5099–5108. 1, 4

[11] Seonghwa Choi, Anh-Duc Nguyen, Jinwoo Kim, Sewoong Ahn, and Sanghoon Lee, "Point Cloud Deformation for Single Image 3d Reconstruction," *2019 IEEE International Conference on Image Processing (ICIP)*, pp. 2379–2383, 2019. 1

[12] Anh-Duc Nguyen, Seonghwa Choi, Woojae Kim, and Sanghoon Lee, "GraphX-Convolution for Point Cloud Deformation in 2D-to-3D Conversion," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. oct 2019, pp. 8627–8636, IEEE. 1

[13] Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Ruslan R Salakhutdinov, and Alexander J Smola, "Deep sets," in *Advances in neural information processing systems*, pp. 3391–3401. 1, 2, 3, 4

[14] Yangyan Li, Rui Bu, Mingchao Sun, Wei Wu, Xinhan Di, and Baoquan Chen, "Pointcnn: Convolution on x-transformed points," in *Advances in Neural Information Processing Systems*, pp. 828–838. 1

[15] Martin Simonovsky and Nikos Komodakis, "Dynamic edge-conditioned filters in convolutional neural networks on graphs," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3693–3702. 1, 4

[16] Shikun Liu, Lee Giles, and Alexander Ororbia, "Learning a hierarchical latent-variable model of 3d shapes," in *2018 International Conference on 3D Vision (3DV)*. pp. 542–551, IEEE. 1, 4

[17] Li Yi, Vladimir G Kim, Duygu Ceylan, I Shen, Mengyan Yan, Hao Su, Cewu Lu, Qixing Huang, Alla Sheffer, and Leonidas Guibas, "A scalable active framework for region annotation in 3d shape collections," *ACM Transactions on Graphics (TOG)*, vol. 35, no. 6, pp. 210, 2016. 1, 4

[18] Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton, "Dynamic routing between capsules," in *Advances in neural information processing systems*, pp. 3856–3866. 1

[19] George Casella and Roger L Berger, *Statistical inference*, vol. 2, Duxbury Pacific Grove, CA, 2002. 2

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778. 3

[21] Olaf Ronneberger, Philipp Fischer, and Thomas Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. pp. 234–241, Springer. 3

[22] Zhirong Wu, Shuran Song, Aditya Khosla, Xiaoou Tang, and Jianxiong Xiao, "3d shapenets for 2.5 d object recognition and next-best-view prediction," *arXiv preprint arXiv:1406.5670*, vol. 2, no. 4, 2014. 4