Adaptive-Learngene: Continual Expansion and Task-Aware Selection of Learngenes for Dynamic Environments

Shuxia Lin, Qiufeng Wang, Chang Liu, Xu Yang*, Xin Geng *

School of Computer Science and Engineering, Southeast University, Nanjing 210096, China
Key Laboratory of New Generation Artificial Intelligence Technology and Its Interdisciplinary Applications (Southeast University), Ministry of Education, China
{shuxialin, qfwang, liuchang0520, xuyang_palm, xgeng}@seu.edu.cn

Abstract

Pre-trained Vision Transformer (ViT) models have achieved impressive performance across various computer vision tasks. However, most existing pre-trained models are built on fixed datasets and lack the flexibility to incorporate new pre-training data. When additional data becomes available, previous models must typically be retrained on both old and new data, which is costly and impractical, especially in privacy-sensitive or resource-constrained environments. Moreover, direct fine-tuning on downstream tasks does not provide mechanisms to adapt to the specific data distributions of those tasks, and it only supports fixed model sizes. To address these challenges, we propose **Adaptive-Learngene**, a novel framework in which the ancestry model is trained solely on newly available data, and a new component, termed a learngene, is extracted and added to a global learngene pool that expands incrementally. This design enables a dynamically evolving pool of learngenes without requiring access to previous data. For each new downstream task, the Task-Adaptive Learngene Selector (TALS) retrieves a sparse combination of learngenes that best match to the data distribution of the target task. TALS requires only a small amount of downstream data for this selection, enabling descendant models of different sizes to be efficiently initialized and tailored to specific data distributions and resource constraints. Extensive experiments on diverse downstream tasks demonstrate that our method matches or outperforms existing approaches while offering superior scalability, adaptability, and efficiency in dynamic learning environments.

Introduction

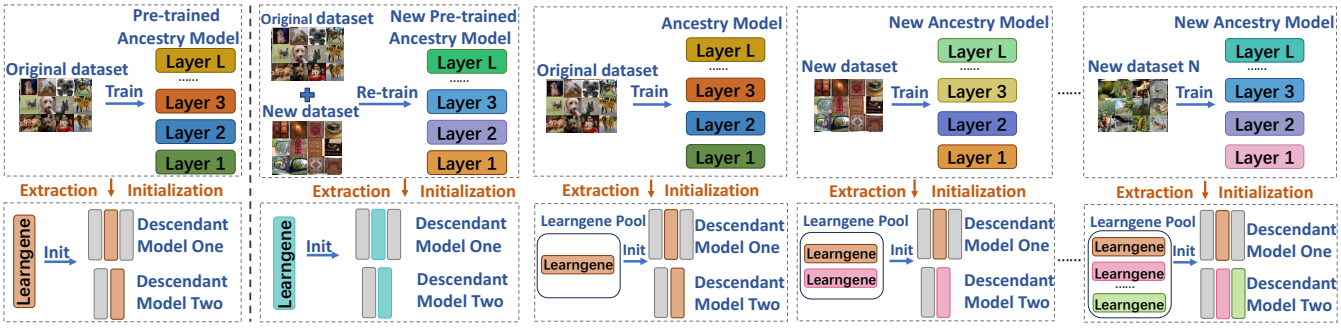
Pre-trained Vision Transformer (ViT) models have achieved remarkable success in various computer vision tasks, including image classification (Alexey 2020; Chen, Fan, and Panda 2021), segmentation (Strudel et al. 2021; Wu et al. 2025a), and detection (Zhang et al. 2021; Fang et al. 2021; Lin et al. 2024a). Model compression methods (Xu and McAuley 2023; Wang et al. 2024c; Zhang, Zhan, and Ye 2025) enable these models to be deployed on resource-constrained downstream tasks. However, both pre-training/fine-tuning and compression methods share core limitations: they are developed under the assumption of a static data environment, and each training yields only a single model with a fixed architecture and size. In dynamic real-world settings such as edge computing, where new pre-training data and downstream tasks continuously emerge, adapting to these changes typically requires retraining the model on both old and new data, followed by repeated compression or fine-tuning to obtain a new model of the desired size. This process incurs substantial computational overhead and hinders flexible deployment.

To address these challenges, the **Learngene** framework (Wang et al. 2022, 2023) is introduced, drawing inspiration from biological inheritance mechanisms in which genes encode and transmit the experiences of ancestors to their descendants. Within this paradigm, inheritable knowledge units, termed *learngenes*, are extracted from an ancestry model and used to initialize smaller descendant models of varying sizes for downstream tasks. A variety of methods have been explored for defining and extracting learngenes, often corresponding to different network modules and inheritance strategies. For instance, Heur-Learngene (Wang et al. 2022) heuristically selects the final layers of the ancestry model as learngenes based on gradient analysis and combines them with randomly initialized lower layers. Learngene Pool (Shi et al. 2024) distils multi-scale transformer layers into a learngene pool from which descendant models are assembled by stitching. PEG (Wang et al. 2024a) applies probabilistic sampling over self-attention and feed-forward to construct learngenes, followed by non-linear mapping to initialize descendant models.

