Cluster-Learngene: Inheriting Adaptive Clusters for Vision Transformers

Qiufeng Wang[1,2], **Xu Yang**[1,2†], **Fu Feng**[1,2], **Jing Wang**[1,2], and **Xin Geng**[1,2†]

[1]School of Computer Science and Engineering, Southeast University, Nanjing 210096, China
[2]Key Laboratory of New Generation Artificial Intelligence Technology and Its Interdisciplinary Applications, Southeast University, Ministry of Education, China
`{qfwang, xuyang_palm, fufeng, wangjing91, xgeng}@seu.edu.cn`

Abstract

In recent years, the merging of vast datasets with powerful computational resources has led to the emergence of large pre-trained models in the field of deep learning. However, the common practices often overgeneralize the applicability of these models, overlooking the task-specific resource constraints. To mitigate this issue, we propose **Cluster-Learngene**, which effectively clusters critical internal modules from a large ancestry model and then inherits them to initialize descendant models of elastic scales. Specifically, based on the density characteristics of attention heads, our method adaptively clusters attention heads of each layer and position-wise feed-forward networks (FFNs) in the ancestry model as the learngene. Moreover, we introduce priority weight-sharing and learnable parameter transformations that expand the learngene to initialize descendant models of elastic scales. Through extensive experimentation, we demonstrate that Cluster-Learngene not only is more efficient compared to other initialization methods but also customizes models of elastic scales according to downstream task resources.

1 Introduction

The evolution of deep learning has been profoundly influenced by the confluence of expansive data sources and robust computational capabilities. This collaboration has given rise to large pre-trained foundation models [11, 10, 41, 5], particularly those built upon the Transformer [47, 11], such as the Vision Transformers (ViTs) [11]. The pre-trained foundation models, being widely deployed in various devices like smartphones or edge devices, serve as the initialization point [16, 2, 19, 64, 49, 50] for diverse downstream applications. However, this dominant methodology implicitly assumes that a one-size-fits-all approach, *i.e.*, the whole foundation model is universally apt for every application, neglecting the specific resource constraints (*e.g.*, memory, FLOPs, or latency) inherent to certain downstream tasks. Furthermore, not all tasks demand the full power of these extensive foundation models. This naturally raises a pivotal question: *Can we extract and harness the condensed part of these foundation models to achieve a harmonious balance between accuracy and resource efficiency?*

To achieve the goal of efficiently initializing models, [49, 50] introduce the innovative *Learngene* framework inspired by the observation of genes (cf. Fig. 1 (a)). As showcased in Fig. 1 (b), *Learngene* framework is designed in two pivotal stages. In the first stage, the significant knowledge is condensed from a large **ancestry model** into a more compact part termed as **learngene**. In the next stage, this learngene is inherited to initialize the **descendant models of elastic scales**. [*] Previous works [49, 50]

[†]Corresponding authors.
[*]The terms "foundation model" and "ancestry model," as well as "downstream model" and "descendant model," are interchangeably utilized unless distinctions are explicitly mentioned.

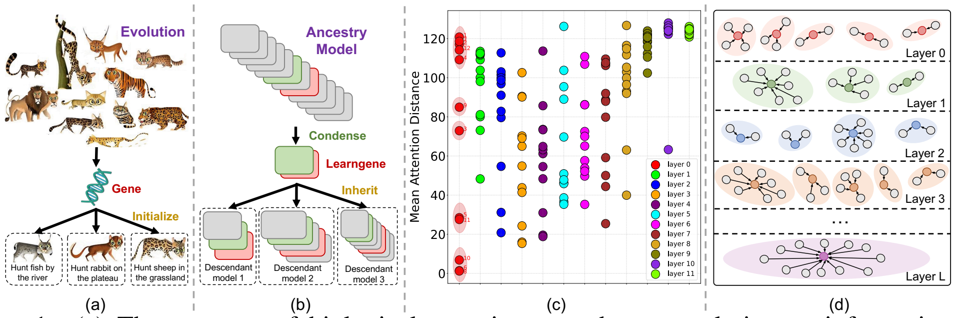

Figure 1: (a) The ancestry of biological organisms condenses evolutionary information into information-dense genes to initialize their diverse descendants [62, 17]. (b) The *Learngene* framework condenses the significant knowledge from an ancestry model into a more compact part termed learngene and then inherited to initialize the descendant models of elastic scales. (c) The density of attention heads across the different layers of the ancestry model, which employs the DeiT-B [46]. (d) An illustration of our idea.

predominantly focus on extracting a few integral layers as the learngene and manually stacking them with the randomly initialized layers.

However, such approaches struggle with inherent limitations: (i) The strategy of extracting certain integral layers overlooks the potential existence of learngene within these layers, leading to the preservation of many redundant weights. (ii) The approach of manually stacking the learngene with randomly initialized layers lacks the adaptability to scale the model, preventing the initialization of downstream models with custom dimensions.

As mentioned earlier, the *Learngene* framework aims to preserve the most generalizable part of the ancestry model while eliminating redundant weights that weaken representational capacity. Recent studies [42, 56] have visualized the mean attention distance of ViTs, offering deeper insights into weight redundancy among attention heads across different layers. As illustrated in Fig. 1 (c), the lower layers focus on both local and global perspectives, leading to a more sparse density of attention heads. Conversely, the higher layers prioritize a global context, resulting in a compact density. A notable observation is the repetitive functionality across many attention heads especially in the higher layers, which inevitably leads to weight redundancy.

Inspired by the above observation, we propose the Cluster-Learngene, an innovative approach that adaptively extracts internal modules in ViTs as the learngene, *e.g.*, attention heads and position-wise feed-forward networks (FFNs). Firstly, to extract the cluster centroids of the attention heads (*i.e.*, **head centroids**) across each layer of the ancestry model, we cluster the attention heads within each layer of the ancestry model based on their density characteristics. As depicted in Fig. 1 (c-d), the attention heads in the first layer exhibit a sparse density, resulting in five clusters, whereas the attention heads in the last layer cluster more compactly, forming a single group. Furthermore, our method includes clustering FFNs by assessing the distance density of head centroids in adjacent layers of the ancestry model. Specifically, when the distance density of head centroids in adjacent layers is similar, we inherit the FFN from the shallower of these adjacent layers (*i.e.*, **FFN centroid**) as the learngene. As illustrated in Fig. 1 (c), the similar densities of attention heads in the 7-th and 8-th layers lead to proximate head centroids, enabling these layers to only require the inheritance from the 7th layer as the FFN centroid. Overall, Cluster-Learngene extracts critical parameters containing significant knowledge, as the extracted part represents attention heads/FFNs with similar semantics.

In the inheriting stage, to achieve the initialization of descendant models with **varying number of attention heads**, we adopt the priority weight-sharing. We start by ranking the head centroids based on the size of their respective clusters, arranging them in descending order of priority. Subsequently, we perform weight-sharing by distributing these head centroids to initialize the attention heads of the descendant models. If the number of attention heads in a specific layer aligns perfectly with the number of centroids, they are evenly shared. However, if they fail to align perfectly, any remaining centroids are shared according to the remainder. Moreover, we apply learnable parameters to transform FFN centroids into multiple FFNs, thus enabling the initialization of descendant models with elastic scales.

