ABSTRACT

Sequence ordering of word vector matters a lot to text reading, which has been proven in natural language processing (NLP). However, the role of different sequence ordering in computer vision (CV) was not well explored, e.g., why the “zigzag” flattening (ZF) is commonly utilized as a default option to get the image patches ordering in vision networks. Notably, when decomposing multi-scale images, the ZF could not maintain the invariance of feature point positions. To this end, we investigate the Hilbert fractal flattening (HF) as another method for sequence ordering in CV and contrast it against ZF. The HF has proven to be superior to other curves in maintaining spatial locality, when performing multi-scale transformations of dimensional space. And it can be easily plugged into most deep neural networks (DNNs). Extensive experiments demonstrate that it can yield consistent and significant performance boosts for a variety of architectures. Finally, we hope that our studies spark further research about the flattening strategy of image reading.

1 Introduction

Humans usually read text by row or by column, but how do you “read” a 2D image? We first look at the area of greatest interest and then the other areas or patches. And how the DNNs do with the image and text? Arguably, regardless of the text or image, many DNNs read it as text. Recently years, inspired by the Transformer extension successes in NLP, Convolution-free architectures, in particular MLP-Mixer [Tolstikhin et al., 2021], have become the model of choice in computer vision.

To apply Transformer from NLP to CV, however, the image embedding scale had to be reduced on account of computational cost of the Transformers would scale quadratically with the number of pixels. An image can simply be treated as $16 \times 16$ words in MLP-Mixer, but is it suitable to utilize a text-reading approach to read the images? In this work, we investigated the question of whether, fundamentally, the “zigzag” flattening strategy (aka line by line) is the optimal solution for image reading. Intuitively, the writing can be compared to a one-way time line, which ranked the key information. On the contrary, the vision system projects the static real world onto a two-dimensional screen, which constitutes the image.

The key information on the image can be considered as an undirected graph [Bronstein et al., 2021]. In other words, all the key information on an image can not be represented in a one-way vector. This also explains that the semantics expressed by the image is unchanged after rotation, see Fig. 1(a) and (b). In addition, the position of the key information
in the text must be fixed. If it was moved, the original message conveyed by the text would be subverted. This also explained that why the position encoding is prominent for Transformer to capture sequence ordering of input tokens. However, the key information of the image which consists of whole blocks of pixels would not change in any way even if it was moved by panning or scaling, see Fig. 1(a) and (c). Notably, the feature points inside an image block can not be moved, just like the position of a keyword in a sentence. If the features were moved, the semantics expressed by the image would change radically, see Fig. 1(a) and (d).

In the nutshell, there is a fundamental difference in the modality text and images represent information. Hence, it is naive to apply the mode of reading text directly to read images in CV. For this purpose, we investigated the Hilbert fractal flattening strategy (aka Pseudo-Hilbert curve flattening or “Hilbert” for short) as another method for image reading in CV and contrast it against ZF. The Hilbert is the only space-filling curve whose Hausdorff–Besicovitch dimension is greater than its topological dimension. And it has been shown to outperform the other curves in remaining the spatial locality, when transforming from a multi-dimensional space to a one-dimensional space [Moon et al. [2001]]. Several related works have applied it simply to the indexing of image pixels in CV. But, its theoretical explanation and application potential have not been well investigated, and even remains controversial.

In this paper, we first discussed the nature of the Hilbert fractal in image dimensional transformation and its scale robustness. Then, extensive experiments including Dynamic Time Warping (DTW) distance, interpolation based image resize, image classification, etc, demonstrated that Hilbert flattening was a better image reading method compared to “zigzag” flattening. Our contributions can be summarized as follows:

• We posed a simple but easily taken-for-granted question. In MLP-Mixer, does the model have to use the same paradigm for reading images as it does for text reading? Through theoretical analysis and fine experimental design, we have attempted to give one answer and to generalize this question to a larger context.

• We answered previous controversial questions. We have theoretically estimated the square-to-linear dilation factor of the finite approximation of Hilbert curve. This indicates that the consecutive parts in sequence are close in the corresponding image, and explains that why Tsinganos et al. [2019] can feed the 1D signals to the CNNs. Meanwhile, The Average Square Distance was proposed to give a quantitative description of comparison between inverse Hilbert flattening and inverse Zigzag flattening on probability of points close in 2-dimension are close in linear sequence. In addition, we empirically demonstrate that the Hilbert flattening can maintain feature consistency in multi-scale images.

• We proposed a new patch embedding method, named Hilbert Patch Embedding (HPE), dedicated to any DNNs, considering both effectiveness and simplicity. The HPE are simple and can be easily plugged into most DNNs. Experiments demonstrate that, without introducing additional hyperparameters, it can improve MLP-Mixer and the proposed Feature Pyramid Network (FPN-MLPs) by 1.2% (Top1 Acc) and 4.29% over their original models on CIFAR-10 [Krizhevsky et al. [2014]], respectively.

2 Related Works

In this section, we first overview the applications of Hilbert curves respectively according to their motivations. Then, we provide a review of the development of MLP-Only architectures, which serve as the backbones of our experimental section.

