REPARAMETERIZE BEFORE YOU EXTRACT: TRAINING-FREE SPECTRAL PRECONDITIONING FOR KNOWLEDGE TRANSFER

Anonymous authors
Paper under double-blind review

ABSTRACT

Large-scale pre-trained models contain transferable knowledge, but their parameters are tied to the architecture and size of the source model, limiting direct reuse across model sizes. Existing learngene-based approaches seek compact, task-agnostic representations for variable-sized model initialization, but many require auxiliary training. FRONT shows that low-frequency components of pre-trained weights are informative for foundational and transferable knowledge. However, standard pre-training does not explicitly organize weights for subsequent knowledge extraction, and the resulting spectra can exhibit limited low-frequency concentration. Under a fixed coefficient budget, this constrains the amount of spectral energy that can be retained. We introduce Spatial Permutation Alignment (SPA), a training-free structural preconditioner for frequency-domain knowledge extraction. Our key observation is that functionally equivalent parameterizations within the same permutation-equivalence class can exhibit different spectral energy distributions. SPA exploits this reparameterization freedom through depth-wise bipartite matching and width-wise anchored joint permutation, reducing inter- and intra-layer structural variation. Integrating SPA with FRONT yields SPA-FRONT, which captures a larger fraction of spectral energy in the retained low-frequency coefficients under the same coefficient budget. Across vision and language models, SPA-FRONT substantially improves variable-sized model initialization, reducing vision-model training epochs to reach target accuracy by 30.0% on average and cutting language-model pre-training FLOPs by up to 58.7%. The code is available at `https://anonymous.4open.science/r/SPA-FRONT`.

1 INTRODUCTION

Large-scale pre-trained models contain transferable knowledge that can be adapted to a wide range of downstream tasks (Kolesnikov et al., 2020; Hu et al., 2022). However, this knowledge is typically tied to the architecture and size of the source model, limiting its direct reuse in target models with different configurations, as illustrated in Figure 1(a). This limitation has motivated efforts to seek compact, task-agnostic knowledge representations, often referred to as "learngenes," for cross-size model initialization (Wang et al., 2022; Feng et al., 2025a; Shen et al., 2026b). Although prior methods demonstrate the feasibility of this paradigm, many rely on auxiliary training to construct transferable parameter structures (Xia et al., 2024; Feng et al., 2025b), as shown in Figure 1(b).

A relevant observation from FRONT (Shen et al., 2026a) is that the low-frequency components of pre-trained weights are informative for foundational and transferable knowledge. By applying the Discrete Cosine Transform (DCT) (Wallace, 1991) and retaining only a subset of low-frequency coefficients, FRONT obtains a compact representation without additional training. With a fixed coefficient budget, extraction quality also depends on how strongly the transferable knowledge is concentrated within the retained low-frequency coefficients (Jähne, 2005). In standard pre-trained models, however, the weight spectra remain relatively dispersed, leaving substantial useful information outside the region retained by fixed low-frequency truncation, shown in Figure 1(c). FRONT further shows that regularizing the model toward stronger low-frequency concentration can improve subsequent extraction, but requires auxiliary training (Shen et al., 2026a). More broadly, this points to a mismatch between pre-training and subsequent extraction: conventional pre-training optimizes the model's

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

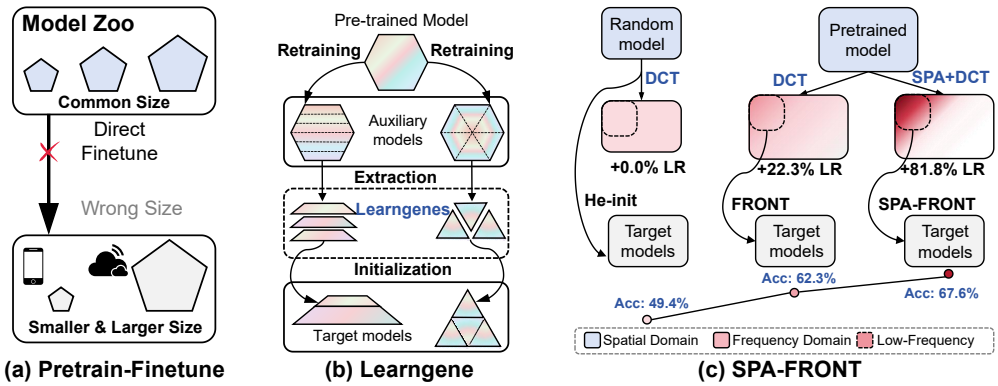


Figure 1: (a) Standard pre-training and fine-tuning paradigms remain tightly coupled to monolithic architecture sizes, precluding adaptation to different target sizes. (b) Learngene decouples knowledge but requires computationally prohibitive retraining to induce specific weight structures. (c) Applying SPA prior to DCT compacts dispersed spectral energy into low-frequency bands. Compared with FRONT, SPA increases Low-frequency Ratio (LR) by 81.8% relative to the original dispersed-energy level and improves accuracy from 62.3% to 67.6%, without retraining overhead.

learning objective alone, without shaping the parameterization into a form that post-hoc extractors can readily exploit. This raises a natural question: *can we efficiently obtain, from a pre-trained model, a parameterization that is better suited to knowledge extraction without auxiliary training?*

Permutation symmetry offers a natural way to address this question (Sussmann, 1992; Barkai et al., 1992). It defines a family of equivalent parameterizations obtained by coordinated reordering of hidden units and their coupled parameters, while preserving the source network function. Prior work has mainly exploited the symmetry to establish neuron correspondences across different trained models, particularly for model fusion and model merging (Singh & Jaggi, 2020; Ainsworth et al., 2022). Here, we instead use it to reorganize the parameters within a single pre-trained model. Our key observation is that different members of the same permutation-equivalence class can exhibit markedly different spectral distributions. This motivates a reparameterization-based perspective: rather than retraining the model, we search within its equivalence class for the parameter organization best suited to subsequent knowledge extraction. For frequency-domain extraction, this corresponds to seeking functionally equivalent reorderings with more compact spectral structure.

Building on this insight, we propose **Spatial Permutation Alignment (SPA)**, which reorganizes pre-trained weights before frequency-domain extraction by reducing local structural variation across adjacent layers and within the hidden dimensions of individual layers. Because the DCT represents slowly varying spatial patterns primarily through low-frequency coefficients, reducing such variation tends to concentrate more spectral energy toward the low-frequency region (Raid et al., 2014). To this end, *depth-wise alignment* uses minimum-cost bipartite matching to place structurally similar neurons at corresponding positions across adjacent layers, thereby reducing inter-layer variation. *Width-wise alignment* jointly reorders coupled projections through anchored path-based optimization, placing similar hidden units closer together to reduce intra-layer variation. Both alignments operate solely through coordinated weight permutations, requiring neither training data nor gradient-based updates. When SPA is integrated with FRONT, the resulting **SPA-FRONT** concentrates a larger fraction of spectral content in the low-frequency coefficients retained by FRONT, allowing more information to be preserved under the same fixed coefficient budget. More broadly, SPA suggests a lightweight route to post-hoc reparameterization of pre-trained models.

Extensive empirical evaluations across vision and language models demonstrate the effectiveness of SPA-FRONT for variable-sized model initialization. In vision tasks, SPA-FRONT consistently outperforms direct frequency-domain extraction and achieves performance competitive with retraining-based methods that require up to 150 epochs of auxiliary optimization, while introducing only tens of seconds of one-time CPU preprocessing. SPA-FRONT further reduces the training epochs to reach target accuracy by 30.0% on average and consistently improves downstream transfer across both DeiT and ResNet architectures. In language tasks, SPA-FRONT accelerates the pre-training of BERT, RoBERTa, and GPT, reducing the FLOPs required to reach the target loss by up to 58.7%, while achieving the best average performance among the evaluated initialization methods on the GLUE benchmark. Overall, these results demonstrate that SPA provides an efficient structural preconditioner for training-free knowledge transfer across different model architectures.

2 Related Work

Model Initialization. Neural network initialization has evolved from statistical distributions (Glorot & Bengio, 2010; Chen et al., 2021) to the pre-training and fine-tuning paradigm (He et al., 2016; Devlin et al., 2019). While pre-trained weights provide strong initialization, their fixed dimensionality limits direct transfer across model sizes. Existing methods either manipulate weights directly, as in Wt Select (Xu et al., 2023) and LiGO (Wang et al., 2023a), or use generative frameworks, as in GHN-3 (Knyazev et al., 2023). Such approaches may disrupt structural parameter correlations or incur substantial computational overhead, with a risk of negative transfer (Feng et al., 2025b). Compared with FRONT (Shen et al., 2026a), which directly performs spectral extraction on the parameterization obtained from standard pre-training, SPA-FRONT instead reorganizes the pre-trained weights to promote a more compact spectral structure before extraction, allowing more useful information to be retained under the same coefficient budget without auxiliary retraining.

Learngene. Drawing inspiration from biological inheritance, the learngene framework advocates for extracting condensed, informative representations from source networks to initialize structurally diverse target models (Feng et al., 2025a; Shen et al., 2026b). Existing methods generally follow two strategies. HeurLG and AutoLG (Wang et al., 2022; 2023b) extract selected network components according to empirical criteria, such as gradient robustness or feature alignment, but their reliance on fixed architectural fragments limits flexibility across target architectures. In contrast, TLEG (Xia et al., 2024) and WAVE (Feng et al., 2025b) bridge dimensional differences through cascaded linear transformations, at the cost of substantial auxiliary training. To overcome these bottlenecks, SPA-FRONT aligns the hidden neurons of the pre-trained weights, so that FRONT extracts the low-frequency coefficients with more energy and no auxiliary training is required.

3 Methods

3.1 Preliminaries: Permutation Symmetry and Frequency-Domain Transfer

Permutation Symmetry in Neural Networks. Neural networks exhibit permutation symmetry along their hidden dimensions. Consider a Multi-Layer Perceptron (MLP) block $y = W_{out}\sigma(W_{in}x+b_{in})+b_{out}$, where $\sigma(\cdot)$ is an element-wise nonlinear activation, $W_{in} \in \mathbb{R}^{d_h \times d_{in}}$, and $W_{out} \in \mathbb{R}^{d_{out} \times d_h}$. For any permutation matrix $P \in \{0,1\}^{d_h \times d_h}$ satisfying $PP^\top = P^\top P = I$, the element-wise activation commutes with permutation, i.e., $\sigma(Pz) = P\sigma(z)$. Hence,

$$y = W_{\text{out}}\sigma(W_{\text{in}}x + b_{\text{in}}) + b_{\text{out}} = (W_{\text{out}}P^\top)\sigma(PW_{\text{in}}x + Pb_{\text{in}}) + b_{\text{out}}. \tag{1}$$

Thus, the transformations $W_{in} \leftarrow PW_{in}$, $b_{in} \leftarrow Pb_{in}$, and $W_{out} \leftarrow W_{out}P^\top$ reorder the hidden units without changing the network function. The same principle extends to consistent permutations of coupled Transformer projections (e.g., (W_q, W_k), (W_v, W_o), and the feed-forward network (FFN)) and to synchronized channel reordering in Convolutional Neural Networks (CNNs).

Frequency-Domain Knowledge Transfer. FRONT (Shen et al., 2026a) extracts task-agnostic *learngenes* from the low-frequency components of pre-trained weights. It stacks compatible network parameters into a three-dimensional tensor $W \in \mathbb{R}^{L \times d_{in} \times d_{out}}$, where L denotes network depth, and applies a 3D Discrete Cosine Transform (3D-DCT), denoted by $\mathcal{D}$: $\Phi = \mathcal{D}(W)$. A binary low-pass mask M_r, controlled by a retention ratio $r \in (0, 1]$, then selects the low-frequency coefficients:

$$\mathcal{G} = M_r \odot \Phi, \quad M_r[l,i,j] = \begin{cases} 1, & l \le \lfloor rL \rfloor,\ i \le \lfloor rd_{in} \rfloor,\ j \le \lfloor rd_{out} \rfloor, \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

The extracted learngene $\mathcal{G}$ is resized through zero-padding or truncation, represented by $\mathcal{P}_{\text{resize}}$, and transformed back to the spatial domain to initialize the target model $W_{\text{tgt}} = \mathcal{D}^{-1}\left(\mathcal{P}_{\text{resize}}(\mathcal{G})\right)$.

Standard pre-training optimizes the learning objective rather than subsequent parameter-space operations, thus imposing no explicit topological constraint on neuron ordering across layers or hidden dimensions. Semantically related weights may therefore remain spatially separated in W, leading to weak low-frequency concentration, as illustrated in Figure 1(c). Consequently, direct low-pass truncation in Equation (2) may discard substantial useful information dispersed across the spectrum. Since greater similarity among neighboring elements generally yields a more compact low-frequency representation (Jähne, 2005), we investigate whether the spatial organization of pre-trained weights can be improved via a training-free, function-preserving transformation.

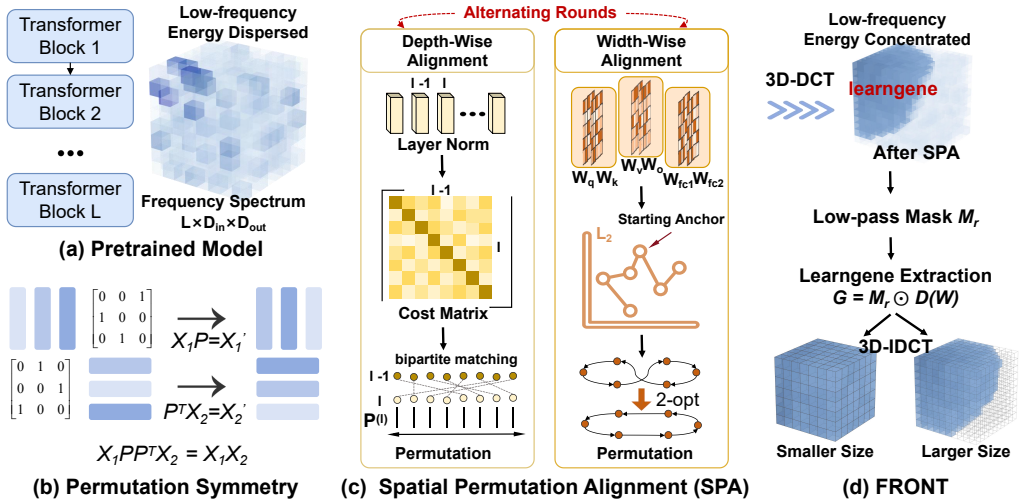


Figure 2: **Overview of SPA-FRONT.** SPA acts as a training-free structural preconditioner before learngene extraction. It alternates depth-wise alignment based on minimum-weight bipartite matching with width-wise alignment based on anchored joint permutation and 2-opt refinement. FRONT subsequently extracts low-frequency learngenes from the reorganized weights.