Our **contributions** can be summarized as follows: (i) We propose the adaptive clustering to extract the head and FFN centroids as the learngene, ensuring the preservation of significant knowledge within the ancestry model. (ii) To achieve the initialization of descendant models, we introduce priority weight-sharing that favors head centroids within larger clusters and employ learnable parameters to transform the FFN centroids into multiple FFNs. (iii) Comprehensive experimental evaluations across datasets of different scales reveal that Cluster-Learngene not only outperforms traditional initialization strategies but also stands toe-to-toe with more resource-demanding fine-tuning methodologies.

2 Related Work

Model Initialization: Over the years, various initialization techniques have been proposed including the popular random initialization, Xavier initialization [13] and the Kaiming initialization [19]. Recently, the use of pre-trained foundation models has gained prominence as an initialization strategy before fine-tuning for specific tasks [11, 10, 41, 57, 37, 5, 61, 22, 15, 53, 33, 32, 7, 18, 39, 54, 29, 25, 26, 51, 28, 44, 55, 59, 40, 58, 36, 31, 21, 60]. However, such an approach necessitates pre-training separate models for each downstream task, which can lead to substantial computational resource consumption. In contrast, Cluster-Learngene presents a unique model initialization method that alleviates the need for multiple pre-training steps.

Density-based Clustering: Clustering aims to group similar data points together while separating dissimilar ones. A wide array of approaches has been explored, including partitioning-based clustering [14, 1, 27], hierarchical clustering [35, 8, 65, 52], and density-based clustering [23, 43, 6, 3], and so on. In particular, density-based clustering operates by taking into account the density relationships between data points to form clusters. Inspired by this, our method adopts a similar principle by assessing the density of attention heads to retain essential head centroids that represent significant knowledge.

3 Methodology

Learngene framework is primarily divided into two phases in Fig. 1 (b): the significant knowledge is condensed from an ancestry model into a more compact part termed as learngene and then inherited to initialize the descendant models of assorted scales. Specifically, in phase 1, our Cluster-Learngene selects mean attention distance as the density metric and uses it to cluster the attention heads of each layer and FFNs in the ancestry model as the **learngene**, because they can effectively represent attention heads/FFNs with similar semantics. The pseudocode for this phase is presented in Algorithm 1. In phase 2, to initialize the descendant models, we employ priority weight-sharing of head centroids for varying number of attention heads as illustrated in Fig.2 and leverage learnable parameters to expand the FFN centroids into multiple FFNs. Next, we briefly introduce some preliminaries related to ViTs.

3.1 Preliminary

In the ViT architecture, an input image is first divided into N non-overlapping patches, and each patch is linearly embedded into a flat vector of size D. The ViT encoder consists of alternating layers of multi-head self-attention (MSA) and FFN blocks. Let H denote the total number of heads in each layer. For the h^{th} head, the query $\mathbf{Q}_h \in \mathbb{R}^{N\times d_k}$, key $\mathbf{K}_h \in \mathbb{R}^{N\times d_k}$, and value $\mathbf{V}_h \in \mathbb{R}^{N\times d_v}$ are linearly generated through learned weight matrices $\mathbf{W}_h^Q \in \mathbb{R}^{D\times d_k}$, $\mathbf{W}_h^K \in \mathbb{R}^{D\times d_k}$, and $\mathbf{W}_h^V \in \mathbb{R}^{D\times d_v}$, where d_k and d_v are the dimensions of the key and value vectors, respectively. The SA mechanism of the i-th head can be represented as:

$$\mathbf{A}^h = \text{Attention}(\mathbf{Q}_h, \mathbf{K}_h, \mathbf{V}_h) = \text{softmax}\left(\frac{\mathbf{Q}_h\mathbf{K}_h^\top}{\sqrt{d_k}}\right)\mathbf{V}_h. \tag{1}$$

MSA allows the model to jointly attend to information at different positions from different representational subspaces at different positions:

$$\text{MultiHead}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{Concat}(\mathbf{A}^1, \ldots, \mathbf{A}^H)\mathbf{W}^O, \tag{2}$$

where $\mathbf{W}^O \in \mathbb{R}^{Hd_v\times D}$ is a learned weight matrix. Besides, the FFN can be formulated as:

$$\text{FFN}(\mathbf{x}) = \text{ReLU}(\mathbf{x}\mathbf{W}_1 + \mathbf{b}_1)\mathbf{W}_2 + \mathbf{b}_2, \tag{3}$$

Algorithm 1: Adaptively Cluster for MSA

```
Input: Number of layers as $L$, set of attention heads in the $l^{th}$ layer as $S_l$, radius as $Eps$, density threshold as $MinHds$, and distance function as $Dist$.
Output: The centroids of attention head in all clusters.
Initialize all attention heads as unvisited and an empty list for clusters
for $l = 1, \dots, L$ do
    foreach attention head $a$ in $S_l$ do
        if $a$ is not visited then
            Mark $a$ as visited, $NeighborHds \leftarrow$ all attention heads within $Eps$ distance of $a$
        end
        if number of $NeighborHds \geq MinHds$ then
            $C \leftarrow$ new cluster, Add $a$ to cluster $C$                // Start a new cluster
            foreach attention head $b$ in $NeighborHds$ do
                // Expand neighborhood
                if $b$ is not visited then
                    Mark $b$ as visited
                    $NeighborHds' \leftarrow$ all attention heads within $Eps$ distance of $b$
                end
                if number of $NeighborHds' \geq MinHds$ then
                    $NeighborHds = NeighborHds \cup NeighborHds'$
                end
                if $b$ is not yet a member of any cluster then Add $b$ to cluster $C$
            end
            Add $C$ to the list of clusters                // Consolidate clusters
        end
    end
end
```

where $\mathbf{x} \in \mathbb{R}^{N\times D}$ is the input, $\mathbf{W}_1 \in \mathbb{R}^{D\times d_{ff}}$ and $\mathbf{W}_2 \in \mathbb{R}^{d_{ff}\times D}$ are the weight matrices, and $\mathbf{b}_1 \in \mathbb{R}^{d_{ff}}$ and $\mathbf{b}_2 \in \mathbb{R}^{D}$ are the bias vectors. d_{ff} is the dimension of the intermediate layer.

3.2 Adaptively Learngene Clustering

3.2.1 Density metric on attention heads

Given a pre-trained ancestry model with L layers and H_a attention heads per layer, let the attention weights for the h^{th} head in the l^{th} layer be denoted by the matrix $\mathbf{A}^{(l,h)} \in \mathbb{R}^{N\times N}$. The element $A_{i,j}^{(l,h)}$ represents the attention weight from position i to position j. The distance matrix $\mathrm{T} \in \mathbb{R}^{N\times N}$ is defined with the Euclidean distance between any two positions i and j in the sequence, given by $T_{i,j} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$. The mean attention distance for the h^{th} head in the l^{th} layer, encapsulating the weighted distance for each position i across the sequence, is given by:

$$\tilde{d}^{(l,h)} = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{N} A_{i,j}^{(l,h)} \times T_{i,j}. \tag{4}$$

To deduce this metric for every head across all layers, iterate the above computation for every $l \in \{1, \dots, L\}$ and $h \in \{1, \dots, H_a\}$. As depicted in Fig. 1 and Appendix A.1, while the lower layers simultaneously attend to both local and global features, leading to a more dispersed distribution of attention heads, the higher layers predominantly focus on global aspects, causing a tighter concentration of attention heads. As a result, there is a significant overlap in the semantic representations among many attention heads, especially in the higher layers, leading to weight redundancy.