2.1 Applications of Hilbert Curves

Such prominent works in the field of Mathematics as Jagadish [1990], Gotsman and Lindenbaum [1996], Moon et al. [2001] have evidenced that the locality between objects in multi-dimensional space is preserved in linear space. Inspired
Hilbert Flattening: a Locality-Preserving Matrix Unfolding Method

**Figure 2**: Multi-scale transformation of dimensional space with Zigzag curve and Hilbert curve flattening/folding, respectively. Specifically, suppose that two points separated by $d_L \in (0^+, 1]$ are taken at random on the real number axis in the interval $I = [0, 1]$, which can always be taken if $2^n \times 2^n$ ($n \in \{3, 4, 5, 6, 7\}$) points are equally spaced on the number axis. By dividing the interval of the numerical axis by $2^n$ equal parts and using different unfolding operations (ZF and HF), we find that the distance $d_L$ between two points changes to $d_L^Z$ and $d_L^H$. Similarly, if $2^6$ equal divisions are performed, the distance between the two points again changes to $d_L^Z$ and $d_L^H$. The two points with fixed distance $d_L$ on $[0, 1]$ mapped to 2-D space under different types of unfolding operations will have various distances.

by such idea, recent widely works consist of [Tsinganos et al. 2019], [Bappy et al. 2019], [Zhang et al. 2021a] have been proposed to introduce the Hilbert curves into a CV application. Bappy et al. [2019] noted that the order of the image patches has a significant impact on the performance of the Long-Short Term Memory (LSTM), and if the zigzag flattening was performed in the horizontal direction, the neighboring blocks in the vertical direction are far apart. Eventually, the LSTM may not establish the connection between those patches well. To improve the performance of localization in the detection of image forgeries, they utilized the Hilbert curves to arrange image patches before the block sequences were fed into the LSTM. With the same idea, to extract the better spatial features, FDPT [Zhang et al. 2021a] also utilized the Hilbert curves to flatten image patches before feed them into the Gated Recurrent Unit (GRU). By contrast, [Tsinganos et al. 2019] employed the Hilbert curves to generate 2D image representations from 1D surface electromyography (sEMG) signals, then the features of the sEMG signals were extracted by the CNN based backbones. But, the above methods only apply the Hilbert curve to a CV task without in-depth theoretical analysis and fine empirical experimental proofs.

2.2 MLPs

[Tolstikhin et al. 2021] proposed a new architecture named MLP-Mixer that differs from CNNs and Transformers by eliminating the need for convolution and self-attention, which relies only on the repeated implementations of MLPs across the spatial or feature channels. Those works all employed Zigzag flattening to expand 2-D images or features into 1-D patch or token sequences. With the same motivation, ResMLP [Tolstikhin et al. 2021] exploited the effect of data augmentation and knowledge distillation on training a MLPs based architecture. Those works above all employed the zigzag flattening to expand 2D images or features into 1D patch or pixel sequences. But the ZF would move the initially adjacent image blocks (semantically related patches) away from each other, but HF does not, see Fig. 3 for details. Hence, in this paper, we explored Hilbert curves whose cluster property outperform zigzag curves for those MLPs based architectures.

3 Hilbert Flattening

The proposed Hilbert flattening is inspired by the Hilbert curve of space-filling curves (SFC) from the geometric theory of Fractals [Sprecher and Draghici 2002], just like Zigzag one mimics how people read text from left to right, row by row. Hilbert curves preserve the most locality of the input features among all candidates when flattening such a
multi-dimensional matrix input into a 1-D vector. In this section, we first introduce the definition of SFC and how it can be introduced into image unfolding operations. Next, we analyze the properties of HF, in the comparison with existing unfolding operations. With the property analysis of HF, we show how to apply it to popular models in practical applications.

3.1 Preliminaries

Space-Filling Curves  A continuous curve is called space-filling curve if it can pass through every point of a closed square [Peano 1890]. More precisely, a space-filling curve is a continuous mapping from a closed unit interval \( I = [0, 1] \) to a closed unit square \( Q = [0, 1]^2 \). It is defined as follows:

**Definition 3.1** A mapping \( f : I \rightarrow E^n (n \geq 2) \) is continuous and \( f(I) \) has positive Peano–Jordan measure, then \( f(I) \) is called a space-filling curve, where \( E^n \) denotes an \( n \)-dimensional Euclidean space.

Hilbert Curve Hilbert [1935] is the first generic geometric program that allows to construct entire classes of space-filling curves. Compared to zigzag curves and Gray-encoded curves, Hilbert curves were the best at minimizing the number of clusters Jagadish [1990]. More precisely, as shown in Fig. 2, a space-filling curve Simmons [1963] is a continuous mapping from a closed unit interval \( I = [0, 1] \) to a closed unit square \([0, 1] \times [0, 1]\). The definitions and known theorems used in this paper mainly come from Sagan [2012].

Hilbert Curve We assume \( I \) and \( Q \) as the interval \([0, 1]\) and square \([0, 1] \times [0, 1]\) respectively. The generating process of the Hilbert curve is driven by the following:

\[
\mathcal{H} : t \in [0, 1] \rightarrow \mathcal{H}(t) \in [0, 1] \times [0, 1],
\]

\[
t = 0, q_1, q_2 \ldots, 0 \leq q_j \leq 3,
\]

\[
\mathcal{H}(t) = \left( \frac{Re}{Im} \right) \lim_{n \to \infty} T_{q_1} T_{q_2} \ldots T_{q_n} Q,
\]

where \( t \) is represented in quaternary form. The definition of \( \{ T_i | 0 \leq i \leq 3 \} \) is defined as follows:

\[
T_i z = \frac{1}{2} H_i z + h_i, 0 \leq i \leq 3,
\]

\[
H_0 z = -i, H_1 z = z, H_2 z = z, H_3 z = -i,
\]

\[
h_0 = 0, h_1 = \frac{i}{2}, h_2 = \frac{1+i}{2}, h_3 = \frac{2+i}{2},
\]

where we consider complex numbers \( z \in \mathbb{C} \) as \((Re(z), Im(z)) \in Q\). The transformations \( \{ T_i | 0 \leq i \leq 3 \} \) defined above correspond to different geometric deformations. Take transformation \( T_3 \) as an example, we first shrink the original \( Q \) towards the original point under the ratio \( \frac{1}{2} \), then reflect on the imaginary axis by multiplying with \(-1\) and rotate the square through 90° by multiplying with imaginary number \( i \).

