SyncSpecCNN: Synchronized Spectral CNN for 3D Shape Segmentation Supplementary Material

Li Yi[1] Hao Su[1] Xingwen Guo[2] Leonidas Guibas[1]
[1]Stanford University [2]The University of Hong Kong

This document provides a list of supplemental materials that accompany the main paper.

- **SpecTN Initialization** - We discuss technical details about how we initialize SpecTN including how we generate a set of pre-computed functional maps (see Section A).
- **Implementation Details** - We describe implementation details as well as hyperparameter choices in Section B.
- **Partial Shape Part Segmentation** - We provide additional per-category IoU for partial shape part segmentation task and compare our approach with ACNN [1] in Section C.
- **Spectral Dilated Kernel Parametrization** - In Section D, We provide more diagnosis of dilated kernel parametrization in analogy to Table 4 of the main paper. Differently, we use vertex geometric features as network input and these features are computed as is in [4].
- **Robustness to Sampling Density Variation** - We show the robustness of our approach w.r.t point cloud density variation in Section E.
- **Normal Prediction** Another graph vertex function prediction task is conducted in Section F to validate the generalizability of our approach to very local properties. We predict shape vertex normals and the results are visualized in Figure 2.

A. SpecTN Initialization

A set of precomputed functional maps are used for SpecTN pretraining. We mentioned in the main paper that the functional map C from S to the average shape $\bar{S}$ could be induced from the spatial correspondences between S and $\bar{S}$, by the primal-dual relationship. Once we have the bases of S and $\bar{S}$, as well as the rough spatial correspondences between them from the volumetric occupancy, the functional map can then be discovered by the approach proposed in [3].

To be specific, we use $\boldsymbol{B}_v$ to denote the volumetric reparametrization of graph Laplacian eigenbases $\boldsymbol{B}$ for each shape S, and use $\bar{\boldsymbol{B}}_v$ to denote the graph Laplacian eigenbases of $\bar{S}$. $\boldsymbol{B}_v$ and $\bar{\boldsymbol{B}}_v$ both lie in the volumetric space and their spatial correspondence is natural to acquire. The functional map C_{pre} aligning $\boldsymbol{B_v}$ with $\bar{\boldsymbol{B}}_v$ could be computed through simple matrix multiplication $C_{pre} = \bar{\boldsymbol{B}}_v^T \boldsymbol{B}_v$. The computed functional map will serve as supervision and SpecTN is pretrained to minimize the loss function $||C - C_{pre}||_F^2$.

It is worth mentioning that, if the shapes under consideration are diverse in topology and geometry, i.e. shapes from different categories, aligning every shape to a single "average" shape might cause unwanted distortion. Therefore we leverage multiple "average" shapes $\{\bar{S}_i\}_{i=1}^n$ and use a combination of their spectral domains as the canonical domain. Specifically, we assign each shape S to its closest "average" shape under some global similarity measurement (i.e. lightfield descriptor) and use $\{a_i\}_{i=1}^n$ to represent such assignment, namely $a_i = 1$ if S is assigned to $\bar{S}_i$ and $a_i = 0$ otherwise. Also we use $\bar{\boldsymbol{B}}_{vi}$ to denote the spectral bases of $\bar{S}_i$. Then the functional map C_{pre} for each shape S could be computed through $C_{pre} = [a_1\bar{\boldsymbol{B}}_{v1}\ a_2\bar{\boldsymbol{B}}_{v2}\ ...\ a_n\bar{\boldsymbol{B}}_{vn}]^T\boldsymbol{B}_v$. The SpecTN is pretrained to predict a functional map which only synchronizes spectral domain of each shape to its most similar "average" shape.

B. Implementation Details

In most of our experiments, input shapes are represented as point cloud with around $2000 - 3000$ points. Given an input shape point cloud, we build a k-nearest neighbor graph $\mathcal{G}$ first. We use $k = 6$ in all our experiments. Then a graph weight matrix W could be constructed in which $W_{i,j} = \frac{1}{d_{i,j}^2}$ if point i and j are connected, 0 otherwise. We then compute the symmetric normalized graph Laplacian L as $L = I - D^{-1/2}WD^{-1/2}$, where D is the degree matrix and I denotes identity matrix. Since many natural functions we care about could be depicted by a small number of low-frequency Laplacian eigenbases, we compute and use the smallest 100 eigenvalues as well as the corresponding eigenbases for each L in all our experiments.

