Vision Transformers as Probabilistic Expansion from Learngene

Qiufeng Wang [1 2] **Xu Yang** [1 2] **Haokun Chen** [1 2] **Xin Geng** [1 2]

Abstract

Deep learning has advanced through the combination of large datasets and computational power, leading to the development of extensive pre-trained models like Vision Transformers (ViTs). However, these models often assume a one-size-fits-all utility, lacking the ability to initialize models with elastic scales tailored to the resource constraints of specific downstream tasks. To address these issues, we propose **P**robabilistic **E**xpansion from Learn**G**ene (PEG) for mixture sampling and elastic initialization of Vision Transformers. Specifically, PEG utilizes a probabilistic mixture approach to sample Multi-Head Self-Attention layers and Feed-Forward Networks from a large ancestry model into a more compact part termed as learngene. Theoretically, we demonstrate that these learngene can approximate the parameter distribution of the original ancestry model, thereby preserving its significant knowledge. Next, PEG expands the sampled learngene through non-linear mapping, enabling the initialization of descendant models with elastic scales to suit various resource constraints. Our extensive experiments demonstrate the effectiveness of PEG and outperforming traditional initialization strategies.

1. Introduction

The trajectory of deep learning has been significantly shaped by the integration of vast data resources and powerful computational technologies. This synergy has led to the emergence of extensive pre-trained foundation models (Dosovitskiy et al., 2021; Devlin et al., 2019; Radford et al., 2021; Bubeck et al., 2023; Yang et al., 2023b), notably those based on Transformer architecture (Vaswani et al., 2017; Dosovitskiy et al., 2021), including Vision Transformers (ViTs)(Dosovitskiy et al., 2021). These foundation models, which are widely utilized across a variety of devices from smartphones to edge computing devices, act as starting points(Hanin & Rolnick, 2018; Arpit et al., 2019; He et al., 2016; Zhang et al., 2021; Wang et al., 2022; 2023) for a broad range of downstream tasks.

Nevertheless, such a method of model initialization often assumes that the whole model fits all purposes, ignoring

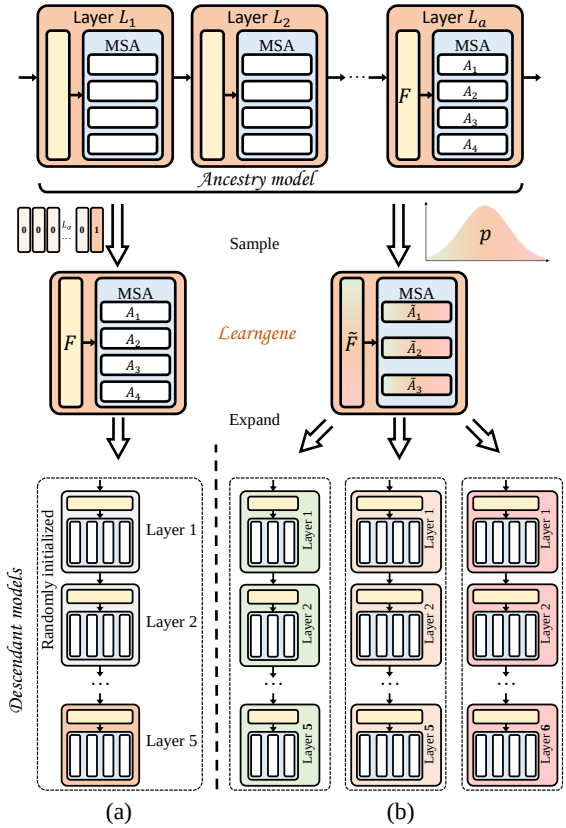

Figure 1. (a) Existing works (Wang et al., 2022; 2023) primarily involve selecting a few integral layers as the learngene and manually integrating them with layers that are randomly initialized. (b) Our PEG implements a probabilistic mixture to sample MSA of each layer and FFNs as the learngene and then expands them to initialize descendant models of elastic scales.

[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. Correspondence to: Xin Geng <xgeng@seu.edu.cn>, Xu Yang <xuyang_palm@seu.edu.cn>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

specific limitations like memory usage, processing power, or response time that are important in some downstream tasks. This one-size-fits-all mindset may not be elastic, particularly when deploying models on devices with limited resources. For example, even the 86M Base-Scale ViT (Touvron et al., 2021) needs to consider how well it fits with the available resources. Moreover, employing these foundational models at their full scale runs the risk of negative transfer (Wang et al., 2019b; Zhang et al., 2022b), potentially carrying over less favorable aspects to downstream tasks.

To effectively initialize models, (Wang et al., 2022; 2023) have innovatively introduced the *Learngene* framework, inspired by observations of biological genes. This framework is designed in two stages: significant knowledge is condensed from a large **ancestry model** into a more compact part termed as **learngene** and then the learngene serves as a start point for initializing **descendant models of elastic scales**. As depicted in Fig. 1 (a), existing works (Wang et al., 2022; 2023) primarily involve selecting a few integral layers as the learngene and manually integrating them with layers that are randomly initialized.

However, these methods present two key limitations: (i) Extracting only certain integral layers directly overlooks the distribution of knowledge across the unselected layers of the ancestry model. (ii) The approach of manually integrating the learngene with randomly initialized layers falls short in adaptively scaling the model to initialize downstream models of any customized size.

To tackle these challenges, we propose **P**robabilistic **E**xpansion from Learn**G**ene (PEG) for mixture sampling and elastic initialization of Vision Transformers. As illustrated in Fig. 1 (b), for Multi-Head Self-Attention (MSA) layers, we implement a probabilistic mixture $\sum_{i=1}^{H_a} p_{ih}\mathbf{A}_i$ to derive MSA layers from each layer of an ancestry model, thereby forming MSA learngene. Here, $\mathbf{A}_i$ denotes the parameters of attention heads within a specific layer of the ancestry model. Each MSA learngene is a composite of the original H_a heads, where the weights p_k are determined by probabilistic factors from a standard Gaussian distribution. This deliberate choice of Gaussian distribution for generating p_k capitalizes on its widespread utility for its straightforwardness and flexibility, thus amplifying the effectiveness of learngene in condensing knowledge from an ancestry model while preserving its performance and generalizability. Similarly, for Feed-Forward Networks (FFNs), we employ a probabilistic mixture $\sum_{i=1}^{L_a} p_{il}\mathbf{F}_i$ to sample FFN layers as the FFN learngene, where $\mathbf{F}_i$ denotes the parameters of FFNs in the i^{th} layer of an ancestry model with depth L_a.

During the expanding phase, we further employ probabilistic mixture to sample from both MSA and FFN learngene, thereby initializing descendant models with flexible scales tailored to downstream resource constraints. Specifically, the probabilistic mixture is relaxed to nonlinear learnable parameters. By simply fine-tuning these parameters according to downstream data, we achieve an elastic expansion in the number of attention heads and FFNs. This process equips the initialized descendant models with scalability in both width and depth. Our **contributions** are summarized as follows:

- We propose a probabilistic mixture to sample MSA layers and FFNs as learngene from the ancestry model and theoretically demonstrate that these learngene can approximate the parameter distribution of the original ancestry model, thereby preserving its significant knowledge.
- We further introduce non-linear mapping to expand MSA learngene and FFN learngene for initializing models with elastic scales.
- Extensive experiments across various datasets validate that PEG not only surpasses traditional initialization strategies but also competes effectively with more resource-intensive fine-tuning methodologies.