3.2.2 Cluster for MSA

Motivated by the empirical observations, we extract cluster centroids [43, 6, 3] of attention heads in ViTs as the learngene inherited into the descendant models, thus aggregating similar semantics into the head centroids. To realize this, we select $\tilde{d}$ as a density metric for adaptively clustering the attention heads of the ancestry model at each layer, without setting the number of clusters in advance. This realization prompts the formulation of the definitions and lemmas, which scaffold our adaptive clustering approach.

Definition 1 *(Eps-neighborhood of an attention head). The Eps-neighborhood of an attention head a, denoted as $N_{Eps}(a)$, is defined as: $N_{Eps}(a) = \{b \in S \mid Dist(a,b) \leq Eps\}$, where $Dist(a,b) = \left|\tilde{d}^{(a)} - \tilde{d}^{(b)}\right|$ denotes the difference in $\tilde{d}$ values between attention heads a and b.*

Our approach could require for each head in a cluster that there are at least a **Min**imum number of **Heads** ($MinHds$) in an Eps-neighborhood of that head.

Definition 2 *(density-reachable). Transitioning from the neighborhood concept, an attention head a is considered density-reachable from another head b with respect to Eps and $MinHds$ if there is a sequence of heads $a_1, \ldots, a_n$ such that $a_1 = b, a_n = a$, and each head in this sequence lies within the Eps-neighborhood of its preceding head.*

Definition 3 *(density-connected). Broadening our purview, attention heads a and b are labeled density-connected with respect to Eps and $MinHds$ if there exists an intermediary head o from which both a and b are density-reachable.*

Considering all attention heads in layer l as S_l, a cluster C based on Eps and $MinHds$ is identified as a non-empty subset of S_l that satisfies the conditions: (i) **Maximality:** For any heads a and b in the sequence, if a resides within C and b is *density-reachable* from a dictated by Eps and $MinHds$, then b seamlessly becomes part of C. (ii) **Connectivity**: Within C, each pairing a, b maintains a *density-connection*, anchored by Eps and $MinHds$.

Therefore, upon satisfying these two conditions, we cluster all attention heads of each layer into different lists of clusters. The pseudo-code for this process is summarized in Algorithm 1. For each cluster C in the list of clusters, we select the attention head C_{lg} closest to the cluster centroid as the learngene. This is because it effectively represents the functionality of all attention heads in the cluster. Formally, the formula can be expressed as follows:

$$C_{\text{lg}} = \arg\min_{c \in C} \left|\tilde{d}^{(c)} - \bar{d}^{(C)}\right| \tag{5}$$

where $\bar{d}^{(C)}$ is computed as the average $\tilde{d}$ across all attention heads within this cluster C. The lemma presented below is pivotal in substantiating the correctness of our clustering algorithm.

Lemma 1 *Presuming an attention head a belongs to S_l and satisfies the condition $|N_{Eps}(a)| \geq MinHds$. Then, the set $O = \{o \mid o \in S_l$ and o is density-reachable from a with respect to Eps and $MinHds\}$ collectively shapes a cluster.*

3.2.3 Cluster for FFN

Based on the distances between attention heads discussed earlier, we can further cluster FFNs as the learngene. For adjacent layers sharing the same number N_C of head centroids, a key criterion is established: if the average distance across all head centroids between these layers is less than the threshold ε, it suggests redundant similarity in their representational capabilities. This criterion is formulated as:

$$\frac{1}{N_C} \sum_{k=1}^{N_C} Dist\left(C_{l,k}, C_{l+1,k}\right) < \varepsilon. \tag{6}$$

Under this condition, we enhance model efficiency by preserving only the FFN from the shallower of these adjacent layers as the FFN centroid, thereby maintaining essential functionality while eliminating redundancy. Moreover, we also select those FFNs that do not fit this established criterion as the learngene because they exhibit representational capacities independent of adjacent layers.

3.3 Learngene Inheriting

3.3.1 Expanding head centroids with priority weight-sharing

Since the head centroids have been extracted as the learngene, we further expand them for initializing the descendant models with varying number of attention heads. For an ancestry model with L

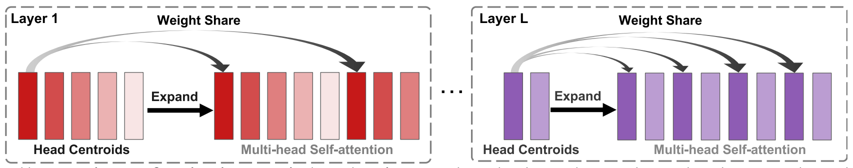

Figure 2: Illustration of priority weight-sharing. The darker the color, the larger the cluster size associated with the head centroid.

layers, the l^{th} layer has c_l head centroids of weight $\mathbf{A}^{(l,1)}, \ldots, \mathbf{A}^{(l,c_l)}$. Importantly, the head centroids at each layer are sorted in descending order based on the size of their respective cluster, *i.e.*, centroids representing more attention heads in the ancestry model are ranked higher. These head centroids condense significant knowledge and ensure the initialization of descendant models without performance degradation. Assume the descendant model has H_d attention heads for each layer. To achieve the desired expansion of heads to initialize the descendant models, we adopt the priority weight-sharing and Fig. 2 illustrates two scenarios:

- When H_d is divisible by c_l: The weights of head centroids are shared $\frac{H_d}{c_l}$ times in sequence. For instance, centroids of weights $\mathbf{A}^{(L,1)}$ and $\mathbf{A}^{(L,2)}$ each share their weights across four attention heads, which are then directly assigned to eight attention heads of the descendant model in layer L.
- When H_d is not divisible by c_l: The weights of the head centroids are sequentially shared $\left\lfloor \frac{H_d}{c_l} \right\rfloor$ times, followed by appending $\mathbf{A}^{(l,1)}, \ldots, \mathbf{A}^{(l,H_d \mod c_l)}$ at the end. As an illustration, we share the centroids of weights $\mathbf{A}^{(1,1)}, \ldots, \mathbf{A}^{(1,5)}$ once and then append $\mathbf{A}^{(1,1)}, \ldots, \mathbf{A}^{(1,3)}$, thus initializing eight attention heads of the descendant model in the first layer.

For the attention heads in the descendant models, we introduce the hyperparameter $\omega = \frac{H_a}{H_d}$ to denote the factor by which the number of attention heads is reduced compared to the ancestry model. In addition to uniformly setting the number of attention heads for each layer with the hyperparameter ω, we also explore two other possibilities in Appendix A.7: incrementing and decrementing the count of attention heads with layer depth. According to the adjustments in the number of attention heads, the weights $\mathbf{W}^O$ of the projection layer are also proportionally pruned and then inherited by the descendant models. [†]

3.3.2 Expanding FFN

As outlined in Section 3.2, we retain the shallowest FFN determined by distance-based clustering. For this FFN centroid, we introduce learnable parameters to transform it into multiple FFNs, facilitating the initialization of descendant models at elastic scales. This process is summarized in the formula:

$$\text{FFN}_t(\mathbf{x}) = (\text{ReLU}((\mathbf{x}(\mathbf{W}_1 + \mathbf{b}_1)\widehat{\mathbf{W}}_t)\mathbf{W}_2 + \mathbf{b}_2)\widehat{\mathbf{W}}_t, \tag{7}$$

Here, $\widehat{\mathbf{W}}_t$ denotes the newly introduced learnable parameters to expand the t^{th} FFN. These parameters are designed for minimal learning overhead, yet they are highly effective in quickly adapting descendant models to downstream tasks.