During the generating process of the Hilbert curve, the sub-squares shrink into points, which claims that \( \mathcal{H}(t) \) is a point in \( \mathbb{R}^2 \). We construct the \( n \)-th approximation of the Hilbert curve by \( n \)-th iteration, which is denoted as \( \mathcal{H}_n \),

\[
\mathcal{H}_n(0, q_1, q_2 \ldots, q_n) = \left( \frac{Re}{Im} \right) \sum_{j=1}^{n} \frac{1}{2^j} H_{q_0} H_{q_1} H_{q_2} \ldots H_{q_{j-1}} h_{q_j},
\]

\[
= \sum_{j=1}^{n} \frac{1}{2^j} (-1)^{e_{k_j}} \text{sgn}(q_j) \left( \frac{1 - d_j}{1 - d_j q_j} \right)
\]

\[
\text{sgn}(x) = \begin{cases} 
1, & \text{if } x > 0, \\
0, & \text{if } x = 0.
\end{cases}
\]

\[
e_{k_j} = \#("k" \text{ preceding } q_j) \mod 2,
\]

\[
d_j = e_{q_j} + e_{3j} \mod 2,
\]

where \( \# \) is the counting function and \( k \in \{0, 3\} \). We have drawn the image points of finite quaternary \((2 \leq n \leq 3)\) connected by straight lines in Figure flattening curves left. Note that the order \( n \) approximation of the Hilbert curve originates in the lower-left sub-square and terminates in the lower-right sub-square. The exit point from each sub-square coincides with the point that goes into the following sub-square.
Hilbert Flattening: a Locality-Preserving Matrix Unfolding Method

A PREPRINT

Figure 3: Zigzag flattening (red lines and arrows) VS. Hilbert flattening (blue lines and arrows) in ViT for image patch expanding. When flattening a 2D image into a 1D patch sequence, ZF will move the initially adjacent image blocks (semantically related patches) away from each other, but HF does not. That is, the head of cat remained clustered together after slicing at different flattening scales, and the position of the head on the 1D sequence was not change.

Zigzag curve We study the Zigzag curve on the image with a size of $H \times W$. For convenience, we assume that both $H$ and $W$ are equal to 1 and divided uniformly into $2^n$ parts. Given a real number $t \in [0,1]$ which can be represented in quaternary form with finite length: $t = 0.q_1 q_2 \cdots q_n$, it is defined by $Z$ as follows:

$$Z: 0.q_1 q_2 \cdots q_n \mapsto \left(\sum_{k=1}^{n} q_k 4^{n-k} \% 2^n * \frac{1}{2^n} + \frac{1}{2^n+1}\right)$$

Morton Curve The generating process of the Morton curve is similar to the Hilbert curve’s. We first denote the $n$-th approximation of the Morton curve as $M_n$. Morton curve is the limit of $M_n$ as $n$ goes to infinity. The conversion function from binary to decimal is denoted by $B$. The generating process of the Morton curve is driven as follows:

$$M_n: t \in I \mapsto M_n(t) \in Q$$

$$M_n(B(q_1 q_2 \cdots q_n) / 2^n - 1) = \left(\frac{B(q_1 q_3 \cdots q_n)}{2^n/2}, \frac{B(q_2 q_4 \cdots q_{n-1})}{2^n/2}\right) \triangleq p \in Q$$

$$M_n(t) = p_0 * (1 - s) + p_1 * s, t \in [M_n^{-1}(p_0), M_n^{-1}(p_1)]$$

where $s = \frac{t - M_n^{-1}(p_0)}{M_n^{-1}(p_1) - M_n^{-1}(p_0)} \in [0, 1], q_i \in \{0, 1\}$

Matrix Unfolding In computer vision, the matrix unfolding operation takes place when a tensor undergoes a dimensional change. For example, as illustrated in Fig. 3, the flattening of 2-dimensional grid data into 1-dimensional linear data is a matrix unfolding operation. We can observe that it is precisely the inverse of SFC operation. In this paper, three SFCs will be introduced, namely Hilbert curves, Morton curves, and Zigzag curves. Each of the three SFCs corresponds to a specific matrix flattening method: HF for Hilbert curves, MF for Morton curves, and ZF for Zigzag curves. The Hilbert curves [Hilbert, 1935] is the first generic geometric program that allows the construction of entire classes of space-filling curves. The Morton curves (aka, Z-order [Lebesgue, 2003]) map multidimensional data to one dimension while preserving the locality of the data points in data structures. And, the Zigzag curve is the default operation for flattening tensors in deep learning (i.e., torch.flatten()).

We give the expressions for the three matrix unfolding methods as follows.
We analyzed locality preserving and scale robustness about the three aforementioned unfolding methods using toy examples, respectively. Specifically, by preserving the 2-D geometric structure in a 1-D format, we believe the HF can enhance MLP-like architectures in visual discriminations, as evidenced in Section 4.3.

### Properties

We analyzed locality preserving and scale robustness about the three aforementioned unfolding methods using toy examples, respectively. Specifically, by preserving the 2-D geometric structure in a 1-D format, we believe the HF can enhance MLP-like architectures in visual discriminations, as evidenced in Section 4.3.

#### Hilbert Flattening

Hilbert flattening is built upon the Hilbert curve, with the approximation of the Hilbert curve of order \( n \), we defined its operation. Consider an image with resolution \( n \times n \), the inverse map of the approximation of Hilbert curve at order \( n \) provides the mechanism of HF:

\[
\mathcal{H}^{-1}_n : \left( \frac{i}{2^n} + \frac{1}{2^{n+1}}, \frac{j}{2^n} + \frac{1}{2^{n+1}} \right) \mapsto 0.q_1q_2 \cdots q_n, \tag{6}
\]

where \( \mathcal{H}_n(0.q_1q_2 \cdots q_n) = \left( \frac{i}{2^n} + \frac{1}{2^{n+1}}, \frac{j}{2^n} + \frac{1}{2^{n+1}} \right) \mapsto \) \( 0.q_1q_2 \cdots q_n \). All the pixels on the images will be ordered by their values, which in fact gives the definition of Hilbert Flattening.