2. Related Work

Model Initialization: Model initialization plays a vital role in training deep neural networks, affecting their convergence speed and final performance. Over the years, various initialization techniques have been proposed. These techniques encompass the widely used random initialization, as well as more sophisticated methods like Xavier initialization (Glorot & Bengio, 2010) and the Kaiming initialization (He et al., 2016). Recently, the utilization of pre-trained foundation models has garnered significant attention as an initialization strategy prior to fine-tuning for specific tasks (Dosovitskiy et al., 2021; Devlin et al., 2019; Radford et al., 2021; Yang et al., 2022; Ni et al., 2022; Bubeck et al., 2023; Wang et al., 2019a; Yuan et al., 2022; LI Daiyi, 2022; NAN Yucen, 2022; LIU Qinbo, 2022; JI Yuhe, 2023; Yang et al., 2023a; Shi et al., 2024; Xia et al., 2024; Meng et al., 2023a;b; 2024). Nevertheless, this approach necessitates pre-training separate models for each downstream task, as depicted in Figure 2, which does not take into account the resource constraints and customization of model sizes required for downstream tasks. Additionally, the reuse of the entire original model can introduce the risk of negative transfer effects (Wang et al., 2019b; Zhang et al., 2022b), leading to unstable performance on downstream tasks. In contrast, our approach only requires the sampling of a learngene containing significant knowledge once and can rapidly expand it into models of elastic sizes to adapt to the varying complexities and performance for the downstream tasks.

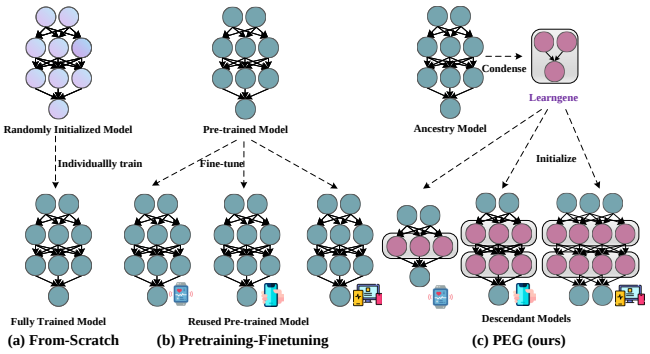

Figure 2. (a) Training from scratch involves randomly initializing models of different scales to adapt to downstream tasks, with complexity increasing linearly with the number of models. (b) Pretraining-Finetuning entails reusing the original whole model each time to adapt to various downstream scenarios. (c) Our PEG allows us to sample a learngene containing significant knowledge just once and rapidly expand it to models of elastic scales, accommodating the varying model complexities and performance requirements of downstream scenarios.

Mixture Models for Transformers. Recently, there has been a growing interest in leveraging Mixture Models to enhance Transformers. For instance, in the works (Guo et al., 2019; Cho et al., 2020), Gaussian priors are introduced to facilitate a better capture of knowledge for downstream tasks. Another notable development is the use of Switch Transformers (Fedus et al., 2022), which employs a routing algorithm inspired by Mixture of Experts (MoE) to reduce communication and computational costs in transformers. Moreover, (Nguyen et al., 2022) has derived a Gaussian mixture model, focusing specifically on attention heads. This approach aims to reduce redundant heads and enhance the training efficiency of transformers. However, it's worth noting that these methods continue to build upon the Pretraining-Finetuning initialization paradigm. In contrast, our PEG takes a different approach by considering the flexible initialization of models with elastic scales based on the requirements of downstream tasks.

3. Methodology

The *Learngene* framework operates in two distinct phases: significant knowledge is condensed from a large ancestry model into a more concise part termed as learngene, which then forms the basis for initializing descendant models of elastic scales. Initially, we analyze the Vision Transformers with a Probabilistic Mixture of MSA and FFNs, demonstrating that the learngene can closely approximate the parameter distribution of the original ancestry model, thus retaining its significant knowledge. Building on this theoretical foundation, in the learngene sampling phase as illustrated in Fig. 3, we employ a probabilistic mixture to sample MSA and FFN from the ancestry model. In the learngene expansion phase, non-linear mapping is introduced to enlarge the MSA learngene and FFN learngene, facilitating the initialization of descendant models with elastic scales. Next, we briefly introduce some preliminary concepts related to ViTs.

3.1. Preliminary

In the architecture of Vision Transformer, an initial step involves segmenting the input image into N distinct patches. Each patch is then linearly transformed into a D-dimensional vector. The core of the ViT encoder is a sequence of layers, each comprising a MSA mechanism and FFN blocks.

Let us denote the total count of attention heads in each layer by H. For the h^{th} head, the query matrix $\mathbf{Q}_h$, key matrix $\mathbf{K}_h$, and value matrix $\mathbf{V}_h$, each of dimension $\mathbb{R}^{N\times d_k}$ for the query and key, and $\mathbb{R}^{N\times d_v}$ for the value, are computed linearly with learned weights matrices $\mathbf{W}_h^Q$, $\mathbf{W}_h^K$, and $\mathbf{W}_h^V$, of dimensions $\mathbb{R}^{D\times d_k}$ and $\mathbb{R}^{D\times d_v}$ respectively. The SA process for each head is captured as:

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

The MSA framework empowers the model to concurrently process information from various positions and representation subspaces:

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

where $\mathbf{W}^O \in \mathbb{R}^{Hd_v\times D}$ is a learned weight matrix of dimension $\mathbb{R}^{Hd_v\times D}$. Additionally, the FFN is defined as:

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

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

3.2. Vision Transformers with a Probabilistic Mixture of MSA and FFNs

3.2.1. Theoretical Analysis for MSA

In the ancestry model, for each layer of attention heads $\mathbf{A}_1,\ldots,\mathbf{A}_i,\ldots,\mathbf{A}_{H_a}$, we hypothesize that each $\mathbf{A}_i$ is drawn from a distribution $\mathbb{Q}_i$. Due to the high dimensionality and computational complexity of the distribution $\mathbb{Q}$, we naturally opt for a Probabilistic Mixture of Gaussian distributions to approximate $\mathbb{Q}$. The theoretical justification is as follows:

Theorem 3.1. *Let's consider $\mathbb{Q} \in \mathbb{R}^{D'}$ as a probability distribution contained within a compact set, characterized*

by a differentiable and bounded density function q. For any scale parameter $\alpha > 0$ and any threshold $\epsilon > 0$, there exists a universal constant C and a number of components $H_a \leq (C \log(1/\epsilon))^{D'}$, enabling the construction of a mixture $\sum_{i=1}^{H_a} p_i \mathcal{N}(\theta_i^{MSA}, \alpha^2 \mathbf{I}_{D'})$. Here, $p_1, \ldots, p_{H_a}$ are mixture weights, and $\theta_1^{MSA}, \ldots, \theta_{H_a}^{MSA}$ are the MSA parameters. This mixture satisfies the inequality:

$$\sup_{x \in \mathbb{R}^d} \left| q(x) - \sum_{i=1}^{H_a} p_i \phi\left(x \mid \theta_i^{MSA}, \alpha^2 \mathbf{I}_{D'}\right) \right| \leq \epsilon,$$

where $\phi(\cdot|\theta^{MSA}, \alpha^2\mathbf{I})$ is the Gaussian density function with mean θ^{MSA} and covariance matrix $\alpha^2\mathbf{I}_D$.