Although existing learngene-based methods can generate descendant models of varying sizes to accommodate different computational resources, they still share a key limitation: the fixed, pre-trained model is used as the ancestry model which is trained on a static dataset. In practice, the data available for pre-training evolves over time. When new datasets arrive, the ancestry model must be retrained on both the new and previous data, followed by re-extraction of learngenes to update the model parameters (see Figure 1a). This process is computationally expensive and impractical, particularly in dynamic or privacy-sensitive environments. Furthermore, current methods lack effective mechanisms for selecting learngenes that are well aligned with the data distribution of new downstream tasks, often resulting in suboptimal descendant model initialization and degraded task performance.

To overcome these challenges, we propose **Adaptive-Learngene**, a framework that enables the continual evolution

*Co-corresponding auther.



(a) Existing Learngene methods

(b) Adaptive-Learngene

Figure 1: (a) Existing methods: Most use a fixed pre-trained ancestry model. When new data arrives, the model must be retrained on both previous and new data, followed by re-extraction of learngenes. (b) Adaptive-Learngene (ours): The ancestry model is trained only on the current data to extract new learngenes, which are added to the **learngene pool**. The descendant models can be initialized by selectively combining learngenes, enabling efficient adaptation to downstream tasks.

of the learngene pool in response to dynamically arriving datasets and supports data distribution-aware initialization of descendant models. As illustrated in Figure 1b, our approach trains the ancestry model exclusively on each newly available dataset, extracting the corresponding learngenes and adding them to a global **learngene pool**. This design allows the pool to be dynamically expanded with new knowledge without revisiting or storing previous data. When a new downstream task arises, descendant models are initialized by selectively retrieving relevant learngenes from the learngene pool, providing tailored initialization strategies for the specific requirements of the task.

Our method builds on recent findings (Lin et al. 2024c) demonstrating that ViT layers can be linearly decomposed into shared modular components across all layers. Leveraging this insight, we design the ancestry model architecture to flexibly incorporate new components into each layer as new data become available. This expandable structure enables the effective learning of new knowledge as learngenes. After training, the newly extracted learngenes are added to the learngene pool, which accumulates knowledge from previous and new data and can be used to initialize descendant models.

To facilitate data-aware initialization, we introduce the **Task-Adaptive Learngene Selector (TALS)**. TALS selects a sparse, relevant subset of learngenes from the pool based on a reward-driven mechanism that considers: (1) alignment of selected learngenes with the data distribution of the downstream task to improve the performance, and (2) a sparsity constraint to ensure efficiency. Notably, TALS requires only a small amount of downstream data to identify suitable learngenes. This allows for flexible and efficient adaptation of the descendant models to different downstream applications, providing a trade-off between accuracy and resource cost.

In summary, the main contributions of this work are:

- We propose an expandable ancestry model framework that can be trained on the current dataset to extract new learngenes, without requiring retraining on previous data.
- We develop a dynamic selection mechanism, TALS, that adaptively selects a subset of task-relevant learngenes to achieve computational efficiency. For example, selecting only **1** out of 10 learngenes on CIFAR10 achieves 95% of the full model's performance.
- Extensive experiments across diverse tasks validate that descendant models initialized using our method achieve comparable or superior performance to traditional approaches, demonstrating the generalizability and scalability of Adaptive-Learngene.

Related Work

Model Initialization Model initialization (Zhou 2016; Zhou and Tan 2024) plays a pivotal role in deep neural network training by influencing convergence and generalization. Classic schemes like Xavier (Glorot and Bengio 2010) and Kaiming (He et al. 2016) ensure stable gradient flow during early training. More recent approaches leverage large pre-trained models (He et al. 2020; Zhang et al. 2023; Yao, Zhang, and Xu 2025) to transfer general knowledge and provide strong starting points for downstream tasks.

However, pre-trained models are typically large and inflexible, limiting their applicability in dynamic or resource-constrained scenarios. Model compression techniques such as pruning, quantization, and knowledge distillation (Xu and McAuley 2023; Zhu et al. 2024; Peng et al. 2025) reduce model size but produce static models tied to specific datasets. Parameter-efficient tuning methods (Hu et al. 2021; Houlsby et al. 2019; Wu et al. 2025b; Lin et al. 2024b) reduce adaptation overhead but still depend on fixed backbones. When new pre-training data becomes available, existing approaches require retraining the model on the combined old and new data, followed by compression to transfer updated knowledge to downstream tasks.

These limitations highlight the need for more flexible initialization strategies that enable efficient training only on new data, without access to previous datasets, and support adaptable model sizes for downstream deployment.

Learngene The **Learngene** paradigm (Wang et al. 2022, 2023) extracts inheritable knowledge from an ancestry model