#### Zigzag Flattening

Same as HF, given a real number \( t \in [0, 1] \) which can be represented in quaternary form with finite length: \( t = 0.q_1q_2 \cdots q_n \), the ZF is defined by \( Z^{-1} \) as follows:

\[
Z^{-1} : \left[ \frac{i}{2^n}, \frac{j}{2^n} \right] \mapsto 0.q_1q_2 \cdots q_n = Z^{-1}(\left[ \frac{i}{2^n}, \frac{j}{2^n} \right]), \tag{7}
\]

where \( 0 \leq i, j \leq 2^n - 1 \).

#### Morton Flattening

According to the definition of Morton Curve in Equation 5, we define the Morton Flattening of order \( n \) by

\[
\mathcal{M}^{-1}_n : \left( \frac{B(q_1^0q_2^0 \cdots q_{n-1}^0)}{2^{n/2}}, \frac{B(q_1^1q_2^1 \cdots q_{n-1}^1)}{2^{n/2}} \right) \mapsto B(q_1^0q_2^0 \cdots q_{n-1}^0q_{n-1}^1) \tag{8}
\]

where \( \mathcal{M}^{-1}_n \) is the inverse mapping of \( \mathcal{M}_n \).

### 3.2 Properties

We analyzed locality preserving and scale robustness about the three aforementioned unfolding methods using toy examples, respectively. Specifically, by preserving the 2-D geometric structure in a 1-D format, we believe the HF can enhance MLP-like architectures in visual discriminations, as evidenced in Section 4.3.

#### 3.2.1 Locality Preserving

Theoretically, we demonstrate that HF maximizes the preservation of 2-D topological structure among the other flattening methods, notably surpassing the orthodox Zigzag method. This holds true whether flattening 2-D matrices into 1-D vectors or folding 1-D vectors into 2-D matrices, confirming symmetry in dimension reduction or expansion.

#### Flattening 2-D to 1-D

Flattening techniques can transform high-dimensional data such as matrices or tensors into flattened representations that can be processed by neural networks, highlighting its significance in deep learning. Indeed, the transformation process inevitably disrupts the original grid structure. For instance, when applying the ZF operation, only the neighboring pixel points within each row maintain their adjacency, while the structural coherence across rows is lost. To facilitate the learning of locality representations in images, it is essential to ensure the local smoothness of the input data matrix. This concept often referred to as locality bias, is a widely adopted assumption in computer vision. And the locality bias assumes that nearby pixels or regions in an image exhibit similar characteristics or share common patterns. By retaining the spatial relationships between pixels or regions, the model can better capture local dependencies and patterns, leading to improved performance in various computer vision tasks [Sun et al., 2022, Nguyen et al., 2020].
Figure 4: Suppose we compute the DeGrid at each point on a $8 \times 8$ grid structure data with $K = 2$. The results are expressed as a heat map, in which the brightness of the pixels indicates the grid structure info preservation. The closer the pixels are to black, the better.

Table 2: Theoretical derivation of the dilation factor and limits for the three space-filling curves.

| Flattening Methods | Dilation Factor Lower Bound | Limits |
|--------------------|-------------------------------|--------|
| Zigzag curves      | $4^n - 2^{n+1} + 2$           | $+\infty$ |
| Morton curves      | $2^n - 2^{-n}$                | $+\infty$ |
| Hilbert curves     | 6 (refer to Bauman [2006a])   | 6      |

Now we explore how much the flattened 1-D sequence can preserve the original 2-D structure info. As depicted in Fig. 4, the process begins by selecting a neighborhood with a radius of $K = 2$ from the flattened 1-D data. Within this neighborhood, a central node $p$ is identified, i.e., the points circled in cyan. The degree of deformation of this neighborhood is determined by calculating the sum of the squares of the distances between each pixel (i.e., the points circled in green) and the central node $p$ over the 2-D grid data. This measure provides insight into how much the neighborhood has been altered or transformed. To describe the degree of destruction of the grid structure within this neighborhood, the ratio of the degree of deformation to the sum of the elements in the neighborhood is computed, we call it the Deformation Value of 2-D Grid Structure.

**Formal Definition of DeGrid** The steps for calculating the DeGrid have been given in Section 3.2 Grid Structure Preservation, and we give its formal expression as follows. Suppose we have a sequence of points of length $N$. For each point $p$ at position $i$ ($1 \leq i \leq N$), we collect the neighbors which are $K$ steps away from $p$. The DeGrid definition of these pixel points with respect to $p$ is

$$\text{DeGrid}(p) = \frac{\sum_{1 \leq i \leq N} \| F^{-1}(p_k) - F^{-1}(p) \|_2}{\#(K \text{ step neighbors})}. \quad (9)$$

A higher DeGrid indicates a lower degree of retention of the grid structure, while a lower ratio suggests a less significant deformation or disruption of the original grid structure.

Lastly, as presented in Table 2, we establish a threshold for the DeGrid. Nodes with a DeGrid below this threshold $\epsilon$ are deemed to have a manageable deformation. The grid structure preservation within the DeGrid threshold is determined by the ratio between the total number of these points and the total number of nodes in the flattened 2-D grid data. Indeed, as illustrated in Fig. 4, it is evident that when the $\epsilon$ exceeds 0.045, the grid structure preservation of the HF can surpass that of other SFCs entirely.

**Folding 1-D to 2-D** When performing a folding operation on a 1-D vector, it leads to an expansion of the spatial distance between the original data points, i.e., loss of local-range correlation info. As depicted in Fig. 2 when transforming 1-D sequence data into a 2-D grid using unfolding methods, the distance between two points (i.e., $d_L$) is expanded, resulting in an increased spatial distance of $d_Z$ and $d_H$, respectively. We want the process to increase $d_L$ by as small a factor (i.e., $d_Z/d_L$, where $F$ denotes a matrix flattening method) as possible to preserve the locality info.
Similar to [Estevez-Rams et al., 2013], we formulate this dilation factor (DF) to describe the preservation of the 1-D local-range correlation information.