4 Experiments

4.1 Experimental Setting

Datasets. To condense the learngene, we employ the ImageNet-1K, a collection of 1.2 million training images and 50,000 validation images distributed across 1,000 classes as part of the ILSVRC2012 competition [9]. After initializing the descendant models with the learngene, we proceed to fine-tune these models on diverse downstream tasks. These tasks include iNaturalist-2019 [45], Food101 [4], Oxford Flowers [38], Stanford Cars [12], CIFAR-10 [24], CIFAR-100 [24], CUB-200-2011 [48]. For detailed dataset descriptions, see Appendix A.2.

[†]Please see Appendix A.3 for more details.

Training settings. During the learngene clustering, We set $Eps, \varepsilon = 10, MinHds = 1$ [‡], ensuring that each attention head is included in a unique cluster. In the learngene inheriting phase, we train the descendant models on downstream tasks for 500 epochs, including a 10-epoch warm-up period, except for iNaturalist-2019, where we train for 100 epochs with a 5-epoch warm-up. The initial learning rate is set to 5×10^{-4} for most tasks, except for Stanford Cars where it is 5×10^{-3}, and a weight decay of 0.05.

Architectures. Both the ancestry model and descendant models are variants derived from DeiT [46]. In terms of width, there are three types of DeiT: **Tiny**, **Small**, and **Base**. Furthermore, as detailed in Section 4.2.4, we implement experiments on Swin Transformer [30] to demonstrate the applicability of our method across various backbones. The learnable parameters in Eqns. (7) are implemented through a nonlinear mapping such as a neural network with the rectified linear units (ReLU).

4.2 Main Results of Model Initialization

In this section, we validate the capabilities of Cluster-Learngene in efficiently initializing models and measure model performance with Top-1 accuracy.

4.2.1 Initializing Descendant Models of Elastic Scales

As illustrated in Fig. 3, for the Tiny-scale descendant models with only 32 attention heads, Cluster-Learngene outperforms Pretraining-Finetuning on ImageNet. Moreover, the performance of Cluster-Learngene improves with an increase in the number of attention heads and FFNs. This shows that Cluster-Learngene can adaptively initialize the descendant models with varying number of attention heads and FFNs, catering to downstream resource constraints. In contrast, Pretraining-Finetuning requires retraining for each model variant, significantly raising storage and training costs, particularly when dealing with models of elastic scales. Therefore, our method resolves the limitations of the one-size-fits-all approach seen in Pretraining-Finetuning.

Table 1: Comparisons of performance on ImageNet-1K between models trained From-Scratch with 100 epochs and those initialized via Cluster-Learngene fine-tuned for 50 epochs.

Model	H_d	L_d	Params (M)	FLOPs (G)	From-Scratch	Ours
Tiny	2	6	1.3	0.3	50.06	**52.73**
		9	1.9	0.4	54.64	**59.60**
		12	2.5	0.5	57.99	**61.84**
	3	6	3	0.6	58.16	**59.58**
		9	4.4	0.9	60.58	**65.47**
		12	5.7	1.2	61.44	**70.28**
Small	4	6	5	1	61.32	**64.98**
		9	7.3	1.4	63.17	**70.90**
		12	9.5	1.9	64.25	**73.20**
	6	6	11.4	2.3	64.91	**67.72**
		9	16.8	3.4	67.02	**73.06**
		12	22	4.6	68.56	**78.43**
Base	12	3	22.8	4.5	68.77	67.82
		4	29.9	5.9	70.32	**70.41**
		5	36.9	7.4	72.04	**72.13**
		6	44	8.8	73.73	**73.95**
		7	51.2	10.2	74.42	**74.87**
		8	58.2	11.7	76.14	**76.90**
		9	65.3	13.1	76.46	**77.11**
		10	72.4	14.6	76.81	**77.99**
		11	79.5	16	77.03	**78.82**
		12	86.6	17.5	77.22	**79.50**

Moreover, we initialize 22 descendant models on ImageNet-1K, each with different configurations, such as the number of attention heads H_d per layer and the quantity of FFNs L_d in descendant models. As shown in Tab. 1, Cluster-Learngene swiftly initializes models of varying scales and proves competitive in overall performance. For example, the Small-scale descendant model with $H_d = 6$ and $L_d = 12$ illustrates this point. Cluster-Learngene not only outperforms From-Scratch by **9.87%** but also manages to reduce 2× training times.

4.2.2 Efficiently Initializing Large Models on ImageNet

The results in Fig. 4 reveal that Cluster-Learngene outperforms Pretraining-Finetuning with fewer inherited parameters. For example, in the case of Small-scale descendant models, Cluster-Learngene inherits only 15.1M parameters and attains a performance of 78.43%, while Pretraining-Finetuning with 18.0M parameters achieves less than 75%. These results demonstrate the advanced initialization ability of Cluster-Learngene, as the clustered learngene retains the critical generalizable knowledge.

4.2.3 Initialization Results on Different Downstream Tasks

We conduct a comparative analysis of our approach for initializing descendant or downstream models, as follows: (i) Pretraining-Finetuning: This approach pre-trains DeiT on ImageNet and

[‡]Please see Appendix A.6 for more analysis.

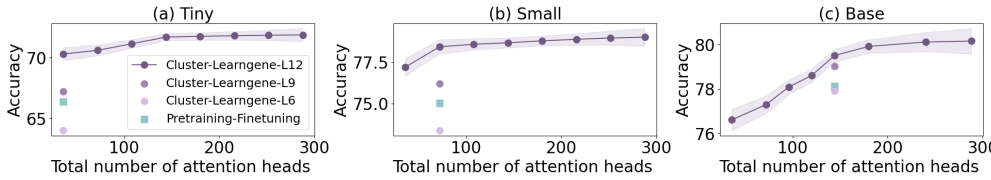

Figure 3: **Initializing descendant models of elastic scales.** "L6/9/12" denote descendant models with 6, 9, and 12 layers, respectively. For a fair comparison, the downstream models in Pretraining-Finetuning inherit parameters from 12 layers of the pre-trained model, with the inherited number of attention heads matching those in Cluster-Learngene. We fine-tune **50 epochs** for all models. In (a), the hyperparameter ω takes values ranging from 1 to $\frac{1}{8}$ (i.e., the number of attention heads in descendant models is eight times that of the ancestry model). In (b), ω ranges from 2 to $\frac{1}{4}$. Continuing this pattern, in (c), ω ranges from a maximum of 4 to a minimum of $\frac{1}{2}$.

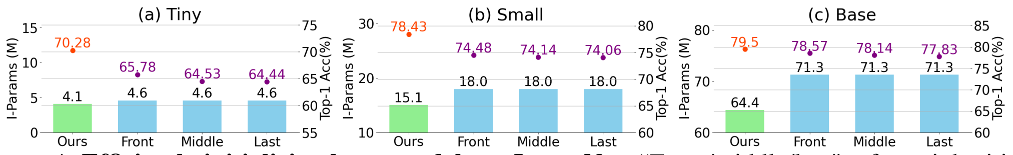

Figure 4: **Efficiently initializing large models on ImageNet.** "Front/middle/last" refer to inheriting parameters from the front, middle, or last 10 layers of a pretrained model to initialize 12-layer descendant models. All approaches are fine-tuned for 50 epochs. "I-Params" means the number of **I**nherited parameters in the downstream/descendant models, measured in MB.