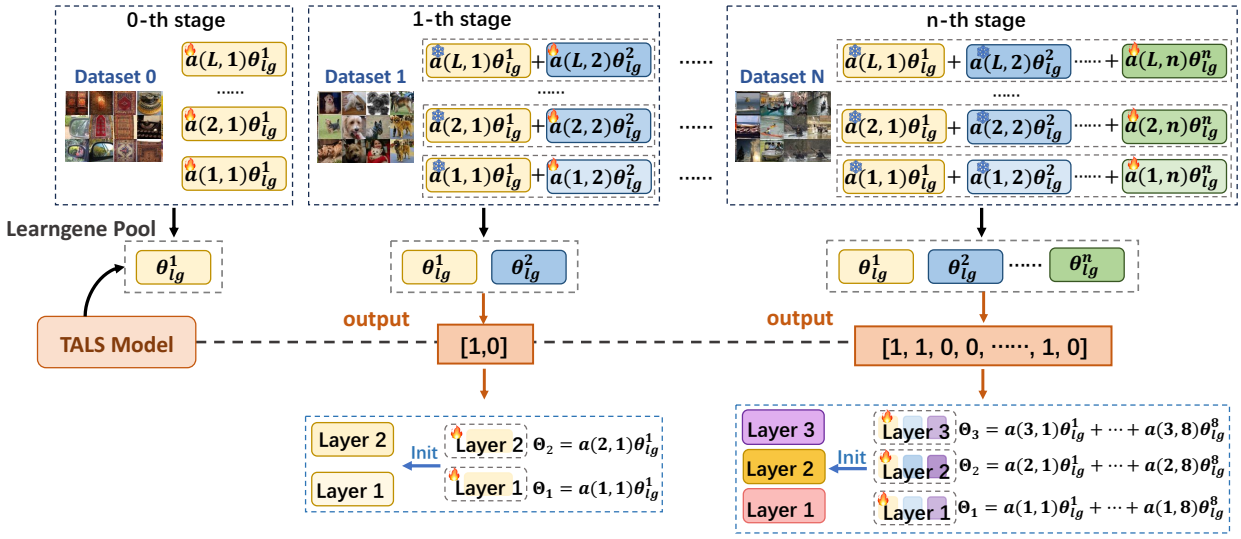


Figure 2: Framework. The ancestry model is trained incrementally across streaming datasets. For each dataset, a new learngene is introduced and shared across all layers of the model, with previous learngenes kept fixed. The parameters of each layer are represented as a polynomial-weighted linear combination of all accumulated learngenes. After training, the new learngene is added to a global learngene pool. When a downstream task arises, the Task-Adaptive Learngene Selector (TALS) selects a sparse combination of learngenes to initialize the descendant model, following the same composition method as in the ancestry model.

as learngenes to efficiently initialize smaller descendant models of various sizes. TLEG (Xia et al. 2024) employs an auxiliary model to identify learngenes from a fixed pre-trained transformer, while Learngene Pool (Shi et al. 2024) distills layers of different scales into a pool, enabling descendant models to be assembled by stitching together learngenes. Cluster-Learngene (Wang et al. 2024b) clusters attention heads and feed-forward networks as learngenes based on representational similarity and uses for descendant model initialization.

Although effective for knowledge transfer, these methods rely on the static, pre-trained ancestry model. When new data becomes available, updating learngenes requires full retraining of the ancestry model and re-extraction, which is computationally expensive. Furthermore, descendant models are constructed in a task-agnostic manner, without considering the specific data distributions of downstream tasks.

In contrast, our Adaptive-Learngene framework enables the ancestry model to learn from new datasets and extract corresponding learngenes without accessing previous data. These learngenes are added to a global pool that accumulates knowledge from all data and supports initialization of descendant models. For each downstream task, TALS selectively combines relevant learngenes to construct a tailored descendant model, enabling task-aware initialization.

Method

We propose Adaptive-Learngene, a framework with two key components: (1) an expandable ancestry model that learns from new data and extracts new learngenes without accessing previous data, while preserving existing learngenes; and (2) a Task-Adaptive Learngene Selector (TALS) that identifies optimal combinations of learngenes to initialize descendant models, as illustrated in Figure 2. We first describe the training and structure of the ancestry model, followed by the learngene selection strategy and the initialization of descendant models.

Expanding the Ancestry Model via Learngenes

To support dynamic training data environments, the ancestry model is expanded and trained only on newly arriving data. For each new dataset, a data-specific *learngene* is extracted to capture new knowledge and added to a global **learngene pool**. This approach enables continuous accumulation of knowledge while preserving previously acquired information without revisiting previous data.

The Structure of the Ancestry Model To facilitate learning in dynamic environments, we design the ancestry model as an expandable Vision Transformer (ViT) architecture that is continuously trained on new data. Inspired by (Lin et al. 2024c), which shows that each ViT layer can be represented as a linear combination of shared components across all layers. We extend this idea by adding a new cross-layer shared component to each layer of the ancestry model whenever new data arrives. This component is used to learn about the new dataset and serves as a dataset-specific **learngene**.

Formally, for the first dataset, a single learngene θ_1^{lg} is initialized and used across all layers:

$$\Theta_l = a(l,1) \cdot \theta_1^{lg}. \tag{1}$$

When the second new dataset arrives, a new learngene θ_2^{lg} is added to each layer. The parameters are then represented as:

$$\Theta_l = a(l,1) \cdot \theta_1^{lg} + a(l,2) \cdot \theta_2^{lg}. \tag{2}$$

This process continues iteratively for each new dataset. At n-th dataset, the layer parameters become:

$$\Theta_l = \sum_{n=1}^{N} a(l,n) \cdot \theta_n^{lg}, \tag{3}$$

where θ_{lg}^n is the n-th learngene and $a(l,n)$ is the predefined coefficient for layer l. Each learngene corresponds to a full ViT block, including Multi-Head Self Attention (MHSA), Feed-Forward Network (FFN), and Layer Normalization.

Training of Learngenes When a new dataset arrives, the ancestry model is expanded by adding a new learngene while keeping all previously learned learngenes fixed. For the first dataset, the ancestry model is initialized with a single learngene θ_{lg}^1 and optimized using standard cross-entropy loss:

$$\mathcal{L}_{\text{CE}} = -\frac{1}{N} \sum_{i=1}^{N} \sum_{c=1}^{C} y_{i,c} \log(\hat{y}_{i,c}), \tag{4}$$

where N is the number of training data, C is the number of classes, $y_{i,c}$ is the one-hot label, and $\hat{y}_{i,c}$ is the predicted probability.