For elaboration, let’s consider folding the interval \( I = [0, 1] \) into a \( 2^n \times 2^n \) grid data. As depicted in Fig. 2 when using the ZF method, the original distance between two points is indefinitely and continuously enlarged as the dimension of the 2-D grid increases. Instead, that distance converges to a definite value with the HF method. In Table 2, we also provide the dilation factor expressions for the above three flattening methods, along with their corresponding limit values. We find that as \( n \) tends to positive infinity, only the limit of HF is a constant (i.e., the \( C \) is equal to 6 [Bauman 2006a]) for all three expansion methods.

The properties of different matrix flattening have been discussed in above, where we derived some qualitative remarks. Here, we give formal definitions and proofs of these claims as follows.

**Dilation Factor and Limits** We introduce a definition of the limit, namely the dilation factor. Specifically, given two points \( t_1, t_2 \in [0, 1] \), the square-to-linear dilation factor of the MF curve is

\[
\begin{align*}
|H(t_1) - H(t_2)|^2 &\leq \sum_{k=j+1}^{n} \frac{8}{2^k} \leq \frac{8}{2^n},
\end{align*}
\]

Then we get Remark 3.3. And, our proposed Remark has been cited in the point cloud classification and segmentation tasks [Chen et al., 2022].

For Morton flattening, let

\[
\begin{align*}
M^{-1}(0) = \left( \begin{array}{c}
B(01\ldots1/2^{n/2}) \\
B(1\ldots1/2^{n/2})
\end{array} \right),
\end{align*}
\]

And, we have

\[
M^{-1}(0) = 2n^2/2^n - 1.
\]

Then we get Remark 3.4.

**Theorem 3.2** The square-to-linear dilation factor of the Peano-Hilbert curve is equal to 6 [Bauman 2006b], which means that the maximum value of \( \frac{|H(t_1) - H(t_2)|^2}{|t_1 - t_2|^2} \) \leq 6.

**Remark 3.3** The square-to-linear dilation factor of the ZF curve is \( \infty \). (lim \( n \to \infty \) \( 4n^2 - 2^n + 2 = \infty \)).

**Remark 3.4** The square-to-linear dilation factor of the MF curve is \( \infty \). (lim \( n \to \infty \) \( 2^n - 2^{-n} = \infty \)).

As mentioned in Section 3.2 Scale Robustness of Folding, as \( n \to \infty \), we employ the value of the ratio between the distances after multiscale folding to indicate the scale robustness. Below we give the steps for calculating the ratios for HF and ZF, respectively.

\[
\begin{align*}
\lim_{n \to \infty} \frac{d_H^2}{d_L^{2(n+1)}} &= \lim_{n \to \infty} \frac{d_H^2}{d_L^{2n}} = 6 \times \frac{1}{6} = 1, \\
\lim_{n \to \infty} \frac{d_L^2}{d_Z^{2(n+1)}} &= \lim_{n \to \infty} \frac{d_L^2}{d_Z^{2n}} = \lim_{n \to \infty} \frac{d_L^2}{d_Z^{2n+1}} = \lim_{n \to \infty} \frac{4n^2 - 2^n + 2}{4n^2 - 2^{n+2}} = \frac{1}{4}.
\end{align*}
\]
Symmetry Group Definition of Scale Robustness  According to the general equivariant [Bronstein et al., 2021, Wang et al., 2022] of the convolution operator which is defined in Definition 3.5, we give a corollary about $S$-robust of the flattening operator. Take $\Omega = \mathbb{Z}_{2^n} \times \mathbb{Z}_{2^n}$ to be a 2-D grid, and $\Omega' = \mathbb{Z}_{2^{n+1}}$ to be a 1-D sequence. Consider the $n$-th order and $(n+1)$-th order approximation of Hilbert flattening as reported in Eq. 3, geometrically, the HF operation just divides the $n$-th order approximation Hilbert curve uniformly between every pair of endpoints into three parts, then moves the second part away from the original curve with distance $\frac{1}{2n+1}$. Finally, it connects the moving part with the endpoints of the second part (please turn to Fig. 2 for details).

Given an image $I$ with size $2^{n+1} \times 2^{n+1}$, we utilize the $(n+1)$-th order HF to unfold it. We denote the pixel set after flattening as $H_{n+1}(I)$. Also, we first scale down the image $I$ into image $I_{1/2}$ with size $2^n \times 2^n$. We denote the pixel set after $n$-th order HF as $H_n(I_{1/2})$. According to the previous two paragraphs, $H_n(I_{1/2})$ and $H_{n+1}(I)$ satisfy the following condition:

$$ (H_{n+1}(I))_{1/2} \approx H_n(I_{1/2}) $$

where $1/2$ means that image scaling ratio. Consider the scale operation group $S = \{(\cdot)_{2^{-m}} | m \in \mathbb{Z}\}$, we have

$$ (F_{n+m}(I))_{2^{-m}} \approx F_n(I_{2^{-m}}) $$

where $F$ is a flattening operator, and we get the Corollary 3.6. In conclusion, as $n$ approaches a sufficiently large number (i.e., $n \to +\infty$), we only find the Hilbert flattening is $S$-robust.

Definition 3.5  A function $f : \mathcal{X}(\Omega) \to \mathcal{X}(\Omega)$ is $G$-robust if $f(\rho(g)x) \approx \rho(g)f(x)$ for all $g \in G$, i.e., group action on the input affects the output in the same way, where $\rho$ is a representation of group $G$, $\mathcal{X}(\Omega)$ denotes all signals on domain $\Omega$.

Corollary 3.6  A flattening function $F : \mathcal{X}(\Omega) \to \mathcal{X}(\Omega')$ is $S$-robust if $F(\rho(g)x) \approx \rho'(g)F(x)$ for all $g \in S$, i.e., group action on the input affects the output in the same way with input and output spaces having different domains $\Omega, \Omega'$ and representations $\rho, \rho'$ of the same group $S$.

