Analogous to Evolutionary Algorithm: Designing a Unified Sequence Model

Anonymous Author(s)
Affiliation
Address
email

Abstract

Inspired by biological evolution, we explain the rationality of Vision Transformer by analogy with the proven practical Evolutionary Algorithm (EA) and derive that both of them have consistent mathematical representation. Analogous to the dynamic local population in EA, we improve the existing transformer structure and propose a more efficient EAT model, and design task-related heads to deal with different tasks more flexibly. Moreover, we introduce the spatial-filling curve into the current vision transformer to sequence image data into a uniform sequential format. Thus we can design a unified EAT framework to address multi-modal tasks, separating the network architecture from the data format adaptation. Our approach achieves state-of-the-art results on the ImageNet classification task compared with recent vision transformer works while having smaller parameters and greater throughput. We further conduct multi-model tasks to demonstrate the superiority of the unified EAT, *e.g.*, Text-Based Image Retrieval, and our approach improves the rank-1 by +3.7 points over the baseline on the CSS dataset.

1 Introduction

Since Vaswani *et al*. [61] introduce the Transformer that achieves outstanding success in the machine translation task, many improvements have been made to this method [15, 32, 20]. Recent works [56, 73, 36] led by ViT [21] have achieved great success in the field of many vision tasks by replacing CNN with transformer structure. In general, these works are experimentally conducted to verify the effectiveness of modules or improvements, but they may lack other forms of supporting evidence.

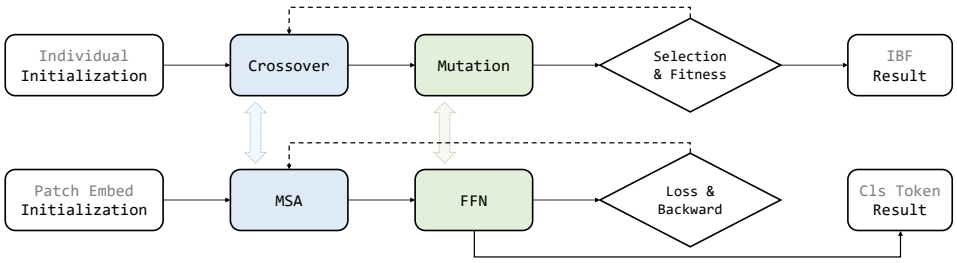

Figure 1: Analogy of EA (top) and Transformer (bottom) pipelines. For simplicity, only one layer of the Transformer structure is displayed here.

Inspired by biological population evolution, we explain the rationality of Vision Transformer by analogy with the proven effective, stable, and robust Evolutionary Algorithm (EA), which has been widely used in practical applications. Through analogical analysis, we observe that the training

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

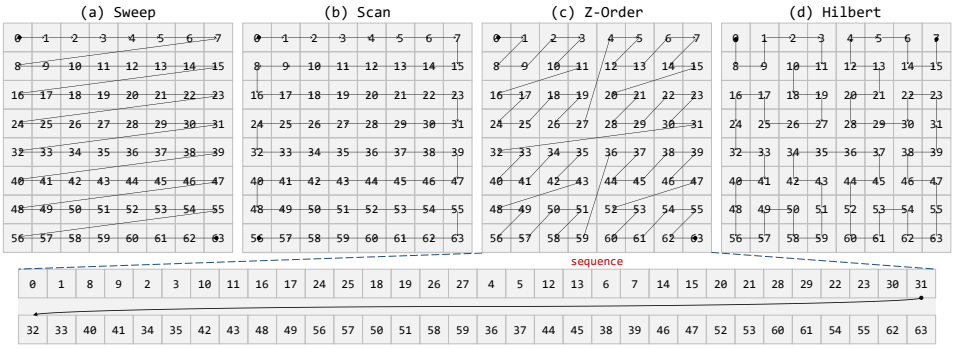

Figure 2: Different SFC indexing methods, taking 2D images with side length of 8 as an example.

procedure of the transformer has similar attributes to the naive EA, as shown in Figure 1. Take the one-tier transformer (*abbr.*, TR) as an example. ***1)*** TR processes a sequence of patch embeddings while EA evolutes individuals, both of which have the same vector formats and necessary initialization. ***2)*** The Multi-head Self-Attention (MSA) among patch embeddings in TR is compared with that of (sparse) global individual crossover among all individuals in EA, in which local and dynamic population concepts are introduced to increase running speed and optimize results [55, 52]. ***3)*** Feed-Forward Network (FFN) in TR enhances embedding features that is similar to the individual mutation in EA. ***4)*** During training, TR optimizes the network through backpropagation while EA optimizes individuals through selection and fitness calculation. ***5)*** TR chooses the enhanced Classification Token (Cls Token) as the target output, while EA chooses Individual with the Best Fitness (IBF). Meanwhile, we deduce the mathematical characterization of crossover and mutation operators in EA (*c.f.*, Equations 5,8) and find that they have the same mathematical representation as MSA and FFN in TR (*c.f.*, Equations 6,9), respectively. Inspired by the characteristics in the crossover step of EA, we propose a novel *EA-based Transformer* (EAT) that intuitively designs a local operator in parallel with global MSA operation, in which the local operator can be instantiated as 1D convolution, local MSA, *etc*. Subsequent experiments demonstrate the effectiveness of this design in that it could reduce parameters and improve the running speed and boost the network performance, which is consistent with the results of EAs in turn. Current TR-based models would initialize different tokens for different tasks, and they participate in every level of calculation that is somewhat incompatible with other tokens for internal operations. Therefore, we design task-related heads docked with transformer backbone to complete final information fusion, which is more flexible for different tasks learning and suitable for the transfer learning of downstream tasks.

Our designed local operator receives the same data format as the global MSA branch, *i.e.*, a sequence that is conform to NLP. Therefore, the generally used high dimension operations such as 2D reshape [70, 69, 71] are not required, which brings the possibility to standardize multi-modal data (*e.g.*, 1D sentence, 2D image, and 3D video) of input into consistent sequence data in *one* unified model. To accomplish this target, we introduce the space-filling curve (SFC) concept to standardize the multi-model data format and design an SFC module. As shown in Figure 2, taking 2D image as an example, the top half represents four kinds of SFCs: Sweep, Scan, Z-Order, and Hilbert, while the bottom two lines represent the sequenced image by Z-Order SFC. Thus the image ($\in \mathbb{R}^{8\times8\times3}$) is specifically re-indexed and arranged ($\in \mathbb{R}^{64\times3}$) by the predefined SFC before feeding the network, realizing uniform sequence input. The difference between SFCs mainly reflects how the 1D sequence preserves the 2D spatial structure, and the SFC can be extended to 3D and higher dimensions.