For each new dataset, a new learngene θ_{lg}^n is introduced and optimized, while all existing $\theta_{\text{lg}}^1, \ldots, \theta_{\text{lg}}^{n-1}$ are frozen. This procedure ensures that only the new learngene adapts to the current dataset, while previous learngenes retain previously acquired knowledge. After training, the newly added learngene is stored in the global learngene pool.

By continually expanding the model with modular and dataset-specific learngenes, the ancestry model achieves incremental training without the need to access previous data, effectively addressing scalability and privacy concerns. Therefore, the accumulated learngene pool maintains knowledge of all datasets and provides a robust basis for initializing downstream descendant models.

Learngene Selection and Descendant Initialization

After constructing the learngene pool, the next step is to identify an optimal subset of learngenes to initialize a data-aware descendant model. Instead of heuristically reusing all extracted learngenes, we propose a learning-based selection strategy called the **Task-Adaptive Learngene Selector (TALS)**. TALS is designed to select a subset of learngenes that balances task accuracy and model efficiency.

TALS is implemented as a policy network that generates a binary vector $\mathcal{G} \in \{0,1\}^N$, indicating which of the N learngenes are selected. The goal is to learn a policy that maximizes task-specific accuracy while minimizing the number of active components, which is essential for adapting to downstream tasks with varying resource constraints.

To address issues of reward discontinuity and optimization instability, we adopt a smooth reward function inspired by the focal loss (Lin et al. 2017). Specifically, for a given selection $\mathcal{G}$ and data point (x_i, y_i), the reward is defined as:

$$r(\mathcal{G}|(x_i,y_i)) = (1-p_{y_i}(x_i;\theta_{\mathcal{G}}))^{\gamma} \cdot \log(p_{y_i}(x_i;\theta_{\mathcal{G}})) - \lambda \cdot \|\mathcal{G}\|_1, \tag{5}$$

where $p_{y_i}(x_i;\theta_{\mathcal{G}})$ denotes the softmax probability assigned to the ground-truth label y_i by the model initialized with the selected learngenes $\mathcal{G}$. The first term promotes confident and correct predictions while emphasizing hard examples; the second term penalizes the number of selected modules, encouraging sparsity. Here, $\gamma > 0$ controls the modulation strength, and λ trades off sparsity against accuracy.

We optimize the TALS policy using the REINFORCE algorithm (Williams 1992), a Monte Carlo policy gradient method. Let $\pi_\phi(\mathcal{G})$ denote the selection policy parameterized by ϕ. The optimization objective is to maximize the expected reward:

$$J(\phi) = \mathbb{E}_{\mathcal{G}\sim\pi_\phi}[r(\mathcal{G})] \tag{6}$$

with gradient estimated by:

$$\nabla_\phi J(\phi) = \mathbb{E}_{\mathcal{G}\sim\pi_\phi}\left[\nabla_\phi \log \pi_\phi(\mathcal{G}) \cdot (r(\mathcal{G}) - b)\right] \tag{7}$$

where b is a moving average baseline used to reduce variance.

Initialization of descendant models Once TALS selects a data-aware subset of learngenes $\mathcal{G}$, the descendant model is constructed using the same compositional principle as the ancestry model. Formally, for each transformer layer l, the layer parameters Θ_l are computed as:

$$\Theta_l = \sum_{n=1}^{N} \mathcal{G}_n \cdot a(l,n) \cdot \theta_n^{\text{lg}}, \tag{8}$$

where $\mathcal{G}_n \in \{0,1\}$ indicates whether the n-th learngene is selected, $a(l,n)$ is the predefined polynomial coefficient associated with layer l and learngene n.

This initialization enables descendant models to inherit knowledge from the ancestry model without requiring access to previous datasets. After initialization, the descendant models can be trained using two methods, similar to the methods proposed in (Lin et al. 2024c). The first one uses Equation 8 solely for initialization, then each layer are updated independently without adhering to the initial constraints. The second one maintains the constraints of Equation 8 during training, updating only the parameters of the learngenes used for initialization. This flexibility allows the descendant model to adapt to various computational budgets and generalization requirements in downstream tasks.

Experiment

Experimental Setup

Datasets. The ancestry model is trained on ImageNet-1K (Deng et al. 2009), which contains 1.2 million training and 50,000 validation images across 1,000 classes. To simulate a dynamic learning environment, we randomly partition the dataset into 10 non-overlapping subsets, each containing 100 classes. And the order is fixed across runs to emulate a consistent streaming setup.