Layer	1	2	3	4	5	6	7	8	9	10
Dilation (γ)	1	1	4	4	16	16	64	64	1	1
SpecTN	No	No	No	No	No	No	Yes	Yes	No	No
#Kernel Param	7	1	7	1	7	1	45	45	7	1
#Out Channel	c	c	c	c	2c	2c	2c	2c	2c	2c

Table 1. Parameters used in different layers of the architecture, including dilation parameter γ which controls convolution kernel size, whether use spectral transformer network (SpecTN), the number of learnable parameters in convolution kernels, the number of output channels after each convolution operation.

The choice of dilation parameters γ, number of output channels after each convolution layer, number of learnable parameters in each convolution kernel are shown in Table 1. We choose $c = 50$ in all of our experiments. As is mentioned, we only consider the problem of synchronizing the low-frequency end of different spectral domains, so we choose to predict a functional map $C \in \mathbb{R}^{15\times 45}$ in all our experiments, which maps the first 15 eigenbases of each individual spectral domain into a canonical domain of dimension 45. Notice the dimension of canonical domain is larger that each individual domain to allow very different shapes to be mapped into different subspaces.

C. Partial Shape Part Segmentation

We provide per-category IoU for partial shape part segmentation experiments in this section, which is not included in the main paper due to space limit. We compare our approach with baseline method ACNN [1] and the results are shown in Table 2

Notice our approach outperforms ACNN on most categories and the margin between ACNN and ours is large on average. Our approach is also more robust to data incompleteness since its performance drop from complete data segmentation results is less significant than ACNN.

D. Spectral Dilated Kernel Parametrization

We redo the diagnosis experiment about spectral dilated kernel parametrization, with geometric features as network input instead of XYZ coordinates of graph vertices. These geometric features are computed as is in [4]. Similarly we generate Table 3 in analogy to Table 4 of the main paper. Compared with Table 4 of the main paper, small kernels alone deliver better performance. This is because some multi-scale information is already captured in geometric features used. Note that our multi-scale kernel construction still performs the best, partially due to the advantage that our dilated kernel design achieves a spectral counterpart for spatial pooling.

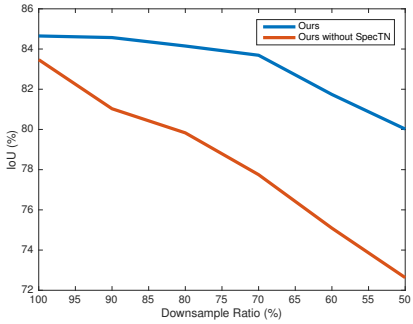

Figure 1. We evaluate the robustness of our model to sampling density change. Test shapes are downsampled by different ratios and fed into our network. We compute the segmentation IoU for different downsample ratios and show it here. With SpecTN, our framework becomes more robust to sampling density change.

E. Robustness to Sampling Density Variance

In this experiment, we evaluate the robustness of our approach w.r.t point cloud density variation. To be specific, we train our SyncSpecCNN for shape segmentation on the point cloud provided by [4] first. Then we downsample the point cloud under different downsample ratio and evaluate our trained model to check how segmentation performance would change. Again we evaluate our approach with/without SpecTN and the result is shown in Figure 1.

By introducing SpecTN, our framework becomes more robust to sampling density variation. Our conjecture is that sampling density variation may result in large spectral space perturbation, therefore being able to synchronize different spectral domains becomes especially important.

F. Normal Prediction

Our framework could learn generic graph functions not limited to part segmentation or keypoint prediction. To validate this point, we leverage our proposed SyncSpecCNN to learn another type of graph vertex function, vertex normal function. Specifically, our SyncSpecCNN takes the XYZ coordinate function of graph vertices as network input and predicts vertex normal as output. The network is trained to minimize the L2 loss between ground truth normals and predicted normals. We use the official train/test split provided by [2] and visualize some of the normal prediction results from test set in Figure 2.