Massive experiments are conducted to demonstrate the superiority of our approach over SOTA methods on classification tasks, and we further apply EAT to multi-modal tasks, *e.g.*, Text-based Image Retrieval (TIR) and Reinforcement Learning Based Vision Language Navigation (VLN), to verify the robustness of our designed model and flexibility of our proposed unified paradigm.

2 Related Work

2.1 Evolution Algorithms

Evolution algorithm is a subset of evolutionary computation in computational intelligence, and it belongs to modern heuristics [53, 62]. Inspired by biological evolution, general EA contains reproduction, crossover, mutation, and selection steps, and has been proven to be effective and stable

in many application scenarios [26]. Moscato *et al*. [40] propose the Memetic Algorithm (MA) for the first time in 1989, which applies a local search process to refine solutions. Later Differential Evolution (DE) appears in 1995 [54] is arguably one of the most competitive improved variant [17, 42, 45, 22]. The core of DE is a differential mutation operator, which differentiates and scales two individuals in the same population and interacts with the third individual to generate a new individual. In contrast to the aforementioned global optimization, Local Search Procedures (LSPs) aim to find a solution that is as good as or better than all other solutions in its "neighborhood" [37, 14, 9, 35, 25]. LSP is more efficient than global search in that a solution can be verified as a local optimum quickly without associating the global search space. Also, some works [37, 14, 35] apply the Multi-Population Evolutionary Algorithm (MPEA) to solve the constrained function optimization problems relatively efficiently. Recently, Li *et al*. [35] point out that the ecological and evolutionary history of each population is unique, and the capacity of Xylella fastidiosa varies among subspecies and potentially among populations. Therefore, we argue that the local population as a complement to the global population can speed up the EA convergence and obtain better results. In this paper, we explain and improve the naive transformer structure by analogy with the validated evolutionary algorithms, where a parallel local path is designed inspired by the local population concept in EA.

2.2 Vision Transformers

Since Transformer structure is proposed for the machine translation task [61], many improved language models [44, 20, 46, 47, 7] follow it and achieve great results. Some later works [28, 66, 15, 32, 3] improve the basic transformer structure for better efficiency. Inspired by the high performance of transformer in NLP and benefitted from abundant computation, recent ViT [21] introduces the transformer to vision classification for the first time and sparks a new wave of excitement for many vision tasks, *e.g*., detection [8, 76], segmentation [72, 11, 58], generative adversarial network [31, 64], low-level tasks [10], video processing [4], general backbone [68, 67, 36], self-supervision [12, 2, 13, 41], neural architecture search [65, 34], *etc*. Many researchers have focused on improving the basic transformer structure [56, 73, 69, 23, 57, 70, 18, 71, 16, 60], which is more challenging than other application-oriented works. Among these methods, DeiT [56] is undoubtedly a star job that makes the transformer performance better and training more data-efficient. Based on DeiT, this work focuses on improving the basic structure of the vision transformer, making it perform better considering both accuracy and efficiency. Besides, our approach supports multi-modal tasks using only one unified model while other transformer-based methods such as DeiT do not.

2.3 Space Filling Curve

A space-filling curve maps the multi-dimensional space into the 1D space, and we mainly deal with 2D SFCs in this paper. SFC acts as a thread that passes through all the points in the space while visiting each point only once, *i.e*., every point except the starting and ending points is connected to two adjacent line segments. There are numerous kinds of SFCs, and the difference among them is in their way of mapping to the 1D space. Peano [43] first introduces a mapping from the unit interval to the unit square in 1890. Hilbert [29] generalizes the idea to a mapping of the whole space. G.M.Morton [39] proposes Z-order for file sequencing of a static two-dimensional geographical database. Subsequently, many SFCs are proposed [38, 59, 49], *e.g*., Sweep, Scan, Sierpiński, Lebesgue, Schoenberg, *etc*. Some researchers further apply SFC methods to practical applications [5, 74, 6], such as data mining and bandwidth reduction, but so far, almost none has been applied to computer vision. By introducing SFC in our EAT, this paper aims to provide a systematic and scalable paradigm that uses a unified model to deal with multi-modal data, keeping the network architecture and data structure independent of the data dimensionality.

3 Preliminary Transformer

The Transformer structure in vision tasks usually refers to the encoder and mainly builds upon the MSA layer, along with FFN, Layer Normalization (LN), and Residual Connection (RC) operations.

MSA is equivalent to the fusion of several SA operations that jointly attend to information from different representation subspaces, formulated as:

$$\begin{aligned} \mathrm{MultiHead}(Q,K,V) = & \text{ Concat } (\text{ Head }_1, \text{ Head }_2, \ldots, \text{ Head }_\mathrm{h})\, W^O, \\ \text{here Head }_\mathrm{i} = & \mathrm{Attention}\left(QW_i^Q, KW_i^K, VW_i^V\right) \\ = & \mathrm{softmax}\left[\frac{QW_i^Q\left(KW_i^K\right)^T}{\sqrt{d_k}}\right] VW_i^V = AVW_i^V, \end{aligned} \tag{1}$$

where $W_i^Q \in \mathbb{R}^{d_m \times d_k}, W_i^K \in \mathbb{R}^{d_m \times d_k}, W_i^V \in \mathbb{R}^{d_m \times d_v}, W^O \in \mathbb{R}^{hd_v \times d_m}$ are parameter matrices for feature projections; d_m is the input dimension, while d_k and d_v are hidden dimensions of each projection subspace; $A \in \mathbb{R}^{l \times l}$ is the attention matrix and l is the sequence length.

FFN consists of two cascaded linear transformations with a ReLU activation in between:

$$\mathrm{FFN}(x) = \max\left(0, xW_1 + b_1\right) W_2 + b_2, \tag{2}$$

where W_1 and W_2 are weights of two linear layers, while b_1 and b_2 are corresponding biases.

LN is applied before each layer of MSA and FFN, and the transformed $\hat{x}$ is calculated by:

$$\hat{x} = x + [\mathrm{MSA} \mid \mathrm{FFN}](\mathrm{LN}(x)). \tag{3}$$

4 EA-based Transformer

In this section, we firstly expand the association between operators in EA and modules in Transformer, and derive a unified mathematical representation for each set of corresponding pairs, expecting an evolutionary explanation for *why Transformer architecture works*. Then we introduce the SFC module to sequence 2D image that conforms to standard NLP format. Thus we may only focus on designing one unified model to solve multi-modal data. Finally, an EA-based Transformer only containing 1D operators is proposed for vision tasks, and we argue that this property is more suitable for hardware optimization in different scenarios.