To evaluate the transferability and generalization of learngenes, we fine-tune descendant models on a range of downstream tasks spanning different domains: (1) fine-grained classification: Oxford Flowers (Nilsback and Zisserman 2008), Stanford Cars (Gebru et al. 2017), and Food101 (Bossard, Guillaumin, and Van Gool 2014);

Model	Method	iNat-2019	CIFAR100	CIFAR10	Food-101	Stanford Cars	Flowers
Tiny	Pre-Fine	58.12	80.81	96.65	83.24	75.71	84.79
	From Scratch	37.16	67.44	88.30	61.54	67.32	68.82
	Heur-Learngene	41.55	70.19	91.66	72.54	70.68	78.67
	Auto-Learngene	52.46	75.83	93.02	79.12	74.20	80.84
	PEG	56.80	79.85	96.33	85.37	71.75	87.85
	TLEG	55.64	78.66	95.32	82.80	-	-
	Adapt-Learngene	**58.74** (↑ 21.58)	**80.03** (↑ 12.59)	**96.76** (↑ 8.46)	**86.12** (↑ 24.58)	**78.57** (↑ 11.25)	82.62 (↑ 13.80)
Small	Pre-Fine	68.48	84.43	97.59	87.80	86.81	91.13
	From Scratch	50.79	73.32	92.49	74.64	71.63	72.91
	Heur-Learngene	53.21	78.13	93.12	77.09	81.52	82.84
	Auto-Learngene	59.92	79.49	93.58	80.25	84.98	87.02
	PEG	67.73	83.59	97.38	87.15	82.57	91.01
	TLEG	66.70	83.64	97.68	87.27	-	-
	Adapt-Learngene	**69.11** (↑ 18.32)	**84.06** (↑ 10.74)	**97.88** (↑ 5.39)	87.21 (↑ 12.57)	**86.82** (↑ 15.19)	89.59 (↑ 16.68)

Table 1: Comparison of different methods on different classification datasets. All models are based on the Deit-Tiny and DeiT-Small and the number of parameters is the same for all methods. In the Adaptive-Learngene column, the bolded results indicate that our method outperforms the baseline Learngene methods. The arrows indicate the performance improvement over "From Scratch". And "-" in TLEG indicates that the corresponding dataset was not included in the original paper.

(2) general object classification: CIFAR-10 and CIFAR-100 (Krizhevsky, Hinton et al. 2009); (3) large-scale classification: ImageNet-1K; (4) long-tailed recognition: iNaturalist-2019 (iNat-2019) (Zhou et al. 2020); (5) semantic segmentation: ADE20K (Zhou et al. 2017). This diverse benchmark allows us to evaluate both in-domain and cross-domain generalization capabilities of the proposed method.

Training Settings. The ancestry model is trained for 300 epochs per dataset subset. Descendant models are fine-tuned for 50 epochs on ImageNet-1K and 500 epochs on other classification datasets, except iNat-2019 (100 epochs, following (Wang et al. 2024a)). For ADE20K, models are trained for 50 epochs. In ablation studies, all descendant models are trained for 100 epochs for consistency.

For TALS, we sample 10% of the downstream task's training set as the reward evaluation set to estimate the data distribution. We set the parameter $\gamma = 2.0$ and the sparsity trade-off coefficient $\lambda = 0.01$. A moving average baseline is used to reduce the variance of the gradient.

Architectures. We adapt DeiT (Touvron et al. 2021) as the backbone for both ancestry and descendant models, including DeiT-Tiny/-Small/Base variants. Each learngene corresponds to a transformer block (comprising MSA, FFN, and MLP modules). Descendant models are composed of variable-depth stacks of such blocks, initialized by learngenes selected from the learngene pool by TALS. To further validate the generality of our method, we also conduct experiments on the Swin Transformer (Liu et al. 2021) as backbone.

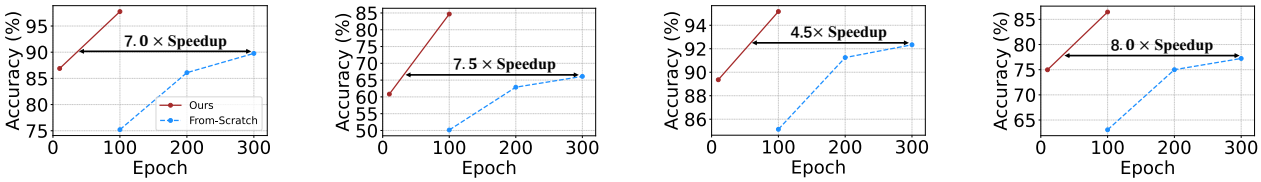
Baselines. We compare our method with the following representative baselines: (i) *From-Scratch*: The model is randomly initialized and trained directly on the downstream datasets. (ii) *Pre-Fine*: The DeiT model is pre-trained on the full ImageNet-1K and fine-tuned on each downstream task. (iii) *Heur-Learngene* (Wang et al. 2022): Learngenes are defined as the last three blocks of a pre-trained ancestry model, and combined with randomly initialized lower layers. (iv)*Auto-Learngene*: The layers treated as learngene are extracted from the ancestry model by meta learning, and then stacked with randomly initialized higher layers to initialize the descendant models. (v) *PEG* (Wang et al. 2024a): Learngenes are sampled via a probabilistic mixture of MSA and FFN layers from the ancestry model. (vi) *TLEG* (Xia et al. 2024): An auxiliary model is trained via distillation to extract learngenes, which are then used to initialize the descendant model. All baselines follow their original hyperparameter settings.

Results and Analyses

In this section, we first compare the descendant models initialized using our method with several baselines, including pre-trained models, learning from scratch, and previous learngene-based methods, on various downstream datasets. In addition, we perform ablation studies to validate the effectiveness of the learngene combinations selected by the proposed Task-Adaptive Learngene Selector.