The proof of Theorem 3.1 is provided in Appendix C. Based on Theorem 3.1, we can sample H_b attention heads of Gaussian mixture from the each layer of an ancestry model. Let $h = 1, 2, \ldots, H_b$. For the j^{th} Gaussian mixture, $\mathbb{Q}'_h = \sum_{i=1}^{H_a} p_{ih} \mathcal{N}\left(\theta_{ih}^{\text{MSA}}, \alpha^2 \mathbf{I}_{D'}\right)$ approximate the distribution $\mathbb{Q}$ with a given accuracy ϵ, along with their corresponding weights $p_{1h}, \ldots, p_{H_a h}$, and the MSA parameters $\theta_{1h}^{\text{MSA}}, \ldots, \theta_{H_a h}^{\text{MSA}}$. This theoretical assurance enables us to sample a more compact number of attention heads from the ancestry model as learngene (*i.e.*, MSA learngene), while ensuring that the MSA learngene retains as much of the significant knowledge from the ancestry model as possible.

3.2.2. Theoretical Analysis for FFNs

In the ancestry model, each FFN layer consists of transformation functions $\mathbf{F}_1, \ldots, \mathbf{F}_i, \ldots, \mathbf{F}_{L_a}$, where we posit that each $\mathbf{F}_i$ is influenced by a distribution $\mathbb{P}_i$. Likewise, a Probabilistic Mixture of Gaussian distributions is selected to approximate $\mathbb{P}$, with the theoretical foundation for this method elaborated as follows:

Theorem 3.2. *Consider $\mathbb{P} \in \mathbb{R}^{D'}$ representing a probability distribution within a bounded and compact set, characterized by a differentiable and limited density function f. Given a scale parameter $\beta > 0$ and a threshold $\delta > 0$, there is a universal constant C and a number of components $L_a \leq (C \log(1/\delta))^{D'}$, allowing the formation of a mixture $\sum_{i=1}^{L_a} p_i \mathcal{N}(\theta_i^{FFN}, \beta^2 \mathbf{I}_D)$. In this expression, $p_1, \ldots, p_{L_a}$ denote mixture weights, and $\theta_1^{FFN}, \ldots, \theta_{L_a}^{FFN}$ are FFN parameters. This mixture fulfills the conditions:*

$$\sup_{x \in \mathbb{R}^d} \left| f(x) - \sum_{i=1}^{L_a} p_i \phi\left(x \mid \theta_i^{\text{FFN}}, \beta^2 \mathbf{I}_{D'}\right) \right| \leq \delta,$$

where $\phi(\cdot|\theta^{FFN}, \beta^2\mathbf{I})$ is the Gaussian density function with mean θ^{FFN} and covariance matrix $\beta^2\mathbf{I}_{D'}$.

Following the rationale of Theorem 3.2, we can effectively sample L_b FFNs of Gaussian mixtures. Let $l = 1, 2, \ldots, L_b$. For the l^{th} Gaussian mixture, $\mathbb{P}'_l = \sum_{i=1}^{L_a} p_{il} \mathcal{N}\left(\theta_{il}^{\text{FFN}}, \beta^2 \mathbf{I}_{D'}\right)$ approximates the distribution $\mathbb{P}$ with a designated accuracy δ, accompanied by corresponding weights $p_{1l}, \ldots, p_{L_a l}$, and FFN parameters $\theta_{1l}^{\text{FFN}}, \ldots, \theta_{L_a l}^{\text{FFN}}$. Similarly, this theoretical assurance allows us to sample a critical set of FFN layers from the ancestry model as learngene (*i.e.*, FFN learngene). To sum up, based on the Probabilistic Mixture of MSA and FFNs, we are going to elaborate on the learngene sampling in the next sections.

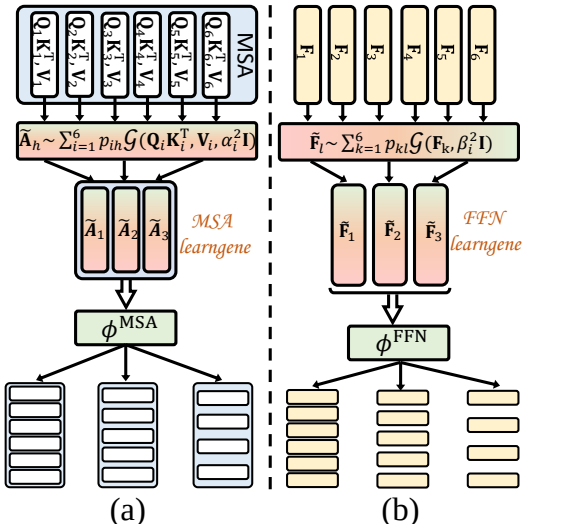

Figure 3. **Illustration of Learngene Sampling and Expanding.** (a) For MSA, the attention heads of each layer in the ancestry model are sampled with Gaussian mixtures into the MSA learngene. Subsequently, the MSA learngene is expanded with learnable parameters to initialize the descendant models of varying widths. (b) For FFN, Gaussian mixtures are employed to sample the FFN learngene from all layers of the ancestry model. The FFN learngene is then expanded with learnable parameters to facilitate the initialization of descendant models with varying depths.

3.3. Learngene Sampling

3.3.1. MSA Sampling

As illustrated in Fig. 3, we introduce an innovative method to sample the MSA layers from a pre-trained ancestry model. Originally, each layer of the ancestry model consists of H_a attention heads. To efficiently sample and refine the attention heads, we apply the following procedure H_b times:

$$\begin{aligned} \tilde{\mathbf{A}}_h &= \text{Attention}\left(\sum_{i=1}^{H_a} p_{ih}\left(\mathbf{Q}_i \mathbf{K}_i^\top, \mathbf{V}_i\right)\right) \\ \text{s.t., } p_{ih} &\sim \mathcal{G}(0, \mathbf{I}), \quad \sum_{i=1}^{H_a} p_{ih} = 1, \quad p_{ih} \geq 0, \end{aligned} \tag{4}$$

where $h = 1, 2, \ldots, H_b$. This formula aims to synthesize a reduced number of H_b attention head as the MSA learn-

gene from the ancestry model. Each MSA learngene is a weighted combination of the original H_a heads, where the weights p_k are probabilistic factors drawn from a standard Gaussian distribution. The selection of the Gaussian distribution for generating probabilities p_k is deliberate, rooted in its widespread use in machine learning for simplicity and versatility, enhancing the ability of learngene to efficiently condense knowledge from a larger model while preserving performance and generalization.

3.3.2. FFN SAMPLING

Building on the approach used for MSA, we apply a similar method to the FFNs in a pretrained ancestry model, which initially consists of L_a FFN layers. In this context, the FFN parameters at the k^{th} layer of the ancestry model are denoted by $\mathbf{F}_k = (\mathbf{W}_1, \mathbf{W}_2, \mathbf{b}_1, \mathbf{b}_2)$. The following formula is employed to sample and refine the FFN layers:

$$\begin{aligned} \tilde{\mathbf{F}}_l &= \sum_{k=1}^{L_a} p_{kl} \mathbf{F}_k \\ \text{s.t., } p_{kl} &\sim \mathcal{G}(0, \mathbf{I}), \quad \sum_{k=1}^{L_a} p_{kl} = 1, \quad p_{kl} \geq 0, \end{aligned} \tag{5}$$

where $l = 1, 2, \ldots, L_b$. This equation condenses the FFN layers into L_b layers as the FFN learngene. Similar to the MSA learngene, each FFN learngene layer is a weighted combination of the original L_a layers, with the weights p_{il} being probabilistic factors derived from a standard Gaussian distribution.