4.1 EA Interpretation of Transformer Structure

Analogous to EA, the transformer structure has conceptually similar sub-modules, as shown in Figure 3. For both of methods, we define the individual (patch embedding) as $\boldsymbol{x}_i = [x_{i,1}, x_{i,2}, \ldots, x_{i,d}]$, where d indicates sequence depth. Denoting l as the sequence length, the population (patch embeddings) can be defined as $X = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_l]^{\mathrm{T}}$.

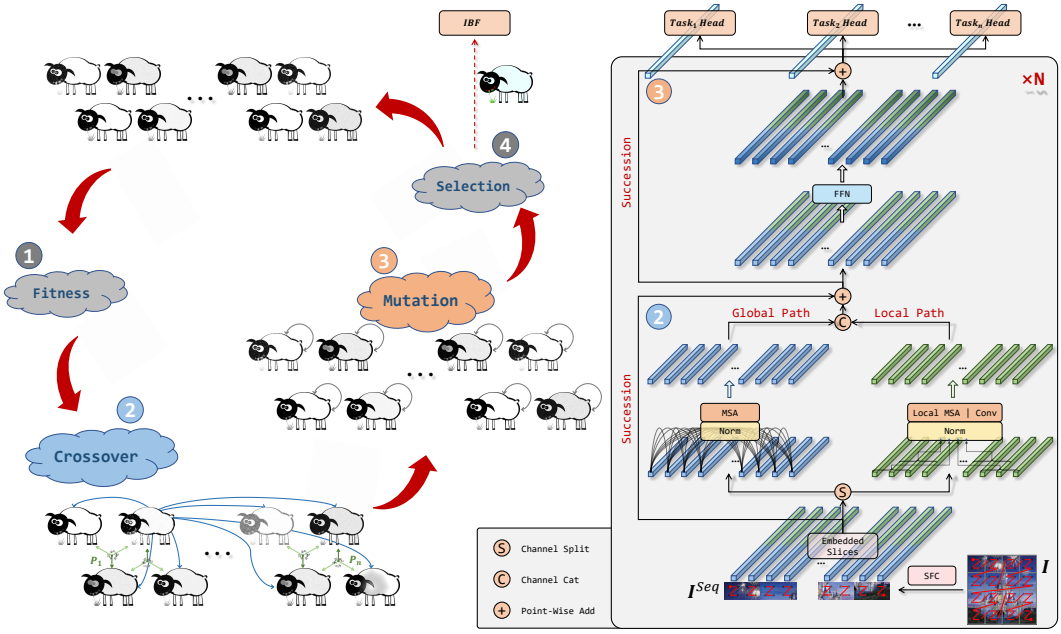

Figure 3: Structure of the proposed EAT. The right part illustrates the detailed procedure inspired by the left evolutionary algorithm. The blue and green lines in EA represent information interactions among individuals in global and local populations, which feeds back the design for global and local paths in our EAT. $P_1, \cdots, P_n$ represent biological evolution inside each local population. The proposed SFC module maps the 2D image into a 1D sequence so that only 1D operators are required.

Crossover Operator *vs.* **SA Module**. In EA, the crossover operator aims at creating new individuals by combining parts of other individuals. In detail, for an individual $\boldsymbol{x}_1 = [x_{1,1}, x_{1,2}, \ldots, x_{1,d}]$, the

operator will randomly pick another individual $\boldsymbol{x}_i = [x_{i,1}, x_{i,2}, \ldots, x_{i,d}](1 \leq i \leq l)$ in the global population, and then randomly replaces features of $\boldsymbol{x}_1$ with $\boldsymbol{x}_i$ to form the new individual $\hat{\boldsymbol{x}}_1$:

$$\hat{\boldsymbol{x}}_{1,j} = \begin{cases} \boldsymbol{x}_{i,j} & if\ randb(j) \leqslant CR \\ \boldsymbol{x}_{1,j} & otherwise \end{cases}, \quad s.t.\ j = 1, 2, \ldots, d, \tag{4}$$

where $randb(j)$ is the j-th evaluation of a uniform random number generator with outcome $\in [0, 1]$, CR is the crossover constant $\in [0, 1]$ that is determined by the user. We re-formulate this process as:

$$\begin{aligned} \hat{\boldsymbol{x}}_1 &= \boldsymbol{w}_1 \cdot \boldsymbol{x}_1 + \boldsymbol{w}_i \cdot \boldsymbol{x}_i \\ &= \boldsymbol{w}_1 \cdot \boldsymbol{x}_1 + 0 \cdot \boldsymbol{x}_2 + \cdots + \boldsymbol{w}_i \cdot \boldsymbol{x}_i + \cdots + 0 \cdot \boldsymbol{x}_l \\ &= W_1^{cr} X_1 + 0 X_2 + \cdots + W_i^{cr} X_i + \cdots + 0 X_l, \end{aligned} \tag{5}$$

where $\boldsymbol{w}_1$ and $\boldsymbol{w}_i$ are vectors filled with zeros or ones, representing the selection of different features of $\boldsymbol{x}_1$ and $\boldsymbol{x}_i$. W_1^{cr} and W_i^{cr} are corresponding diagonal matrix representations.

For the SA module in transformer (MSA degenerates to SA when there is only one head), each patch embedding interacts with all embeddings. Without loss of generality, $\boldsymbol{x}_1 = [x_{1,1}, x_{1,2}, \ldots, x_{1,d}]$ interacts with the whole population, *i.e.*, X, as follows:

$$\begin{aligned} \hat{\boldsymbol{x}}_1 &= A_1 V_1 + A_2 V_2 + \cdots + A_l V_l \\ &= A_1 W^V X_1 + A_2 W^V X_2 + \cdots + A_l W^V X_l, \end{aligned} \tag{6}$$

where $A_i (i = 1, 2, \cdots, l)$ are attention weights of all patch embedding tokens that are calculated from queries and keys, W^V is the parameter metric for the value projection. By comparing Equations 5 with 6, we find that they share the same formula representation, and the crossover operation is a sparse global interaction while SA has more complex computing and modeling capabilities.