3.2 Overview of SPA-FRONT

We propose Spatial Permutation Alignment (SPA), a training-free structural preconditioner applied to pre-trained weights before frequency-domain extraction. SPA reorganizes weights along both the depth and width dimensions through consistent permutation propagation while preserving the source-model function without any auxiliary training. Depth-wise alignment matches structurally similar hidden units across adjacent layers, reducing local inter-layer variation and increasing correspondence between neighboring layers. Width-wise Anchored Joint Permutation (AJP) jointly reorders coupled projections within their hidden dimensions, placing similar hidden units closer together to reduce local intra-layer variation. The reorganized weights are then processed by FRONT, forming the complete SPA-FRONT framework, as illustrated in Figure 2.

3.3 Depth-Wise Alignment: Inter-Layer Smoothing

Standard pre-training does not explicitly establish neuron correspondence between adjacent layers, allowing structurally related hidden units to appear at different positions along the depth axis. To reduce such inter-layer variation, we sequentially align each layer with its preceding layer. Let $W_{in}^{(l)}$ and $W_{out}^{(l)}$ denote the input and output weights associated with the hidden units of layer l. Their incoming and outgoing weight descriptors are concatenated and standardized to form:

$$\tilde{F}^{(l)} = \mathrm{Norm}\left([W_{in}^{(l)}, (W_{out}^{(l)})^T]\right). \tag{3}$$

For two adjacent layers with compatible hidden dimensions, we construct a pairwise Euclidean distance matrix $C^{(l)}$ (Danielsson, 1980): $C_{i,j}^{(l)} = \|\tilde{F}_i^{(l-1)} - \tilde{F}_j^{(l)}\|_2$, where $C_{i,j}^{(l)}$ measures the structural difference between neuron i in layer $l-1$ and neuron j in layer l.

The correspondence is obtained by solving the following minimum-cost bipartite matching problem:

$$P^{(l)} = \arg\min_{P \in \{0,1\}^{d_h \times d_h}} \sum_{i=1}^{d_h}\sum_{j=1}^{d_h} C_{i,j}^{(l)} P_{i,j} \quad \text{subject to} \quad P\mathbf{1} = \mathbf{1}, \quad P^\top \mathbf{1} = \mathbf{1}. \tag{4}$$

These constraints enforce a one-to-one assignment, so no neuron is removed or duplicated. The Hungarian algorithm computes the exact minimum-cost matching for each adjacent-layer pair (Kenesei et al., 2002). The resulting permutation is applied as $W_{in}^{(l)} \leftarrow P^{(l)} W_{in}^{(l)}$ and $W_{out}^{(l)} \leftarrow W_{out}^{(l)} (P^{(l)})^\top$, together with the corresponding transformation of other coupled parameters. Processing the layers from shallow to deep aligns each layer with its already aligned predecessor, thereby reducing structural variation along the depth dimension.

3.4 WIDTH-WISE ALIGNMENT: ANCHORED JOINT PERMUTATION

Depth-wise alignment reduces variation across layers but does not optimize the ordering of hidden units within each layer. Moreover, a hidden-dimension permutation must be propagated consistently through all coupled projections to increase neuron similarity while preserving functional equivalence. In an FFN, for example, optimizing the ordering of W_{fc1} alone may increase the variation of its coupled projection W_{fc2}. We therefore construct a joint descriptor for each hidden unit:

$$F_i = [W_{fc1}[i,:], (W_{fc2}[:,i])^\top], \quad i \in 1, \ldots, d_h, \tag{5}$$

where $\mathcal{F} \in \mathbb{R}^{d_h \times (d_{in}+d_{out})}$ collects both the incoming and outgoing weight information.

We formulate width-wise reordering as an approximate shortest Hamiltonian path (Sanz-Serna & Calvo, 2018) problem over $\mathcal{F}$, referred to as Anchored Joint Permutation (AJP). The hidden unit with the largest ℓ_2-norm is selected as the starting anchor: $\pi(1) = \arg\max_i |F_i|_2$. Starting from this unit, AJP seeks an ordering π that minimizes the total distance between adjacent descriptors:

$$\mathcal{L}(\pi) = \sum_{k=1}^{d_h-1} \|F_{\pi(k)} - F_{\pi(k+1)}\|_2. \tag{6}$$

Because exact shortest-path optimization is NP-hard (Woeginger, 2003), we use a two-stage approximation. A greedy nearest-neighbor procedure first constructs an initial ordering from the anchor. The ordering is then refined using 2-opt local search (Chiang et al., 2010). For valid indices $1 < i < j < d_h$, the subpath $\pi(i, \ldots, j)$ is reversed whenever the corresponding edge exchange decreases the total path length:

$$|F_{\pi(i-1)} - F_{\pi(j)}|_2 + |F_{\pi(i)} - F_{\pi(j+1)}|_2 < |F_{\pi(i-1)} - F_{\pi(i)}|_2 + |F_{\pi(j)} - F_{\pi(j+1)}|_2. \tag{7}$$

The resulting permutation is consistently applied to both W_{fc1} and W_{fc2}, as well as to analogous coupled projections in other modules, detailed in Appendix A. This joint reordering reduces local variation along the width dimension without changing the network function.

3.5 INTEGRATION: ALTERNATING ALIGNMENT OF THE 3D WEIGHT TENSOR

To construct a globally smooth 3D weight tensor W, we iteratively coordinate depth-wise alignment (Section 3.3) and width-wise alignment (Section 3.4). Each iteration alternates depth-wise and width-wise alignment to refine inter-layer correspondence and intra-layer organization. Repeating this process reduces structural variation along both dimensions and increases low-frequency energy concentration, with only lightweight one-time CPU preprocessing.

The resulting tensor $\widehat{W} \in \mathbb{R}^{L \times d_{in} \times d_{out}}$ obtained by SPA remains functionally equivalent to the original model because each permutation is consistently propagated to its coupled weights. Within the FRONT framework, $\widehat{W}$ is transformed as $\widehat{\Phi} = \mathcal{D}(\widehat{W})$. The improved spatial smoothness concentrates more spectral energy in the low-frequency region, enabling the truncation mask M_r in Equation 2 to retain more transferable knowledge without additional training.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Following FRONT (Shen et al., 2026a), we evaluate SPA-FRONT across both vision and language domains, covering initialization stability and generalization. We use depth-only alignment for depth vision expansion, width-only alignment for width vision expansion, and one iteration of joint depth–width alignment for all other experiments. Additional implementation details and other results are provided in Appendix D and Appendix E.

Vision. We evaluate SPA-FRONT on DeiT and ResNet architectures using ImageNet-1K (Deng et al., 2009) and seven downstream datasets. Baselines comprise direct initialization methods, including He-Init (Chen et al., 2021), Mimetic-Init (Trockman & Kolter, 2023), Wt Select (Xu et al., 2023), Heur-LG (Wang et al., 2022), Cluster-LG (Wang et al., 2024), LiGO (Wang et al., 2023a), and FRONT,

Table 1: Performance of models initialized at various depths on ImageNet-1K for 10 epochs. "Para.(M)" denotes the average number of parameters transferred during initialization, in millions. "Aux. Epochs" indicates the auxiliary training cost required *prior to* the actual initialization. Specifically, the multiplier $\times 3$ signifies that the method requires independent retraining for each of the three architecture sizes (Ti, S, B). Similarly, the $\times 15$ multiplier denotes mandatory retraining for all 15 configurations (3 sizes $\times$ 5 depths). " / " denotes failed initialization, and "N/A" indicates that the metric is not applicable. The **best** result is highlighted in bold, second-best is underlined, and Δ denotes the performance difference relative to FRONT.

			W_{192} (DeiT-Ti)						W_{384} (DeiT-S)						W_{768} (DeiT-B)				
			L_4	L_6	L_8	L_{10}	L_{12}		L_4	L_6	L_8	L_{10}	L_{12}		L_4	L_6	L_8	L_{10}	L_{12}
Methods	Aux. Epochs	Para.	2.2	3.1	4.0	4.9	5.8	Para.	7.9	11.5	15.0	18.6	22.2	Para.	29.9	44.1	58.3	72.5	86.7
He-Init	0	0	34.7	40.6	43.7	46.8	48.3	0	42.2	49.4	52.1	53.7	55.5	0	47.9	53.1	54.4	55.0	56.7
Mimetic	0	0	35.1	40.2	43.2	46.3	48.1	0	43.3	49.1	53.0	54.1	55.6	0	50.2	54.3	56.5	58.5	58.6
Wt Select	0	3.9	46.1	52.3	55.0	56.8	58.9	15.0	50.1	56.0	57.5	61.9	63.3	58.2	56.7	61.1	63.6	64.8	65.9
Heur-LG	0	1.7	41.5	47.4	50.5	53.5	55.5	6.1	52.3	57.3	61.7	64.4	65.9	22.8	60.5	68.7	72.2	73.6	74.0
Cluster-LG	0	2.4	46.6	51.5	51.9	55.8	57.2	5.1	53.2	54.9	52.6	52.1	55.5	19.6	52.7	60.4	60.6	58.6	62.7
LiGO	0	2.2	/	59.0	60.2	59.8	60.9	7.9	/	68.6	69.9	69.7	70.0	29.9	/	74.2	74.4	75.3	75.4
GHN-3	N/A	N/A	40.9	45.0	46.6	49.1	48.9	N/A	45.4	49.0	50.2	52.3	53.2	N/A	49.5	52.5	53.8	54.2	54.3
Share Init	150×3	0.8	55.2	59.8	62.5	64.3	65.3	2.5	65.0	69.7	71.7	72.7	73.3	8.6	71.7	75.3	76.4	77.4	77.6
Auto-LG	50×3	2.2	52.4	61.8	64.6	65.9	66.8	7.9	63.2	70.5	72.2	73.3	73.8	29.9	60.9	70.0	72.4	73.5	73.8
TLEG	150×3	1.3	55.0	60.5	62.9	64.4	65.4	4.3	65.4	70.5	72.1	73.2	73.8	15.7	71.6	74.9	76.2	77.0	77.1
WAVE	150×3	1.3	**58.6**	63.2	65.4	66.6	67.3	4.3	**68.9**	**72.7**	**74.1**	74.9	75.3	15.7	**74.5**	**77.5**	**78.2**	**78.9**	**79.2**
FRONT	0	2.2	55.3	63.3	64.4	64.7	65.3	8.1	63.4	68.9	70.5	70.9	71.2	32.4	72.1	74.6	75.3	75.8	76.3
SPA-FRONT	**0**	2.2	58.0	**64.1**	**66.6**	**69.1**	**69.5**	8.1	68.4	72.2	**74.1**	**75.1**	**75.4**	32.4	73.6	76.3	78.0	78.7	**79.2**
Δ	-	-	**↑2.7**	**↑0.8**	**↑2.2**	**↑4.4**	**↑4.2**	-	**↑5.0**	**↑3.3**	**↑3.6**	**↑4.2**	**↑4.2**	-	**↑1.5**	**↑1.7**	**↑2.7**	**↑2.9**	**↑2.9**
Direct PT	150×15	4.0	50.4	57.7	62.7	66.2	68.6	15.0	62.6	70.1	73.8	76.0	77.6	58.3	70.7	76.2	79.1	80.5	81.5

as well as retraining-based methods, including GHN-3 (Knyazev et al., 2023), Share-Init (Lan et al., 2019), Auto-LG (Wang et al., 2023b), TLEG (Xia et al., 2024), and WAVE (Feng et al., 2025b).

Language. We further evaluate SPA-FRONT on BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), and GPT-2 (Radford et al., 2019), transferring knowledge from Base-scale 12-layer models to Small-scale 6-layer counterparts. Generalization is measured on GLUE (Wang et al., 2018). Baselines include Scratch, knowledge distillation (KD) (Hinton et al., 2015), Wt Select and FRONT.

4.2 Results on Vision Tasks

4.2.1 Initialization Ability on Variable-sized Models

Depth Expansion. Table 1 evaluates variable-depth initialization on ImageNet-1K. Among direct initialization methods, LiGO and FRONT substantially outperform random initialization and Wt Select. However, LiGO introduces additional randomly initialized parameters, while FRONT directly operates on the original weight spectra without addressing spectral dispersion. By preconditioning the weight topology before extraction, SPA-FRONT consistently improves upon FRONT across all expanded-depth settings, achieving an average gain of 3.1% and the best training-free performance.

SPA-FRONT also remains highly competitive with retraining-based methods. Notably, the advantage of SPA-FRONT becomes more pronounced as the target depth increases. At L_{12}, it matches or surpasses WAVE across all three DeiT scales without auxiliary training. In contrast, WAVE requires 150 epochs of retraining for each scale. For example, on DeiT-Ti-L_{12}, SPA-FRONT improves accuracy from 67.3% to 69.5%. These results show that structural preconditioning can recover much of the benefit of costly retraining through training-free weight reorganization alone.

Table 2: Performance of initializing models with variable widths on ImageNet-1K.

			L_6				
			W_{384}	W_{576}	W_{768}	W_{1152}	W_{1536}
Methods	Aux. Epoch	Para.	11.5	25.1	44.1	98.0	173.1
He-Init	0	0	49.4	51.5	53.1	46.6	31.3
Mimetic	0	0	49.1	48.0	54.3	47.7	33.0
Wt Select	0	26.8	48.6	50.7	55.4	/	/
LiGO	0	10.0	63.6	63.1	69.5	70.0	73.7
GHN-3	N/A	N/A	49.0	51.8	52.5	52.7	51.0
WAVE	150	5.4	64.8	67.0	72.4	73.5	**78.3**
FRONT	0	8.9	62.3	63.6	69.8	70.7	73.9
SPA-FRONT	**0**	8.9	**67.6**	**68.8**	**72.7**	**74.1**	76.6
Δ	-	-	**↑5.3**	**↑5.2**	**↑2.9**	**↑3.4**	**↑2.7**

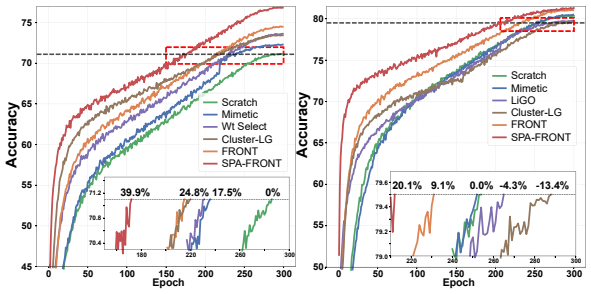
Width Expansion. Table 2 evaluates variable-width model initialization on ImageNet-1K. While direct frequency-domain extraction with FRONT already provides a strong baseline, it does not explicitly optimize the coupled structure of weights. SPA-FRONT addresses this limitation through Anchored Joint Permutation, thereby improving width-wise structural coherence. As a result, SPA-FRONT consistently improves upon FRONT across all evaluated target widths.

