A. Universality of WAVE

Proposition 1. *Heur-LG [36], Auto-LG [37] and TLEG [40] are special cases of WAVE.*

Proof. To prove Proposition 1, we establish a correspondence between weight templates and ViT layers since the learngenes in Heur-LG, Auto-LG and TLEG are structured in the form of ViT layers.

Consider the set of weight templates:

$$\mathcal{T} = \{T_{qkv}^{(1\sim N_{qkv})}, T_{o}^{(1\sim N_{o})}, T_{in}^{(1\sim N_{in})}, T_{out}^{(1\sim N_{out})})\}$$

These templates can be used to construct N_l learngene layers in a ViT model:

$$\mathcal{G} = \{G_{qkv}^{(1\sim N_l)}, G_{o}^{(1\sim N_l)}, G_{in}^{(1\sim N_l)}, G_{out}^{(1\sim N_l)}\}$$

via the following operation:

$$G_{\star}^{(l)} = \sum_{t=1}^{s_1 s_2} T_{\star}^{(s_1 s_2 \cdot (l-1)+t)} \otimes \mathring{\mathbb{1}}_{(i,j)} \tag{A.1}$$

where $G_{\star}^{(l)} \in \mathbb{R}^{M_1 \times M_2}$ is the constructed weight matrix for the l-th layer of type $\star$. $T_{\star}^{(t)} \in \mathbb{R}^{w_1 \times w_2}$ is the t-th weight template of type $\star$, where $\star \in \{qkv, o, in, out\}$. $\mathring{\mathbb{1}}_{(i,j)} \in \mathbb{R}^{s_1 \times s_2}$ is a padding matrix with 1 at position (i, j) and 0 elsewhere, where $s_1 = \frac{M_1}{w_1}$ and $s_2 = \frac{M_2}{w_2}$.

The indices i and j in $\mathring{\mathbb{1}}_{(i,j)}$ are calculated as follows:

$$i = \lfloor \frac{t-1}{s_2} \rfloor, \quad j = (t-1) \bmod s_2 \tag{A.2}$$

Now, consider a ViT model with L_{tar} layers whose weight matrices are

$$\theta_{\text{tar}} = \{W_{qkv}^{(1\sim L_{\text{tar}})}, W_{o}^{(1\sim L_{\text{tar}})}, W_{in}^{(1\sim L_{\text{tar}})}, W_{out}^{(1\sim L_{\text{tar}})}\}$$

We will demonstrate how Heur-LG, Auto-LG, and TLEG can be represented as special cases of WAVE:

- **Heur-LG** extracts the last N_l layers from a pre-trained model and then stacks randomly initialized layers $R_{\star}$ in the lower layers to construct target models:

$$W_{\star}^{(l)} = \begin{cases} R_{\star}^{(l)} & l < L_{\text{tar}} - N_l \\ G_{\star}^{(N_l + l - L_{\text{tar}})} & l \geq L_{\text{tar}} - N_l \end{cases} \tag{A.3}$$

- **Auto-LG** extracts the first N_l layers from a pre-trained model and then stacks randomly initialized layers $R_{\star}$ in the higher layers to construct target models:

$$W_{\star}^{(l)} = \begin{cases} G_{\star}^{(l)} & l \leq N_l \\ R_{\star}^{(l)} & l > N_l \end{cases} \tag{A.4}$$

- **TLEG** adopts linear expansion on two shared learngene layers $G_{\star}^{(\mathcal{A})}$ and $G_{\star}^{(\mathcal{B})}$:

$$W_{\star}^{(l)} = G_{\star}^{(\mathcal{A})} + \frac{l}{L_{\text{tar}}} G_{\star}^{(\mathcal{B})} \tag{A.5}$$

Algorithm 1 Integration of Structured Size-agnostic Knowledge into Weight Templates