3.3 Arbitrary Hilbert Flattening

For clarity, we focus on the grid with equal size of height and width ($2^n$). As shown in Fig. 5 following a simple recursive algorithm proposed in [Tautenhahn, 2003, Zhang et al., 2006], the general Hilbert flattening can cover the grid with arbitrary size.

4 Experiments

In this section, two analysis experiments were first reported to compare the scale robustness of ZF and HF. Second, a FPN-MLPs architecture was proposed to compare the multi-scale representations stability of both. Third, We introduce
Figure 6: Qualitative results of the 1D interpolation-based image scaling. (a) After expanding the image from 2D to 1D with ZF, we downsample the original image to $256 \times 256$ by the nearest neighbor 1D interpolation algorithm. (b) Again, we do the dimensional transform with ZF first, upsample the original image to $512 \times 512$ by the same algorithm. (c) Same as (a), but with HF. (d) Same as (b), but with HF. (e) $512 \times 512$ resolution original “Lena” image. (f) $256 \times 256$ resolution original image.

an implementation of patch embedding for the MLP-only architectures, see Fig. 3, which sets the flattening strategy between image patches.

**Experimental Setup** We utilized the common settings to compare the performance for fairness. With limited computational resources, we are not motivated by practice-based CV tasks. Notably, the settings including software (i.e., the virtual environment of Python 3.7 and Torch 1.7.1) and hardware (i.e., $4 \times $ NVIDIA 1080Ti GPUs) are strictly consistent.

4.1 Image Scaling.

Image scaling is a common operation in digital image processing (DIP). Two interpolations by row and column respectively are the standard practice for image scaling. But, if we expand the image and interpolate it only once, will this scale the image properly? As shown in Fig. 6(a) and (b), with ZF method, neither up-sampling nor down-sampling operations result in a normal image. On the contrary, with HF strategy, see Fig. 6(c) and (d), the normal results are obtained regardless of the upsampling or downsampling operation. Moreover, the scaling effect of HF can perform favorably against the result of 2D interpolation algorithms.

4.2 Scale Robustness of Folding

The scale robustness of folding refers to the extent to which the relationship between two chosen points changes in different folding scales. Generally, we suppose HF is robust to the folding scale while ZF is not. Referring to Fig. 2, assume that the interval $I$ is folded into $2^n \times 2^n$ grid data again. As $n$ increases, if the distance $d_F^{2n}$ between the two selected points converges to a nonzero constant value (i.e., $\lim_{n \to \infty} d_F^{2n} \in \mathbb{C}^+$), which makes the ratio (i.e., $\lim_{n \to \infty} d_F^{2n}/d_F^{2(n+1)}$) between the distances after multiscale folding is 1, we suppose this folding process is robust to scale. Specifically, when performing a multiscale folding operation using the ZF, the ratio between the two distances becomes
Figure 7: A toy dataset owning 18 synthetic images to measure scale robustness empirically.

Table 3: The DTW distance for different flattening methods on the proposed toy dataset (consisting of multiple scale object and image resolutions). Lower is better. As presented in Fig. 7, “L32” means Large scale object with a resolution of 32 × 32 and “S128” means Small scale one with a resolution of 128 × 128.

| Scale Methods | L32 vs S32 | L32 vs L64 | L64 vs S64 | L64 vs L128 | L128 vs S128 | L32 vs S128 |
|---------------|------------|------------|------------|-------------|--------------|-------------|
| Circle        | 5.19       | 4.13       | 6.14       | 6.79        | 9.13         | 6.74        |
| Square        | 3.40       | 5.42       | 7.00       | 7.39        | 10.58        | 7.39        |
| Triangle      | 4.61       | 3.63       | 6.93       | 6.06        | 7.99         | 6.06        |

Table 4: On CIFAR-10, recognition accuracy of different flattening methods with the proposed FPN-MLPs. “Residual-H-Backbone-Z” indicates that the flattening approaches in the residual branch and backbone are HF and ZF, respectively. Same for the other settings.

| Flattening Methods        | Top-1% | Top-5% |
|---------------------------|--------|--------|
| Residual-Z-Backbone-Z     | 81.42  | 99.18  |
| Residual-H-Backbone-Z     | 85.45  | 99.54  |
| Residual-Z-Backbone-H     | 81.67  | 99.14  |
| Residual-H-Backbone-H     | **85.71** | 99.58  |

indeterminate, i.e., \( \lim_{n \to \infty} \frac{d^n_Z}{d^{n+1}_Z} \neq 1 \), s.t., \( \lim_{n \to \infty} d^n_Z = \infty \). In contrast, when applying the same operation using the HF, that ratio tends to become convergent, i.e., \( \lim_{n \to \infty} \frac{d^n_H}{d^{n+1}_H} = 1 \), s.t., \( \lim_{n \to \infty} d^n_H = 6 \).

We show how ZF, MF, and HF respond to flattening 2-D toy examples in scale robustness. A toy dataset comprising three common shapes (circles, squares, and triangles) with two scales (small and large) is constructed, as illustrated in Fig. 7. Further, to compute the correlation between different 1-D folding vectors, dynamic time warping (DTW) distance [Berndt and Clifford, 1994; Fang et al., 2020] is employed as it allows for the comparison and alignment of sequences with variable lengths. The computed DTW distances for various sequence pairs are presented in Table 3 providing insights into the pairwise dissimilarity of the flattened images. When the resolution of image pairs is increased, the DTW distance between large and small targets also increases. This observation indicates that higher resolution amplifies the differences between multi-scale targets within an image. It is worth noting that the variation in HF is relatively small when comparing the DTW distance of different strategies. This suggests that the HF is more robust to resolution changes and preserves the structural consistency of the flattened sequences.
**4.3 Image Classification**

We apply HF to the MLP-only models, constructing advanced MLP-Mixer and MLP-FPN. As depicted in Fig. 8, the HF can be readily applied in various deep-learning operations whenever there is a need for matrix resampling or alignment, e.g., it was employed in the patch embedding module.