3.4. Learngene Expanding

3.4.1. MSA EXPANDING

We expand upon the approach introduced in the previous section, where we sample MSA learngene $\tilde{\mathbf{A}}_1, \ldots, \tilde{\mathbf{A}}_{\mathrm{H}_b}$ from each layer of the ancestry model. For each layer of the descendant models, we propose the learnable parameters to expand the number of attention heads. This process is defined by the following equation:

$$\text{MultiHead}(\cdot) = \text{Concat}\left(\phi^{\text{MSA}}\left(\tilde{\mathbf{A}}_1, \ldots, \tilde{\mathbf{A}}_{H_b}\right)\mathbf{W}^O\right), \tag{6}$$

Here, ϕ is a learnable transformation function capable of expanding the number of attention heads from H_b to H_c. This expansion is essential for initializing the descendant model with H_c attention heads in each layer, offering flexibility to adapt the number of initialized attention heads according to the resource constraints of downstream tasks. To implement this, we utilize a nonlinear mapping such as a neural network employing rectified linear units (ReLU). 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. [1]

3.4.2. FFN EXPANDING

We further discuss the process of expanding L_b layers of FFN learngene as described in Section 3.3. Building upon the methodology used for expanding MSA, we also employ learnable parameters to expand the number of FFN learngene, which helps in initializing descendant models with an elastic depth. The formulation of this process is as follows: $\phi^{\text{FFN}}\left(\tilde{\mathbf{F}}\right)$. Here, ϕ^{FFN} serves as a transformative function that flexibly expands the number of FFNs from L_b to L_c layers. In summary, our method ensures that the model can dynamically adjust both its depth and width to meet the resource requirements of downstream tasks during initialization.

4. Experiments

4.1. Experimental Setting

Datasets. After initializing the descendant models with the learngene, we fine-tune them on various downstream tasks, including Oxford Flowers (Nilsback & Zisserman, 2008), CUB-200-2011 (Wah et al., 2011), Stanford Cars (Gebru et al., 2017), CIFAR-10 (Krizhevsky et al., 2009), CIFAR-100 (Krizhevsky et al., 2009), Food101 (Bossard et al., 2014), iNaturalist-2019 (Tan et al., 2019), ImageNet-1K (Deng et al., 2009). For detailed dataset descriptions, see Appendix A.

Training settings. During the learngene expanding phase, we train the learnable parameters for 100 epochs before expanding them into descendant models of elastic scales. After this, we fine-tune these descendant models on downstream tasks for 500 epochs, which includes a 10-epoch warm-up period. The only exception is iNaturalist-2019, where we train for 100 epochs with a 5-epoch warm-up. For all tasks, the initial learning rate is set to 5×10^{-4} and a weight decay of 0.05 is applied.

Architectures. Both the ancestry model and descendant models are based on DeiT (Touvron et al., 2021), which comes in three width variants: **Tiny**, **Small**, and **Base**. Additionally, as discussed in Sections 4.2.3 and 4.2.4, we conduct experiments on Swin Transformer (Liu et al., 2021) to showcase the versatility of our approach across different architectures.

4.2. Main Results of Model Initialization

In this section, we evaluate the effectiveness of the PEG framework in efficiently initializing models, assessing their performance with Top-1 accuracy.

[1] Please see Appendix B for more details.

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

Model	H_c	L_c	Params (M)	FLOPs (G)	From-Scratch	PEG
Tiny	2	6	1.3	0.3	50.06	50.03
		9	1.9	0.4	54.64	54.56
		12	2.5	0.5	57.99	**58.01**
	3	6	3	0.6	58.16	57.92
		9	4.4	0.9	60.58	60.45
		12	5.7	1.2	61.44	**61.55**
Small	4	6	5	1	61.32	**62.28**
		9	7.3	1.4	63.17	**63.91**
		12	9.5	1.9	64.25	**64.46**
	6	6	11.4	2.3	64.91	**64.94**
		9	16.8	3.4	67.02	**69.49**
		12	22	4.6	68.56	**70.24**
Base	12	3	22.8	4.5	68.77	**69.24**
		4	29.9	5.9	70.32	**70.66**
		5	36.9	7.4	72.04	**72.3**
		6	44	8.8	73.73	**73.98**
		7	51.2	10.2	74.42	**74.63**
		8	58.2	11.7	76.14	**76.19**
		9	65.3	13.1	76.46	**76.82**
		10	72.4	14.6	76.81	**76.95**
		11	79.5	16	77.03	**77.16**
		12	86.6	17.5	77.22	**77.39**

4.2.1. Initializing Descendant Models of Elastic Scales

In real-world scenarios, various downstream tasks often demand models of different scales to accommodate their specific requirements. Our PEG addresses this challenge by offering a flexible solution that expands the sampled learngene into models of elastic scales. To illustrate, we conduct the initialization of 22 descendant models on the ImageNet-1K, each characterized by different configurations, such as the number of attention heads H_c per layer and the quantity of FFNs L_c in descendant models. As indicated in Tab. 1, PEG not only quickly initializes models of varying scales but also proves to be a competitive performer in terms of overall performance. Let's take Base-scale descendant models as a case in point. We find that PEG not only achieves better performance but also reduces training costs by approximately $1.7\times$ (comparing 10×100 epochs to $100 + 10 \times 50$ epochs). Furthermore, as we upscale the initialized descendant models, the performance of PEG continues to improve. Specifically, while PEG may initially lag behind fully-trained From-Scratch models in the case of Tiny-Scale models, it shines with superior performance for Small and Base-Scale models. This demonstrates that our PEG effectively addresses the one-size-fits-all problem, where each model of a novel scale typically requires retraining from scratch to achieve a good initialization.

4.2.2. Initialization Results on Different Downstream Tasks

We conduct a comparative analysis of our approach for initializing descendant or downstream models as follows: (i) Fine-tuning: This approach pre-trains DeiT on ImageNet and subsequently fine-tunes the entire model on downstream tasks. (ii) From-Scratch: We commence with a randomly initialized DeiT model and exclusively train it on the downstream datasets. (iii) Heur-Learngene (Wang et al., 2022): This strategy involves extracting the last three layers from a DeiT model pre-trained on ImageNet. These layers are then stacked with randomly initialized lower layers to construct a new model. (iv) Weight-Multi (Zhang et al., 2022a): 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. Moreover, Weight-Multi utilizes distillation as a trick. (v) Auto-Learngene (Wang et al., 2023): 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 PEG significantly outperforms both From-Scratch and Weight-Multi. When compared to other Learngene methods, such as Auto-Learngene, PEG exceeds by **7.81%** on the iNaturalist-2019 (iNat-2019) for the Small-scale descendant models. These results highlight the superior capability of PEG in efficiently initializing descendant models. Moreover, the performance of PEG outperforms that of Fine-tuning on Flowers for the Tiny-scale descendant models, where the entire model is fine-tuned. This phenomenon can be attributed to the more universally significant knowledge within the learngene, allowing it to adapt effectively to various downstream tasks. In contrast, Fine-tuning risks negative transfer effects (Wang et al., 2019b; Zhang et al., 2022b) by reusing the entire model, potentially carrying over less favorable aspects to downstream tasks.