Input: Training dataset $\{(x^{(i)}, y^{(i)})\}_{i=1}^{m}$, number of epochs N_{ep}, batch size B, learning rate η, pre-trained model (i.e., ancestry model) f_{pre}, auxiliary model f_{aux} with weight matrices θ_{aux}
Output: Weight Templates $\mathcal{T}$

```
Randomly initialize weight matrices θ_aux, weight templates T and weight scalers S
for ep = 1 to N_ep do
    for each batch {(x_i, y_i)}_{i=1}^{B} do
        Update θ_aux with T and S under the rule of Eq. (4)
        For each x_i, forward propagate ŷ_i = f_aux(x_i)
        Calculate L_batch = (1/B) Σ_{i=1}^{B} L(ŷ_i, y_i)
        Backward propagate L(ŷ_i, y_i) to compute the gradients with respect to T and S: ∇_T L_batch, ∇_S L_batch
        Update T and S :
            T := T − η · ∇_T L_batch
            S := S − η · ∇_S L_batch
    end for
end for
```

The above formulations demonstrate that Heur-LG, Auto-LG, and TLEG are all specific cases of WAVE, where different rules are applied to concatenate and weight these templates in $\mathcal{T}$.

B. Initialization of Weight Scalers

For the initialization of $\mathcal{S}$, we simulate linear initialization [40] and Net2Net [5], and propose **linear padding initialization** to better preserve the structured knowledge of original weight templates during initialization, thereby providing a suitable starting point for target networks.

For a target network with L_{tar} layers, we consider its weight matrix $W_{\star}^{(l)} \in \mathbb{R}^{M_1 \times M_2}$ and corresponding weight templates $T_{\star}^{(1\sim N_{\star})} \in \mathbb{R}^{w_1 \times w_2}$, where $M_1 > w_1$ and $M_2 > w_2$. The corresponding $S_{\star}^{(l,t)} \in \mathbb{R}^{s_1 \times s_2}$ is initialized as:

$$S_{\star}^{(l,t)} = \alpha_t \cdot \mathring{\mathbb{1}}_{(i,j)} + \epsilon \mathcal{N}(\mu, \sigma^2) \tag{B.1}$$

Here, $\alpha_t = \begin{cases} 1, & \text{if } t \leq \frac{N_{\star}}{2} \\ \frac{l}{L_{\text{tar}}}, & \text{otherwise} \end{cases}$ is a linear weight and $\mathring{\mathbb{1}}_{(i,j)} \in \mathbb{R}^{s_1 \times s_2}$ is a padding matrix with 1 at (i, j) and 0 elsewhere. The indices (i, j) are given by $i = \lfloor \frac{(t-1) \bmod (s_1 \times s_2)}{s_2} \rfloor$, $j = (t-1) \bmod s_2$ with $s_i = \frac{M_i}{w_i}$. ϵ denotes a small value (e.g., 10^{-6}) and $\mathcal{N}(\mu, \sigma^2)$ represents Gaussian noise.

C. Training Details

C.1. Details of Knowledge Integration

Algorithm 1 presents the pseudo code for integrating structured size-agnostic knowledge into weight templates (i.e.,

Table C.1. Hyper-parameters for WAVE integrating structured knowledge on ImageNet-1K.

Training Settings	Configuration
optimizer	AdamW
base learning rate	Ti: 5e-4 | S: 2.5e-4 | B: 1.25e-4
warmup learning rate	1e-6
weight decay	0.05
optimizer momentum	0.9
batch size	Ti: 512 | S: 256 | B: 128
training epochs	150
learning rate schedule	cosine decay
warmup epochs	5
color jitter	0.4
auto augment	rand-m9-mstd0.5-inc1
mixup	0.8
cutmix	1.0
label smoothing	0.1
drop path	0.1

learngenes).

C.2. Hyper-parameters

Table C.1 and Table C.4 present the basic settings, including batch size, warmup epochs, training epochs and other settings for WAVE integrating structured common knowledge into weight templates and training the models initialized with weight templates on various datasets, respectively.