Flexibility and Effectiveness of Descendant Models Downstream applications often require descendant models of varying scales to meet resource constraints. To evaluate the flexibility of our framework, we construct descendant models of different depths by learngenes selected from the global pool and fine-tuning them on the full ImageNet-1K. We experiment with DeiT-Tiny, DeiT-Small, and DeiT-Base backbones under multiple layer configurations. Table 2 reports comparisons with From-Scratch, PEG, and TLEG. Our method consistently outperforms all baselines across 13 configurations. In particular, for the 9-layer DeiT-Small model, it achieves a **7%** gain over training from scratch.

These results show that our method effectively inherits knowledge from the continually trained ancestry model and enables strong initialization even for computationally constrained descendant models. Unlike methods that rely on statically pre-trained ancestry models, our adaptive framework evolves with incoming data, improving adaptability to distribution shifts. Moreover, descendant models can be assem-



(a) CIFAR10/DeiT-Base (b) CIFAR100/DeiT-Base (c) CIFAR10/Swin-Tiny (d) CIFAR100/Swin-Tiny

Figure 3: Convergence comparison between Adapt-Learngene and From-Scratch training on DeiT and Swin-Transformer. Each plot shows model accuracy over epochs.

Model	L	From-Scratch	PEG	TLEG	Ours
Tiny	6	58.16	57.92	58.2	**58.43**
	9	60.58	60.45	62.5	**62.65**
	12	61.44	61.55	65.4	**65.68**
Small	6	64.91	64.94	69.5	**70.21**
	9	67.02	69.49	73.2	**74.22**
	12	68.56	70.24	75.1	**75.92**
Base	6	73.73	73.98	76.2	**76.67**
	7	74.42	74.63	77.3	**77.72**
	8	76.14	76.19	78.1	**78.41**
	9	76.46	76.82	78.7	**79.00**
	10	76.81	76.95	79.1	**79.22**
	11	77.03	77.16	79.6	**79.89**
	12	77.22	77.39	79.9	**79.99**

Table 2: Comparison with different methods on the ImageNet-1K. The first column is the model type (DeiT-Tiny/Small/Base), while the second column is the number of layers. The "From Scratch" method, the models are trained for 100 epochs following the settings in (Wang et al. 2024a).

bled directly from learngenes, without additional full-model training, enabling efficient deployment without additional full-model training.

Generalization of Learngene Across Image Classification and Segmentation Tasks To evaluate whether the learngenes extracted from the ancestry model can generalize across different data domains and tasks, we evaluate learngenes-initialized descendant models on a variety of classification and segmentation datasets.

Image Classification Task. We evaluate our method on six classification datasets, comparing it against a diverse set of initialization baselines: (i) From-Scratch, (ii) Pre-Fine, and (iii) Learngene-based methods including Heur-Learngene, Auto-Learngene, PEG, and TLEG. We use both DeiT-Tiny and DeiT-Small as backbones with the same parameter and training settings across all methods.

As shown in Table 1, Adapt-Learngene consistently outperforms From-Scratch, Heur-Learngene, and Auto-Learngene by a large margin. On iNat-2019, it achieves over **21%** improvement (Tiny) and **18%** (Small) over From-Scratch. Compared with PEG and TLEG, Adapt-Learngene performs better or comparably in most cases. Remarkably, it also matches or exceeds Pre-Fine, which is pretrained on full ImageNet and thus serves as an upper bound. These results highlight the generalizability of learngenes extracted from our continually

Method	Pre-Fine	Heur-LG	PEG	Adapt-Learngene
mIoU	47.08	40.12	48.30	**55.63**

Table 3: Comparison with baselines on the ADE20K dataset. The table reports the mean Intersection over Union (mIoU) scores for each method.

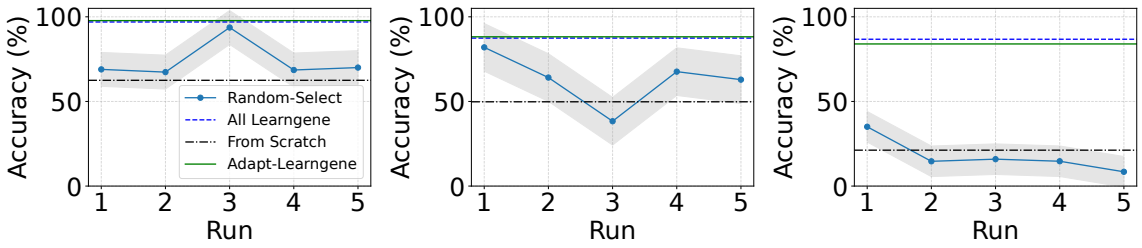
trained ancestry model. Without revisiting previous datasets, our method yields highly effective data-aware initialization, offering clear advantages in dynamic or privacy-constrained learning scenarios.

Image Segmentation Task. To assess the applicability of our method, we evaluate cross-task generalization on the ADE20K semantic segmentation benchmark. We adopt the Segmenter architecture (Strudel et al. 2021) and replace its ViT encoder layers with those composed of learngenes from learngene pool. The segmentation decoder (mask transformer) is randomly initialized and trained from scratch.

We compare our method against Pre-Fine, Heur-Learngene, and PEG. All models are trained for 50 epochs under the same configuration. As reported in Table 3, our method achieves the highest performance with 55.63 mIoU, surpassing PEG by **7.3** points and outperforming Pre-Fine by **8.5** points. These results confirm the strong cross-task generalization capability of the learngene pool and highlight the benefit of incremental training of the ancestry model in producing robust and transferable learngenes.