Mutation Operator *vs.* **FFN Module**. In EA, the mutation operator injects randomness into the population by stochastically changing specific features of individuals. For an individual $\boldsymbol{x}_i = [x_{i,1}, x_{i,2}, \ldots, x_{i,d}](1 \leq i \leq l)$ in the global population, it goes through the mutation operation to form the new individual $\hat{\boldsymbol{x}}_i$, formulated as follows:

$$\hat{x}_{i,j} = \begin{cases} rand(v_j^L, v_j^H) x_{i,j} & if\ randb(j) \leqslant MU \\ x_{i,j} & otherwise \end{cases}, \quad s.t.\ j = 1, 2, \ldots, d, \tag{7}$$

where the MU is the mutation constant $\in [0, 1]$ that the user determines, v_j^L and v_j^H are lower and upper scale bounds of the j-th feature. Similarly, we re-formulate this process as:

$$\begin{aligned} \hat{\boldsymbol{x}}_i &= \boldsymbol{w}_i \cdot \boldsymbol{x}_i = [w_{i,1} x_{i,1}, w_{i,2} x_{i,2}, \cdots, w_{i,l} x_{i,l}] \\ &= W_i^{mu} X_i, \end{aligned} \tag{8}$$

where $\boldsymbol{w}_i$ is a randomly generated vector that represents weights on different characteristic depths, while W_i^{mu} is the corresponding diagonal matrix representation.

For the FFN module in transformer, each patch embedding carries on directional feature transformation through cascaded linear layers (*c.f.*, Equation 2). Take one-layer linear as an example:

$$\hat{\boldsymbol{x}}_i = W_1^{FFN} X_i, \tag{9}$$

where W_1^{FFN} is the weights of the first linear layer of the FFN module, and it is applied to each position separately and identically. By comparing Equations 8 and 9, we also find that they have the same formula representation. FFN module is more expressive because it contains cascaded linear layers and non-linear ReLU activation layers in between, as depicted in Equation 2.

Population Succession *vs.* **Residual Connection**. In the evolution of the biological population, individuals at the previous stage have a certain probability of inheriting to the next stage. This phenomenon is expressed in the transformer in the form of residual connection, *i.e.*, patch embeddings of the previous layer are directly mapped to the next layer.

Best Individual *vs.* **Task-Related Token**. EAT chooses the enhanced task-related token (*e.g.*, classification token) associated with all patch embeddings as the target output, while EA chooses the individual with the best fitness score among the population as the object.

4.2 Detailed Architecture of EAT

Local Path. Inspired by works [37, 14, 35, 55, 52] that introduce local and dynamic population concept to the evolutionary algorithm, we analogically introduce a local operator into the naive transformer structure in parallel with global MSA operation. As shown in Figure 3, the input features are divided into global features (marked blue) and local features (marked green) at the channel level with ratio p, which are then fed into global and local paths to conduct feature interaction, respectively. The outputs of the two paths recover the original data dimension by concatenation operation. Thus the improved module is very flexible and can be viewed as a plug-and-play module for the current transformer structure. This structure also implicitly introduces the design of multi-model fusion, analogous to grouping convolution with the group equaling two. Specifically, the local operator can be 1D convolution, local MSA, 1D DCN [75], *etc*. In this paper, we adopt 1D convolution as the local operator that is more efficient and parameter friendly, and the improved transformer structure owns a constant number of sequentially executed operations and $O(1)$ maximum path length between any two positions, *c.f*., Appendix A. Therefore, the proposed structure maintains the same parallelism and efficiency as the original vision transformer structure. And p is set to 0.5 for containing the minimal network parameters and FLOPs, proved by Proposition 1 and Proposition 2 in Appendix B.

Proposition 1. *The numbers of global and local branches in the $i-th$ mixed attention layer (MA^i) is d_1 and d_2. For the $i-th$ input sequence $F^i \in \mathbb{R}^{l\times d}$, the parameter of MA^i is minimum when $d_1 = d_2 = d/2$, where l and d are sequence length and dimension, respectively.*

Proposition 2. *The numbers of global and local branches in the $i-th$ mixed attention layer (MA^i) is d_1 and d_2. For the $i-th$ input sequence $F^i \in \mathbb{R}^{l\times d}$, the FLOPs of MA^i is minimum when $d_1 = d/2 + l/8, d_2 = d/2 - l/8$, where l and d are sequence length and dimension, respectively.*

Task-Related Head. Furthermore, we remove the Cls token that has been used since ViT [21], and propose a new paradigm to vision transformer that uses the task-related Head to obtain the corresponding task output through the final features. In this way, the transformer-based backbone can be separated from specific tasks. Each transformer layer does not need to interact with the Cls token, so the computation amount reduces slightly from $O((n+1)^2)$ to $O(n^2)$. Specifically, we use cross-attention to implement this head module. As shown in Figure 4, K and V are output features extracted by the transformer backbone, while Q is the task-related token that integrates information through cross-attention. M indicates that each Head contains M layers and is set to 2 in the paper. This design is flexible, with a negligible computation amount compared to the backbone.

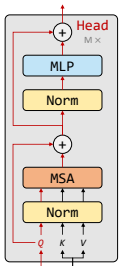

Figure 4: Structure of the proposed task-related Head.

Convergence Analysis. The convergence of Adam has been well studied in previous work [48]. It can be verified that global path and local path are both Lipschitz-smooth and gradient-bounded, as long as the transformer embedding is Lipschitz-smooth and gradient-bounded. In this case, the iteration sequence of the training algorithm converges to a stationary point of the learning objective with a convergence rate $O(1/sqrt(T))$, where T is the number of iterations.

4.3 Space-Filling Curve Module

In this subsection, we mainly deal with two-dimensional SFCs. The mathematical definition of SFC is a surjective function that continuously maps a closed unit interval:

$$\omega = \{z \mid 0 \le z \le 1\} = [0, 1], \text{ where } z \in \omega \tag{10}$$

as the domain of the function and is mapped onto the unit square:

$$\Omega = \{(x, y) \mid 0 \le x \le 1, \quad 0 \le y \le 1\}, \text{ where } (x, y) \in \Omega \tag{11}$$

which is our range of the function. One can easily define a surjective mapping from ω onto Ω, but it is not injective [59], *i.e*., not bijective. Nevertheless, for a finite 1D sequence that maps 2D pixels, it contains bijective property.