4.2.3. Faster Convergence

We conduct an in-depth analysis of training efficiency by directly comparing our method to From-Scratch on the ImageNet dataset. As depicted in Fig. 4 (b), the results reveal that our PEG substantially reduces training time, demanding only **3.6** $\times$ less training time than From-Scratch on the Base-scale descendant models. This significant reduction in training time directly results from the capability of PEG to offer a superior initialization point for the descendant models. With the learngene as an enriched starting point, the models rapidly converge towards their optimal performance. Notably, even after just 100 epochs of training, PEG surpasses the Fine-tuning strategy applied to the entire pre-trained model. This efficiency represents a substantial advantage in real-world scenarios, where faster convergence translates to reduced computational costs and quicker model

Table 2. **Initialization results on different downstream tasks.** "I-Params" represent the number of parameters **I**nherited into the downstream/descendant models. The symbol ↑ denotes the performance improvement achieved by our method compared to the best-performing method excluding Fine-tuning. The results presented for PEG are based on the 6-layer descendant model, while the model depth of other baselines matches that of the descendant model.

Model	Method	I-Params (M)	Flowers	CUB-200	CIFAR-10	CIFAR-100	Food-101	iNat-2019	Cars
Tiny	Fine-tuning	2.8	84.79	71.12	96.65	80.81	83.24	58.12	75.71
	From-Scratch	0	68.82	59.83	88.3	67.44	61.54	37.16	67.32
	Heur-Learngene	1.3	78.67	67.5	91.66	70.19	72.54	41.55	70.68
	Weight-Multi	2.8	80.01	66.25	93.07	74.01	77.36	51.22	74.19
	Auto-Learngene	2.8	80.84	67.51	93.02	75.83	79.12	52.46	74.20
	PEG (ours)	**2.5**	**87.85**(↑7.84)	**67.97**(↑0.46)	**96.33**(↑3.26)	**79.85**(↑4.02)	**85.37**(↑6.25)	**56.8**(↑4.34)	**71.75**
Small	Fine-tuning	10.5	91.13	78.13	97.59	84.43	87.8	68.48	86.81
	From-Scratch	0	72.91	62.75	92.49	73.32	74.64	50.79	71.63
	Heur-Learngene	5.6	82.84	72.64	93.12	78.13	77.09	53.21	81.52
	Weight-Multi	10.5	86.37	70.28	93.67	75.98	81.79	59.83	85.01
	Auto-Learngene	10.5	87.02	73.31	93.58	79.49	80.25	59.92	84.98
	PEG (ours)	**9.7**	**91.01**(↑3.99)	**78.18**(↑5.54)	**97.38**(↑2.21)	**83.59**(↑4.1)	**87.15**(↑5.36)	**67.73**(↑7.81)	**82.57**

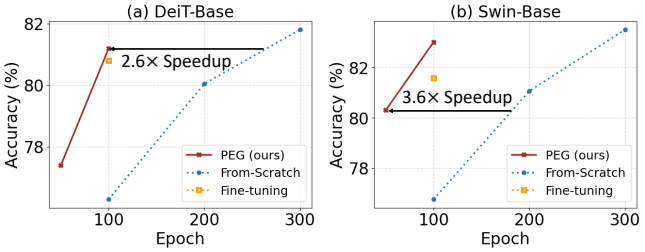

Figure 4. **Faster convergence** of DeiT-Base (a) and Swin-Base (b). Different points represent results for diverse epochs.

deployment.

4.2.4. Higher Data Efficiency

The last advantage of our method is its ability to perform well even when there is a limited amount of training data available. To demonstrate this, we conduct experiments on two subsets of the ImageNet-1K(IN-1K) dataset. One subset contain 25% of the training data, while the other have 50%. We train Base-scale descendant models on these subsets and observe that our method outperform From-Scratch in these scenarios.

As summarized in Tab. 3, our PEG demonstrates increased stability as the volume of training data decreases, while not surpassing From-Scratch performance on the entire dataset. For example, with just 25% of the training data, PEG surpasses From-Scratch by **10.6%** on DeiT-Base, while consuming only $\frac{1}{6}$ of the training resources. This improved data efficiency can be attributed to the significant knowledge encoded within the learngene, which helps descendant models mitigate overfitting, particularly in situations with limited data. Furthermore, we conduct experiments on the Swin-Transformer, yielding similar results. This underscores the versatility and effectiveness of our approach across different model architectures.

Table 3. **Higher data efficiency.** The symbol ↑ indicates the performance gap between our approach and From-Scratch. PEG initializes the descendant model over **50** training epochs, while From-Scratch achieves its results after **300** training epochs.

Training data	DeiT-Base		Swin-Base	
	From-Scratch	PEG	From-Scratch	PEG
100% IN-1K	81.8	77.4	83.5	80.3
50% IN-1K	74.7	77.1(↑2.4)	76.2	79.7(↑3.5)
25% IN-1K	65.7	76.3(↑10.6)	68.1	79.5(↑11.4)

4.3. Analysis and Ablation

/In this section, we delve deeper into the analysis and ablation study of PEG. For our experiments, we primarily focus on the **CIFAR-100** dataset, employing **Small-scale DeiT** as the foundational ancestry model, unless stated otherwise.

4.3.1. Latent Embedding Visualization

Figure 5 displays the T-SNE visualization of feature representations obtained from both the From-Scratch and PEG methods on the CIFAR-10 dataset. This visualization highlights a notable disparity in the performance of these two approaches. In the case of From-Scratch, we observe that it encounters difficulties in effectively distinguishing between different classes within the dataset. This struggle can be attributed to the challenges faced by models trained from scratch, as they often require an extensive amount of data and training time to develop meaningful class-specific representations. In contrast, our PEG demonstrates a remarkable capability to rapidly acquire and encode class-specific information for downstream tasks. At "Epoch 100", PEG showcases clear clustering patterns and well-defined decision

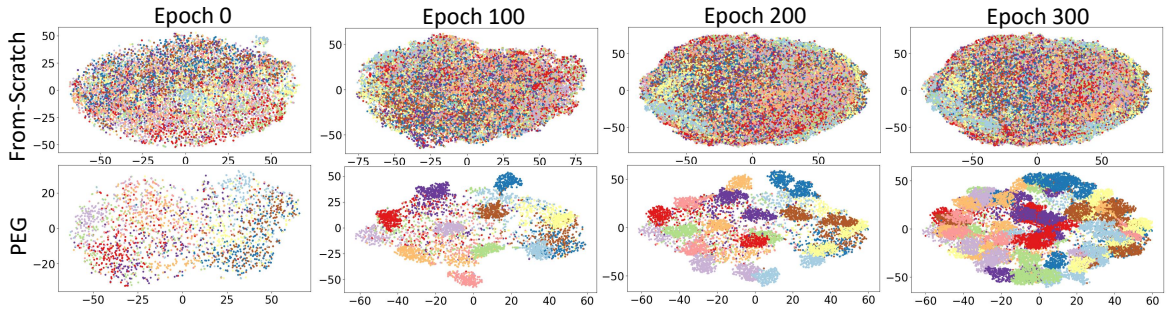

Figure 5. **T-SNE feature visualization** of the descendant model from different epochs.

boundaries. This impressive performance can be attributed to the ability of PEG to leverage the extensive inter-class semantic knowledge stored within the *learngene*, which is inherited from the ancestry model.