**MLP-FPN**  Intuitively, we believe that the scale robustness property of HF can enhance the multi-scale representation of the network containing a feature pyramid network (FPN) branch, resulting in improved performance gains. To this end, we devised a network architecture solely based on the MLP module, while incorporating an FPN branching structure, namely MLP-FPN as presented in Fig. 8. The FPN was originally introduced to address the challenge of multi-scale variation in object detection [Lin et al., 2017; Min et al., 2022]. It accomplishes this by effectively integrating features that possess strong semantic information at lower resolutions and features with abundant spatial details at higher resolutions. Inspired by this, The residual branch of our MLP-FPN is formed by stacking down-sampling pyramid modules, which include our HTS algorithm and 1-D convolutional neural networks (Conv1-D). The MLP consisting of Conv1-D is then assembled into the backbone network. The outline of the proposed network architecture FPN-MLPs is shown in Table 5.

These experiments involved the validation of the proposed FPN-MLP network, supporting the hypothesis that HTS can enhance the multi-scale representation of the networks. The results are presented in Table 4, we see that the MLP-Mixer
Table 6: Recognition accuracy of different patch embedding methods on CIFAR-10. “Inter-Z Intra-H” means that the flattening strategy in the inter-patches and intra-patches are ZF and HF, respectively. Same for the other settings. “Conv1D-H” indicates that encoding the image patches by the 1D convolution, and the flattening method of image patches is HF. “Conv1D-Z” means that the flattening method of image patches is ZF. “Overlap” indicates that the kernel size of Conv1D is bigger than the patch size.

| Methods        | Patch Size | Overlap | Top-1%  | Top-5%  |
|----------------|------------|---------|---------|---------|
| Original       | 8          | -       | 87.05   | 99.15   |
| Inter-H Intra-Z| 8          | -       | 87.35   | 99.19   |
| Inter-Z Intra-Z| 8          | -       | 86.74   | 99.16   |
| Inter-Z Intra-H| 8          | -       | 86.86   | 99.08   |
| Inter-H Intra-H| 8          | -       | 86.75   | 99.18   |
| Conv1D-Z       | 8          | -       | 83.58   | 98.71   |
| Conv1D-H       | 8          | -       | 84.52   | 98.83   |
| Conv1D-Z       | 4          | -       | 79.73   | 98.34   |
| Conv1D-H       | 4          | -       | 80.59   | 98.55   |
| Conv1D-Z       | 4          | ✓       | 80.48   | 98.32   |
| Conv1D-H       | 4          | ✓       | 81.68   | 98.57   |

with the HTS achieves clear gaps over the baseline (e.g., 1.2% on MLP-Mixer-B/4). We also find the FPN-MLP using the HTS can outperform the baseline by an obvious margin (i.e., 4.29% earnings).

MLP-Mixer Both convolution-free and attention-free model MLP-Mixer, which is an architecture based exclusively on MLPs. Similar to ViT [Dosovitskiy et al., 2020], it also treats an image as 16 × 16 words. That is, the patch embedding was an essential component of this model. As shown in Table 6, we conduct numerous experiments with different patch embedding approaches on CIFAR-10 by utilizing the MLP-Mixer. The results note that the proposed HF is effective in MLP-Mixer and also achieves significant improvement based on the original patch embedding method. In addition, a 1D convolution based patch embedding method was proposed to compare HF and ZF. The experiments demonstrated that there is a obvious gap between the best accuracy of HF and ZF. Notably, when we utilized the overlap convolution, this gap was widened.

5 Conclusion

In this paper, we explored whether Hilbert flattening is a better fit for image reading than Zigzag flattening. To answer the above question, we theoretically evaluate the square-to-linear dilation factor of the finite approximation of Hilbert curve, and propose the Average Square Distance to compare inverse HF with ZF. Based on the above theory, we proposed a new patch embedding method for MLPs. Extensive experiments including dynamic time warping distances, interpolation based image resize, and image classification demonstrate that HF is more effective than ZF. The code will be released soon.

References

Wenfeng Liu, Maoguo Gong, Zedong Tang, A Kai Qin, Kai Sheng, and Mingliang Xu. Locality preserving dense graph convolutional networks with graph context-aware node representations. Neural Networks, 143:108–120, 2021a.

Ali Hassani, Steven Walton, Jiachen Li, Shen Li, and Humphrey Shi. Neighborhood attention transformer. arXiv preprint arXiv:2204.07143, 2022.

Craig Gotsman and Michael Lindenbaum. On the metric properties of discrete space-filling curves. IEEE Trans. Image Process., 5(5):794–797, 1996.

H. V. Jagadish. Linear clustering of objects with multiple attributes. In Proceedings of the 1990 ACM SIGMOD International Conference on Management of Data, Atlantic City, NJ, USA, pages 332–342. ACM Press, 1990.

Bongki Moon, H. V. Jagadish, Christos Faloutsos, and Joel H. Saltz. Analysis of the clustering properties of the hilbert space-filling curve. IEEE Trans. Knowl. Data Eng., 13(1):124–141, 2001.
Jawadul H. Bappy, Cody Simons, Lakshmanan Nataraj, B. S. Manjunath, and Amit K. Roy-Chowdhury. Hybrid LSTM and encoder-decoder architecture for detection of image forgeries. *IEEE. Trans. Image Process.*, 28(7):3286–3300, 2019.

Xueqi Zhang, Shuo Wang, Chenyu Liu, Min Zhang, Xiaohan Liu, and Haiyong Xie. Thinking in patch: Towards generalizable forgery detection with patch transformation. In *PRICAI 2021: Trends in Artificial Intelligence - 18th Pacific Rim International Conference on Artificial Intelligence*, volume 13033, pages 337–352, 2021a.