C.3. Details of Weight Templates

Table C.2 presents a detailed overview of weight templates utilized in auxiliary models for integrating structured, size-agnostic knowledge from the ancestry model.

C.4. Details of Downstream Datasets

Additional datasets include Oxford Flowers [24], CUB-200-2011 [34], Stanford Cars [11], CIFAR-10, CIFAR-100 [19], Food-101 [3], and iNaturalist-2019 [29]. Table C.3 presents the details of seven downstream datasets, which are sorted by the size of datasets.

D. Additional Results

D.1. Integration of Knowledge from Larger Pre-trained Models

Weight templates enable structured integration of knowledge from pre-trained ancestor models while filtering out size-specific information that violates the constraints in Eq. (4). This mechanism ensures effective transfer and sharing of size-agnostic knowledge across models of varying sizes.

To evaluate the influence of ancestor models with different architectures and sizes, we incorporate a larger pre-trained model, RegNet-16GF (83.6M) [25], and compare it with WAVE and TLEG, both using LeVit-384 (39.1M) as the ancestor model. Table D.1 presents the results on DeiT-Ti and DeiT-S across various layers.

The results demonstrate that WAVE consistently outperforms TLEG across all model sizes. While the larger ancestor model (RegNet-16GF) provides some improvements, the gains remain limited, suggesting that once a pre-trained model is sufficiently trained and informative, the shared size-agnostic knowledge remains stable. These findings underscore WAVE's robustness in effectively condensing and integrating knowledge from diverse ancestor models.

Table C.2. Configuration of weight templates. $l \times w$ @ n represents that the weight templates of corresponding weight matrices are composed of n templates with the size $l \times w$.

	DeiT-Ti	DeiT-S	DeiT-B
W_{qkv}	192×192 @ 6	384×384 @ 6	768×768 @ 6
W_{o}	192×192 @ 2	384×384 @ 2	768×768 @ 2
W_{in}	192×192 @ 8	384×384 @ 8	768×768 @ 8
W_{out}	192×192 @ 8	384×384 @ 8	768×768 @ 8
W_{norm_1}	192 @ 4	384 @ 4	768 @ 4
W_{norm_2}	192 @ 4	384 @ 4	768 @ 4
$W_{qkv}^{(\text{bias})}$	576 @ 4	1152 @ 4	2304 @ 4
$W_{o}^{(\text{bias})}$	192 @ 4	384 @ 4	768 @ 4
$W_{in}^{(\text{bias})}$	768 @ 4	1536 @ 4	3702 @ 4
$W_{out}^{(\text{bias})}$	192 @ 4	384 @ 4	768 @ 4
$W_{norm_1}^{(\text{bias})}$	192 @ 4	384 @ 4	768 @ 4
$W_{norm_2}^{(\text{bias})}$	192 @ 4	384 @ 4	768 @ 4
W_{irpe_q}	1×64×49 @ 6	1×64×49 @ 6	1×64×49 @ 6
W_{irpe_k}	1×64×49 @ 6	1×64×49 @ 6	1×64×49 @ 6
W_{irpe_v}	1×49×64 @ 6	1×49×64 @ 6	1×49×64 @ 6

Table C.3. Characteristics of downstream datasets.

Dataset	Classes	Total	Training	Testing
Oxford Flowers [24]	102	8,189	2,040	6,149
CUB-200-2011 [34]	200	11,788	5,994	5,794
Stanford Cars [11]	196	16,185	8,144	8,041
CIFAR10 [19]	10	60,000	50,000	10,000
CIFAR100 [19]	100	60,000	50,000	10,000
Food101 [3]	101	101,000	75,750	25,250
iNat-2019 [29]	1010	268,243	265,213	3,030

D.2. Initialization of Deeper Models

We extend our experiments (Table 1) to deeper models with 24 and 36 layers across different widths (W_{192}, W_{384}, and W_{768}). Table D.2 shows that WAVE consistently outperforms He-Init [6] and TLEG [40], achieving higher accuracy across all settings.