Faster Convergence To further evaluate the efficiency of our method during training, we compare the convergence speed of models initialized with our Adapt-Learngene method against those trained from scratch. As shown in Figure 3, descendant models initialized with selected learngenes reach high accuracy significantly faster than those trained without initialization.

For instance, on the CIFAR100 dataset with a DeiT-Base backbone, our method achieves 84.67% accuracy within 100 epochs, while the scratch-trained model requires nearly 300 epochs to reach comparable performance—yielding a 7.5× speedup. Similarly, on Swin-Tiny for CIFAR100, our method achieves over 85% accuracy eight times faster than the baseline. These consistent improvements across datasets and architectures demonstrate that Adapt-Learngene not only improves generalization but also dramatically accelerates training, making it highly suitable for scenarios with limited compute budgets or time constraints.



(a) CIFAR10/Deit-Small (b) Food101/Deit-Small (c) Flowers/Deit-Small

(d) CIFAR10/Deit-Base (e) Food101/Deit-Base (f) Flowers102/Deit-Base

Figure 4: Ablation Study. All models are based on the DeiT-Base/Small initialized with 5 learngenes. "Random-Select" is repeated five times, with shaded areas indicating the standard deviation.

Ablation Study on the Selection of Learngenes

To further evaluate the effectiveness of the TALS-selected learngene combinations, we conduct two ablation studies. The first compares descendant models initialized with learngene combinations selected by TALS to those initialized with randomly selected learngenes (*Random-Select*) and those trained from scratch. The second investigates how many learngenes are required to achieve 95% of the performance of descendant models initialized with all learngenes.

Effectiveness of Learngenes Selected by the TALS Model
To assess the effectiveness of TALS in selecting informative learngene subsets, we conduct an ablation experiment where each descendant model is initialized with 5 learngenes selected from a pool of 10. We compare TALS with a random selection strategy ("Random-Select") where each random selection is repeated five times. As shown in Figure 4, our method significantly outperforms those initialized with "Random-Select" and the models learning from scratch. In addition, the "Random-Select" method shows high variability in performance, with a variance of **18** on CIFAR10. Moreover, on Flowers102, the random strategy performs even worse than training from scratch, underlining its instability. In contrast, TALS maintains performance close to using all learngenes, demonstrating its ability to identify data-aware learngenes that enable efficient and robust initialization.

In practical scenarios with limited computational budgets, the selection of a minimal but effective subset of learngenes is critical. TALS is designed to identify such subsets by balancing data awareness and sparsity. As shown in Table 4, descendant models achieve at least 95% of the performance of models initialized with the full learngene pool, while using only a fraction of the components. For example, on CIFAR-10 with the DeiT-Small architecture, selecting 3 out of 10 learngenes achieves **98.33%** of the full accuracy while reducing the number of parameters from 22M to about 6.6M, saving nearly **70%** of the model size. Similarly, under the DeiT-Base architecture, selecting only one learngene maintains over 95% of the full accuracy while reducing the parameter count from 86M to approximately 8.6M, achieving a **90%** reduction. These results demonstrate that TALS can effectively identify highly task-relevant learngenes, enabling efficient model initialization with a significant reduction in parameter footprint without sacrificing accuracy.

Model	Metric	CIFAR10	CIFAR100	Food-101	iNat-2019
Small	All	95.79	82.59	87.42	70.96
	Num	3	4	3	3
	Ours	94.19	79.49	85.44	67.48
Base	All	98.12	85.88	88.66	73.12
	Num	1	2	2	3
	Ours	97.05	83.48	87.99	71.89

Table 4: The results of the number of learngenes selected and the accuracy of the models, achieving an overall accuracy of 95% of the descendant models initialized by all learngenes.

Conclusion

We present Adaptive-Learngene, a biologically inspired framework enabling continual expansion of knowledge and task-aware initialization. An expandable ViT serves as the ancestry model to integrate new data while preserving prior knowledge, forming a modular learngene pool. The proposed Task-Adaptive Learngene Selector selects sparse, task-relevant learngenes to balance performance and efficiency. Together, these components support scalable, flexible, and data-aware model initialization, offering an effective solution for dynamic and resource-constrained learning environments.

Acknowledgments

This research was supported by the Jiangsu Science Foundation (BK20243012, BG2024036), the National Science Foundation of China (62125602, U24A20324, 92464301), and the Fundamental Research Funds for the Central Universities (2242025K30024). This work was also partially supported by the Southeast University Kunpeng & Ascend Center of Cultivation and the Big Data Computing Center of Southeast University.

References

Alexey, D. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv: 2010.11929*.

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

Chen, C.-F. R.; Fan, Q.; and Panda, R. 2021. Crossvit: Cross-attention multi-scale vision transformer for image classification. In *Proceedings of the IEEE/CVF international conference on computer vision*, 357–366.

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

Fang, Y.; Liao, B.; Wang, X.; Fang, J.; Qi, J.; Wu, R.; Niu, J.; and Liu, W. 2021. You only look at one sequence: Rethinking transformer in vision through object detection. *Advances in Neural Information Processing Systems*, 34: 26183–26197.

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

Glorot, X.; and Bengio, Y. 2010. Xavier initialization. *J. Mach. Learn. Res*.

He, K.; Fan, H.; Wu, Y.; Xie, S.; and Girshick, R. 2020. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 9729–9738.