Table 3: Performance of models on downstream datasets. "Para.(M)" is the average parameter transferred during model initialization.

		DeiT-Ti, 3.0M									DeiT-S, 11.3M								
Methods	Aux. Epoch	Para.	Flow.	CUB	Cars	C10	C100	Food	iNat.	*Aver.*	Para.	Flow.	CUB	Cars	C10	C100	Food	iNat.	*Aver.*
He-Init	0	0	53.9	26.1	19.9	92.4	68.3	68.4	52.3	*54.5*	0	57.2	27.3	23.8	94.0	66.5	70.6	54.0	*56.2*
Mimetic	0	0	52.1	35.0	20.5	88.9	63.4	66.9	49.0	*53.7*	0	57.4	39.6	34.2	91.6	65.7	67.1	52.2	*58.3*
Wt Select	0	2.9	55.0	34.4	21.7	92.5	67.0	67.4	51.2	*55.6*	11.0	58.3	33.1	28.1	94.1	68.1	69.0	54.2	*57.8*
Heur-LG	0	1.5	64.7	44.6	37.7	94.0	71.1	74.7	57.4	*63.5*	5.7	69.1	48.0	51.2	95.1	72.8	76.8	59.3	*67.5*
Cluster-LG	0	2.4	62.8	44.8	36.9	94.1	74.5	78.6	56.4	*64.0*	5.1	65.5	45.4	46.3	95.4	74.6	78.4	60.4	*66.6*
LiGO	0	2.0	94.2	71.8	83.9	95.6	78.5	82.1	61.6	*81.1*	7.5	95.9	74.8	87.9	96.9	81.3	84.0	66.1	*83.8*
GHN-3	N/A	N/A	50.0	41.1	23.2	92.5	70.1	76.2	51.7	*57.8*	N/A	52.7	45.2	30.6	93.9	72.7	76.2	55.5	*61.0*
Share Init	150×2	0.6	92.4	70.1	82.1	96.0	77.2	81.2	63.2	*80.3*	2.2	94.1	72.4	87.2	96.5	78.5	83.0	63.0	*82.1*
Auto-LG	50×2	2.0	93.5	71.4	83.5	96.4	77.1	81.7	62.5	*80.9*	7.5	96.4	75.1	88.2	97.3	81.0	84.6	67.0	*84.2*
TLEG	150×2	1.1	91.0	69.5	78.2	96.1	77.0	82.0	63.4	*79.6*	3.9	93.7	72.6	87.2	97.2	80.2	84.9	66.5	*83.2*
WAVE	150×2	1.1	**94.9**	**74.8**	84.4	**96.6**	**80.7**	83.8	65.2	***82.9***	4.0	**96.9**	**78.1**	**89.4**	97.4	**83.2**	85.5	67.6	***85.4***
FRONT	0	2.2	92.9	70.5	82.3	95.9	78.7	83.7	63.7	*81.1*	8.1	94.5	74.1	87.4	97.2	81.8	85.2	66.6	*83.9*
SPA-FRONT	**0**	2.2	93.1	73.0	**84.8**	96.5	79.5	**83.9**	**65.5**	*82.3*	8.1	94.6	75.2	88.2	97.3	81.9	**85.9**	**68.8**	*84.6*
Δ	-	-	↑0.2	↑2.5	↑2.5	↑0.6	↑0.8	↑0.2	↑1.8	↑1.2	-	↑0.1	↑1.1	↑0.8	↑0.1	↑0.1	↑0.7	↑2.2	↑0.7
Direct FT	150×2	2.9	95.4	75.1	86.5	96.6	80.2	84.0	66.9	*83.5*	11.0	96.4	77.0	89.4	97.5	82.8	85.6	69.3	*85.4*

Despite requiring no auxiliary training, SPA-FRONT also outperforms the retraining-based WAVE at four of the five target widths and remains competitive at the widest configuration. This favorable performance-efficiency trade-off indicates that width-wise structural preconditioning can recover much of the benefit otherwise obtained through costly auxiliary optimization, using only function-preserving weight reorganization.

Initialization Performance during Full Training. Figure 3 compares the 300-epoch training dynamics of different initialization methods on DeiT-Ti and DeiT-S. While prior methods such as Cluster-LG and LiGO initially outperform training from scratch, their advantages gradually diminish, particularly on DeiT-S. In contrast, SPA-FRONT starts from a substantially higher accuracy and maintains a consistent advantage throughout training. It reduces the training cost required to reach the target accuracy by 39.9% and 20.1% on DeiT-Ti and DeiT-S, respectively, exceeding the 24.8% and 9.1% reductions achieved by the second-best method, FRONT. These results show that SPA-based structural preconditioning provides a more favorable initialization and substantially accelerates subsequent convergence.



(a) DeiT-Ti (b) DeiT-S

Figure 3: Comparison of different direct initialization methods during full training on DeiT-Ti and DeiT-S.

4.2.2 Generalization Across Downstream Datasets.

Initialization on DeiT-based Architectures. Table 3 evaluates the downstream performance of initialized DeiT methods. For all methods, target models are initialized from an ImageNet-1K pre-trained source and directly fine-tuned on downstream datasets without intermediate pre-training.

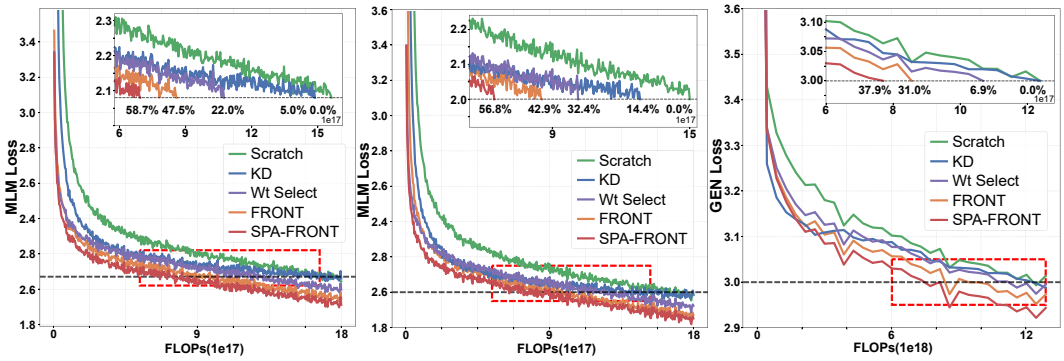
SPA-FRONT consistently improves downstream transfer performance. Wt Select shows limited robustness across datasets and can even underperform He-Init on iNaturalist. FRONT provides a strong training-free baseline through low-frequency learngene extraction, while SPA-FRONT further improves the weight organization before extraction through structural preconditioning. Consequently, SPA-FRONT outperforms FRONT on all evaluated datasets, increasing the average accuracy by 1.2% and 0.7% on DeiT-Ti and DeiT-S, respectively. Despite requiring no auxiliary training, SPA-FRONT also remains competitive with retraining-based methods: on DeiT-Ti, it achieves an average accuracy of 82.3%, close to WAVE's 82.9%, while outperforming WAVE on CUB and Food.

Table 4: Performance comparison on ResNet-50 and ResNet-152 architectures.

	ResNet-50				ResNet-152			
	Scratch	FRONT	**SPA-FRONT**	Δ	Scratch	FRONT	**SPA-FRONT**	Δ
Image.	74.1	76.1	**76.6**	↑0.5	76.8	77.2	**79.1**	↑1.9
Flow.	51.3	91.3	**94.5**	↑3.2	28.0	87.7	**97.6**	↑9.9
CUB	40.7	68.9	**73.2**	↑4.3	22.8	68.2	**79.3**	↑11.1
Cars	48.6	88.2	**88.8**	↑0.6	29.3	87.4	**90.9**	↑3.5
C10	94.8	95.6	**96.5**	↑0.9	95.6	95.7	**98.1**	↑2.4
C100	76.8	78.3	**81.3**	↑3.0	77.7	79.5	**86.6**	↑7.1
Food	82.1	85.8	**87.5**	↑1.7	83.2	85.9	**89.5**	↑3.6
iNat	61.8	69.0	**70.2**	↑1.2	62.7	69.1	**74.4**	↑5.3

Initialization on ResNet-based Architectures. To evaluate architectural generalization beyond Transformers, we extend SPA-FRONT to CNNs. We use an ImageNet-1K pre-trained ResNet-101 to initialize both ResNet-50 and ResNet-152, and evaluate the resulting models on ImageNet-1K and seven downstream datasets. As shown in Table 4, FRONT already enables effective frequency-domain transfer across different ResNet depths, while SPA-FRONT consistently improves upon it by reorganizing the weights before spectral extraction. SPA-FRONT achieves the best performance across all evaluated datasets, with particularly pronounced gains for ResNet-152, including improvements of 11.1 and 9.9 percentage points over FRONT on CUB and Flowers, respectively. These results indicate that SPA can extend beyond Transformer architectures and provide consistent benefits for cross-size initialization in CNNs.

4.3 Results on Language Tasks



(a) BERT-B → BERT-S (b) RoBERTa-B → RoBERTa-S (c) GPT-B → GPT-S

Figure 4: Results for language model pre-training. Relative to models trained from scratch, SPA-FRONT yields maximum FLOP reductions of 58.7%, 56.8%, and 37.9% for BERT-S, RoBERTa-S, and GPT-S, respectively.

More Efficient Pre-training. As illustrated in Figure 4, SPA-FRONT starts with a notably lower initial loss and accelerates model convergence across all three language model architectures. Specifically, SPA-FRONT reaches the target pre-training loss with remarkably fewer computational FLOPs, yielding reductions of 58.7%, 56.8%, and 37.9% for BERT, RoBERTa, and GPT, respectively. These results consistently outperform the FRONT baseline, delivering additional absolute reductions of 11.2%, 13.9%, and 6.9%. While alternative parameter transfer approaches such as Wt Select offer moderate convergence benefits, they lack topological awareness and inevitably disrupt structural parameter relationships. Furthermore, although KD leverages pre-trained knowledge, it imposes a massive cumulative computational overhead due to the continuous teacher inferences required at every training step. By contrast, through the explicit anchoring of spatial dependencies, SPA-FRONT preserves the transferred structural knowledge, fundamentally accelerating the language model pre-training process without any retraining overhead.

Generalization on GLUE Benchmark. As detailed in Table 5, empirical results on the GLUE benchmark demonstrate the exceptional generalization capabilities of SPA-FRONT, which consistently achieves the highest performance across all evaluated tasks and yields a robust 2.01% average improvement over FRONT. While KD provides little overall benefit over training from scratch and Wt Select yields stronger transfer, both fail to fully exploit the structural priors embedded within the source models. Furthermore, although FRONT effectively transfers fundamental competencies, its direct frequency truncation limits its adaptability to complex natural language understanding tasks like MRPC. By contrast, SPA-FRONT preserves critical topological structures before the initialization phase, strictly outperforming all evaluated baselines. Ultimately, these results

Table 5: GLUE benchmark results on BERT-S.

Task	Scratch	KD	Wt Select	FRONT	**SPA-FRONT**	Δ
SST-2	78.97	77.75	79.78	82.44	**84.03**	↑ **1.59**
MNLI	63.65	64.81	72.07	74.08	**74.60**	↑ **0.52**
MRPC	66.42	64.14	69.12	68.46	**70.10**	↑ **1.64**
CoLA	8.14	9.14	10.13	17.38	**18.25**	↑ **0.87**
QNLI	57.21	58.00	64.07	70.86	**76.06**	↑ **5.20**
QQP	81.50	80.07	83.62	84.93	**86.94**	↑ **2.01**
STS-B	17.27	15.76	24.54	54.23	**56.47**	↑ **2.24**
Average	53.31	52.81	57.62	64.63	**66.64**	↑ **2.01**

confirm that SPA-FRONT establishes a highly effective, topology-aware initialization paradigm, ensuring superior generalization across diverse downstream tasks.

4.4 Ablation

Effectiveness of Depth-Wise Alignment Strategies. To validate the design of our inter-layer alignment mechanism, we conduct an ablation study comparing four distinct depth-wise alignment strategies: Optimal Transport (OT), Global Anchor (GA), Spectral Sorting (SS), and our proposed Sequential Euclidean matching detailed in Appendix D.4. He-Init is included as a reference baseline.

Table 6: Ablation of depth-wise alignment strategies on DeiT-Ti. Accuracy (%) is reported for different target depths, and CPU time denotes the overhead of permutation matching.

Strategy	CPU Time (s)	L4	L6	L8	L10	L12
He-Init	0	34.7	40.6	43.7	46.8	48.3
OT	0.20	46.2	51.9	57.5	58.8	60.8
GA	2.21	52.9	58.5	62.7	65.1	65.3
SS	1.42	55.4	58.8	63.3	65.7	66.2
Ours	0.22	**58.0**	**64.1**	**67.4**	**69.1**	**69.5**

As shown in Table 6, our method consistently achieves the best accuracy across all target depths ($L = \{4, 6, 8, 10, 12\}$) on DeiT-Ti. Unlike soft alignment schemes such as OT, our bipartite matching establishes a discrete one-to-one correspondence between neurons in adjacent layers, providing a well-defined permutation for reducing inter-layer structural variation. Notably, our method requires only 0.22 s of CPU time, comparable to OT (0.20 s) and substantially lower than GA (2.21 s) and SS (1.42 s), demonstrating a favorable trade-off between performance and efficiency.

Effectiveness of Anchored Joint Permutation. Table 7 compares AJP with several width-wise permutation strategies detailed in Appendix D.5, while the definition of the reported ΔLR metric is provided in Appendix D.6. The standard pre-trained weight exhibits a 22.33% higher LR, whereas random permutation seeks only an 8.29% gain and reduces the W_{384} accuracy from 62.3% to 58.7% compared with FRONT. These results further support our observation that different parameterizations within the same permutation-equivalence class can exhibit markedly different spectral distributions. Effective frequency-domain extraction therefore depends on selecting a favorable reordering, rather than applying permutation indiscriminately.