For 24-layer models, WAVE surpasses TLEG by 0.8%, 0.7%, and 1.8%, with even greater improvements at 36 layers, confirming its effectiveness in deeper architectures. These results demonstrate WAVE's ability to maintain initialization quality as model depth increases, leveraging size-agnostic weight templates to ensure stable parameter inher-

Table C.4. Hyper-parameters for neural networks trained on downstream datasets.

Dataset	Batch Size	Epoch	Learning Rate	Drop Last	Warmup Epochs	Droppath Rate	Color Jitter	Auto Augment	Random Rrase	Mixup	Cutmix	Scheduler	Optimizer
Oxford Flowers	512	300	3e-4	False	0	0	0.4	rand-m9-mstd0.5-inc1	0.25	0	0	cosine	AdamW
CUB-200-2011	512	300	3e-4	False	0	0.1	0		0.25	0	0	cosine	AdamW
Stanford Cars	512	300	3e-4	False	0	0.1	0		0.25	0	0	cosine	AdamW
CIFAR10	512	300	5e-4	True	0	0.1	0.4		0.25	0	0	cosine	AdamW
CIFAR100	512	300	5e-4	True	0	0.1	0.4		0.25	0	0	cosine	AdamW
Food101	512	300	5e-4	True	0	0.1	0.4		0.25	0	0	cosine	AdamW
iNat-2019	512	100	5e-4	True	0	0.1	0.4		0.25	0	0	cosine	AdamW

Table D.1. Additional results on different ancestry models.

	Ancestry	DeiT-Ti L_4	DeiT-Ti L_8	DeiT-Ti L_{12}	DeiT-S L_4	DeiT-S L_8	DeiT-S L_{12}
TLEG [40]	LeVit-384 (39.1M)	55.0	62.9	65.4	65.4	72.1	73.8
WAVE	LeVit-384 (39.1M)	58.6*	65.4*	**67.3**	**68.9**	**74.1**	75.3*
WAVE	RegNet-16GF (83.6M)	**58.7**	**65.7**	67.0*	68.7*	74.0*	**75.4**

Table D.2. Results on initializing deeper models.

			W_{192} L_{24}	W_{192} L_{36}		W_{384} L_{24}	W_{384} L_{36}		W_{768} L_{24}	W_{768} L_{36}
	Epoch	Para.	11.3	16.7	Para.	43.6	65.0	Para.	171.8	257.0
He-Init [6]	0	0	52.4	53.6	0	57.8	58.2	0	59.2	59.4
TLEG [40]	150	1.3	68.1	68.7	4.3	74.8	75.3	15.7	77.6	77.5
WAVE	150	1.3	**68.9**	**69.5**	4.4	**75.5**	**76.6**	15.8	**79.4**	**79.6**
			↑0.8	↑0.8		↑0.7	↑1.3		↑1.8	↑2.1

itance and robust generalization across variable depths and widths. This scalability establishes WAVE as an efficient initialization strategy for large-scale models.

E. Additional Analysis

E.1. Instincts

Instincts are natural abilities in organisms, brought by genes, that enable quick adaptation to environments with minimal or even no interaction [28]. GRL [9] first defines instincts in RL agents, showing that newborn agents can move toward rewards unconsciously. ECO [10] further extends this definition to supervised learning, demonstrating that networks can quickly classify images with minimal gradient descents, even with a substantial proportion of randomly initialized neurons.

Following the definition of instincts in ECO [10], we demonstrate that weight templates, as a new form of the learngene, provide neural networks with strong instincts. As shown in Figure E.1, WAVE exhibits stronger initial classification ability compared to other learngenes (including Heur-LG, Auto-LG, TLEG), even after just one epoch of training. This is attributed to the structured size-agnostic knowledge encapsulated in WAVE's weight templates.

E.2. Strong Learning Ability

Just as biological instincts enhance learning abilities in organisms, the learning abilities of neural networks are also enhanced by the instincts brought by learngenes.