4.3.2. Analysis on the Selection of Distribution

Tab. 4 presents the results of sampling MSA and FFN learngene with different probabilistic mixtures. Our PEG outperforms outperforms the utilization of alternative probability distributions. This superiority arises from distinct characteristics of these distributions. The uniform distribution uniformly samples values within a specified range, which leads to suboptimal initialization due to this absence of central tendency. The central tendency of the beta distribution is comparatively less stable and is reliant on hyperparameters, making it less consistent for initialization purposes. The Poisson distribution with its discrete nature may yield discrete initialization weight values, constraining the adaptability of initialized weights for downstream tasks. In contrast, the Gaussian distribution not only exhibits central tendencies but also provides superior continuity. When utilized for sampling, it positions initialized weights closer to the mean, facilitating rapid convergence in downstream tasks while preventing excessive weight dispersion.

Table 4. **Analysis on the Selection of Distribution** for sampling MSA and FFN learngene.

Uniform	Beta	Poisson	Gaussian (ours)
83.02	82.13	82.26	**83.59**

4.3.3. Component-wise Analysis

Tab. 5 illustrates the results when descendant models inherit various components from the ancestry model. Our experimental performances lead to the following three conclusions: (i) In the case of MSA, inheriting MSA^Q and MSA^K is more crucial than MSA^V. This observation is based on the fact that MSA^Q and MSA^K play pivotal roles in the self-attention mechanism, carrying significant information for downstream tasks, while MSA^V contributes less to task-related knowledge (Tay et al., 2021; Kim et al., 2021). (ii) Inheriting the complete MSA is more important than FFN. This preference is due to MSA's ability to capture complex relationships, long-range dependencies, and global context within images, making it a fundamental component in Vision Transformers (ViTs). FFN, while important, primarily focuses on local patterns and fine-grained details. Thus, initializing descendant models with MSA as the learngene component provides them with richer spatial dependencies and context information, significantly benefiting downstream tasks. (iii) Our PEG, which simultaneously samples both MSA and FFN as the learngene, attains the most favorable results and provides further validation of the effectiveness of our method.

Table 5. **Ablation study on different components** inheriting from the ancestry model.

MSA^Q	MSA^K	MSA^V	FFN	PEG
✓				75.18
	✓			75.24
		✓		74.74
			✓	78.85
✓	✓	✓		79.02
✓	✓	✓	✓	**83.59**

5. Conclusion

In this paper, we introduce a novel approach termed PEG for mixture sampling and elastic initialization of Vision Transformers. PEG leverages probabilistic mixtures to sample MSA layers and FFNs as learngene from an ancestry model, effectively preserving the significant knowledge of the original model. During the expansion phase, we employ non-linear mapping to flexibly adjust the number of attention heads and FFNs in descendant models, providing scalability in both width and depth. Extensive experiments validate the efficiency and scalability of our initialization method. In the context of initialization methods, PEG not only expedites model convergence but also tailors models

to suit downstream tasks, rendering it a valuable asset for practical applications in the field of Vision Transformers.

Impact Statement

The implementation of learngene in PEG has the potential for broader societal impact. It introduces learngene as a medium for facilitating model interactions, thereby safeguarding data privacy in both upstream and downstream tasks. This technology has the capacity to advance responsible AI development by ensuring data privacy, building trust, and fostering acceptance in applications across various sectors such as healthcare, finance, and personal devices.

Acknowledgements

This research was supported by the National Science Foundation of China (62125602, 62076063, 62206048), Natural Science Foundation of Jiangsu Province (BK20220819), Young Elite Scientists Sponsorship Program of Jiangsu Association for Science and Technology Tj-2022-027 and the Big Data Computing Center of Southeast University.

References

Arpit, D., Campos, V., and Bengio, Y. How to initialize your network? robust initialization for weightnorm & resnets. *Advances in Neural Information Processing Systems*, 32, 2019.

Bossard, L., Guillaumin, M., and Van Gool, L. 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*, pp. 446–461. Springer, 2014.

Bubeck, S., Chandrasekaran, V., Eldan, R., Gehrke, J., Horvitz, E., Kamar, E., Lee, P., Lee, Y. T., Li, Y., Lundberg, S., et al. Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv preprint arXiv:2303.12712*, 2023.

Cho, S. M., Park, E., and Yoo, S. Meantime: Mixture of attention mechanisms with multi-temporal embeddings for sequential recommendation. In *Proceedings of the 14th ACM Conference on Recommender Systems*, pp. 515–520, 2020.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. 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)*, pp. 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

Fedus, W., Zoph, B., and Shazeer, N. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *The Journal of Machine Learning Research*, 23(1):5232–5270, 2022.

Gebru, T., Krause, J., Wang, Y., Chen, D., Deng, J., and Fei-Fei, L. Fine-grained car detection for visual census estimation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, 2017.

Ghosal, S. and Van Der Vaart, A. W. Entropies and rates of convergence for maximum likelihood and bayes estimation for mixtures of normal densities. *The Annals of Statistics*, 29(5):1233–1263, 2001.

Glorot, X. and Bengio, Y. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 249–256. JMLR Workshop and Conference Proceedings, 2010.

Guo, M., Zhang, Y., and Liu, T. Gaussian transformer: a lightweight approach for natural language inference. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 6489–6496, 2019.

Hanin, B. and Rolnick, D. How to start training: The effect of initialization and architecture. *Advances in Neural Information Processing Systems*, 31, 2018.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

JI Yuhe, HAN Jing, Z. Y. Z. S. G. Z. Log anomaly detection through gpt-2 for large scale systems. *ZTE Communications*, 21(3):70, 2023.

Kim, K., Wu, B., Dai, X., Zhang, P., Yan, Z., Vajda, P., and Kim, S. J. Rethinking the self-attention in vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3071–3075, 2021.

Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images. 2009.

LI Daiyi, TU Yaofeng, Z. X. Z. Y. M. Z. End-to-end chinese entity recognition based on bert-bilstm-att-crf. *ZTE Communications*, 20(S1):27, 2022.

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., and Guo, B. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 10012–10022, 2021.

LIU Qinbo, JIN Zhihao, W. J. L. Y. L. W. Msra-fed: A communication-efficient federated learning method based on model split and representation aggregate. *ZTE Communications*, 20(3):35, 2022.

Meng, X., Cao, Y., and Zou, D. Per-example gradient regularization improves learning signals from noisy data. *arXiv preprint arXiv:2303.17940*, 2023a.

Meng, X., Zou, D., and Cao, Y. Benign overfitting in two-layer relu convolutional neural networks for xor data. *arXiv preprint arXiv:2310.01975*, 2023b.

Meng, X., Yao, J., and Cao, Y. Multiple descent in the multiple random feature model. *Journal of Machine Learning Research*, 25(44):1–49, 2024.

NAN Yucen, FANG Minghao, Z. X. D. Y. A. Y. Z. A collaborative medical diagnosis system without sharing patient data. *ZTE Communications*, 20(3):3, 2022.

Nguyen, T. M., Nguyen, T. M., Le, D. D., Nguyen, D. K., Tran, V.-A., Baraniuk, R., Ho, N., and Osher, S. Improving transformers with probabilistic attention keys. In *International Conference on Machine Learning*, pp. 16595–16621. PMLR, 2022.

Ni, Z., Wang, Y., Yu, J., Jiang, H., Cao, Y., and Huang, G. Deep incubation: Training large models by divide-and-conquering. 2022.