Table 7: Ablation study of width-wise alignment. ΔLR denotes the relative increase in the low-frequency ratio with respect to He-Init.

Strategy	CPU Time (s)	ΔLR (%)	W384	W576	W768
He-Init	0.00	0.00%	49.4	51.5	53.1
Random Perm.	0.27	8.29%	58.7	62.6	63.4
FRONT	0.00	22.33%	62.3	63.6	69.8
Smooth	0.49	67.78%	66.2	67.2	69.2
JG	0.79	76.56%	67.0	67.8	70.4
TSP	84.64	81.82%	67.3	67.9	71.0
STSP	74.19	**81.98%**	67.0	68.1	71.9
AJP	66.40	81.82%	**67.6**	**68.8**	**72.7**

The structured permutation strategies reveal substantial variation within the same permutation-equivalence class. Although Smooth, JG, TSP, and STSP progressively increase the relative LR gain, higher spectral concentration does not necessarily translate into better transfer; STSP reaches the highest relative LR gain (81.98%) but remains below AJP in accuracy. By anchoring the path and refining it with 2-opt, AJP selects a more favorable reordering, achieving an 81.82% relative LR gain and the best accuracy across all widths with lower preprocessing time than TSP and STSP.

5 Conclusion

We presented Spatial Permutation Alignment (SPA), a training-free structural preconditioner that reorganizes pre-trained weights before frequency-domain knowledge extraction. Motivated by the observation that functionally equivalent parameterizations within the same permutation-equivalence class can exhibit markedly different spectral distributions, SPA uses coordinated weight permutations to reduce inter-layer and intra-layer structural variation and promotes spectral compactness without training data or gradient-based updates. SPA-FRONT integrates SPA with FRONT, which builds on the observation that low-frequency weight components effectively capture foundational and transferable knowledge. It allows a fixed coefficient budget to better preserve such knowledge and thereby improves variable-sized model initialization across vision and language models. Our results show that function-preserving reparameterization can better align pre-trained weights with subsequent spectral extraction without auxiliary retraining. More broadly, SPA suggests a lightweight route to post-hoc reparameterization of pre-trained models for structured weight processing.

AI USE STATEMENT

Generative AI tools were used solely to improve the clarity, grammar, and readability of the manuscript. They were not used for research ideation, theoretical development, methodology or experiment design, code implementation, data processing, result analysis or interpretation, literature review, or the generation of scientific claims, figures, or other research artifacts. All AI-assisted edits were carefully reviewed and revised by the authors, who take full responsibility for this work.

REPRODUCIBILITY STATEMENT

We provide code and implementation details to facilitate reproduction of our experiments. The main paper describes the SPA preconditioner, the DCT-based spectral extraction of the learngene, and the expansion operator used to initialize the target model. Appendix A details the alignment procedures and Appendix D details experimental settings. The objectives beyond FRONT discussed in Appendix F are conceptual. The code is available at `https://anonymous.4open.science/r/SPA-FRONT`.

REFERENCES

Samuel K Ainsworth, Jonathan Hayase, and Siddhartha Srinivasa. Git re-basin: Merging models modulo permutation symmetries. *arXiv preprint arXiv:2209.04836*, 2022.

Eli Barkai, David Hansel, and Haim Sompolinsky. Broken symmetries in multilayered perceptrons. *Physical Review A*, 45(6):4146, 1992.

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

Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV'21)*, pp. 9640–9649, 2021.

Cheng-Wen Chiang, Wei-Ping Lee, and Jia-Sheng Heh. A 2-opt based differential evolution for global optimization. *Applied Soft Computing*, 10(4):1200–1207, 2010.

Per-Erik Danielsson. Euclidean distance mapping. *Computer Graphics and image processing*, 14(3): 227–248, 1980.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV'09)*, pp. 248–255, 2009.

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

Fu Feng, Jing Wang, Xu Yang, and Xin Geng. Learngene: Inheritable "genes" in intelligent agents. *Artificial Intelligence.*, 348:104421, 2025a.

Fu Feng, Yucheng Xie, Jing Wang, and Xin Geng. Wave: Weight template for adaptive initialization of variable-sized models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR'25)*, 2025b.

Timnit Gebru, Jonathan Krause, Yilun Wang, Duyun Chen, Jia Deng, and Li Fei-Fei. Fine-grained car detection for visual census estimation. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI'17)*, pp. 4502–4508, 2017.

Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*, pp. 249–256, 2010.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR'16)*, pp. 770–778, 2016.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In *Proceedings of the International Conference on Learning Representations (ICLR'22)*, pp. 1–13, 2022.

Bernd Jähne. *Digital image processing*. Springer, 2005.

István Kenesei, Robert M Vago, and Anna Fenyvesi. *Hungarian*. Routledge, 2002.

Boris Knyazev, Doha Hwang, and Simon Lacoste-Julien. Can we scale transformers to predict parameters of diverse imagenet models? In *Proceedings of the International Conference on Machine Learning (ICML'23)*, pp. 17243–17259, 2023.

Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Joan Puigcerver, Jessica Yung, Sylvain Gelly, and Neil Houlsby. Big transfer (bit): General visual representation learning. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part V 16*, pp. 491–507, 2020.

Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. *Handbook of Systemic Autoimmune Diseases*, 1, 2009.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*, 2019.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

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

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

AM Raid, WM Khedr, Mohamed A El-Dosuky, and Wesam Ahmed. Jpeg image compression using discrete cosine transform-a survey. *arXiv preprint arXiv:1405.6147*, 2014.

Jesus-Maria Sanz-Serna and Mari-Paz Calvo. *Numerical hamiltonian problems*, volume 7. Courier Dover Publications, 2018.

Jianlu Shen, Fu Feng, Yucheng Xie, Jiaqi Lv, and Xin Geng. Breaking the scale barrier: One-shot knowledge transfer via frequency transform. In *Proceedings of the International Conference on Machine Learning (ICML'26)*, pp. 1–28, 2026a.

Jianlu Shen, Fu Feng, Jiaze Xu, Yucheng Xie, Jiaqi Lv, and Xin Geng. A unified framework for knowledge transfer in bidirectional model scaling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR'26)*, pp. 34535–34545, 2026b.

Sidak Pal Singh and Martin Jaggi. Model fusion via optimal transport. *Proceedings of the Annual Conference on Neural Information Processing Systems (NeurIPS'20)*, 33:22045–22055, 2020.

Héctor J Sussmann. Uniqueness of the weights for minimal feedforward nets with a given input-output map. *Neural networks*, 5(4):589–593, 1992.

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

Asher Trockman and J Zico Kolter. Mimetic initialization of self-attention layers. In *Proceedings of the International Conference on Machine Learning (ICML'23)*, pp. 34456–34468, 2023.

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

Gregory K Wallace. The jpeg still picture compression standard. *Communications of the ACM*, 34: 30–44, 1991.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*, 2018.

Peihao Wang, Rameswar Panda, Lucas Torroba Hennigen, Philip Greengard, Leonid Karlinsky, Rogerio Feris, David Daniel Cox, Zhangyang Wang, and Yoon Kim. Learning to grow pretrained models for efficient transformer training. *arXiv preprint arXiv:2303.00980*, 2023a.

Qiu-Feng Wang, Xin Geng, Shu-Xia Lin, Shi-Yu Xia, Lei Qi, and Ning Xu. Learngene: From open-world to your learning task. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI'22)*, pp. 8557–8565, 2022.

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

Qiufeng Wang, Xu Yang, Fu Feng, Jing wang, and Xin Geng. Cluster-learngene: Inheriting adaptive clusters for vision transformers. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NeurIPS'24)*, 2024.

Gerhard J Woeginger. Exact algorithms for np-hard problems: A survey. In *Combinatorial Optimization—Eureka, You Shrink! Papers Dedicated to Jack Edmonds 5th International Workshop Aussois, France, March 5–9, 2001 Revised Papers*, pp. 185–207. Springer, 2003.

Shiyu Xia, Miaosen Zhang, Xu Yang, Ruiming Chen, Haokun Chen, and Xin Geng. Transformer as linear expansion of learngene. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI'24)*, pp. 16014–16022, 2024.

Zhiqiu Xu, Yanjie Chen, Kirill Vishniakov, Yida Yin, Zhiqiang Shen, Trevor Darrell, Lingjie Liu, and Zhuang Liu. Initializing models with larger ones. *arXiv preprint arXiv:2311.18823*, 2023.

A Detailed Alignment Procedures

This section details the implementation of Spatial Permutation Alignment (SPA). All permutation operations are performed without gradient computation and modify only functionally symmetric hidden coordinates. Algorithm 1 describes the depth-wise alignment across adjacent Transformer blocks. Algorithm 3 specifies the module-dependent width-wise procedure under the Anchored Joint Permutation (AJP), including the additional synchronization required by attention modules equipped with relative positional encoding (RPE). Algorithm 4 combines these two operations into the alternating alignment procedure used before frequency-domain knowledge extraction.

A.1 Depth-Wise Inter-Layer Neuron Alignment

Let $\mathcal{M}^{(l)}{}_{l=1}^{L}$ denote the MLP blocks ordered from shallow to deep, where these blocks share the same embedding dimension d and hidden width d_h. The MLP in block l contains an input projection $W_1^{(l)} \in \mathbb{R}^{d_h \times d}$, an optional hidden bias $b_1^{(l)} \in \mathbb{R}^{d_h}$, and an output projection $W_2^{(l)} \in \mathbb{R}^{d \times d_h}$. The output bias of the second projection, when present, is indexed by the output dimension and therefore remains unchanged.

For each hidden neuron, we concatenate its incoming and outgoing weights to form the joint descriptor

$$F^{(l)} = \left[W_1^{(l)}, \left(W_2^{(l)} \right)^\top \right] \in \mathbb{R}^{d_h \times 2d}. \tag{8}$$

Each descriptor coordinate is independently standardized across the hidden neurons within the same block:

$$\widetilde{F}_{:,k}^{(l)} = \frac{F_{:,k}^{(l)} - \mu_k^{(l)}}{\sigma_k^{(l)} + \epsilon}, \qquad \epsilon = 10^{-8}, \tag{9}$$

where $\mu_k^{(l)}$ and $\sigma_k^{(l)}$ denote the mean and standard deviation of the k-th descriptor coordinate in block l.

The first MLP block retains its original hidden-neuron ordering and serves as the initial reference. For each $l = 2, \ldots, L$, the current block is aligned with its already aligned predecessor. We first construct the pairwise cost matrix

$$C_{i,j}^{(l)} = \left| \widetilde{F}_i^{(l-1)} - \widetilde{F}_j^{(l)} \right|_2, \tag{10}$$

where row i corresponds to neuron i in block $l-1$ and column j corresponds to neuron j in block l. The Hungarian algorithm then determines the minimum-cost assignment

$$\pi_{\text{dep}}^{(l)} = \arg\min_{\pi \in S_{d_h}} \sum_{i=1}^{d_h} C_{i,\pi(i)}^{(l)}, \tag{11}$$

where $\pi_{\text{dep}}^{(l)}(i)$ is the neuron in the original ordering of block l matched to position i of block $l-1$. The corresponding permutation matrix is defined by

$$\left(P_{\text{dep}}^{(l)} \right)_{i,j} = \begin{cases} 1, & j = \pi_{\text{dep}}^{(l)}(i), \\ 0, & \text{otherwise.} \end{cases} \tag{12}$$

The current MLP parameters are updated as

$$W_1^{(l)} \leftarrow P_{\text{dep}}^{(l)} W_1^{(l)}, \qquad W_2^{(l)} \leftarrow W_2^{(l)} \left(P_{\text{dep}}^{(l)} \right)^\top, \tag{13}$$

and, when the hidden bias is present,

$$b_1^{(l)} \leftarrow P_{\text{dep}}^{(l)} b_1^{(l)}. \tag{14}$$

Under the standard MLP structure, whose intermediate activation is shared and applied element-wise across hidden coordinates, these coupled updates preserve the deterministic MLP mapping.

Algorithm 1 Depth-wise inter-layer neuron alignment

Require: Ordered MLP blocks $\mathcal{M}^{(l)}{}_{l=1}^{L}$ with common embedding dimension d and hidden width d_h; stabilizer $\epsilon = 10^{-8}$
Ensure: MLP blocks sequentially aligned with their immediate predecessors

function DESCRIPTOR(W_1, W_2)
  $F \leftarrow \left[W_1, W_2^\top\right]$
  for $k \leftarrow 1$ **to** $2d$ **do**
    $\widetilde{F}_{:,k} \leftarrow \left(F_{:,k} - \mu_k\right) / \left(\sigma_k + \epsilon\right)$
  end for
  return $\widetilde{F}$
end function

for $l \leftarrow 2$ **to** L **do**
  $\widetilde{F}^{(l-1)} \leftarrow$ DESCRIPTOR($W_1^{(l-1)}, W_2^{(l-1)}$)
  $\widetilde{F}^{(l)} \leftarrow$ DESCRIPTOR($W_1^{(l)}, W_2^{(l)}$)
  for $i \leftarrow 1$ **to** d_h **do**
    for $j \leftarrow 1$ **to** d_h **do**
      $C_{i,j}^{(l)} \leftarrow \left|\widetilde{F}_i^{(l-1)} - \widetilde{F}_j^{(l)}\right|_2$
    end for
  end for
  $\pi_{\text{dep}}^{(l)} \leftarrow$ HUNGARIAN($C^{(l)}$) ▷ $\pi_{\text{dep}}^{(l)}(i)$ is matched to reference position i
  Construct $P_{\text{dep}}^{(l)}$ such that $\left(P_{\text{dep}}^{(l)}\right)_{i,\pi_{\text{dep}}^{(l)}(i)} = 1$
  $W_1^{(l)} \leftarrow P_{\text{dep}}^{(l)} W_1^{(l)}$
  if $b_1^{(l)}$ exists **then**
    $b_1^{(l)} \leftarrow P_{\text{dep}}^{(l)} b_1^{(l)}$
  end if
  $W_2^{(l)} \leftarrow W_2^{(l)} \left(P_{\text{dep}}^{(l)}\right)^\top$ ▷ The output bias, if present, is unchanged
end for
return $\mathcal{M}^{(l)}{}_{l=1}^{L}$

Processing the blocks from shallow to deep ensures that each current block is matched against a predecessor that has already been aligned within the same depth-wise pass. Algorithm 1 summarizes the complete procedure.

