Neural Space-filling Curves

Hanyu Wang, Kamal Gupta, Larry Davis, and Abhinav Shrivastava

University of Maryland, College Park
{hywang66@, kampta@, lsd@umiacs., abhinav@cs.}umd.edu

Supplementary Material

A Implementation details

Neural SFCs. The architectures of the weight generator $\mathrm{F}_{\mathcal{G}}$, and the weight evaluator $\mathrm{E}_{\mathcal{G}}$ are shown in Table 1, and Table 2 respectively. Note that the weight evaluator $\mathrm{E}_{\mathcal{G}}$ takes both an image I and a set of SFC weights $\mathbf{W}_{\mathcal{G}'}$ as inputs. The image I is passed to $\mathrm{E}_{\mathrm{enc}}$ followed by $\mathrm{E}_{\mathrm{pool}}$ which computes feature maps F_{map}. Next, together with the SFC weights $\mathbf{W}_{\mathcal{G}'}$, they are taken as inputs by $\mathrm{E}_{\mathrm{Line}}$ to regress the negative autocorrelation. The number of Residual Blocks and the number of GNN Blocks are denoted by m_1 and m_2, respectively. In all our experiments, we use $m_1 = 8$ and $m_2 = 6$. GCN [1] is used as the GNN block for MNIST and Fashion-MNIST datasets. Residual GAT [2] block is used as GNN block for FFHQ and TGIF datasets. More implementation details are available in the code attached.

Table 1: **Architecture Overview -** $\mathrm{F}_{\mathcal{G}}$

	Weight Generator $\mathrm{F}_{\mathcal{G}}$
$\mathrm{F}_{\mathrm{enc}}$	2×2 Conv2D (dual graph conv) $m_1\times$ Residual Block
$\mathrm{F}_{\mathrm{pool}}$	Parallel 1×2 Pooling and 2×1 Pooling Pooling Results Concatenation
$\mathrm{F}_{\mathrm{Line}}$	$m_2\times$ GNN Block

B Qualitative Evaluation

In Figure 3, we show more examples to compare the Neural SFCs with the Hilbert curves. In each image pair, left image shows the original image overlayed with Space-filling Curves the red. The right image shows a 1D representation of the image obtained by just flattening the pixel colors in the SFC order. Although Neural SFCs are averaged on each class label (MNIST and Fashion-MNIST) or even the entire dataset (FFHQ), it is still clear to see how Neural SFCs keep better long-range spatial coherence than Dafner SFCs.

Table 2: **Architecture Overview -** $\mathrm{E}_{\mathcal{G}}$

	Weight Generator $\mathrm{F}_{\mathcal{G}}$
$\mathrm{E}_{\mathrm{enc}}$	2×2 Conv2D (dual graph conv) $m_1\times$ Residual Block
$\mathrm{E}_{\mathrm{pool}}$	Parallel 1×2 Pooling and 2×1 Pooling Pooling Results Concatenation
$\mathrm{E}_{\mathrm{Line}}$	Addition(Linear($\mathbf{W}_{\mathcal{G}}\cdot +F_{\mathrm{map}}$)) $m_2\times$ GNN Block Global Average Pooling Linear & Sigmoid

C Quantitative Evaluation

We provide additional autocorrelation results on the class conditional MNIST datasets in Figure 4. In this case, we train multiple Neural SFC models on the subsets of MNIST corresponding to each class labels separately using the lag-6 autocorrelation objective. Then we evaluate these models on the corresponding test sets. We can observe the similar trends as described in Section 4.4 in the main paper. We observe that Neural SFCs perform the best at $k = 6$ which is also the value we used during training. Also that the class conditional image sets are about 10 times smaller than the full MNIST dataset, so it's understandable that the performance of Neural SFCs are not that good on certain subsets.

D Ablation for Lag-k Autocorrelation

In Figure 1, we show lag-k autocorrelation for Neural SFCs trained using different values of k or combinations of different values of k. When training using multiple values of k, the loss values are averaged evenly on them for both the weight generator $\mathrm{F}_{\mathcal{G}}$ and the weight evaluator $\mathrm{E}_{\mathcal{G}}$. We can see $k = 6$ is generally the best choice among all single k training settings. But if we train NerualSFCs using $k = 4, 6$ simultaneously, we can obtain even better autocorrelations from $k = 4$ to 6. However, training using $k = 4, 6, 8$ results in worse performance.

E Performance of the Weight Evaluator

Fig. 2 shows a typical loss (MSE) of the Weight Evaluator $\mathrm{E}_{\mathcal{G}}$ and the LZW code length resulting from the Weight Generator during training. In above experiment, NeuralSFC is trained on FFHQ dataset using the LZW code length objective. As the loss values get close to 0, they can provide sufficient signal to guide the weight generator $\mathrm{F}_{\mathcal{G}}$ which is apparent in Fig. 2.