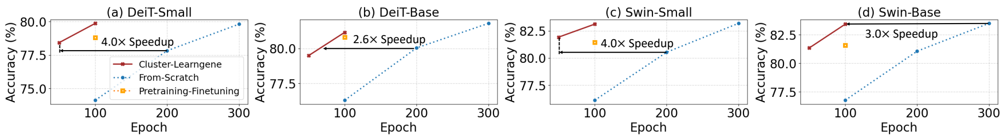

Figure 5: **Faster convergence.** Different points represent results for varying epochs and the hyperparameter ω is set to 1.0 for our method.

Table 2: **DeiT-Small Results on downstream datasets.** ↑ represents the performance improvement achieved by Cluster-Learngene, when compared to the best method excluding Pretraining-Finetuning. All results are derived from the 6-layer downstream models.

Method	I-Params	iNat-2019	Food-101	Flowers	Cars	CIFAR-10	CIFAR-100	CUB-200
Pretraining-Finetuning	10.5	68.48	87.8	91.13	86.81	97.59	84.43	78.13
From-Scratch	0	50.79	74.64	72.91	71.63	92.49	73.32	62.75
Heuristic-Learngene	5.6	53.21	77.09	82.84	81.52	93.12	78.13	72.64
Weight-Transformation	10.5	59.83	81.79	86.37	85.01	93.67	75.98	70.28
Auto-Learngene	10.5	59.92	80.25	87.02	84.98	93.58	79.49	73.31
Cluster-Learngene	7.5	**71.09(↑11.17)**	**89.53(↑7.74)**	**92.31(↑5.29)**	**89.87(↑4.86)**	**97.79(↑4.12)**	**85.38(↑5.89)**	**75.98(↑2.67)**

subsequently fine-tunes the entire model on downstream tasks. (ii) From-Scratch: We commence with a randomly initialized DeiT model on the downstream datasets. (iii) Heuristic-Learngene [49]: This strategy involves extracting the last six layers from a DeiT model pre-trained on ImageNet and then stacking them with randomly initialized lower layers to construct a new model. (iv) Weight-Transformation [63]: This method employs Weight Transformation to pre-train DeiT on ImageNet, followed by fine-tuning the entire model to adapt it to specific downstream tasks. (v) Auto-Learngene [50]: The first six layers are extracted from the DeiT and then stacked with randomly initialized higher layers to initialize the descendant models.

As illustrated in Tab. 2, our Cluster-Learngene significantly outperforms both From-Scratch and Weight-Transformation. When compared to other Learngene methods, such as Auto-Learngene, Cluster-Learngene exceeds by **11.17%** on the iNaturalist-2019 (iNat-2019). These results highlight the superior capability of Cluster-Learngene in efficiently initializing descendant models. Moreover, on six datasets, the performance of Cluster-Learngene outperforms that of Pretraining-Finetuning, where the entire model is fine-tuned. This phenomenon can be attributed to the more universally significant knowledge within learngene, allowing it to adapt effectively to various downstream tasks.

4.2.4 Faster Convergence

We provide a detailed comparison of training efficiency between our approach and From-Scratch on ImageNet. As shown in Fig. 5 (a), Cluster-Learngene requires only **4.0** $\times$ less training overhead compared to From Scratch on Small-scale descendant models. A key advantage of our approach is that descendant models initialized with the learngene achieve faster convergence, owing to a superior initialization point.

4.2.5 Higher Data Efficiency

We further conduct experiments on Base-scale descendant models over different percentages of training data from ImageNet-1K (IN-1K). As shown in Tab. 3, while our method does not outperform the From-Scratch on the entire dataset, its performance exhibits greater stability as the amount of training data decreases. For instance, with only 25% of the training data, Cluster-Learngene outperforms From-Scratch by **7.09%**, while requiring only $\frac{1}{6}$ of the training cost. This higher data efficiency of our method is attributed to the significant knowledge within the learngene, which helps descendant models mitigate overfitting, especially in scenarios with limited data.

Table 3: **Initialization of descendant models with diverse training samples.** The symbol ↑ denotes the performance gap between our approach and the From-Scratch method. Cluster-Learngene initializes the descendant model over 50 training epochs. In contrast, From-Scratch results are achieved after 300 training epochs.

Training data	From-Scratch	Cluster-Learngene
100% IN-1K	81.80	78.43
50% IN-1K	74.70	**76.41(↑1.71)**
25% IN-1K	65.73	**72.82(↑7.09)**

4.3 Analysis and Ablation

In this section, we provide further analysis and ablation of Cluster-Learngene. [§] Unless otherwise specified, we conduct experiments on **CIFAR-100** and use **Small-scale DeiT** as the ancestry model.

4.3.1 Comparison of the Clustering Method

Additionally, we compare the results of *k-means* clustering [20] of attention heads with cluster centroids (k) set at 1, 2, and 3. Tab. 4 shows that Cluster-Learngene not only outperforms in clustering efficiency but also adaptively adjusts the count of cluster centroids for each model layer, unlike *k-means* that requires predefined the numbers of cluster centroids.

Table 4: **Comparison of the clustering method.** All results are from the 6-layer downstream models.

k-means, $k=1$	*k-means*, $k=2$	*k-means*, $k=3$	Ours
80.12	81.02	81.41	**85.38**

4.3.2 Comparison of Priority Weight-sharing

In Table 5, our priority weight-sharing is compared against three alternative weight-sharing methods: following the original sequence of heads, by ascending $\tilde{d}$ of heads, and by descending $\tilde{d}$ of heads. The results validate the effectiveness of priority weight-sharing, as our method accounts for the fact that the larger the cluster a head belongs to, the richer the critical knowledge it represents.

Table 5: **Comparison of priority weight-sharing.** All results are derived from the 6-layer downstream models.

Origin	Increasing $\tilde{d}$	Decreasing $\tilde{d}$	Ours
83.98	84.24	84.63	**85.38**

4.3.3 Qualitative Visualization

We illustrate attention representations in Fig. 6 to explain which significant knowledge is inherited by learngene. To clarify the visualization, we apply a power exponent of $\gamma = 0.25$. In the first layer of the ancestry model, a head centroid is clustered from heads 1, 2, 4, and 5 to initiate the head 1 in

[§]Please see Appendix A.9 and A.10 for more visualization.

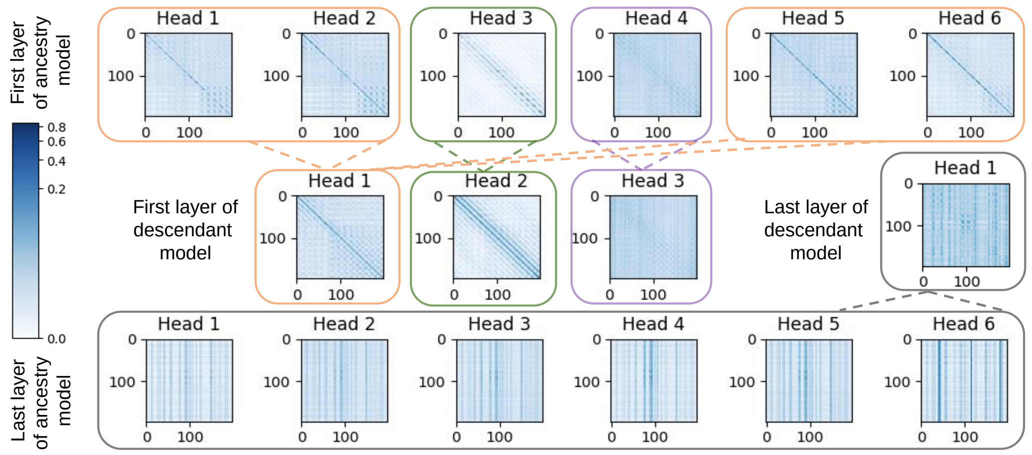

Figure 6: **Visualization of attention representations** (197×197)**.** We perform the following normalization operation on all attention heads $\mathbf{A}$ of the ancestry model and descendant model: $\left(\frac{\mathbf{A}_{i,j}}{255}\right)^{\gamma}$. The descendant model is trained for 50 epochs, and ω is set to $\frac{1}{4}$.

the descendant model, and so forth. Then, weight-sharing is applied to expand head centroids, *e.g.*, sharing twice to initialize the descendant model.