For architectures containing additional parameters whose axes index the same hidden coordinate, such as channel-wise activation parameters or the affine parameters of a hidden normalization layer, the same permutation must be propagated to the corresponding axes. Non-parameterized operations must also be permutation equivariant for the deterministic block mapping to remain unchanged.

A.2 MODULE-AWARE WIDTH-WISE ANCHORED JOINT PERMUTATION

Width-wise alignment optimizes the local ordering of functionally symmetric coordinates within each module. Given two coupled matrices $A \in \mathbb{R}^{n \times d_a}$ and $B \in \mathbb{R}^{d_b \times n}$, AJP constructs the joint descriptor

$$F_i = \left[A[i,:],\ (B[:,i])^\top\right], \qquad i = 1, \ldots, n. \tag{15}$$

It then computes the Euclidean distance matrix between all descriptor pairs and selects the maximum-energy descriptor as a fixed anchor:

$$\pi(1) = \arg\max_i \|F_i\|_2 . \tag{16}$$

Starting from this anchor, a nearest-neighbor traversal produces an initial Hamiltonian path. The path is subsequently refined by 2-opt updates. A subpath is reversed only when the corresponding edge exchange strictly decreases the total adjacent distance. The first position is excluded from reversal,

Algorithm 2 Anchored Joint Permutation (AJP) solver

Require: Coupled matrices $A \in \mathbb{R}^{n \times d_a}$ and $B \in \mathbb{R}^{d_b \times n}$; maximum number of 2-opt passes T_2
Ensure: Permutation $\pi \in S_n$ whose first position holds the anchor coordinate

for $i \leftarrow 1$ **to** n **do**
  $F_i \leftarrow \left[\, A[i,:],\ (B[:,i])^\top \,\right]$ ▷ Eq. 15
end for
$D_{i,j} \leftarrow \left\| F_i - F_j \right\|_2$ **for all** i, j
$\pi(1) \leftarrow \arg\max_i \|F_i\|_2$; $\mathcal{V} \leftarrow \{\pi(1)\}$ ▷ Eq. 16; anchor stays fixed
for $t \leftarrow 2$ **to** n **do**
  $\pi(t) \leftarrow \arg\min_{j \notin \mathcal{V}} D_{\pi(t-1),\, j}$; $\mathcal{V} \leftarrow \mathcal{V} \cup \{\pi(t)\}$ ▷ greedy nearest-neighbor path
end for
$p \leftarrow 0$; *improved* $\leftarrow$ **true**
while *improved* **and** $p < T_2$ **do**
  improved $\leftarrow$ **false**; $p \leftarrow p + 1$
  for $i \leftarrow 2$ **to** $n - 1$ **do** ▷ position 1 excluded, preserving the anchor
    for $j \leftarrow i + 1$ **to** n **do**
      $\Delta \leftarrow D_{\pi(i-1),\pi(j)} + D_{\pi(i),\pi(j+1)} - D_{\pi(i-1),\pi(i)} - D_{\pi(j),\pi(j+1)}$ ▷ terms with index $j+1$ dropped when $j = n$
      if $\Delta < 0$ **then**
        reverse the subpath $\pi(i{:}j)$; *improved* $\leftarrow$ **true**
      end if
    end for
  end for
end while
return π

thereby preserving the selected anchor. Refinement terminates when no improving exchange remains or when the maximum number of 2-opt passes is reached. Algorithm 2 states the solver.

The same AJP solver is used for FFN and attention modules, but their coupled coordinates and parameter updates differ. In an FFN, one permutation jointly acts on W_{fc1} and W_{fc2}. In multi-head attention, two independent permutations are required: π_{QK} jointly reorders the query and key coordinates, whereas π_{VO} jointly reorders the value and output-projection coordinates. Each attention permutation is shared across all heads within the current attention module, while different attention modules compute their own permutations. Algorithm 3 summarizes the complete module-aware procedure.

The attention implementation assumes the conventional fused QKV layout $[W_q; W_k; W_v]$, a common head width within each module, and $d_{\text{embed}} = H\, d_{\text{head}}$. The head width is read from the RPE module when this metadata is available. Otherwise, it must be supplied by the attention architecture; the current implementation uses 64 as its fallback value. For correctness across arbitrary architectures, the head width should be obtained directly from the attention-module configuration rather than inferred from a fixed default.

A.3 Alternating Depth–Width Integration

Depth-wise and width-wise alignment are not independent. Conversely, subsequent depth-wise matching can change the within-layer MLP neighborhoods produced by AJP. We consequently alternate the two procedures and terminate every round with depth-wise realignment.

Let $\mathcal{A}_{\text{dep}}$ denote the procedure in Algorithm 1 and $\mathcal{A}_{\text{wid}}$ denote the procedure in Algorithm 3. Starting from the source-model weights W, we define

$$W^{(0)} = \mathcal{A}_{\text{dep}}(W), \qquad W^{(t)} = \mathcal{A}_{\text{dep}}\left(\mathcal{A}_{\text{wid}}\left(W^{(t-1)}\right)\right), \quad t = 1, \ldots, T. \tag{17}$$

Thus, the first depth-wise pass establishes an initial layer-to-layer reference, and every width-wise update is followed by a new depth-wise pass. This ordering ensures that the final tensor is always produced by inter-layer realignment rather than by an unconstrained layer-specific width update. We use a predefined number T of alternating rounds; no gradient-based optimization is introduced.

Algorithm 3 Module-aware width-wise alignment

Require: Source model with modules $\{\mathcal{N}^{(m)}\}_{m=1}^{M}$ traversed from shallow to deep; embedding width d_{embed}; fallback head width $d_{\text{head}}^{\text{def}} = 64$
Ensure: Width-aligned source model

for $m \leftarrow 1$ **to** M **do**
  if $\mathcal{N}^{(m)}$ exposes an FFN pair $\big(W_{fc1}, W_{fc2}\big)$ **then**
    $\pi_{\text{FFN}} \leftarrow \text{AJP}(W_{fc1},\, W_{fc2})$ ▷ one descriptor per hidden neuron
    $W_{fc1} \leftarrow P_{\text{FFN}} W_{fc1}; \quad W_{fc2} \leftarrow W_{fc2} P_{\text{FFN}}^{\top}$
    if b_{fc1} exists **then**
      $b_{fc1} \leftarrow P_{\text{FFN}} b_{fc1}$
    end if
  else if $\mathcal{N}^{(m)}$ exposes an attention pair $\big(W_{qkv}, W_o\big)$ **then**
    $\big[W_q; W_k; W_v\big] \leftarrow W_{qkv}$ ▷ conventional fused QKV layout
    $d_{\text{head}} \leftarrow$ head width from the RPE metadata, the attention configuration, or $d_{\text{head}}^{\text{def}}$
    $H \leftarrow d_{\text{embed}} / d_{\text{head}}$
    reshape W_q, W_k, W_v into head-wise tensors $\big\{W_q^{(h)}, W_k^{(h)}, W_v^{(h)}\big\}_{h=1}^{H}$ and W_o into $\big\{W_o^{(h)}\big\}_{h=1}^{H}$
    for $c \leftarrow 1$ **to** d_{head} **do**
      $Q_{\text{stack}}[c,:] \leftarrow \big[\, W_q^{(1)}[c,:], \ldots, W_q^{(H)}[c,:] \,\big]$ ▷ aggregate coordinate c across heads
      $K_{\text{stack}}[c,:] \leftarrow \big[\, W_k^{(1)}[c,:], \ldots, W_k^{(H)}[c,:] \,\big]$
      $V_{\text{stack}}[c,:] \leftarrow \big[\, W_v^{(1)}[c,:], \ldots, W_v^{(H)}[c,:] \,\big]$
      $O_{\text{stack}}[:,c] \leftarrow \big[\, W_o^{(1)}[:,c]; \ldots; W_o^{(H)}[:,c] \,\big]$
    end for
    $\pi_{QK} \leftarrow \text{AJP}(Q_{\text{stack}},\, K_{\text{stack}}^{\top})$ ▷ $\pi_{QK} \in S_{d_{\text{head}}}$, shared by all heads
    $\pi_{VO} \leftarrow \text{AJP}(V_{\text{stack}},\, O_{\text{stack}})$ ▷ $\pi_{VO} \in S_{d_{\text{head}}}$, computed independently
    for $h \leftarrow 1$ **to** H **do**
      $W_q^{(h)} \leftarrow P_{QK} W_q^{(h)}; \quad W_k^{(h)} \leftarrow P_{QK} W_k^{(h)}$
      $W_v^{(h)} \leftarrow P_{VO} W_v^{(h)}; \quad W_o^{(h)} \leftarrow W_o^{(h)} P_{VO}^{\top}$
      apply P_{QK} to $b_q^{(h)}, b_k^{(h)}$ and P_{VO} to $b_v^{(h)}$ when the QKV biases are present
    end for
    reassemble $W_{qkv} \leftarrow \big[W_q; W_k; W_v\big]$
    if $\mathcal{N}^{(m)}$ has learnable RPE parameters **then**
      apply π_{QK} to every within-head coordinate axis of rpe_q and rpe_k
      apply π_{VO} to every within-head coordinate axis of rpe_v ▷ keeps RPE synchronized with Q, K, V
    end if
  end if
end for
return the width-aligned model

After the final round, the aligned tensor $\widehat{W} = W^{(T)}$ is passed once to the FRONT frequency-domain transfer stage. Corresponding parameters are normalized and transformed along their spatial dimensions; coefficients associated with the same module position are then stacked and transformed along the depth dimension. The retained low-frequency coefficients are resized to the target architecture, transformed back by the inverse DCT, denormalized, and copied to the compatible target parameters. This transfer stage is executed only after SPA has completed and is not included inside the alternating loop. Algorithm 4 gives the complete execution order.

Every individual permutation in Algorithms 1 and 3 is propagated through its coupled parameters. Hence, the SPA stage preserves the source-model mapping under the stated architectural assumptions. The alternating procedure changes only the coordinate organization of the source weights; frequency truncation and architecture resizing occur afterward for the knowledge transfer.

Algorithm 4 Alternating SPA-FRONT integration

Require: Pre-trained source weights W; number of alternating rounds T; target architecture with depth L_{tgt} and width d_{tgt}; retention configuration ρ
Ensure: Initialized target model W_{tgt}

// Stage 1: structural permutation alignment (SPA)
identify all MLP, attention, and optional RPE modules of W supported by $\mathcal{A}_{\mathrm{dep}}$ and $\mathcal{A}_{\mathrm{wid}}$
$W^{(0)} \leftarrow \mathcal{A}_{\mathrm{dep}}(W)$ ▷ Algorithm 1; block 1 fixes the initial reference
for $t \leftarrow 1$ **to** T **do**
  $\widetilde{W}^{(t)} \leftarrow \mathcal{A}_{\mathrm{wid}}\big(W^{(t-1)}\big)$ ▷ Algorithm 3; per-module orderings, RPE synchronized
  $W^{(t)} \leftarrow \mathcal{A}_{\mathrm{dep}}\big(\widetilde{W}^{(t)}\big)$ ▷ repairs inter-layer correspondence under the new within-layer orderings
end for
$\widehat{W} \leftarrow W^{(T)}$ ▷ Eq. 17; every round ends with depth alignment

// Stage 2: frequency-domain transfer (FRONT), executed once
for all compatible parameters $\theta \in \widehat{W}$ **do**
  $\mu_\theta \leftarrow \mathrm{mean}(\theta)$; $\sigma_\theta \leftarrow \mathrm{std}(\theta)$
  $\bar{\theta} \leftarrow \big(\theta - \mu_\theta\big)/\sigma_\theta$
  $\Phi_\theta \leftarrow \mathrm{DCT}_{\mathrm{spatial}}\big(\bar{\theta}\big)$
end for
for all module positions p shared across depth **do**
  $\Phi_p \leftarrow \mathrm{stack}\big(\{\Phi_\theta : \theta \text{ at position } p\}\big)$
  $\widehat{\Phi}_p \leftarrow \mathrm{DCT}_{\mathrm{depth}}\big(\Phi_p\big)$ ▷ $\widehat{\Phi} = \mathcal{D}(\widehat{W})$
  $\widehat{\Phi}_p \leftarrow \textsc{RetainLowFrequency}(\widehat{\Phi}_p,\ \rho)$
  $\widehat{\Phi}_p \leftarrow \textsc{Resize}(\widehat{\Phi}_p,\ L_{\mathrm{tgt}},\ d_{\mathrm{tgt}})$ ▷ truncation, zero-padding, or the specified transfer rule
end for
for all target parameters $\theta_{\mathrm{tgt}} \in W_{\mathrm{tgt}}$ **do**
  if θ_{tgt} is incompatible or explicitly skipped **then**
    continue ▷ e.g. classification heads of mismatched shape
  end if
  $\bar{\theta}_{\mathrm{tgt}} \leftarrow \mathrm{IDCT}\big(\widehat{\Phi} \text{ entry of } \theta_{\mathrm{tgt}}\big)$
  $\theta_{\mathrm{tgt}} \leftarrow \sigma_\theta\, \bar{\theta}_{\mathrm{tgt}} + \mu_\theta$ ▷ restore the recorded statistics
end for
return W_{tgt}

B Permutation Symmetry

B.1 Function-Preserving Permutations in Transformer Blocks

Figure 5 illustrates how the learned permutations are propagated through the coupled projections of a Transformer block. For clarity, biases are omitted from the figure but are permuted together with the corresponding output coordinates. Consider first a single attention head. We apply the same permutation P_{qk} to the query and key coordinates:

$$W_q' = P_{qk} W_q, \qquad W_k' = P_{qk} W_k. \tag{18}$$

For any query and key vectors q and k, the resulting attention score is unchanged because

$$(P_{qk} q)^\top (P_{qk} k) = q^\top P_{qk}^\top P_{qk} k = q^\top k. \tag{19}$$

Thus, the softmax attention coefficients remain identical. The value coordinates are independently reordered by P_{vo}, while the inverse permutation is absorbed into the input dimension of the output projection:

$$W_v' = P_{vo} W_v, \qquad W_o' = W_o P_{vo}^\top. \tag{20}$$

Since attention aggregation is a linear combination of the value vectors, the permuted values produce $z' = P_{vo} z$. Consequently,

$$W_o' z' = W_o P_{vo}^\top P_{vo} z = W_o z, \tag{21}$$

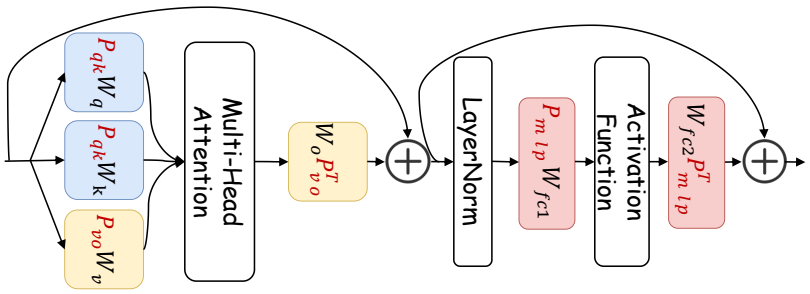


Figure 5: **Function-preserving permutations in a Transformer block.** The same permutation P_{qk} is applied to the output coordinates of the query and key projections, preserving their inner products. The value coordinates are reordered by P_{vo}, with the inverse permutation absorbed into the input dimension of the output projection. Similarly, P_{mlp} is applied to the hidden coordinates of the first MLP projection and canceled by the corresponding inverse permutation in the second projection.

and the complete attention output is preserved. In multi-head attention, the same within-head permutation is shared across all heads in the corresponding module. When learnable relative positional encoding is present, the same coordinate permutations are also propagated to the associated RPE parameters to preserve their correspondence with Q, K, and V.