Panagiotis Tsinganos, Bruno Cornelis, Jan Cornelis, Bart Jansen, and Athanassios Skodras. A hilbert curve based representation of seng signals for gesture recognition. In *International Conference on Systems, Signals and Image Processing, IWSSIP 2019, Osijek, Croatia, June 5-7, 2019*, pages 201–206. IEEE, 2019.

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.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, pages 10012–10022, 2021b.

Ilya O. Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Andreas Steiner, Daniel Keysers, Jakob Uszkoreit, Mario Lucic, and Alexey Dosovitskiy. Mlp-mixer: An all-mlp architecture for vision. *CoRR*, abs/2105.01601, 2021.

Ofir Press, Noah A Smith, and Mike Lewis. Train short, test long: Attention with linear biases enables input length extrapolation. *arXiv preprint arXiv:2108.12409*, 2021.

Shun Kiyono, Sosuke Kobayashi, Jun Suzuki, and Kentaro Inui. Shape: Shifted absolute position embedding for transformers. *arXiv preprint arXiv:2109.05644*, 2021.

Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *arXiv preprint arXiv:2104.09864*, 2021.

Ta-Chung Chi, Ting-Han Fang, Peter J Ramadge, and Alexander Rudnicky. Kerple: Kernelized relative positional embedding for length extrapolation. *Advances in Neural Information Processing Systems*, 35:8386–8399, 2022.

Pengchuan Zhang, Xi Yang, Jianwei Yang, Bin Xiao, Lu Yuan, Lei Zhang, and Jianfeng Gao. Multi-scale vision longformer: A new vision transformer for high-resolution image encoding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2998–3008, 2021b.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021c.

Kan Wu, Houwen Peng, Minghao Chen, Jianlong Fu, and Hongyang Chao. Rethinking and improving relative position encoding for vision transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10033–10041, 2021.

David A Sprecher and Sorin Draghici. Space-filling curves and kolmogorov superposition-based neural networks. *Neural Networks*, 15(1):57–67, 2002.

Giuseppe Peano. Sur une courbe, qui remplit toute une aire plane. *Mathematische Annalen*, 36(1):157–160, 1890.

George F Simmons. *Introduction to topology and modern analysis*, volume 44. Tokyo, 1963.

Hans Sagan. *Space-filling curves*. Springer Science & Business Media, 2012.

David Hilbert. Über die stetige abbildung einer linie auf ein flächenstück. In *Dritter Band: Analysis· Grundlagen der Mathematik· Physik Verschiedenes*, pages 1–2. Springer, 1935.

Henri Lebesgue. *Leçons sur l’intégration et la recherche des fonctions primitives*, volume 267. American Mathematical Soc., 2003.

Weixuan Sun, Zhen Qin, Hui Deng, Jianyuwan Wang, Yi Zhang, Kaishao Zhang, Nick Barnes, Stan Birchfield, Lingpeng Kong, and Yiran Zhong. Vicinity vision transformer. *arXiv preprint arXiv:2206.10552*, 2022.

Thao Nguyen, Maithra Raghu, and Simon Kornblith. Do wide and deep networks learn the same things? uncovering how neural network representations vary with width and depth. *arXiv preprint arXiv:2010.13237*, 2020.

Konstantin Evgen’evich Bauman. The dilation factor of the peano-hilbert curve. *Mathematical Notes*, 80:609–620, 2006a.

E Estevez-Rams, C Perez-Demydenko, B Aragón Fernández, and R Lora-Serrano. Visualizing long vectors of measurements by use of the hilbert curve. *Computer Physics Communications*, 197:118–127, 2015.
Donald J Berndt and James Clifford. Using dynamic time warping to find patterns in time series. In *KDD workshop*, volume 10, pages 359–370. Seattle, WA, USA., 1994.

Huihui Fang, Jianjun Zhu, Danni Ai, Yong Huang, Yurong Jiang, Hong Song, Yongtian Wang, and Jian Yang. Greedy soft matching for vascular tracking of coronary angiographic image sequences. *IEEE Transactions on Circuits and Systems for Video Technology*, 30(5):1466–1480, 2020. doi:[10.1109/TCSVT.2019.2903883](https://doi.org/10.1109/TCSVT.2019.2903883)

Haoqi Fan, Bo Xiong, Karttikeya Mangalam, Yanghao Li, Zhicheng Yan, Jitendra Malik, and Christoph Feichtenhofer. Multiscale vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6824–6835, 2021.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255. Ieee, 2009.

Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *Proceedings of the European conference on computer vision (ECCV)*, pages 418–434, 2018.

Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 633–641, 2017.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021.
Wanli Chen, Xinge Zhu, Guojin Chen, and Bei Yu. Efficient point cloud analysis using hilbert curve. In Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part II, pages 730–747. Springer, 2022.

Konstantin Bauman. The dilation factor of the peano-hilbert curve. Mathematical Notes, 80:609–620, 11 2006b.

Michael M. Bronstein, Joan Bruna, Taco Cohen, and Petar Veličković. Geometric deep learning: Grids, groups, graphs, geodesics, and gauges, 2021.

Zhenyu Wang, Xuemei Xie, Qinghang Zhao, and Guangming Shi. Filter clustering for compressing cnn model with better feature diversity. IEEE Transactions on Circuits and Systems for Video Technology, 2022.

Maithra Raghu, Thomas Unterthiner, Simon Kornblith, Chiyuan Zhang, and Alexey Dosovitskiy. Do vision transformers see like convolutional neural networks? Advances in Neural Information Processing Systems, 34:12116–12128, 2021.

Jian Zhang, Sei-ichiro Kamata, and Yoshifumi Ueshige. A pseudo-hilbert scan algorithm for arbitrarily-sized rectangle region. In International Workshop on Intelligent Computing in Pattern Analysis and Synthesis, pages 290–299. Springer, 2006.

Lutz Tautenhahn. Draw a space-filling curve of arbitrary size. http://lutanho.net/pic2html/draw_sfc.html, 2003.