In the first layer, heads 1, 2, 4, and 5 with similar semantics form the largest cluster, mainly focusing on attention patterns along the main diagonal. Additionally, heads 3 and 4 present distinct semantics, with head 4 reflecting more abstract and high-level features akin to the final layer. Thus, the first learngene layer integrates three principal representation patterns from the ancestry model. In contrast, the representations in the final layer of the ancestry model exhibit significant repetition, leading to the clustering of a single-head centroid for initializing the attention heads of the descendant model.

5 Conclusion

We propose Cluster-Learngene to adaptively cluster attention heads, extracting head centroids and FFN centroids as the learngene. Subsequently, we adopt the priority weight-sharing of head centroids for varying number of attention heads and leverage learnable parameters to expand the FFN centroids into multiple FFNs, enabling adaptation to diverse downstream resource constraints. Extensive experiments validate the efficiency and scalability of our initialization method.

Acknowledgments and Disclosure of Funding

This research was supported by the National Science Foundation of China (62125602, 62076063), the Key Program of Jiangsu Science Foundation (BK20243012), the Fundamental Research Funds for the Central Universities (2242024k30035) and the Big Data Computing Center of Southeast University.

References

[1] Mohiuddin Ahmed, Raihan Seraj, and Syed Mohammed Shamsul Islam. The k-means algorithm: A comprehensive survey and performance evaluation. *Electronics*, 9(8):1295, 2020.

[2] Devansh Arpit, Víctor Campos, and Yoshua Bengio. How to initialize your network? robust initialization for weightnorm & resnets. *Advances in Neural Information Processing Systems*, 32, 2019.

[3] Panthadeep Bhattacharjee and Pinaki Mitra. A survey of density based clustering algorithms. *Frontiers of Computer Science*, 15:1–27, 2021.

[4] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101–mining discriminative components with random forests. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part VI 13*, pages 446–461. Springer, 2014.

[5] Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv preprint arXiv:2303.12712*, 2023.

[6] Adil Abdu Bushra and Gangman Yi. Comparative analysis review of pioneering dbscan and successive density-based clustering algorithms. *IEEE Access*, 9:87918–87935, 2021.

[7] Shiming Chen, Ziming Hong, Guo-Sen Xie, Wenhan Yang, Qinmu Peng, Kai Wang, Jian Zhao, and Xinge You. Msdn: Mutually semantic distillation network for zero-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7612–7621, June 2022.

[8] Vincent Cohen-Addad, Varun Kanade, Frederik Mallmann-Trenn, and Claire Mathieu. Hierarchical clustering: Objective functions and algorithms. *Journal of the ACM (JACM)*, 66(4):1–42, 2019.

[9] 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.

[10] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[11] 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.

[12] Timnit Gebru, Jonathan Krause, Yilun Wang, Duyun Chen, Jia Deng, and Li Fei-Fei. Fine-grained car detection for visual census estimation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, 2017.

[13] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256. JMLR Workshop and Conference Proceedings, 2010.

[14] Greg Hamerly and Charles Elkan. Learning the k in k-means. *Advances in neural information processing systems*, 16, 2003.

[15] Yizeng Han, Gao Huang, Shiji Song, Le Yang, Honghui Wang, and Yulin Wang. Dynamic neural networks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(11):7436–7456, 2021.

[16] Boris Hanin and David Rolnick. How to start training: The effect of initialization and architecture. *Advances in Neural Information Processing Systems*, 31, 2018.

[17] Uri Hasson, Samuel A Nastase, and Ariel Goldstein. Direct fit to nature: an evolutionary perspective on biological and artificial neural networks. *Neuron*, 105(3):416–434, 2020.

[18] Haoyu He, Zizheng Pan, Jing Liu, Jianfei Cai, and Bohan Zhuang. Efficient stitchable task adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 28555–28565, June 2024.

[19] 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.

[20] Abiodun M Ikotun, Absalom E Ezugwu, Laith Abualigah, Belal Abuhaija, and Jia Heming. K-means clustering algorithms: A comprehensive review, variants analysis, and advances in the era of big data. *Information Sciences*, 622:178–210, 2023.

[21] ZHAO Yongxin ZHANG Shenglin GONG Zican JI Yuhe, HAN Jing. Log anomaly detection through gpt-2 for large scale systems. *ZTE Communications*, 21(3):70, 2023.

[22] Runhao Jiang, Jie Zhang, Rui Yan, and Huajin Tang. Few-Shot Learning in Spiking Neural Networks by Multi-Timescale Optimization. *Neural Computation*, 33(9):2439–2472, 08 2021. ISSN 0899-7667. doi: 10.1162/neco_a_01423.

[23] Hans-Peter Kriegel, Peer Kröger, Jörg Sander, and Arthur Zimek. Density-based clustering. *Wiley interdisciplinary reviews: data mining and knowledge discovery*, 1(3):231–240, 2011.

[24] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[25] Lujun Li, Peijie, Zhenheng Tang, Xiang Liu, Qiang Wang, Wenhan Luo, Wei Xue, Qifeng Liu, Xiaowen Chu, and Yike Guo. Discovering sparsity allocation for layer-wise pruning of large language models. In *NeuIPS*, 2024.

[26] Wei Li, Lujun Li, Mark Lee, and Shengjie Sun. Als: Adaptive layer sparsity for large language models via activation correlation assessment. In *NeuIPS*, 2024.

[27] Yunfan Li, Peng Hu, Zitao Liu, Dezhong Peng, Joey Tianyi Zhou, and Xi Peng. Contrastive clustering. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 8547–8555, 2021.

[28] Shuxia Lin, Miaosen Zhang, Ruiming Chen, Xu Yang, Qiufeng Wang, and Xin Geng. Linearly decomposing and recomposing vision transformers for diverse-scale models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[29] Biao Liu, Ning Xu, and Xin Geng. Progressively selective label enhancement for language model alignment. *arXiv preprint arXiv:2408.02599*, 2024.

[30] 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, 2021.

[31] WANG Jiabo LIU Yang LUO Wenjian LIU Qinbo, JIN Zhihao. Msra-fed: A communication-efficient federated learning method based on model split and representation aggregate. *ZTE Communications*, 20 (3):35, 2022.

[32] Xuran Meng and Jianfeng Yao. Impact of classification difficulty on the weight matrices spectra in deep learning and application to early-stopping. *J. Mach. Learn. Res.*, 24(1), March 2024. ISSN 1532-4435.