Nilsback, M.-E. and Zisserman, A. Automated flower classification over a large number of classes. In *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*, pp. 722–729. IEEE, 2008.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., and Sutskever, I. Learning transferable visual models from natural language supervision. In Meila, M. and Zhang, T. (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8748–8763. PMLR, 2021.

Shi, B., Xia, S., Yang, X., Chen, H., Kou, Z., and Geng, X. Building Variable-Sized Models via Learngene Pool. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(13):14946–14954, March 2024.

Tan, K. C., Liu, Y., Ambrose, B., Tulig, M., and Belongie, S. The herbarium challenge 2019 dataset. *arXiv preprint arXiv:1906.05372*, 2019.

Tay, Y., Bahri, D., Metzler, D., Juan, D.-C., Zhao, Z., and Zheng, C. Synthesizer: Rethinking self-attention for transformer models. In *International conference on machine learning*, pp. 10183–10192. PMLR, 2021.

Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., and Jégou, H. Training data-efficient image transformers & distillation through attention. In *International conference on machine learning*, pp. 10347–10357. PMLR, 2021.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Wah, C., Branson, S., Welinder, P., Perona, P., and Belongie, S. The caltech-ucsd birds-200-2011 dataset. 2011.

Wang, Q., Liu, M., Zhang, W., Guo, Y., and Li, T. Automatic Proofreading in Chinese: Detect and Correct Spelling Errors in Character-Level with Deep Neural Networks. pp. 349–359. September 2019a. ISBN 978-3-030-32235-9. doi: 10.1007/978-3-030-32236-6_31.

Wang, Q., Yang, X., Lin, S., and Geng, X. Learngene: Inheriting condensed knowledge from the ancestry model to descendant models. *arXiv preprint arXiv:2305.02279*, 2023.

Wang, Q.-F., Geng, X., Lin, S.-X., Xia, S.-Y., Qi, L., and Xu, N. Learngene: From open-world to your learning task. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 8557–8565, 2022.

Wang, Z., Dai, Z., Póczos, B., and Carbonell, J. Characterizing and avoiding negative transfer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11293–11302, 2019b.

Xia, S.-Y., Zhu, W., Yang, X., and Geng, X. Exploring learngene via stage-wise weight sharing for initializing variable-sized models. *arXiv preprint arXiv:2404.16897*, 2024.

Yang, X., Zhou, D., Liu, S., Ye, J., and Wang, X. Deep model reassembly. In Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., and Oh, A. (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 25739–25753. Curran Associates, Inc., 2022.

Yang, Y., Huang, Y., Guo, W., Xu, B., and Xia, D. Towards global video scene segmentation with context-aware transformer. In *Thirty-Seventh AAAI Conference on Artificial Intelligence*, pp. 3206–3213, Washington, DC, USA, 2023a.

Yang, Z., Li, L., Lin, K., Wang, J., Lin, C.-C., Liu, Z., and Wang, L. The dawn of lmms: Preliminary explorations with gpt-4v (ision). *arXiv preprint arXiv:2309.17421*, 9 (1):1, 2023b.

Yuan, W., Yin, H., He, T., Chen, T., Wang, Q., and zhen Cui, L. Unified question generation with continual lifelong learning. *Proceedings of the ACM Web Conference 2022*, 2022.

Zhang, J., Peng, H., Wu, K., Liu, M., Xiao, B., Fu, J., and Yuan, L. Minivit: Compressing vision transformers with weight multiplexing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12145–12154, 2022a.

Zhang, L., Bao, C., and Ma, K. Self-distillation: Towards efficient and compact neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44 (8):4388–4403, 2021.

Zhang, W., Deng, L., Zhang, L., and Wu, D. A survey on negative transfer. *IEEE/CAA Journal of Automatica Sinica*, 10(2):305–329, 2022b.

Dataset	# Total	#Training	#Validation	#Testing	#Classes
Oxford Flowers (Nilsback & Zisserman, 2008)	8,189	1,020	1,020	6,149	102
CUB-200-2011 (Wah et al., 2011)	11,788	5,394	600	5,794	200
Stanford Cars (Gebru et al., 2017)	16,185	7,329	815	8,041	196
CIFAR10 (Krizhevsky et al., 2009)	65,000	50,000	5,000	10,000	10
CIFAR100 (Krizhevsky et al., 2009)	65,000	50,000	5,000	10,000	100
Food101 (Bossard et al., 2014)	101,000	75,750	25,250	0	101
iNat-2019 (Tan et al., 2019)	268,243	265,213	3030	/	1010

Table 6. Characteristics of the downstream datasets

A. Downstream Datasets

Tab. 6 presents the details of all downstream tasks, with the eight datasets sorted by data size.

B. 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 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.

C. Derivation of Theorem

For ease of presentation in this proof, we denote, for any probability distribution G:

$$q_G(x) := \int f(x-\theta)dG(\theta) = \int \phi\left(x \mid \theta, \sigma^2\mathbf{I}\right) dG(\theta),$$

where $x \in \mathbb{R}^d$ and $f(x) = \frac{1}{(\sqrt{2\pi}\sigma)^d} \exp\left(-\frac{|x|^2}{2\sigma^2}\right)$ for a given $\sigma > 0$. Here, q_G represents the function of f and the probability distribution G. Given that the space of Gaussian mixtures is dense in the space of continuous probability measures (Ghosal & Van Der Vaart, 2001), we can conclude that there exists a probability distribution G_1 such that:

$$\sup_{x\in\mathbb{R}^d} |q(x) - q_{G_1}(x)| \leq \frac{\epsilon}{2}. \tag{7}$$

Our subsequent objective is to establish the existence of a probability measure G_2 with at most K supports, where $K \leq (C\log(1/\epsilon))^d$ for some universal constant C. This new measure satisfies:

$$\sup_{x\in\mathbb{R}^d} |q_{G_1}(x) - q_{G_2}(x)| \leq \frac{\epsilon}{2}. \tag{8}$$

Exploiting Lemma A. 1 from (Ghosal & Van Der Vaart, 2001; Nguyen et al., 2022), we can find a probability distribution G_2 with at most $(2k-2)^d$ supports, where:

$$\int \theta^\alpha d\left(G_1 - G_2\right)(\theta) = 0, \tag{9}$$

for any $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_d) \in \mathbb{N}^d$ with $0 \leq |\alpha| = \sum_{j=1}^d \alpha_j \leq 2k-2$, where $\theta^\alpha = \prod_{j=1}^d \theta_j^{\alpha_j}$. Now, for any $M \geq 2a\sqrt{d}$, we can derive:

$$\|x-\theta\| \geq \|x\| - \|\theta\| > M - a\sqrt{d} > M/2, \tag{10}$$

as long as $|x| > M$ and $\theta \in [-a, a]^d$. This implies:

$$\begin{aligned}\sup_{\|x\|>M}|q_{G_1}(x)-q_{G_2}(x)| &= \sup_{\|x\|>M}\left|\int f(x-\theta)d\left(G_1-G_2\right)(\theta)\right| \\ &\leq \sup_{\|x\|>M}\int \frac{1}{(\sqrt{2\pi}\sigma)^d}\exp\left(-\frac{\|x-\theta\|^2}{2\sigma^2}\right) \\ &\leq \frac{2}{(\sqrt{2\pi}\sigma)^d}\exp\left(-\frac{M^2}{8\sigma^2}\right).\end{aligned} \tag{11}$$