Remark 1. *According to the definition of SFC, it must be surjective. For a finite 1D to 2D mapping, there is bound to be an intersection if the SFC is not injective, where there are at least two 1D points mapping to one same 2D point, i.e., $z_1 = SFC(x, y)$ and $z_2 = SFC(x, y)$. This is in contradiction to the finite SFC filling definition. Therefore, a finite SFC is bijective, and we can carry out two-way data lossless transformation through the pre-defined SFC.*

We propose an SFC module to map 2D images to the uniform 1D sequential format based on the above content. Therefore, we can address sequence inputs to handle multi-modal tasks in *one* unified model, *e.g.*, TIR and VLN. Taking image sequence as an example. As shown in the right bottom of Figure 3, the input image $\boldsymbol{I} \in \mathbb{R}^{H \times W \times 3}$ goes through the SFC module to be $\boldsymbol{I}^{Seq} \in \mathbb{R}^{HW \times 3}$, formulated as:

$$\boldsymbol{I}_k^{Seq} = \boldsymbol{I}_{i,j}, \text{ where } k = SFC(x, y), \tag{12}$$

where $1 \leq i \leq H$, $1 \leq j \leq W$, $1 \leq k \leq HW$, and SFC can be any manually specified space-filling curve. The difference among SFCs mainly reflects how the 1D sequence preserves the 2D spatial structure. As an extension, SFC can be defined to three or higher dimensional data space and can also serialize the intermediate network features, *e.g.*, the output feature maps of stage 3 in ResNet [27]. In this paper, we use *Embedded Slices* instead of *Embedded Patches* as the input of backbone that is consistent with NLP input format. As shown in Figure 5, we further propose a new SFC named SweepInSweep (SIS) that is equivalent to the input mode of the existing transformer-based methods, *e.g.*, ViT and DeiT, where the slice size equaling 16 is equivalent to that patch size equaling 4. The paper uses this mode for SFC by default. Moreover, we present visualizations of the SFC in Appendix C.

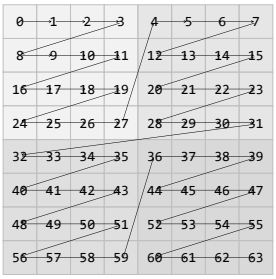

Figure 5: Indexing illustration of the proposed SweepInSweep SFC.

5 Experiments

5.1 Implementation Details.

We choose SOTA DeiT [56] as the baseline and follow the same experimental setting. By default, we train each model for 300 epoch from scratch without pre-training and distillation. The classification task is evaluated in ImageNet-1k dataset [19], and we conduct all experiments on a single node with 8 V100 GPUs. TIR is conducted on Fashion200k [24], MIT-States [30], and CSS [63] datasets, while VLN is performed on the R2R navigation dataset [1]. Detailed EAT variants can be viewed in Appendix D, and we supply the train and test codes in the supplementary material.

Table 1: Comparison with SOTA CNN-based and Transformer-based methods on ImageNet-1k dataset. Reported results are from corresponding papers.

Network	Params. ↓ (M)	Imeges/s ↑ GPU	Imeges/s ↑ CPU	Image Size	Top-1	Inference Memory
CNN-Based Nets						
ResNet-18	11.7M	4729	77.9	224^2	69.8	1728
ResNet-50	25.6M	1041	21.1	224^2	76.2	2424
ResNet-101	44.5M	620	13.2	224^2	77.4	2574
ResNet-152	60.2M	431	9.4	224^2	78.3	2694
RegNetY-4GF	20.6M	976	22.1	224^2	80.0	2828
RegNetY-8GF	39.2M	532	12.9	224^2	81.7	3134
RegNetY-16GF	83.6M	316	8.4	224^2	82.9	4240
EfficientNet-B0	5.3M	2456	49.4	224^2	77.1	2158
EfficientNet-B2	9.1M	1074	22.6	260^2	80.1	2522
EfficientNet-B4	19.3M	313	6.1	380^2	82.9	5280
EfficientNet-B5	30.4M	145	3.0	456^2	83.6	7082
EfficientNet-B7	66.3M	48	1.0	600^2	84.3	14650
NFNet-F0	71.5M	574	12.5	256^2	83.6	2967
Transformer-Based Nets						
ViT-B/16	86.6M	291	10.3	384^2	77.9	2760
ViT-L/16	304M	92	2.7	384^2	76.5	4618
DeiT-Ti	5.7M	2437	89.8	224^2	72.2	1478
DeiT-S	22.1M	927	31.4	224^2	79.8	1804
DeiT-B	86.6M	290	10.2	224^2	81.8	2760
EAT-Ti	4.8M + 0.9M	2442	95.4	224^2	72.7	1448
EAT-S	18.5M + 3.6M	1001	34.4	224^2	80.4	1708
EAT-M	41.0M + 8.0M	519	18.4	224^2	82.1	2114
EAT-B	72.4M +14.2M	329	11.7	224^2	82.0	2508

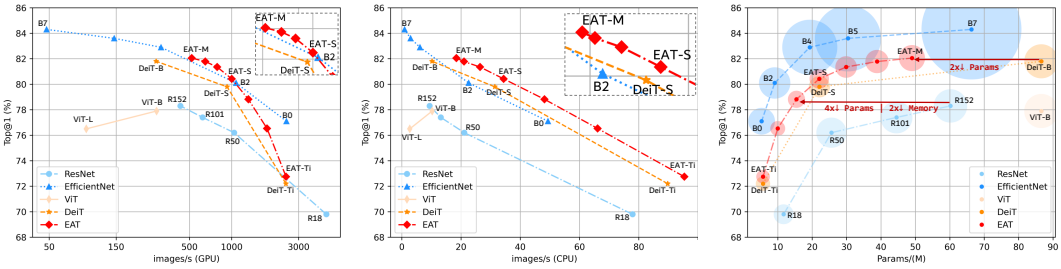

Figure 6: Comparison of different methods under different evaluation indexes. From left to right are GPU throughput, CPU throughput, and model parameters. The smaller the circle in the right sub-figure, the smaller the inference memory occupation. Please zoom in for a better visual effect.

5.2 Comparison with SOTA Methods

The classification task is conducted on the ImageNet-1k dataset without external data. As noted in Table 1, we list four kinds of EAT of different magnitudes with *backbone* + *head* format, *i.e.*, Tiny, Small, Middle, and Base. They have comparable parameters with SOTA DeiT, but *a higher accuracy, a lower inference memory occupancy, and more throughput* in both GPU (tested with batch size equaling 256 in single V100) and CPU (tested with batch size equaling 16 in i7-8700K @3.70GHz). Our method is trained in 224 resolution for 300 epochs without distillation, so our result is slightly lower than large CNN-based EfficientNet in terms of Top-1.

We visualize comprehensive Top-1 results of different methods under various evaluation indexes in Figure 6 and present the results of a series of EAT models with different model sizes. ***1)*** Left two subgraphs show that our EAT outperforms the strong baseline DeiT, and even obtains better results than EfficientNet in both GPU and CPU throughput, *e.g.*, EAT-M. ***2)*** Besides, our approach is very competitive for resource-restrained scenarios, *e.g.*, having a lower inference memory occupancy and parameters while obtaining considerable accuracy. ***3)*** Also, we find accuracy saturation in transformer structure that is also mentioned in recent works [57, 73], and we look forward to follow-up works to solve this problem. Furthermore, some techniques to help improve the accuracy of the model are applied, and our EAT model obtains a better accuracy. Detailed results can be viewed in Appendix E.