[33] Xuran Meng, Difan Zou, and Yuan Cao. Benign overfitting in two-layer relu convolutional neural networks for xor data. In *Forty-first International Conference on Machine Learning*.

[34] Paul Michel, Omer Levy, and Graham Neubig. Are sixteen heads really better than one? *Advances in neural information processing systems*, 32, 2019.

[35] Fionn Murtagh and Pedro Contreras. Algorithms for hierarchical clustering: an overview. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 2(1):86–97, 2012.

[36] ZOU Xiaojing DOU Yutao Albert Y. ZOMAYA NAN Yucen, FANG Minghao. A collaborative medical diagnosis system without sharing patient data. *ZTE Communications*, 20(3):3, 2022.

[37] Zanlin Ni, Yulin Wang, Jiangwei Yu, Haojun Jiang, Yu Cao, and Gao Huang. Deep incubation: Training large models by divide-and-conquering. 2022.

[38] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*, pages 722–729. IEEE, 2008.

[39] Zizheng Pan, Jianfei Cai, and Bohan Zhuang. Stitchable neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16102–16112, June 2023.

[40] Yingzhe Peng, Chenduo Hao, Xu Yang, Jiawei Peng, Xinting Hu, and Xin Geng. Learnable in-context vector for visual question answering. *arXiv preprint arXiv:2406.13185*, 2024.

[41] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR, 2021.

[42] 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.

[43] Erich Schubert, Jörg Sander, Martin Ester, Hans Peter Kriegel, and Xiaowei Xu. Dbscan revisited, revisited: why and how you should (still) use dbscan. *ACM Transactions on Database Systems (TODS)*, 42(3):1–21, 2017.

[44] Chongjie Si, Zhiyi Shi, Shifan Zhang, Xiaokang Yang, Hanspeter Pfister, and Wei Shen. Unleashing the power of task-specific directions in parameter efficient fine-tuning. *arXiv preprint arXiv:2409.01035*, 2024.

[45] Kiat Chuan Tan, Yulong Liu, Barbara Ambrose, Melissa Tulig, and Serge Belongie. The herbarium challenge 2019 dataset. *arXiv preprint arXiv:1906.05372*, 2019.

[46] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International conference on machine learning*, pages 10347–10357. PMLR, 2021.

[47] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[48] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011.

[49] Qiu-Feng Wang, Xin Geng, Shu-Xia Lin, Shi-Yu Xia, Lei Qi, and Ning Xu. Learngene: From open-world to your learning task. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 8557–8565, 2022.

[50] Qiufeng Wang, Xu Yang, Shuxia Lin, and Xin Geng. Learngene: Inheriting condensed knowledge from the ancestry model to descendant models. *arXiv preprint arXiv:2305.02279*, 2023.

[51] Qiufeng Wang, Xu Yang, Haokun Chen, and Xin Geng. Vision transformers as probabilistic expansion from learngene. In *Forty-first International Conference on Machine Learning*, 2024.

[52] Yu Wang, Xinjie Yao, Pengfei Zhu, Weihao Li, Meng Cao, and Qinghua Hu. Integrated Heterogeneous Graph Attention Network for Incomplete Multi-modal Clustering. *International Journal of Computer Vision*, 132(9):3847–3866, September 2024. ISSN 1573-1405. doi: 10.1007/s11263-024-02066-y.

[53] Yulin Wang, Rui Huang, Shiji Song, Zeyi Huang, and Gao Huang. Not all images are worth 16x16 words: Dynamic transformers for efficient image recognition. *Advances in neural information processing systems*, 34:11960–11973, 2021.

[54] Yongrong Wu, Jingyu Lin, Houjin Chen, Dinghao Chen, Lvqing Yang, and Jianbing Xiahou. A graph-transformer network for scene text detection. In *International Conference on Intelligent Computing*, pages 680–690. Springer, 2023.

[55] Shiyu Xia, Miaosen Zhang, Xu Yang, Ruiming Chen, Haokun Chen, and Xin Geng. Transformer as linear expansion of learngene. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 16014–16022, 2024.

[56] Zhenda Xie, Zigang Geng, Jingcheng Hu, Zheng Zhang, Han Hu, and Yue Cao. Revealing the dark secrets of masked image modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14475–14485, 2023.

[57] Xingyi Yang, Daquan Zhou, Songhua Liu, Jingwen Ye, and Xinchao Wang. Deep model reassembly. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 25739–25753. Curran Associates, Inc., 2022.

[58] Xu Yang, Yingzhe Peng, Haoxuan Ma, Shuo Xu, Chi Zhang, Yucheng Han, and Hanwang Zhang. Lever lm: Configuring in-context sequence to lever large vision language models. *arXiv e-prints*, pages arXiv–2312, 2023.

[59] Yao Yao, Zuchao Li, and Hai Zhao. Sirllm: Streaming infinite retentive llm. *arXiv preprint arXiv:2405.12528*, 2024.

[60] Hua Yuan, Yu Shi, Ning Xu, Xu Yang, Xin Geng, and Yong Rui. Learning from biased soft labels. *Advances in Neural Information Processing Systems*, 36, 2023.

[61] Wei Yuan, Hongzhi Yin, Tieke He, Tong Chen, Qiufeng Wang, and Lizhen Cui. Unified question generation with continual lifelong learning. In *Proceedings of the ACM Web Conference 2022*, WWW '22, page 871–881, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450390965. doi: 10.1145/3485447.3511930.

[62] Anthony M Zador. A critique of pure learning and what artificial neural networks can learn from animal brains. *Nature communications*, 10(1):1–7, 2019.

[63] Jinnian Zhang, Houwen Peng, Kan Wu, Mengchen Liu, Bin Xiao, Jianlong Fu, and Lu Yuan. Minivit: Compressing vision transformers with weight multiplexing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12145–12154, 2022.

[64] Linfeng Zhang, Chenglong Bao, and Kaisheng Ma. Self-distillation: Towards efficient and compact neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(8):4388–4403, 2021.

[65] Pengfei Zhu, Xinjie Yao, Yu Wang, Binyuan Hui, Dawei Du, and Qinghua Hu. Multi-view deep subspace clustering networks. *arXiv preprint arXiv:1908.01978*, 2019.

Dataset	# Total	#Training	#Validation	#Testing	#Classes
Oxford Flowers [38]	8,189	1,020	1,020	6,149	102
CUB-200-2011 [48]	11,788	5,394	600	5,794	200
Stanford Cars [12]	16,185	7,329	815	8,041	196
CIFAR10 [24]	65,000	50,000	5,000	10,000	10
CIFAR100 [24]	65,000	50,000	5,000	10,000	100
Food101 [4]	101,000	75,750	25,250	0	101
iNaturalist-2019 [45]	268,243	265,213	3030	/	1010

Table 6: Characteristics of the downstream datasets

A Appendix / supplemental material

A.1 Mean Attention Distance in DeiT-S and DeiT-Ti

Fig. 7 illustrates the mean attention distance for two other variants of DeiT. In both variants, the lower layers exhibit a dual focus on both local and global aspects, resulting in a relatively sparse distribution of attention heads. Conversely, the higher layers prioritize the global context, leading to a more compact distribution of attention heads. Importantly, many attention heads in these layers exhibit repetitive functionality, contributing to weight redundancy. In the process of FFN clustering, the shallowest layers are preserved as the FFN centroids across different configurations: in DeiT-Tiny, this applies to layers (2,3) and (11,12); in DeiT-Small, to layers (10,11,12); and in DeiT-Base, to layers (7,8) and (10,11,12).