Figure E.1 records the classification accuracy of different learngene methods (10 epochs) and models trained from scratch (150 epochs). We can see that WAVE outperforms other learngenes (including Heur-LG, Auto-LG and TLEG) and significantly improves training efficiency.

Compared with the networks trained from scratch, the WAVE-initialized neural networks achieve comparable performance to the neural networks trained from scratch with 150 epochs even after only one epoch of training. Taking the DeiT (-Ti, S and B) of 12 layers as an example, the WAVE reduces the training costs around 11× compared to training from scratch, and such training efficiency is more pronounced in smaller models (37.5× in DeiT-Ti L_4).

Such strong learning ability is also evident in models initialized by WAVE on downstream datasets. We visualize the curve of training loss on small and medium datasets (i.e., Oxford Flowers, CUB-200-2011, Stanford Cars, CIFAR-10 and CIFAR-100). As shown in Figure E.2, the models initialized by WAVE show faster loss reduction, indicating enhanced learning ability in downstream datasets.

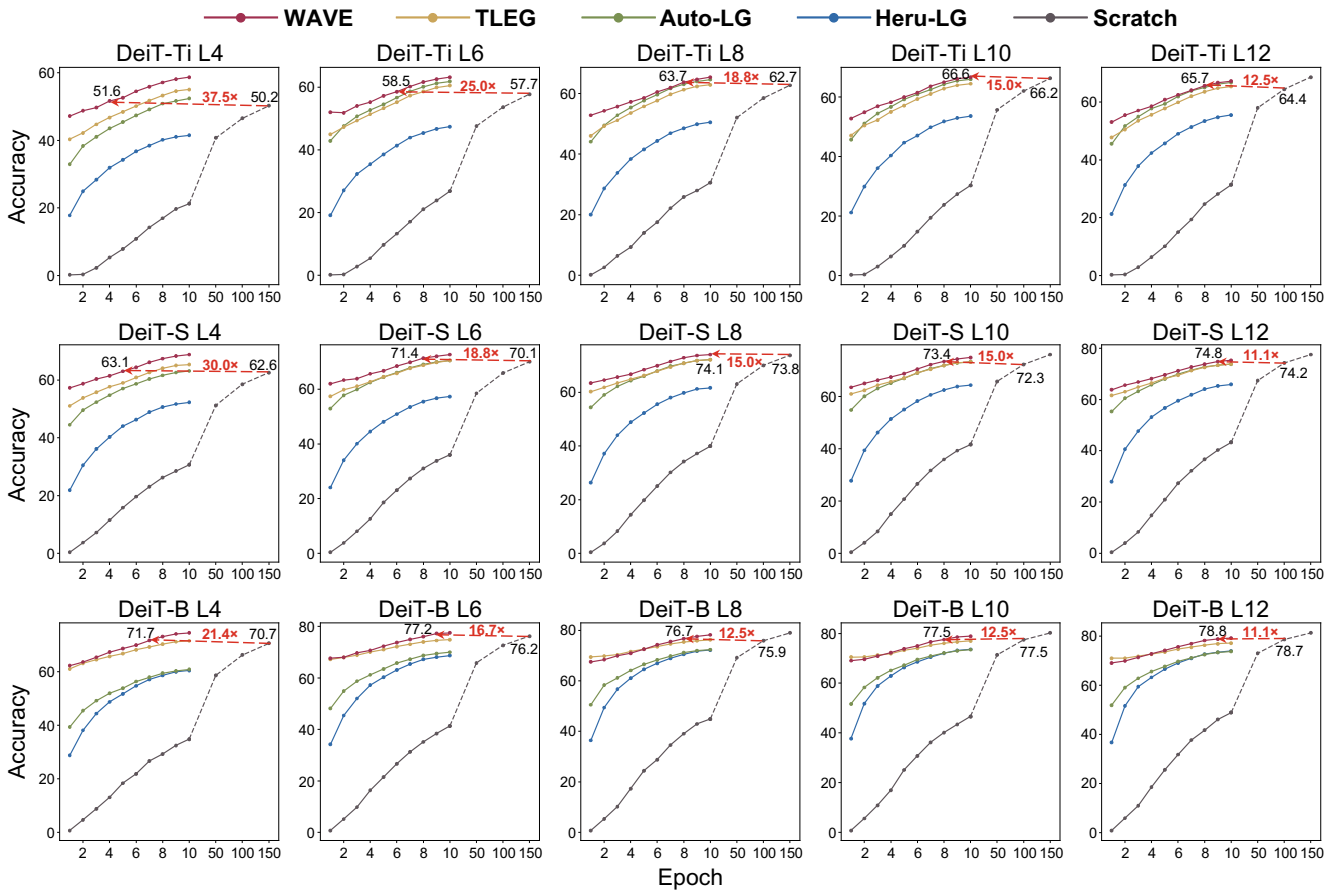

Figure E.1. Performance comparisons on ImageNet-1K among WAVE and other learngene methods.

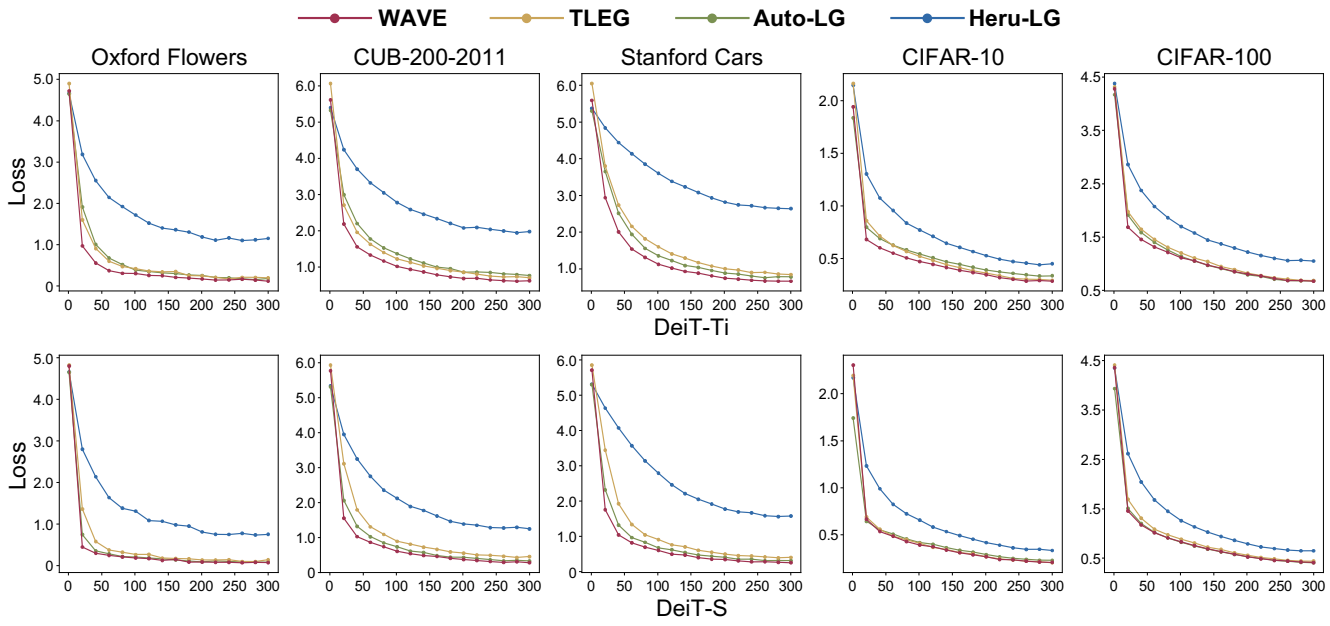

Figure E.2. Performance comparisons on small and medium downstream datasets among WAVE and other learngene methods.