5.3 Multi-Modal Experiments

Text-based Image Retrieval. TIR is the task of searching for semantically matched images in a large image gallery according to the given search query, *i.e.*, an image and a text string. Table 2 illustrates the quantitative results. Naive transformer and EAT_L have the same number of layers, and EAT_V has similar parameters as R101_V. The results show that EAT brings positive benefits by replacing either original sub-network, even though vision and language models share one structure.

Vision Language Navigation. We further assess our approach by VLN experiments, where an agent needs to follow a language-specified path to reach a target destination with the step-by-step visual observation. We choose the PTA [33] as the base method, and Table 3 shows results under the same experimental setting. Our unified EAT obtains comparable results with PTA in the seen mode but achieves greater improvement in the unseen mode, meaning better robustness. When only replacing the vision EAT, the result changes very little. We analyze the reason that the used visual feature dimension of EAT-B is smaller than the original ResNet-152 (from 2048 to 384), and reinforcement learning is sensitive to this. Overall, our unified EAT still contributes to the PTA method.

Table 2: Retrieval performance on three datasets in rank-1. The vision and language models are replaced by other nets, marked in the lower right footnote as V and L. *: Our reproduction; -: No corresponding result. The best number is in bold.

Method	CSS	MIT-States	Fashion200k
TIRG [63]	-	12.2	14.1
TIRG*	70.1	13.1	14.0
+ TR_L	70.7	13.2	14.3
+ EAT_L	71.0	13.4	14.9
+ R101_V	73.0	14.3	19.0
+ EAT_V	73.5	14.9	19.9
+ EAT_{V+L}	**73.8**	**15.0**	**20.1**

Table 3: VLN experiment on the R2R dataset. *: Our reproduction with suggested parameters by authors. NE: Navigation Error; SPL: Success weighted by (normalized inverse) Path Length; nDTW: normalized Dynamic Time Warping.

	Method	NE ↓	SPL ↑	nDTW ↑
Seen	PTA*	3.98	0.61	0.71
	+ EAT_L	**3.84**	**0.62**	**0.72**
	+ EAT_V	3.95	0.61	0.71
	+ EAT_{V+L}	3.95	0.61	0.71
Unseen	PTA*	6.61	0.28	0.48
	+ EAT_L	**6.41**	**0.33**	**0.51**
	+ EAT_V	6.63	0.29	0.49
	+ EAT_{V+L}	6.43	0.32	**0.51**

Table 4: Ablation study for several items on ImageNet-1k. *Default* represents the baseline method based on EAT-B. The gray font indicates that the corresponding parameter is not be modified.

Ablation Items	Head Layer	Local Ratio	FFN Ratio	Top-1	Ablation Items	Kernel Size	Local Operator	SFC Mode	Image Size	Top-1
Default	1	0.50	4			3	1D Conv	SIS	224^2	80.264
Head Layer	0	0.50	4	79.692	Kernel Size	1	1D Conv	SIS	224^2	79.128
	2	0.50	4	80.422		5	1D Conv	SIS	224^2	80.256
	3	0.50	4	80.435		7	1D Conv	SIS	224^2	80.052
	4	0.50	4	80.454	Local Operator	3	Local MSA	SIS	224^2	79.940
	5	0.50	4	80.446		3	DCN	SIS	224^2	68.070
Local Ratio	1	0.25	4	80.280	SFC Mode	3	1D Conv	Sweep	256^2	71.280
	1	0.75	4	79.518		3	1D Conv	Scan	256^2	74.164
FFN Ratio	1	0.50	2	77.422		3	1D Conv	Hilbert	256^2	79.842
	1	0.50	3	79.176		3	1D Conv	Z-Order / SIS	256^2	80.438

5.4 Ablation Study

Table 4 shows ablation results for several items on ImageNet-1k. ***1) Head Layer.*** Within a certain range, the model performance increases with the increase of the head layer, and it equals 2 in the paper for trading off model effectiveness and parameters. ***2) Local Ratio.*** Large local ratio causes performance falling and increases the parameters (*c.f.*, Section 4.2), so p equaling 0.5 is a better choice. ***3) FFN Ratio.*** Similar to the conclusion of work [61], a lower FFN ratio results in a drop in accuracy. ***4) Kernel Size.*** Large kernel can lead to accuracy saturation, so we set it to 3 in the paper. ***5) Local Operator.*** We replace the local operator with local MSA and 1D DCN. The former slightly reduces the accuracy (a structure similar to the global path may result in learning redundancy), while the latter greatly reduces the accuracy (possibly hyper-parameters are not adapted). ***6) SFC Mode.*** Since some SFCs, *e.g.*, Hilbert and Z-Order, only deal with images with the power of side lengths, we set the image size to 256 here. Results show that the images serialized by Z-Order or SIS get better results because they can better preserve 2D spatial information that is important for 1D-convolution.

5.5 Model Interpretation by Attention Visualization

We average attention maps of each head layer and visualize them to illustrate which parts of the image the model is focusing on. As shown in Figure 7, Head1 pays more attention to subjects that are meaningful to the classification results, while the deeper Head2 integrates features of Head1 to form the final vector for classification that focuses on more comprehensive areas. Also, Grad-CAM [51] is applied to highlight concerning regions by our model and results consistently demonstrate the effectiveness of our proposed Head module. Note that the Cls Head contains two head layers, denoted as Head1 and Head2 in the figure. We also visualize attention maps of middle layers in Appendix G and find that the global path prefers to model long-distance over DeiT.

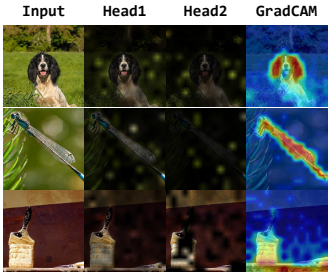

Figure 7: Visualization of attention maps for head layers and Grad-CAM results.

5.6 Influence of Positional Encoding

Benefit from spatial structure-preserving SFC and local operators, we accidentally find that EAT can alleviate the adverse effects of position-free encoding. As shown in Table 5, our EAT-Ti only decreases by 1.8 points after removing the positional encoding that is much lower than DeiT, *i.e.*, 4.0 points.

Table 5: Assessing the impact of PE on different models.

Model	Encoding	Top-1
DeiT-Ti	✗	68.2
DeiT-Ti	✓	72.2
EAT-Ti	✗	70.9
EAT-Ti	✓	72.7

6 Conclusions

This paper explains the rationality of vision transformer by analogy with EA, which brings inspiration for the neural architecture design. Besides, the introduced SFC module serializes the multi-modal data into a sequential format, so that we achieve the paradigm of using one unified model to address multi-modal tasks. Abundant experimental results demonstrate the effectiveness and flexibility of our approach. We hope this work may bring some enlightenment to network interpretability and thinking to design the unified model for multi-modal tasks. At the same time, we observe that neural evolution has been well applied in reinforcement learning [50], and perhaps it can be further mutually promoted with the back-propagation algorithm in the future.