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

Houlsby, N.; Giurgiu, A.; Jastrzebski, S.; Morrone, B.; De Laroussilhe, Q.; Gesmundo, A.; Attariyan, M.; and Gelly, S. 2019. Parameter-efficient transfer learning for NLP. In *International conference on machine learning*, 2790–2799. PMLR.

Hu, E. J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; and Chen, W. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

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

Lin, J.; Wu, Y.; Wang, Z.; Liu, X.; and Guo, Y. 2024a. PairID: A dual modal framework for identity preserving image generation. *IEEE Signal Processing Letters*.

Lin, J.; Zhao, G.; Xu, J.; Wang, G.; Wang, Z.; Dantcheva, A.; Du, L.; and Chen, C. 2024b. DiffTV: Identity-preserved thermal-to-visible face translation via feature alignment and dual-stage conditions. In *Proceedings of the 32nd ACM International Conference on Multimedia*, 10930–10938.

Lin, S.; Zhang, M.; Chen, R.; Yang, X.; Wang, Q.; and Geng, X. 2024c. Linearly Decomposing and Recomposing Vision Transformers for Diverse-Scale Models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Lin, T.-Y.; Goyal, P.; Girshick, R.; He, K.; and Dollár, P. 2017. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, 2980–2988.

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

Nilsback, M.-E.; and Zisserman, A. 2008. Automated flower classification over a large number of classes. In *2008 Sixth Indian conference on computer vision, graphics & image processing*, 722–729. IEEE.

Peng, Y.; Zhang, G.; Zhang, M.; You, Z.; Liu, J.; Zhu, Q.; Yang, K.; Xu, X.; Geng, X.; and Yang, X. 2025. Lmm-r1: Empowering 3b lmms with strong reasoning abilities through two-stage rule-based rl. *arXiv preprint arXiv:2503.07536*.

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

Strudel, R.; Garcia, R.; Laptev, I.; and Schmid, C. 2021. Segmenter: Transformer for semantic segmentation. In *Proceedings of the IEEE/CVF international conference on computer vision*, 7262–7272.

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

Wang, Q.; Yang, X.; Chen, H.; and Geng, X. 2024a. Vision Transformers as Probabilistic Expansion from Learngene. In *Forty-first International Conference on Machine Learning*.

Wang, Q.; Yang, X.; Feng, F.; Geng, X.; et al. 2024b. Cluster-Learngene: Inheriting Adaptive Clusters for Vision Transformers. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

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

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

Wang, Y.; Cheng, L.; Duan, M.; Wang, Y.; Feng, Z.; and Kong, S. 2024c. Improving knowledge distillation via regularizing feature direction and norm. In *European Conference on Computer Vision*, 20–37. Springer.

Williams, R. J. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8: 229–256.

Wu, F.; Cheng, L.; Tang, S.; Zhu, X.; Fang, C.; Zhang, D.; and Wang, M. 2025a. Navigating Semantic Drift in Task-Agnostic Class-Incremental Learning. *arXiv preprint arXiv:2502.07560*.

Wu, Y.; Zhou, Y.; Ziheng, Z.; Peng, Y.; Ye, X.; Hu, X.; Zhu, W.; Qi, L.; Yang, M.-H.; and Yang, X. 2025b. On the generalization of sft: A reinforcement learning perspective with reward rectification. *arXiv preprint arXiv:2508.05629*.

Xia, S.; Zhang, M.; Yang, X.; Chen, R.; Chen, H.; and Geng, X. 2024. Transformer as Linear Expansion of Learngene. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 16014–16022.

Xu, C.; and McAuley, J. 2023. A survey on model compression and acceleration for pretrained language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, 10566–10575.

Yao, J.; Zhang, Z.; and Xu, Z.-Q. J. 2025. An Analysis for Reasoning Bias of Language Models with Small Initialization. *arXiv preprint arXiv:2502.04375*.

Zhang, Y.-K.; Huang, T.-J.; Ding, Y.-X.; Zhan, D.-C.; and Ye, H.-J. 2023. Model spider: Learning to rank pre-trained models efficiently. *Advances in Neural Information Processing Systems*, 36: 13692–13719.

Zhang, Y.-K.; Zhan, D.-C.; and Ye, H.-J. 2025. Capability Instruction Tuning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 25958–25966.

Zhang, Z.; Lu, X.; Cao, G.; Yang, Y.; Jiao, L.; and Liu, F. 2021. ViT-YOLO: Transformer-based YOLO for object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, 2799–2808.

Zhou, B.; Cui, Q.; Wei, X.-S.; and Chen, Z.-M. 2020. Bbn: Bilateral-branch network with cumulative learning for long-tailed visual recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 9719–9728.

Zhou, B.; Zhao, H.; Puig, X.; Fidler, S.; Barriuso, A.; and Torralba, A. 2017. Scene parsing through ade20k dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 633–641.

Zhou, Z.-H. 2016. Learnware: on the future of machine learning. *Frontiers Comput. Sci.*, 10(4): 589–590.

Zhou, Z.-H.; and Tan, Z.-H. 2024. Learnware: Small models do big. *Science China Information Sciences*, 67(1): 112102.

Zhu, X.; Li, J.; Liu, Y.; Ma, C.; and Wang, W. 2024. A survey on model compression for large language models. *Transactions of the Association for Computational Linguistics*, 12: 1556–1577.