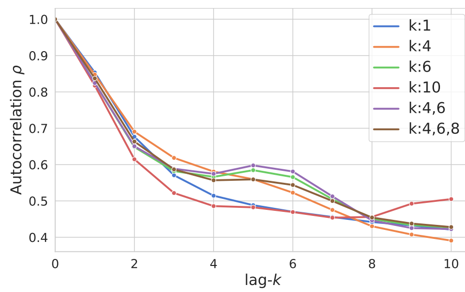


Fig. 1: **Image-set** SFCs with different training k on MNIST

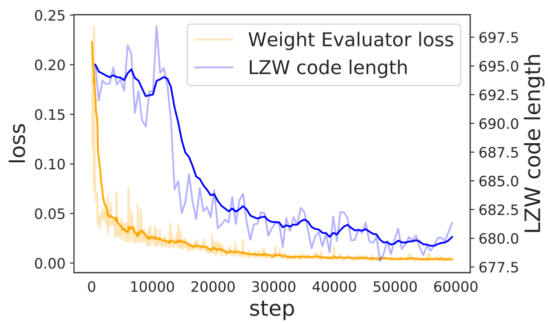
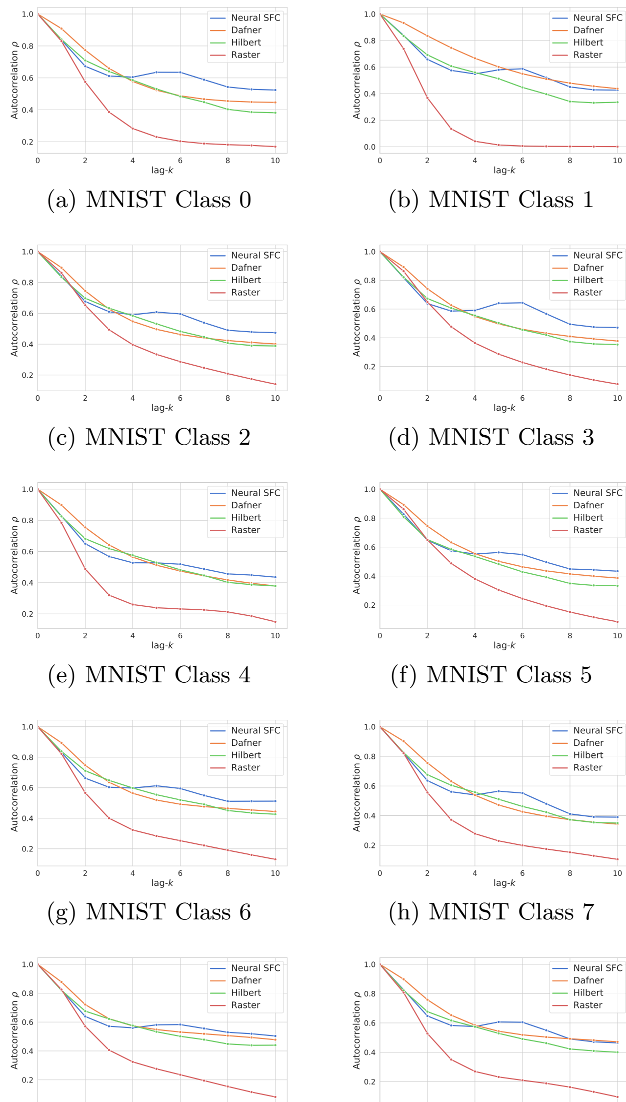


Fig. 2: Weight Evaluator Loss and the LZW code length.

Fig. 3: Additional qualitative comparison between Hilbert curves and Neural SFCs. Left: SFC (in red color) overlayed on the image. Right: Image flattened according to the SFC and visualized in 1-dimension. Images in the top four rows are from MNIST, the ones in the middle four rows are from Fashion-MNIST, and the ones in the bottom four rows are from FFHQ Faces. Neural SFCs on images from MNIST and Fashion-MNIST are class-conditional, *i.e.*, computed for each class. Best viewed in color.



(a) MNIST Class 0

(b) MNIST Class 1

(c) MNIST Class 2

(d) MNIST Class 3

(e) MNIST Class 4

(f) MNIST Class 5

(g) MNIST Class 6

(h) MNIST Class 7

(i) MNIST Class 8

(j) MNIST Class 9

Fig. 4: lag-k autocorrelations of SFCs on class conditional MNIST

References

1. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. ICLR (2016)
2. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., Bengio, Y.: Graph attention networks. In: International Conference on Learning Representations (2018), https://openreview.net/forum?id=rJXMpikCZ