References

[1] Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian Reid, Stephen Gould, and Anton Van Den Hengel. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3674–3683, 2018.

[2] Sara Atito, Muhammad Awais, and Josef Kittler. Sit: Self-supervised vision transformer. *arXiv preprint arXiv:2104.03602*, 2021.

[3] Irwan Bello. Lambdanetworks: Modeling long-range interactions without attention. In *International Conference on Learning Representations*, 2021.

[4] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? *arXiv preprint arXiv:2102.05095*, 2021.

[5] T. Bially. Space-filling curves: Their generation and their application to bandwidth reduction. *IEEE Transactions on Information Theory*, 15(6):658–664, 1969.

[6] Christian Böhm. Space-filling curves for high-performance data mining. *arXiv preprint arXiv:2008.01684*, 2020.

[7] Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.

[8] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European Conference on Computer Vision*, pages 213–229. Springer, 2020.

[9] Andreina I Castillo, Carlos Chacón-Díaz, Neysa Rodríguez-Murillo, Helvecio D Coletta-Filho, and Rodrigo PP Almeida. Impacts of local population history and ecology on the evolution of a globally dispersed pathogen. *BMC genomics*, 21:1–20, 2020.

[10] Hanting Chen, Yunhe Wang, Tianyu Guo, Chang Xu, Yiping Deng, Zhenhua Liu, Siwei Ma, Chunjing Xu, Chao Xu, and Wen Gao. Pre-trained image processing transformer. *arXiv preprint arXiv:2012.00364*, 2020.

[11] Jieneng Chen, Yongyi Lu, Qihang Yu, Xiangde Luo, Ehsan Adeli, Yan Wang, Le Lu, Alan L Yuille, and Yuyin Zhou. Transunet: Transformers make strong encoders for medical image segmentation. *arXiv preprint arXiv:2102.04306*, 2021.

[12] Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, and Ilya Sutskever. Generative pretraining from pixels. In *International Conference on Machine Learning*, pages 1691–1703. PMLR, 2020.

[13] Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised visual transformers. *arXiv preprint arXiv:2104.02057*, 2021.

[14] Ziyi Chen and Lishan Kang. Multi-population evolutionary algorithm for solving constrained optimization problems. In *IFIP International Conference on Artificial Intelligence Applications and Innovations*, pages 381–395. Springer, 2005.

[15] Krzysztof Marcin Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Quincy Davis, Afroz Mohiuddin, Lukasz Kaiser, David Benjamin Belanger, Lucy J Colwell, and Adrian Weller. Rethinking attention with performers. In *International Conference on Learning Representations*, 2021.

[16] Xiangxiang Chu, Zhi Tian, Bo Zhang, Xinlong Wang, Xiaolin Wei, Huaxia Xia, and Chunhua Shen. Conditional positional encodings for vision transformers. *Arxiv preprint 2102.10882*, 2021.

[17] Swagatam Das and Ponnuthurai Nagaratnam Suganthan. Differential evolution: A survey of the state-of-the-art. *IEEE transactions on evolutionary computation*, 15(1):4–31, 2010.

[18] Stéphane d'Ascoli, Hugo Touvron, Matthew Leavitt, Ari Morcos, Giulio Biroli, and Levent Sagun. Convit: Improving vision transformers with soft convolutional inductive biases. *arXiv preprint arXiv:2103.10697*, 2021.

[19] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[20] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[21] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

[22] Manolis Georgioudakis and Vagelis Plevris. A comparative study of differential evolution variants in constrained structural optimization. *Frontiers in Built Environment*, 6:102, 2020.

[23] Kai Han, An Xiao, Enhua Wu, Jianyuan Guo, Chunjing Xu, and Yunhe Wang. Transformer in transformer. *arXiv preprint arXiv:2103.00112*, 2021.

[24] Xintong Han, Zuxuan Wu, Phoenix X Huang, Xiao Zhang, Menglong Zhu, Yuan Li, Yang Zhao, and Larry S Davis. Automatic spatially-aware fashion concept discovery. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1463–1471, 2017.

[25] Phan Thi Hong Hanh, Pham Dinh Thanh, and Huynh Thi Thanh Binh. Evolutionary algorithm and multifactorial evolutionary algorithm on clustered shortest-path tree problem. *Information Sciences*, 553:280–304, 2021.

[26] William Eugene Hart, Natalio Krasnogor, and James E Smith. Memetic evolutionary algorithms. In *Recent advances in memetic algorithms*, pages 3–27. Springer, 2005.

[27] 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*, pages 770–778, 2016.

[28] Ruining He, Anirudh Ravula, Bhargav Kanagal, and Joshua Ainslie. Realformer: Transformer likes residual attention. *arXiv e-prints*, pages arXiv–2012, 2020.

[29] 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.

[30] Phillip Isola, Joseph J Lim, and Edward H Adelson. Discovering states and transformations in image collections. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1383–1391, 2015.

[31] Yifan Jiang, Shiyu Chang, and Zhangyang Wang. Transgan: Two transformers can make one strong gan. *arXiv preprint arXiv:2102.07074*, 2021.

[32] Nikita Kitaev, Lukasz Kaiser, and Anselm Levskaya. Reformer: The efficient transformer. In *International Conference on Learning Representations*, 2020.

[33] Federico Landi, Lorenzo Baraldi, Marcella Cornia, Massimiliano Corsini, and Rita Cucchiara. Perceive, transform, and act: Multi-modal attention networks for vision-and-language navigation. *arXiv preprint arXiv:1911.12377*, 2019.

[34] Changlin Li, Tao Tang, Guangrun Wang, Jiefeng Peng, Bing Wang, Xiaodan Liang, and Xiaojun Chang. Bossnas: Exploring hybrid cnn-transformers with block-wisely self-supervised neural architecture search. *arXiv preprint arXiv:2103.12424*, 2021.

[35] Xiaoyu Li, Lei Wang, Qiaoyong Jiang, and Ning Li. Differential evolution algorithm with multi-population cooperation and multi-strategy integration. *Neurocomputing*, 421:285–302, 2021.

[36] 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. *arXiv preprint arXiv:2103.14030*, 2021.

[37] David E McCauley. Genetic consequences of local population extinction and recolonization. *Trends in Ecology & Evolution*, 6(1):5–8, 1991.