For the MLP, the hidden coordinates are reordered using P_{mlp}:

$$\begin{aligned} W'_{fc1} &= P_{\text{mlp}} W_{fc1}, \qquad b'_{fc1} = P_{\text{mlp}} b_{fc1}, \\ W'_{fc2} &= W_{fc2} P_{\text{mlp}}^{\top}. \end{aligned} \tag{22}$$

Because the activation function $\sigma(\cdot)$ is applied element-wise, it is permutation equivariant, i.e., $\sigma(P_{\text{mlp}} u) = P_{\text{mlp}} \sigma(u)$. Therefore,

$$\begin{aligned} W'_{fc2}\, \sigma\big(W'_{fc1} x + b'_{fc1}\big) &= W_{fc2} P_{\text{mlp}}^{\top} P_{\text{mlp}}\, \sigma(W_{fc1} x + b_{fc1}) \\ &= W_{fc2}\, \sigma(W_{fc1} x + b_{fc1})\,. \end{aligned} \tag{23}$$

The output biases are unaffected because the applied permutations only reorder internal hidden coordinates and are canceled by the corresponding inverse transformations in adjacent projections. Residual connections remain unchanged because the external feature dimensions of each sublayer are preserved. Normalization layers are handled according to their coordinate transformations as discussed in Appendix B.2.

B.2 Extension to Normalization Layers, Activations, and Gauge Constraints

The permutation transformations described above apply to hidden coordinates that are internal to a computational block. In practice, modern Transformer architectures contain normalization and nonlinear activation modules that operate on these hidden dimensions. The permutation equivalence is preserved as long as all coordinate-dependent parameters are transformed consistently.

Normalization layers. For LayerNorm,

$$\mathrm{LN}(x) = \gamma \odot \frac{x - \mu(x)}{\sigma(x)} + \beta, \tag{24}$$

a permutation P applied to the feature dimension yields

$$\gamma' = P\gamma, \qquad \beta' = P\beta. \tag{25}$$

The normalized representation remains equivalent because the mean and variance are invariant to coordinate reordering. Similarly, for RMSNorm,

$$\mathrm{RMSNorm}(x) = \gamma \odot \frac{x}{\mathrm{RMS}(x)}, \tag{26}$$

the scale parameter is transformed as

$$\gamma' = P\gamma. \tag{27}$$

Activation functions. The permutation symmetry considered in SPA relies on the permutation equivariance of element-wise nonlinearities. For a permutation matrix P and an element-wise activation function $\sigma(\cdot)$,

$$\sigma(Px) = P\sigma(x). \tag{28}$$

Therefore, common Transformer activations, including GELU, ReLU, and the element-wise gating operations in SwiGLU, preserve permutation equivalence when the surrounding linear projections are transformed consistently. This property enables the hidden-unit permutations described for MLP blocks, where the first projection and activation output are reordered by the same permutation and the inverse permutation is absorbed into the following projection.

Extended Gauge Freedoms and Signed Permutations. The symmetry exploited by SPA corresponds to the standard permutation group over hidden units, which is broadly applicable to Transformer architectures. In some architectures, this symmetry can be extended to a larger signed-permutation group by additionally allowing coordinate-wise sign flips. Such extensions provide additional reparameterization freedom, but require stronger conditions on the nonlinearities and parameter coupling. In particular, a sign transformation requires

$$\sigma(-x) = \pm\sigma(x) \tag{29}$$

or an equivalent compensation mechanism to preserve the represented function. Common Transformer activations, such as GELU and SwiGLU, do not generally satisfy these sign-equivariance conditions. Therefore, SPA focuses on the more general permutation symmetry and leaves the exploitation of signed-permutation freedoms to future extensions.

Consequently, SPA applies to architectures containing LayerNorm, RMSNorm, and standard Transformer nonlinearities as long as permutations are consistently propagated through all modules sharing the same hidden coordinates. More generally, any reparameterization used by SPA must correspond to a valid symmetry of the architecture; transformations that lack a consistent propagation rule across coordinate-dependent modules are not considered.

B.3 Permutation Constraints for Attention Variants

The validity of attention permutations depends on the positional encoding mechanism and the parameter sharing pattern of the attention module. The synchronization rules described in this work are directly applicable to standard multi-head attention, where query, key, and value projections have explicit coordinate correspondence.

For attention mechanisms with additional positional structures, the permutation must be propagated to all parameters defined on the permuted coordinates. For relative positional encoding (RPE), the corresponding positional parameters are reordered together with the query and key dimensions to preserve their coordinate correspondence. For rotary positional embeddings (RoPE), the permutation is constrained to preserve the paired dimensions used by the rotary transformation; arbitrary channel shuffling would break the rotational structure and is therefore not permitted. For ALiBi, permutations over attention heads are valid only when the associated bias slopes are reordered consistently with the corresponding heads.

Attention variants with asymmetric parameter sharing require additional constraints. In multi-query attention (MQA) and grouped-query attention (GQA), query heads and key-value heads do not share identical parameterizations. Therefore, permutations are applied only within their respective groups: query permutations are synchronized among query heads, while key-value permutations are restricted to the shared key-value projections. If a specific architecture contains coordinate-dependent modules without an equivalent permutation propagation rule, the corresponding dimensions are excluded from SPA alignment.

Overall, SPA operates on the subset of hidden dimensions that admit consistent permutation propagation. This constraint ensures that the resulting reparameterization remains function-preserving while allowing SPA to explore valid alternatives within the permutation-equivalence class.

C Theoretical Motivation for Spectral Preconditioning

This section provides a concise theoretical motivation for SPA from the perspective of spectral preconditioning. We focus on three properties: (i) reducing local structural variation suppresses high-frequency DCT components, (ii) stronger low-frequency concentration reduces fixed-budget spectral approximation error, and (iii) function-preserving permutations can alter spectral energy distributions while leaving the represented network unchanged.

C.1 Structural Variation and Low-Frequency Concentration

SPA is designed to reduce local structural variation along the dimensions of the stacked weight tensor. The detailed depth-wise and width-wise alignment procedures are given in Section A. Here, we explain why reducing such variation is naturally connected to stronger low-frequency concentration.

Consider a one-dimensional sequence $\mathbf{x} = [x_0, \ldots, x_{n-1}]^\top$ and define its discrete variation as

$$\mathcal{V}(\mathbf{x}) = \sum_{i=0}^{n-2} (x_{i+1} - x_i)^2. \tag{30}$$

Let $\widehat{\mathbf{x}} = \mathbf{D}\mathbf{x}$ denote its orthonormal DCT representation. The discrete difference operator is diagonalized by the DCT basis, giving

$$\mathcal{V}(\mathbf{x}) = \sum_{k=0}^{n-1} \lambda_k |\widehat{x}_k|^2, \qquad \lambda_k = 4 \sin^2\left(\frac{\pi k}{2n}\right), \tag{31}$$

where λ_k increases monotonically with the frequency index k. Therefore, high-frequency coefficients are penalized more strongly by local variation than low-frequency coefficients.

For a multi-dimensional weight tensor, the separability of the DCT yields the same relation along each transformed dimension. Consequently, reducing variation between neighboring entries decreases a frequency-weighted measure of spectral energy that emphasizes high-frequency components. SPA does not directly optimize the DCT coefficients; instead, its depth-wise and width-wise alignment objectives act as tractable surrogates for reducing local variation in the reorganized weight tensor. This provides a spectral interpretation of SPA: smoother parameter organization is expected to shift a larger fraction of spectral energy toward lower frequencies.

C.2 Low-Frequency Concentration under a Fixed Coefficient Budget

We next clarify why low-frequency concentration is relevant when only a fixed set of coefficients can be retained. Let $\mathbf{W}$ denote a weight tensor, let $\widehat{\mathbf{W}} = \mathcal{D}(\mathbf{W})$ be its orthonormal DCT representation, and let $\mathbf{M}$ be a binary mask selecting the retained low-frequency coefficients. The truncated reconstruction is

$$\widetilde{\mathbf{W}} = \mathcal{D}^{-1}\left(\mathbf{M} \odot \widehat{\mathbf{W}}\right). \tag{32}$$

Because the DCT is orthonormal, Parseval's identity gives

$$\|\mathbf{W} - \widetilde{\mathbf{W}}\|_F^2 = \|(\mathbf{1} - \mathbf{M}) \odot \widehat{\mathbf{W}}\|_F^2. \tag{33}$$

Equivalently,

$$\frac{\|\mathbf{W} - \widetilde{\mathbf{W}}\|_F^2}{\|\mathbf{W}\|_F^2} = 1 - \frac{\|\mathbf{M} \odot \widehat{\mathbf{W}}\|_F^2}{\|\widehat{\mathbf{W}}\|_F^2}. \tag{34}$$

Thus, for a fixed coefficient mask, increasing the fraction of spectral energy contained in the retained low-frequency region directly decreases the normalized reconstruction error of the weight tensor. This gives a precise interpretation of the low-frequency ratio used in our experiments: it measures how well a fixed spectral budget can approximate the original parameter tensor.

Importantly, this result concerns approximation fidelity rather than semantic information. A higher low-frequency energy fraction does not by itself imply that the retained coefficients contain more transferable knowledge. The usefulness of the resulting approximation for variable-sized initialization is therefore evaluated empirically in our experiments.

C.3 Function-Preserving Permutations and Spectral Redistribution

As shown in Appendix B, coordinated permutations of hidden units and their coupled parameters preserve the network function. For an admissible permutation operator $\mathcal{P}$ applied consistently to all coupled parameters,

$$f_{\mathcal{P}(\theta)}(\mathbf{x}) = f_\theta(\mathbf{x}). \tag{35}$$

Equivalently, a pre-trained model does not come with a single parameterization but with an orbit

$$\mathcal{O}(\theta) = \{\, \mathcal{P}(\theta) \;:\; \mathcal{P} \in \mathcal{G} \,\}, \tag{36}$$

where $\mathcal{G}$ is the group of admissible permutations; every member of $\mathcal{O}(\theta)$ computes the same function. SPA searches this orbit rather than the space of functions or the space of trained models, using the spectral organization of the parameters as its search objective.

However, functional equivalence does not imply spectral equivalence. Let $\mathbf{P}$ denote the matrix representation of a permutation of a weight slice $\mathbf{x} = [x_0, \ldots, x_{n-1}]^\top$, and let $\mathbf{D}$ be the orthonormal DCT-II matrix of the same size. The DCT basis is not invariant to reordering: for a generic permutation,

$$|\mathbf{DPx}|^2 \neq |\mathbf{Dx}|^2 \tag{37}$$

element-wise, so different members of $\mathcal{O}(\theta)$ generally exhibit different spectral energy distributions. We stress that equation 37 concerns the magnitude spectrum and is *not* implied by the mere non-commutativity of $\mathbf{P}$ and $\mathbf{D}$; the sharp criterion follows.

Spectrally inert permutations. A permutation leaves the magnitude spectrum exactly unchanged if and only if

$$\mathbf{DP} = \mathbf{SD} \tag{38}$$

for some diagonal sign matrix $\mathbf{S} = \mathrm{diag}(s_0, \ldots, s_{n-1})$, $s_k \in \{\pm 1\}$. Necessity follows because $|(\mathbf{DPx})_k| = |(\mathbf{Dx})_k|$ for all $\mathbf{x}$ forces the k-th rows of $\mathbf{DP}$ and $\mathbf{D}$ to agree up to sign; sufficiency is immediate from $|\mathbf{SDx}|^2 = |\mathbf{Dx}|^2$.

One such permutation is index reversal, $(\mathbf{Rx})_i = x_{n-1-i}$. Using the DCT-II identity $\cos\big(\pi k - \pi k(2i+1)/2n\big) = (-1)^k \cos\big(\pi k(2i+1)/2n\big)$, we obtain

$$\mathbf{DR} = \mathbf{S}_\pm \mathbf{D}, \qquad \mathbf{S}_\pm = \mathrm{diag}\big((-1)^k\big), \tag{39}$$

hence $|\mathbf{DRx}|^2 = |\mathbf{Dx}|^2$ exactly. Index reversal is admissible: applying it consistently to all coupled parameters preserves the network function by equation 35.

Two consequences deserve emphasis. First, spectral redistribution is not a generic property of the orbit—it is governed by how far $\mathbf{P}$ departs from equation 38, and $\mathbf{R}$ is an explicit member of $\mathcal{O}(\theta)$ for which the low-frequency ratio and the truncation error of Appendix C.2 coincide exactly with those of the original parameterization. Second, because spectrally inert elements exist, the organization that is optimal for extraction is not unique: any two solutions differing by an element of the inert set must yield identical spectral statistics and identical downstream behaviour, a property we use as an exact reproducibility check.

For a multi-dimensional weight tensor, the DCT is applied separably along each transformed axis, and the statements above apply to the permutation restricted to the corresponding axis.

At the same time, both permutation matrices and the orthonormal DCT preserve the total ℓ_2 energy:

$$\|\mathbf{DPx}\|_2^2 = \|\mathbf{Px}\|_2^2 = \|\mathbf{x}\|_2^2. \tag{40}$$

Therefore, at a fixed total energy budget, a permutation neither creates nor removes parameter energy; it can only redistribute that energy across frequency components. SPA exploits this degree of freedom to search $\mathcal{O}(\theta)$ for a functionally equivalent parameterization whose organization is more compatible with subsequent low-frequency extraction.