It can be seen our predictions are very close to the ground truth at most of the time.Even on thin structures the normal predictions are still reasonable. One problem of our prediction is that it tends to generate smoothly transiting normals along the boundary while the ground truth is sharper. This is due to the fact that we are using a small number of eigenbases in our experiments, which is

	mean partial	mean complete	plane	bag	cap	car	chair	ear-phone	guitar	knife	lamp	laptop	motor-bike	mug	pistol	rocket	skate-board	table
ACNN	69.21	79.63	62.73	63.26	58.90	38.25	70.59	**68.68**	**88.08**	74.58	61.49	87.03	31.90	79.92	62.98	35.70	68.41	76.07
Ours1	76.19	83.48	71.01	77.61	64.78	56.05	78.97	68.50	84.63	82.01	73.02	91.40	40.71	87.34	72.60	**42.53**	80.61	79.55
Ours2	**78.02**	**84.74**	**74.55**	**82.58**	**65.36**	**58.12**	**80.41**	65.55	84.75	**82.53**	**77.39**	**93.15**	**43.12**	**90.24**	**74.71**	42.17	**83.22**	**80.51**

Table 2. IoU for part segmentation on incomplete shapes. Note that for comparison, we not only report mean IoU for partial shape part segmentation under "mean partial", but also list mean IoU for complete shape part segmentation under "mean complete". Ours1 represents a variation of our framework without SpecTN and Ours2 corresponds to our full pipeline with SpecTN. On average we outperform ACNN, the baseline approach, by a large margin and we outperforms ACNN on most shape categories. Moreover, our approach is more robust to data incompleteness since its performance drop is lower in comparison with complete shape segmentation.

	small	large	multi-scale
Cubic Spline	0.7509	-	-
Exp Window	0.7527	0.7551	0.7615
Modulated Exp Window	0.7896	0.7819	**0.7965**

Table 3. We compare different kernel basis and kernel size choices, using cross category part segmentation task for evaluation. IoU is reported in the table. This table is similar to Table 4 of the main paper, only different in taking geometric features of graph vertices as input instead of XYZ coordinates alone.

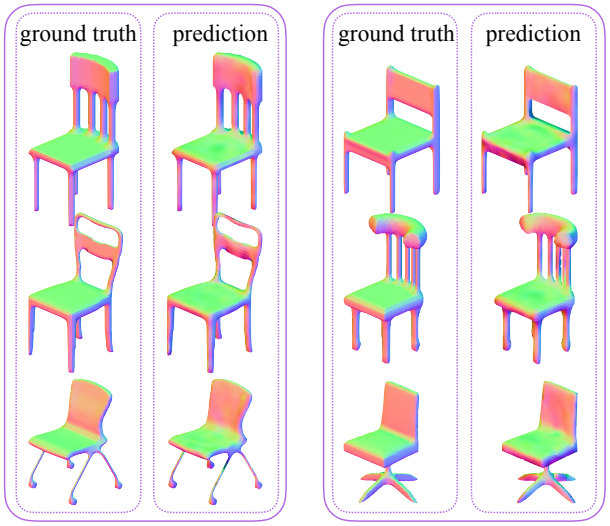

Figure 2. We evaluate our framework on normal prediction task. The colors shown on the 3D shape are RGB-coded normals, namely putting XYZ components of normal directions into RGB channels. Our framework could predict reasonable normal directions even on very thin structures.

not friendly to regression tasks with very high frequency signal as target.

References

[1] D. Boscaini, J. Masci, E. Rodolà, and M. Bronstein. Learning shape correspondence with anisotropic convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 3189–3197, 2016. 1, 2

[2] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, J. Xiao, L. Yi, and F. Yu. ShapeNet: An Information-Rich 3D Model Repository. Technical Report arXiv:1512.03012 [cs.GR], 2015. 2

[3] M. Ovsjanikov, M. Ben-Chen, J. Solomon, A. Butscher, and L. Guibas. Functional maps: a flexible representation of maps between shapes. *ACM Transactions on Graphics (TOG)*, 31(4):30, 2012. 1

[4] 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. 1, 2