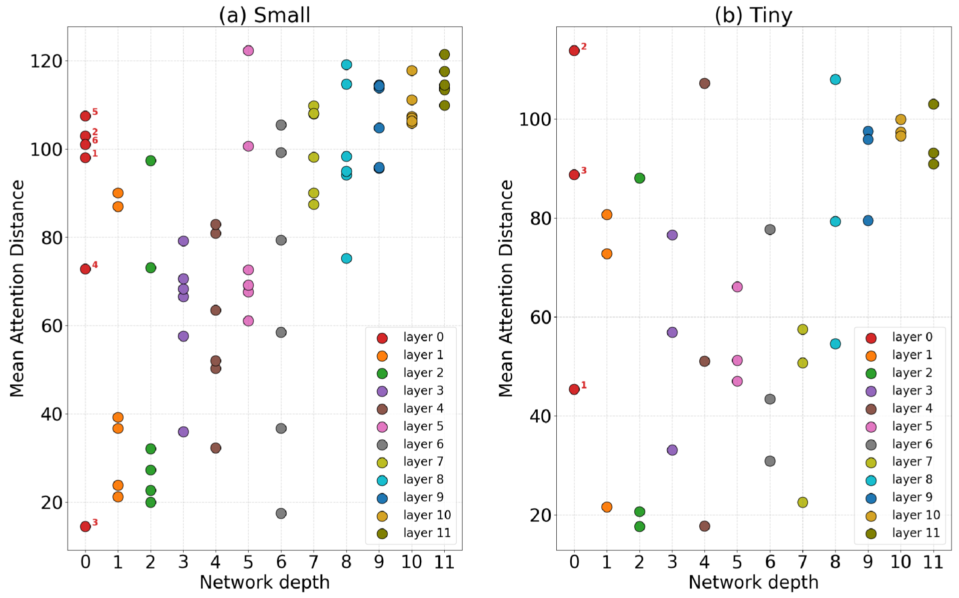

Figure 7: The distribution density of attention heads across the different layers of the ancestry model, which employs the DeiT-S and DeiT-Ti [46].

A.2 Downstream Datasets

Tab. 6 presents the details of all downstream tasks.

A.3 Projection Layer

According to the adjustments in the number of attention heads, the weights $\mathbf{W}^O$ of the projection layer are also proportionally pruned or expanded with the hyperparameter ω and then inherited by

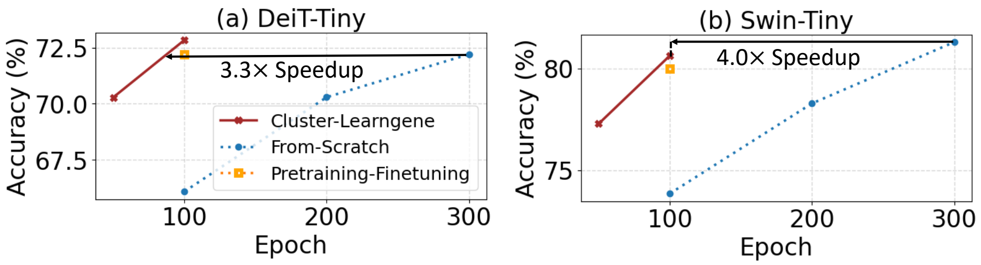

Figure 8: **Faster convergence.** Different points represent results for varying epochs and the hyperparameter ω is set to 1.0 for our method.

the descendant models. Additionally, we directly inherit the weights of layer normalization, patch embeddings, and position embeddings in the ancestry model, which constitute only a small fraction of all weights.

A.4 Faster Convergence

We provide a detailed comparison of training efficiency on ImageNet for the Tiny-scale descendant models. As shown in Fig. 8 (b), Cluster-Learngene requires only **4.0** $\times$ less training overhead compared to From Scratch on Small-scale descendant models. A key advantage of our approach is that descendant models initialized with the learngene achieve faster convergence, owing to a superior initialization point.

A.5 Ablation on the Selection of Head Centroids

We conduct an ablation to assess whether inheriting parameters from the nearest module to the cluster centroid or averaging parameters after clustering heads and FFNs is more effective. Table 7 shows Cluster-Learngene excelling when inheriting from the closest heads/FFNs to the centroid, a logical approach as these modules aptly represent similar semantics within a cluster.

MSA	FFN	Acc (%)
average	average	85.12
argmin	average	84.29
average	argmin	84.76
argmin	argmin	**85.38**

Table 7: **Ablation on the selection of head centroids.** All results are derived from the 6-layer downstream models. we conduct experiments on CIFAR-100 and use DeiT-Small as the ancestry model.

A.6 Ablation on ε

Table 8 shows the results for different values of ε. $\varepsilon = 1$ implies no FFN clustering, potentially causing negative transfer. $\varepsilon = 100$ means clustering all FFNs in adjacent layers with identical head centroid counts, resulting in an excessive cluster of FFNs and subsequent degradation in the performance of initialized descendant models. Our Cluster-Learngene ($\varepsilon = 10$) strikes a good balance between these issues.

$\varepsilon = 1$	$\varepsilon = 10$	$\varepsilon = 100$
85.27	**85.38**	80.45

Table 8: **Ablation on ε.** All results are derived from the 6-layer downstream models.

Model	Decrementing	Incrementing
Tiny	76.56	**78.01**
Small	79.47	**81.29**
Base	80.18	**81.65**

Table 9: **Increment or decrement the count of attention heads.** "Decrementing" denotes halving the number of attention heads in the first four layers, reducing them by a quarter in the middle four layers, and maintaining them in the last four layers relative to the ancestry model. Conversely, "Incrementing" represents the opposite pattern.

A.7 Variation in the Count of Attention Head with Model Depth

Tab. 9 presents two scenarios where the number of attention heads varies across different layers. Across all descendant model configurations, "Incrementing" consistently outperforms "Decrementing" by a margin of 1.45% in terms of accuracy. These findings align with previous research [34, 30], which suggests that setting more attention heads in higher layers can assist these layers in learning more abstract and high-level feature representations.

A.8 Initializing Descendant Models of Elastic Scales

As illustrated in Fig. 3, Cluster-Learngene incurs a total of 300 + 50 * 10 = 800 epochs. In contrast, pre-trained models require retraining a new model of elastic scale for 300 epochs each, followed by fine-tuning. Considering only the training expense of new models, training 10 such models would require at least 300 * 10 = 3000 epochs. Therefore, our algorithm can reduce the computational cost by at least $3\times$. This efficiency demonstrates how our method overcomes the one-size-fits-all limitation inherent in the Pretraining-Finetuning approach.

A.9 Limitations

The performance of descendant models within Cluster-Learngene heavily depends on the quality of the ancestry model. If the ancestry model harbors inherent limitations or biases, these issues may propagate and even amplify within the descendant models, potentially compromising their effectiveness. Therefore, ensuring the robustness and fairness of the ancestry model is paramount for maintaining the performance of descendant models in Cluster-Learngene.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Please see Section 4.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Please see Appendix.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: Please see Section 3.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Please see Section 4.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: Please see Appendix.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: Please see Section 4.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Please see Section 4.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.

- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Please see Section 4.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Please see Appendix.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Please see Appendix.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to

generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.

- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: Please see the Experimtent Section.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [NA]

Justification: The paper does not release new assets.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing or research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.