Taken together, these observations motivate SPA as a structural spectral preconditioner: function-preserving permutations can modify the spectral distribution without changing the source network; reducing local structural variation favours a lower spectral centroid, and—as we verify empirically—a higher low-frequency ratio; and, under a fixed coefficient mask, a higher low-frequency ratio lowers the corresponding truncation error by the identity of Appendix C.2.

D TRAINING DETAILS

D.1 SUPPLEMENTARY EXPERIMENTAL SETUP

We evaluate SPA-FRONT across multiple architectures, modalities, and transfer settings. For vision experiments, we study model scaling by varying the depth and width of DeiT architectures in Section 4.2.1, followed by cross-dataset transfer evaluations in Section 4.2.2. For language experiments, we perform cross-scale initialization on representative Transformer backbones and report downstream results on GLUE in Section 4.3. Additional ablations are provided in Section 4.4.

Datasets. For vision tasks, all source models are pre-trained on ImageNet-1K, which is also used for the scaling experiments. Cross-dataset transfer is evaluated on seven downstream classification benchmarks. For language modeling, BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019) are trained on English Wikipedia, while GPT-2 (Radford et al., 2019) is trained on the concatenation of English Wikipedia and the Toronto Book Corpus (Devlin et al., 2019). Downstream language performance is measured on GLUE (Wang et al., 2018).

Architectural Details. Our experiments cover both vision and language architectures. For vision models, we use publicly available DeiT-Ti/S/B checkpoints pre-trained on ImageNet-1K as source models. To evaluate scalability, target DeiT models are constructed with depths ranging from 4 to 12 layers and widths ranging from 6 to 24 attention heads, corresponding to hidden dimensions from 384 to 1536. For cross-architecture generalization, ResNet-50 and ResNet-152 are initialized from a ResNet-101 source model. For language models, we initialize 6-layer, 384-dimensional Small models using knowledge extracted from their corresponding 12-layer, 768-dimensional Base counterparts. This setting is applied to BERT, RoBERTa, and GPT-2.

Baselines. For vision tasks, we compare SPA-FRONT with two groups of initialization methods. The first group consists of direct initialization approaches that use existing ImageNet-1K pre-trained models without additional optimization, including He-Init (Chen et al., 2021), Mimetic-Init (Trockman & Kolter, 2023), Wt Select (Xu et al., 2023), Heur-LG (Wang et al., 2022), Cluster-LG (Wang et al., 2024), LiGO (Wang et al., 2023a), and FRONT (Shen et al., 2026a). The second group includes methods that require extra training to produce transferable parameters, such as GHN-3 (Knyazev et al., 2023), Share-Init (Lan et al., 2019), Auto-LG (Wang et al., 2023b), TLEG (Xia et al., 2024), and WAVE (Feng et al., 2025b). For language tasks, we use four baselines: training the Small model from scratch, knowledge distillation (Hinton et al., 2015), Wt Select and FRONT, where the corresponding Base model serves as the teacher or source model.

Evaluation Metrics. Top-1 accuracy is used as the primary metric for vision experiments. We additionally report convergence efficiency and parameter transfer efficiency where applicable.

D.2 VISION TASKS

D.2.1 HYPERPARAMETERS

Table 8 reports the optimization settings used for retraining-based methods on ImageNet-1K. Table 9 summarizes the fine-tuning configurations for downstream vision datasets, including batch size, learning rate, training length, augmentation, and optimizer settings.

D.2.2 DETAILS OF DOWNSTREAM DATASETS

The downstream classification benchmarks include Oxford Flowers (Nilsback & Zisserman, 2008), CUB-200-2011 (Wah et al., 2011), Stanford Cars (Gebru et al., 2017), CIFAR-10 and CIFAR-100 (Krizhevsky & Hinton, 2009), Food-101 (Bossard et al., 2014), and iNaturalist-2019 (Tan et al., 2019). Dataset statistics are summarized in Table 10.

D.3 LANGUAGE TASKS

We evaluate SPA-FRONT on cross-scale initialization for language models. Specifically, each 6-layer, 384-dimensional Small model is initialized from its corresponding 12-layer, 768-dimensional Base model. Experiments are conducted on BERT, RoBERTa, and GPT-2. The baselines include training

Table 8: Hyper-parameters for Retraining methods 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

Table 9: Hyperparameters for neural networks trained on downstream datasets.

Dataset	Batch Size	Epochs DeiT/ResNet	Learning Rate	Drop Last	Warmup Epochs	Droppath Rate	Color Jitter	Auto Augment	Random Erase	Mixup	Cutmix	Scheduler	Optimizer
Oxford Flowers	512	300/100	3e-4	False	0	0	0.4	rand-m9-mstd0.5-inc1	0.25	0	0	cosine	AdamW
CUB-200-2011	512	300/100	3e-4	False	0	0.1	0	rand-m9-mstd0.5-inc1	0.25	0	0	cosine	AdamW
Stanford Cars	512	300/100	3e-4	False	0	0.1	0	rand-m9-mstd0.5-inc1	0.25	0	0	cosine	AdamW
CIFAR10	512	300/100	5e-4	True	0	0.1	0.4	rand-m9-mstd0.5-inc1	0.25	0	0	cosine	AdamW
CIFAR100	512	300/100	5e-4	True	0	0.1	0.4	rand-m9-mstd0.5-inc1	0.25	0	0	cosine	AdamW
Food101	512	300/100	5e-4	True	0	0.1	0.4	rand-m9-mstd0.5-inc1	0.25	0	0	cosine	AdamW
iNat-2019	512	100/50	5e-4	True	0	0.1	0.4	rand-m9-mstd0.5-inc1	0.25	0	0	cosine	AdamW

from scratch and knowledge distillation (Hinton et al., 2015),Wt Select and FRONT, where the Base model acts as the teacher or the source.

BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019) are trained on English Wikipedia, while GPT-2 (Radford et al., 2019) uses the union of English Wikipedia and the Toronto Book Corpus. Following RoBERTa (Liu et al., 2019), next-sentence prediction is removed. BERT and RoBERTa use a fixed sequence length of 128. For BERT, the batch size is 256 and the learning rate is 2e-4. For RoBERTa, the batch size is 1024 and the learning rate is 8e-4. GPT-2 is trained with a batch size of 512, a sequence length of 1024, and a learning rate of 1e-3.

We further fine-tune the initialized BERT-S model on GLUE. Fine-tuning uses Adam with a learning rate of 2e-5 and a batch size of 32, and the number of epochs is selected from 3, 6, 10. SPA-FRONT-initialized models are directly fine-tuned on each downstream task, whereas knowledge distillation keeps the corresponding Base model as the teacher throughout training.

Saving Ratio. We use the saving ratio r to quantify the reduction in training computation required to reach a common target performance M. The target is defined separately for each task and architecture but is kept identical across all compared methods. For language-model pre-training, M denotes a target MLM or GEN loss, whereas for vision tasks, it denotes a target classification accuracy. Let ξ_{scratch} be the cumulative FLOPs required by training from scratch to first reach M, and let $\xi_{\mathcal{A}}$ denote the corresponding FLOPs required by an initialization method $\mathcal{A}$. The saving ratio is defined as

$$r(\mathcal{A}) = \frac{\xi_{\text{scratch}} - \xi_{\mathcal{A}}}{\xi_{\text{scratch}}} \times 100\%. \tag{41}$$

Table 10: Characteristics of downstream datasets.

Dataset	Classes	Total	Training	Testing
Oxford Flowers (Nilsback & Zisserman, 2008)	102	8,189	2,040	6,149
CUB-200-2011 (Wah et al., 2011)	200	11,788	5,994	5,794
Stanford Cars (Gebru et al., 2017)	196	16,185	8,144	8,041
CIFAR10 (Krizhevsky & Hinton, 2009)	10	60,000	50,000	10,000
CIFAR100 (Krizhevsky & Hinton, 2009)	100	60,000	50,000	10,000
Food101 (Bossard et al., 2014)	101	101,000	75,750	25,250
iNat-2019 (Tan et al., 2019)	1010	268,243	265,213	3,030

A positive r indicates that method $\mathcal{A}$ reaches the target with less computation than training from scratch, while $r = 0$ indicates no saving. All initialization methods use the same target architecture and per-epoch training configuration; therefore, the FLOPs saving ratio is equivalent to the relative reduction in the number of epochs required to reach the target accuracy.

D.4 Detailed Ablation on Depth-Wise Alignment Strategies

In this section, we provide the comprehensive definitions and extended mechanistic analysis for the depth-wise alignment strategies evaluated in Table 6.

To systematically evaluate the design space of inter-layer smoothing, we implemented the following four distinct alignment strategies:

- **Global Anchor (Global):** Computes a single global mean feature space across all layers, subsequently aligning each layer to this static anchor using cosine similarity.
- **Optimal Transport (OT):** Utilizes the Sinkhorn-Knopp algorithm to compute a soft probability matrix, performing alignment via continuous linear fusion rather than discrete permutation.
- **Spectral Sorting (Spectral):** Independently reorders neurons within each layer based solely on their spectral energy magnitudes.
- **Sequential Euclidean (Ours):** Our proposed method, which sequentially aligns layer l to $l-1$ via strictly bipartite hard permutation based on Euclidean distance.

As demonstrated in the main text, the Sequential Euclidean matching substantially outperforms the alternatives. We attribute this to the inherent mathematical and structural limitations of the suboptimal strategies. Specifically, the Global Anchor strategy severely underperforms because forcing all layers to align with a uniform static mean fundamentally disrupts the natural hierarchical progression of semantic features (e.g., the transition from low-level edges in shallow layers to high-level abstract semantics in deep layers). Although Optimal Transport (OT) exhibits the lowest execution time (0.20s), it yields the worst downstream accuracy. This degradation occurs because OT relies on a soft alignment mechanism: performing matrix multiplication with soft probability matrices intrinsically alters the exact weight magnitudes. Consequently, this continuous linear fusion corrupts the carefully learned pre-trained structural priors, violating the fundamental principle of lossless knowledge transfer. Furthermore, while Spectral Sorting successfully preserves the exact parameter values, sorting neurons independently by energy completely ignores the pairwise topological correlations between adjacent layers. This lack of inter-layer awareness results in suboptimal cross-layer transitions. By contrast, our Sequential Euclidean matching explicitly guarantees no feature fusion or information loss through strict binary permutation. By sequentially anchoring each layer to its immediate predecessor, it smooths structural transitions while perfectly respecting the natural depth-wise semantic hierarchy, ensuring maximum preservation of the transferred knowledge.

D.5 Detailed Ablation on Width-Wise Alignment Strategies

To systematically evaluate the design space of Hamiltonian path optimization across network widths, we compare the following variants:

- **Random:** A random baseline that selects a permutation-equivalent parameterization by uniformly shuffling the hidden-unit ordering along the width dimension. Although this reordering preserves the represented function, the resulting parameterization does not explicitly retain the spatial organization induced during pre-training. Its low-frequency ratio is substantially lower than that of the original pre-trained model, reducing the relative LR gain from FRONT's 22.33% to 8.29%. This demonstrates that functionally equivalent parameterizations can exhibit markedly different spectral properties, and that effective extraction requires a targeted search within the permutation-equivalence class rather than arbitrary reordering.
- **Smooth (1D Magnitude Sorting):** A naive spatial smoothing approach that independently sorts the coupled neurons in descending order based solely on their $\mathcal{L}_2$ norm magnitudes. While highly computationally efficient (0.49s CPU time) and capable of improving relative LR gain (67.78%), this 1D sorting fundamentally ignores the high-dimensional pairwise geometric distances between adjacent features, leaving significant structural optimization potential untapped.

- **JG (Joint Greedy):** A lightweight optimization heuristic that formulates the neuron sequence by simply selecting the nearest unvisited neighbor in the joint feature space $\mathcal{F}$. While it executes rapidly (0.79s CPU time) and improves relative LR gain to 76.56%, the greedy strategy is inherently susceptible to sub-optimal local minima, leading to jagged structural transitions at the tail end of the sequence.
- **TSP (Unanchored Hamiltonian Path):** A standard Traveling Salesperson Problem (TSP) formulation that utilizes both the greedy initialization and the 2-opt local search refinement to minimize the total adjacent Euclidean distance. However, it lacks a deterministic starting node constraint. While it successfully achieves high relative LR gain (81.82%), the arbitrary starting point randomly shifts the highest-energy representations away from the zero-frequency (DC) origin, rendering the subsequent low-pass truncation mathematically misaligned.
- **STSP (Symmetric/Closed Hamiltonian Cycle):** A variant of the TSP formulation that forces the sequence into a closed cycle (i.e., the last neuron connects back to the first neuron). Although this strict cyclic constraint yields the marginally highest relative LR gain (81.98%), creating a closed loop introduces a severe boundary discontinuity when the tensor is unrolled for the discrete cosine transform. This topological mismatch degrades downstream accuracy relative to open-path formulations.
- **AJP (Anchored Joint Permutation):** Our proposed optimal strategy. By strictly anchoring the maximum-norm joint feature to the zero-frequency (DC) origin ($\pi(1) = \arg\max_i \|F_i\|_2$) and optimizing the open Hamiltonian path via 2-opt refinement, AJP perfectly aligns the most critical feature representations with the subsequent low-pass extraction mask. Notably, the deterministic anchor effectively prunes the search space, allowing the 2-opt refinement to converge faster than the unanchored TSP (66.40s vs. 84.64s). Ultimately, this mathematically grounded topological preconditioning secures the highest accuracy across all evaluated width expansions (e.g., 72.7% on W768).

D.6 Definition of Low-Frequency Ratio and Relative LR Gain

We quantify spectral concentration using the *low-frequency ratio*, defined as the proportion of spectral energy retained by the low-frequency mask used for knowledge transfer. Let $\mathcal{G}$ denote the set of compatible parameter groups transferred by FRONT. For each group $p \in \mathcal{G}$, let Φ_p denote its frequency representation after the same normalization and DCT operations used during learngene extraction, and let $M_{\rho,p}$ denote the corresponding low-frequency retention mask. The low-frequency ratio is defined as

$$R_{\mathrm{LF}}(W) = \frac{\sum_{p\in\mathcal{G}} \|M_{\rho,p} \odot \Phi_p\|_F^2}{\sum_{p\in\mathcal{G}} \|\Phi_p\|_F^2}. \tag{42}$$

The numerator measures the spectral energy contained in the coefficients retained for knowledge transfer, whereas the denominator represents the total spectral energy of all transferred parameter groups. Therefore, $R_{\mathrm{LF}}(W)$ measures the fraction of transferable weight energy concentrated in the designated low-frequency region, rather than the absolute spectral magnitude.