Conversely, for any $k \geq 1$, we also have:

$$\begin{aligned}\sup_{\|x\|\leq M}|q_{G_1}(x)-q_{G_2}(x)| &\leq \sup_{\|x\|\leq M}\int\left|f(x-\theta)-\sum_{j=0}^{k-1}\frac{(-1)^j\|x-\theta\|^{2j}}{(\sqrt{2\pi})^d\sigma^{d+2j}j!}\right|d\left(G_1+G_2\right)(\theta) \\ &\leq 2\sup_{\|x\|\leq M,\theta\in[-a,a]^d}\left|f(x-\theta)-\sum_{j=0}^{k-1}\frac{(-1)^j\|x-\theta\|^{2j}}{(\sqrt{2\pi})^d\sigma^{d+2j}j!}\right| \\ &= \sup_{\|x\|\leq M,\theta\in[-a,a]^d}\frac{2}{(\sqrt{2\pi}\sigma)^d}\left|\exp\left(-\frac{\|x-\theta\|^2}{2\sigma^2}\right)-\sum_{j=0}^{k-1}\frac{(-1)^j\|x-\theta\|^{2j}}{\sigma^{2j}j!}\right| \\ &\leq \sup_{\|x\|\leq M,\theta\in[-a,a]^d}\frac{e^k\|x-\theta\|^{2k}}{\sigma^{2k}(2k)^k},\end{aligned} \tag{12}$$

where the final equality arises from:

$$\int\sum_{j=0}^{k-1}\frac{(-1)^j\|x-\theta\|^{2j}}{(\sqrt{2\pi})^d\sigma^{d+2j}j!}d\left(G_1-G_2\right)(\theta)=0, \tag{13}$$

which can be derived from Eqn. 9. To further bound the right-hand-side (RHS) of Eqn. 12, we use the following inequality:

$$\left|\exp(y)-\sum_{j=0}^{k-1}(y)^j/j!\right| \leq |y|^k/k!,$$

for any $y \in \mathbb{R}$. Since $k! \geq (k/e)^k$ for any $k \geq 1$, the above bound can be rewritten as:

$$\left|\exp(y)-\sum_{j=0}^{k-1}(y)^j/j!\right| \leq \frac{|ye|^k}{k^k}. \tag{14}$$

Further simplification of Eqn. 12 leads to

$$\begin{aligned}&\sup_{\|x\|\leq M}|q_{G_1}(x)-q_{G_2}(x)| \leq \sup_{\|x\|\leq M}\int\left|f(x-\theta)-\sum_{j=0}^{k-1}\frac{(-1)^j\|x-\theta\|^{2j}}{(\sqrt{2\pi})^d\sigma^{d+2j}j!}\right|d\left(G_1+G_2\right)(\theta) \\ &\leq 2\sup_{\|x\|\leq M,\theta\in[-a,a]^d}\left|f(x-\theta)-\sum_{j=0}^{k-1}\frac{(-1)^j\|x-\theta\|^{2j}}{(\sqrt{2\pi})^d\sigma^{d+2j}j!}\right|\exp\left(-\frac{\|x-\theta\|^2}{2\sigma^2}\right)-\sum_{j=0}^{k-1}\frac{(-1)^j\|x-\theta\|^{2j}}{\sigma^{2j}j!}\Bigg| \\ &= \sup_{\|x\|\leq M,\theta\in[-a,a]^d}\left|\frac{2}{(\sqrt{2\pi}\sigma)^d}\right| \\ &\leq \sup_{\|x\|\leq M,\theta\in[-a,a]^d}\frac{e^k\|x-\theta\|^{2k}}{\sigma^{2k}(2k)^k}.\end{aligned}$$

In the last inequality, we applied inequality 14 with $y = -|x-\theta|^2/\left(2\sigma^2\right)$. For $|x| \leq M$ and $\theta \in [-a,a]^d$, we have $|x-\theta| \leq |x| + |\theta| \leq M + a\sqrt{d}$. Therefore, we further have:

$$\sup_{\|x\|\leq M} |p_{G_1}(x) - p_{G_2}(x)| \leq \sup_{\|x\|\leq M, \theta\in[-a,a]^d} \frac{e^k \|x-\theta\|^{2k}}{\sigma^{2k}(2k)^k} \leq \frac{e^k (M + a\sqrt{d})^{2k}}{\sigma^{2k}(2k)^k}.$$

Given that $M \geq 2a\sqrt{d}$, it follows that $M + a\sqrt{d} \leq \frac{3M}{2}$. Therefore, the above bound can be simplified as:

$$\sup_{\|x\|\leq M} |q_{G_1}(x) - q_{G_2}(x)| \leq \frac{(9e)^k M^{2k}}{\left(8\sigma^2 k\right)^k}. \tag{15}$$

By choosing $M^2 = 8\sigma^2 \log\left(1/\epsilon'\right)$ for some $\epsilon' > 0$, the bounds in Eqns. 11 and 15 become

$$\sup_{\|x\|\leq M} |q_{G_1}(x) - q_{G_2}(x)| \leq \frac{2}{(\sqrt{2\pi}\sigma)^d}\epsilon'$$

$$\sup_{\|x\|> M} |q_{G_1}(x) - q_{G_2}(x)| \leq \frac{(9e)^k \left(\log\left(1/\epsilon'\right)\right)^k}{k^k} \tag{16}$$

As long as we choose $k = 9e^2 \log\left(1/\epsilon'\right)$ and $\epsilon' \leq 1$, we have

$$\sup_{\|x\|> M} |q_{G_1}(x) - q_{G_2}(x)| \leq e^{-k} = e^{-9e^2 \log\left(1/\epsilon'\right)} = \left(\epsilon'\right)^{9e^2} \leq \epsilon'. \tag{17}$$

By choosing $\epsilon' = \frac{\epsilon}{2\max\left\{\frac{2}{(\sqrt{2\pi}\sigma)^d}, 1\right\}}$, the results from Eqns. 16 and 17 indicate that

$$\sup_{\|x\|\leq M} |q_{G_1}(x) - q_{G_2}(x)| \leq \frac{\epsilon}{2}, \quad \text{and} \quad \sup_{\|x\|> M} |q_{G_1}(x) - q_{G_2}(x)| \leq \frac{\epsilon}{2}.$$

Therefore, if we choose $M = 8\sigma^2 \log\left(\frac{2\max\left\{\frac{2}{(\sqrt{2\pi}\sigma)^d}, 1\right\}}{\epsilon}\right)$ and $k = 9e^2 \log\left(\frac{2\max\left\{\frac{2}{(\sqrt{2\pi}\sigma)^d}, 1\right\}}{\epsilon}\right)$, we have

$$\sup_{x\in\mathbb{R}^d} |q_{G_1}(x) - q_{G_2}(x)| \leq \frac{\epsilon}{2}.$$

This implies that we can establish the conclusion of claim 8 by choosing $K = (2k-2)^d \leq \left(18e^2 \log\left(\frac{2\max\frac{2}{(\sqrt{2\pi}\sigma)^d}, 1}{\epsilon}\right)\right)^d$. Combining the results from Eqns. 7 and 8, we can conclude:

$$\sup_{x\in\mathbb{R}^d} |q(x) - q_{G_2}(x)| \leq \sup_{x\in\mathbb{R}^d} |q(x) - q_{G_1}(x)| + \sup_{x\in\mathbb{R}^d} |q_{G_1}(x) - q_{G_2}(x)| \leq \epsilon.$$

As a consequence, we obtain the conclusion of the theorem.