[38] Mohamed F Mokbel, Walid G Aref, and Ibrahim Kamel. Analysis of multi-dimensional space-filling curves. *GeoInformatica*, 7(3):179–209, 2003.

[39] Guy M Morton. A computer oriented geodetic data base and a new technique in file sequencing. *empty*, 1966.

[40] Pablo Moscato et al. On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms. *Caltech concurrent computation program, C3P Report*, 826:1989, 1989.

[41] Kodai Nakashima, Hirokatsu Kataoka, Asato Matsumoto, Kenji Iwata, and Nakamasa Inoue. Can vision transformers learn without natural images? *arXiv preprint arXiv:2103.13023*, 2021.

[42] Nikhil Padhye, Pulkit Mittal, and Kalyanmoy Deb. Differential evolution: Performances and analyses. In *2013 IEEE Congress on Evolutionary Computation*, pages 1960–1967. IEEE, 2013.

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

[44] Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*, 2018.

[45] Adam P Piotrowski and Jarosław J Napiórkowski. The grouping differential evolution algorithm for multi-dimensional optimization problems. *Control and Cybernetics*, 39:527–550, 2010.

[46] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. *empty*, 2018.

[47] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[48] S Reddi, Manzil Zaheer, Devendra Sachan, Satyen Kale, and Sanjiv Kumar. Adaptive methods for nonconvex optimization. In *Proceeding of 32nd Conference on Neural Information Processing Systems (NIPS 2018)*, 2018.

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

[50] Tim Salimans, Jonathan Ho, Xi Chen, Szymon Sidor, and Ilya Sutskever. Evolution strategies as a scalable alternative to reinforcement learning. *arXiv preprint arXiv:1703.03864*, 2017.

[51] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017.

[52] Anabela Simões and Ernesto Costa. The influence of population and memory sizes on the evolutionary algorithm's performance for dynamic environments. In *Workshops on Applications of Evolutionary Computation*, pages 705–714. Springer, 2009.

[53] Andrew N Sloss and Steven Gustafson. 2019 evolutionary algorithms review. *arXiv preprint arXiv:1906.08870*, 2019.

[54] Rainer Storn and Kenneth Price. Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization*, 11(4):341–359, 1997.

[55] Kay Chen Tan, Tong Heng Lee, and Eik Fun Khor. Evolutionary algorithms with dynamic population size and local exploration for multiobjective optimization. *IEEE Transactions on Evolutionary Computation*, 5(6):565–588, 2001.

[56] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. *arXiv preprint arXiv:2012.12877*, 2020.

[57] Hugo Touvron, Matthieu Cord, Alexandre Sablayrolles, Gabriel Synnaeve, and Hervé Jégou. Going deeper with image transformers. *arXiv preprint arXiv:2103.17239*, 2021.

[58] Jeya Maria Jose Valanarasu, Poojan Oza, Ilker Hacihaliloglu, and Vishal M Patel. Medical transformer: Gated axial-attention for medical image segmentation. *arXiv preprint arXiv:2102.10662*, 2021.

[59] Levi Valgaerts. Space-filling curves an introduction. *Technical University Munich*, 2005.

[60] Ashish Vaswani, Prajit Ramachandran, Aravind Srinivas, Niki Parmar, Blake Hechtman, and Jonathon Shlens. Scaling local self-attention for parameter efficient visual backbones. *arXiv preprint arXiv:2103.12731*, 2021.

[61] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[62] Pradnya A Vikhar. Evolutionary algorithms: A critical review and its future prospects. In *2016 International conference on global trends in signal processing, information computing and communication (ICGTSPICC)*, pages 261–265. IEEE, 2016.

[63] Nam Vo, Lu Jiang, Chen Sun, Kevin Murphy, Li-Jia Li, Li Fei-Fei, and James Hays. Composing text and image for image retrieval-an empirical odyssey. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6439–6448, 2019.

[64] Zhaoyi Wan, Haoran Chen, Jielei Zhang, Wentao Jiang, Cong Yao, and Jiebo Luo. Facial attribute transformers for precise and robust makeup transfer. *arXiv preprint arXiv:2104.02894*, 2021.

[65] Hanrui Wang, Zhanghao Wu, Zhijian Liu, Han Cai, Ligeng Zhu, Chuang Gan, and Song Han. Hat: Hardware-aware transformers for efficient natural language processing. *arXiv preprint arXiv:2005.14187*, 2020.

[66] Sinong Wang, Belinda Li, Madian Khabsa, Han Fang, and Hao Ma. Linformer: Self-attention with linear complexity. *arXiv preprint arXiv:2006.04768*, 2020.

[67] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. *arXiv preprint arXiv:2102.12122*, 2021.

[68] Bichen Wu, Chenfeng Xu, Xiaoliang Dai, Alvin Wan, Peizhao Zhang, Zhicheng Yan, Masayoshi Tomizuka, Joseph Gonzalez, Kurt Keutzer, and Peter Vajda. Visual transformers: Token-based image representation and processing for computer vision. *arXiv preprint arXiv:2006.03677*, 2020.

[69] Haiping Wu, Bin Xiao, Noel Codella, Mengchen Liu, Xiyang Dai, Lu Yuan, and Lei Zhang. Cvt: Introducing convolutions to vision transformers. *arXiv preprint arXiv:2103.15808*, 2021.

[70] Kun Yuan, Shaopeng Guo, Ziwei Liu, Aojun Zhou, Fengwei Yu, and Wei Wu. Incorporating convolution designs into visual transformers. *arXiv preprint arXiv:2103.11816*, 2021.

[71] Li Yuan, Yunpeng Chen, Tao Wang, Weihao Yu, Yujun Shi, Francis EH Tay, Jiashi Feng, and Shuicheng Yan. Tokens-to-token vit: Training vision transformers from scratch on imagenet. *arXiv preprint arXiv:2101.11986*, 2021.

[72] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip HS Torr, et al. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. *arXiv preprint arXiv:2012.15840*, 2020.

[73] Daquan Zhou, Bingyi Kang, Xiaojie Jin, Linjie Yang, Xiaochen Lian, Qibin Hou, and Jiashi Feng. Deepvit: Towards deeper vision transformer. *arXiv preprint arXiv:2103.11886*, 2021.

[74] Liang Zhou, Chris R Johnson, and Daniel Weiskopf. Data-driven space-filling curves. *IEEE Transactions on Visualization and Computer Graphics*, 2020.

[75] Xizhou Zhu, Han Hu, Stephen Lin, and Jifeng Dai. Deformable convnets v2: More deformable, better results. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9308–9316, 2019.

[76] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*, 2020.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [No]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [Yes]
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [Yes]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [No]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]