We further report the relative LR gain with respect to He-Init, which serves as the random-initialization baseline. Since randomly initialized weights do not exhibit an organized spatial topology, their spectral energy is approximately distributed across frequency components without a pronounced low-frequency bias. For a given alignment strategy $\mathcal{A}$, the relative LR gain is defined as

$$\text{Relative LR Gain}(\mathcal{A}) = \frac{R_{\mathrm{LF}}(W_{\mathcal{A}}) - R_{\mathrm{LF}}(W_{\mathrm{He}})}{R_{\mathrm{LF}}(W_{\mathrm{He}})} \times 100\%, \tag{43}$$

where $W_{\mathcal{A}}$ denotes the weights obtained using strategy $\mathcal{A}$ and W_{He} denotes the randomly initialized weights produced by He-Init. Accordingly, a relative LR gain of 0% corresponds to the random-initialization baseline, while a positive value indicates that the considered strategy concentrates a larger fraction of spectral energy in the transferable low-frequency region.

Table 11: Effect of the number of alternating depth–width alignment rounds when transferring knowledge from BERT-B-L12 to BERT-S-L6.

Method	Alignment rounds	Steps					
		1K	5K	10K	50K	200K	400K
Scratch	—	1474.73	603.52	181.28	11.99	9.09	7.79
SPA-FRONT	1	28.34	13.64	11.62	9.17	7.59	6.83
SPA-FRONT	2	28.32	13.62	11.61	9.13	7.59	**6.82**
SPA-FRONT	5	**28.31**	**13.61**	**11.60**	**9.12**	**7.59**	**6.82**

E Additional Results

E.1 Effect of Alternating Alignment Rounds

We evaluate the effect of alternating depth–width alignment rounds when transferring knowledge from BERT-B-L12 to BERT-S-L6. All settings are identical except for the number of SPA rounds before FRONT initialization. Each round applies width-wise alignment followed by depth-wise realignment. We stop when the low-frequency energy no longer increases, which occurs after the third round; a five-round result with early stopping disabled is also reported for comparison. Table 11 presents the validation perplexity at different training steps.

A single alignment round substantially increases low-frequency energy and captures most of the downstream benefit, reducing the final perplexity from 7.79 to 6.83. The improvement is also evident at early training stages, indicating that the aligned initialization accelerates convergence. In contrast, additional rounds increase the low-frequency ratio by less than 0.1 percentage points and yield only marginal perplexity reductions, suggesting that the weight ordering becomes largely stable after the first round. We therefore use **one** round as the default setting for joint depth–width alignment.

E.2 Training Dynamics of Different Initialization Methods

To further complement the results in Table 1, we visualize the optimization trajectories of different initialization methods on ImageNet-1K. Figure 6 reports the accuracy curves of DeiT-Ti, DeiT-S, and DeiT-B under different target depths. We include representative training-free methods (SPA-FRONT and FRONT), retraining-based methods (WAVE, TLEG, Auto-LG, and Heru-LG), as well as training from scratch for comparison.

Across different model scales and target depths, SPA-FRONT consistently starts from a higher accuracy and maintains faster convergence throughout training. Compared with direct spectral extraction by FRONT, SPA-FRONT provides a more favorable initialization state by improving the structural organization of transferred weights before extraction. These training trajectories complement the final performance reported in Table 1, demonstrating that spatial permutation alignment not only improves transfer accuracy but also accelerates the subsequent optimization process.

F Scope and Outlook: SPA Beyond FRONT

The main paper develops SPA as a training-free, function-preserving preconditioner that reorganizes a pre-trained weight tensor by permutations of its shared axes, so as to increase the low-frequency energy concentration on which FRONT's DCT-based spectral extraction relies. A natural question is how far the same principle extends, and in which direction the learngene literature might build on it. This appendix is deliberately forward-looking: rather than adjudicating individual extraction methods, we (i) fix the class of permutations under which a network is functionally unchanged, (ii) state a simple criterion that tells us *which* downstream extractors a permutation search can possibly touch, and (iii) use that criterion as a map of open directions, marking for each one what remains to be solved before it becomes a working method. Everything below is conceptual; none of the variants is implemented or evaluated here.

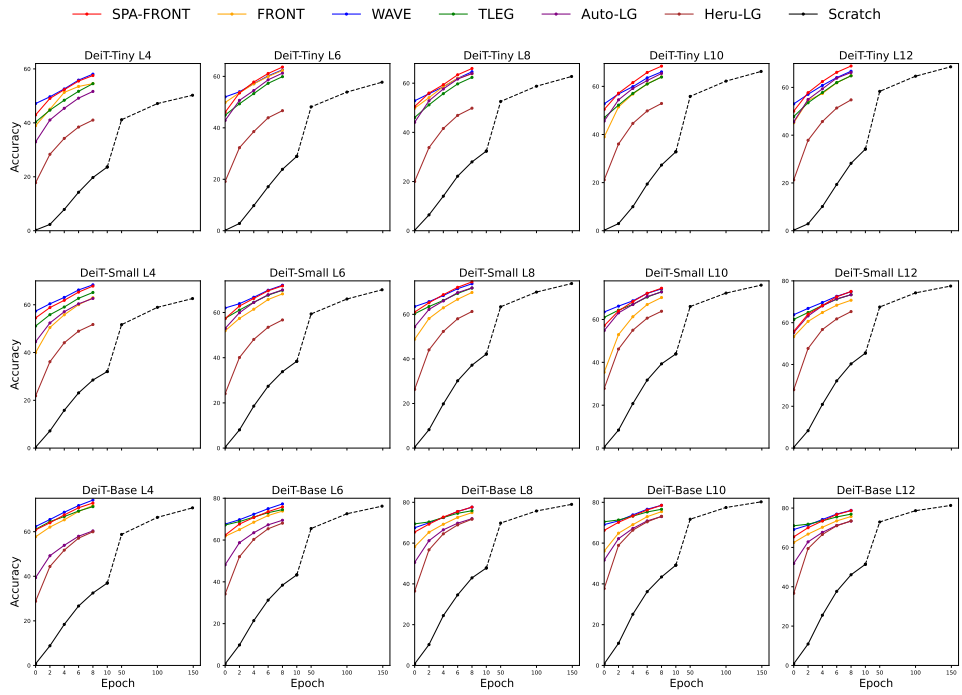


Figure 6: Training trajectories of different initialization methods on ImageNet-1K. Accuracy curves are reported for DeiT-Ti, DeiT-S, and DeiT-B under different target depths. SPA-FRONT consistently provides a stronger initialization and faster convergence compared with existing approaches.

F.1 Admissible permutations

Let $\Phi_{\mathcal{P}}$ denote a reparameterization of a pre-trained network that leaves its computed function unchanged up to a relabeling of channels. Admissible reparameterizations are generated by permutations of *shared axes* subject to the following rules.

(R1) Axis-wise permutations. Our primitive is a permutation of a single shared axis, not a free coupling of the two sides of a matrix. We call an axis *shared* if one weight matrix reads it and another writes it—the channel dims of adjacent layers, and most importantly the Transformer residual stream. A single permutation of a shared axis propagates coherently into every module adjacent to it, and that cross-module propagation is what implements the permutation. At the level of a weight matrix $W \in \mathbb{R}^{d_{\text{in}} \times d_{\text{out}}}$ whose two axes happen to be shared, the induced action reads $W \mapsto P_{\text{out}} W P_{\text{in}}^{\top}$, with the two induced permutations in general *different*.

(R2) Cross-layer consistency. Any dimension shared across layers—most importantly the Transformer residual stream—must be permuted by the *same* permutation in every layer that shares it. Only genuinely layer-local dimensions may be permuted per layer. This rule is the binding one: it is what fixes the effective permutation group, and hence what decides which of the objectives below can be moved at all and which are invariant by construction.

(R3) Structural constraints. Multi-head attention permutations must respect the $(\text{head}, \text{head-dim})$ factorization; normalization parameters, embeddings, classification heads and tied weights must be permuted consistently with the axis they share.

F.2 When can a permutation search matter?

Consider an extractor $\mathcal{E}$ that maps weights W to a learngene. We say $\mathcal{E}$ is *permutation-equivariant* with respect to the admissible group $\mathcal{P}$ if for every $\Phi_{\mathcal{P}}$, $\mathcal{P} \in \mathcal{P}$, there exists an admissible $\Phi'_{\mathcal{P}}$ with

$$\mathcal{E}\big(\Phi_{\mathcal{P}}(W)\big) \;=\; \Phi'_{\mathcal{P}}\big(\mathcal{E}(W)\big). \tag{44}$$

Equivariance is a property of the *pair* $(\mathcal{E}, \mathcal{P})$, not of $\mathcal{E}$ alone, and it is exactly the condition under which permuting before extraction is equivalent to permuting after it: the target model initialized from the resulting learngene is then functionally identical to the one obtained without SPA, and the permutation search can contribute nothing. Conversely, for an extractor that fails equation 44 the search is at least well posed. We stress that non-equivariance is *necessary but not sufficient*: SPA is useful only if the induced objective admits an admissible permutation that improves it, and only if the optimizer finds one. The directions below are therefore sorted by whether this criterion is met, and by whether a constructive recipe is already in hand.

For extractors built on a *fixed* orthonormal transform Φ and a fixed index set $\mathcal{S}$—the setting of FRONT, with Φ the DCT and $\mathcal{S}$ the low-frequency indices—the retained-energy fraction

$$\rho(W) = \frac{\big\|\Pi_{\mathcal{S}}\,\Phi\,\mathrm{vec}(W)\big\|_2^2}{\big\|\mathrm{vec}(W)\big\|_2^2}, \qquad \Pi_{\mathcal{S}} \text{ the projector onto } \mathcal{S}, \tag{45}$$

is invariant under $\Phi_{\mathcal{P}}$ exactly when the induced permutation commutes with the selection operator $\Phi^\top \Pi_{\mathcal{S}} \Phi$. For generic low-pass masks this commutation fails, so ρ is permutation-sensitive and equation 44 is violated: the SPA problem is well posed. This is the property that FRONT possesses, and it is the template we now transfer.

F.3 Open directions beyond FRONT

Fixed orthonormal transforms (e.g. wavelet). Replacing the DCT by an orthonormal wavelet transform preserves the structure of equation 45 and therefore inherits the same non-equivariance, which makes it the closest relative of FRONT among the directions we considered. The natural objective becomes an ℓ_1 penalty on detail coefficients across scales, encouraging a piecewise-smooth, hierarchically ordered weight tensor whose energy sits in the coarse subband. The objective is quadratic in the permutation matrix, so minimizing it is a quadratic assignment problem whose optimiser is necessarily a heuristic, and whether it admits any approximation guarantee is open; and a target-size adaptation rule must be given explicitly. The latter need not be a learned expansion operator—resampling or rescaling of subbands is a legitimate and much simpler route.

TLEG-style depth linearization. The depth component of TLEG-style initialization asks the layer-wise parameters to lie near a straight line in depth, and it is *per-coordinate*: the fitted trajectory is determined by a scalar least-squares problem at each matrix entry. We record a diagnostic first, because it shapes what SPA can and cannot contribute here. By (R2) a permutation of a shared axis relabels the same coordinates at every layer, so the permutation only reorders the summands of the depth objective; the objective is constant along each admissible orbit, and no shared-axis permutation can move pre-trained weights closer to a linear depth trajectory. This is not an argument against the combination—it identifies where the leverage is. Any gain must come from the restricted subgroup that (R2) does admit, namely permutations of genuinely layer-local dimensions, which do alter the per-coordinate fit. The constructive direction we put forward is therefore to let SPA act as a *preconditioner* for TLEG: search over layer-local permutations for a relabeling under which the per-block depth profiles are closer to a common line, and hand the resulting tensor to TLEG as an approximately linear starting point. Whether pre-trained weights admit permutations that materially improve this profile is a conjecture, and we note it as such.

WAVE-style templates. A shared block template must explain many blocks at once, so this objective couples coordinate positions and is therefore *not* invariant in the sense of equation 44: it passes the criterion. The natural design asks a $p \times q$ partition of a permuted weight matrix to admit a common dominant direction, which reduces to maximizing the leading singular value of the stacked, vectorized blocks under the permutation—again a hard combinatorial objective for which we have a heuristic, not a solver. The fitted template inherits the block shape of the source and needs

an explicit resizing rule for targets of different width; as with the wavelet case, a resampling rule suffices, but without one WAVE's size-agnostic motivation is not recovered. And for unstructured weights the blocks are close to mutually orthogonal, in which case no permutation can raise the leading singular value appreciably. Whether pre-trained weights admit permutations that improve this criterion materially is an empirical question we leave open. Read constructively, SPA offers WAVE a preconditioned layout—a relabeling under which the block structure is more coherent—as a starting point for template fitting.

Summary. The equivariance test of equation 44 does not partition extractors into usable and unusable ones; it partitions them by *where the search has to be aimed*. It is automatically satisfied by extractors that apply a fixed, non-data-adaptive selection to the weights—FRONT is the instance developed in the main paper, and a fixed wavelet transform is its closest relative—and by objectives that couple coordinates through a shared structure, which covers WAVE-style width alignment as well as the TLEG-style depth profile once the search is restricted to the layer-local subgroup. Across all of these, the proposals remain conceptual: the optimizers are heuristics for hard combinatorial problems, the size- and width-adaptation rules are sketched rather than defined, and none is evaluated here. We state that they are making a research program available from SPA , not closing it.

G Limitations and Future Directions

Although SPA-FRONT demonstrates the effectiveness of function-preserving weight reorganization for knowledge extraction, two limitations remain. First, the improvement of low-frequency concentration is inherently bounded by the available permutation degrees of freedom. A permutation operation can only reorder coordinates along a specific hidden dimension while preserving functional equivalence, and therefore cannot arbitrarily modify the full two-dimensional organization of individual weight matrices. Although our depth-wise and width-wise alignment jointly exploit existing symmetries across different dimensions, a more expressive function-preserving transformation beyond permutation may further improve spectral compactness.

Second, SPA optimizes structural properties of pre-trained weights, such as spatial smoothness and spectral compactness, rather than directly optimizing downstream task objectives. Consequently, the induced spectral prior is task-agnostic and its effectiveness may depend on the relationship between weight organization and the target task. Developing adaptive structural objectives that incorporate task-specific signals while maintaining the advantages of training-free reparameterization represents an interesting direction for future research.