LINEAR-TIME GLOBAL VISUAL MODELING *without* EXPLICIT ATTENTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Existing research largely attributes the global sequence modeling capability of Transformers to the explicit computation of attention weights, a process that inherently incurs quadratic computational complexity. In this work, we offer a different perspective: we demonstrate that attention can be mathematically reframed as a Multi-Layer Perceptron (MLP) equipped with dynamically predicted parameters. Through this lens, we explain attention's global modeling power not as explicit token-wise aggregation, but as an *implicit* process where dynamically generated parameters act as a compressed representation of the global context. Inspired by this insight, we investigate a fundamental question: *can we achieve Transformer-level sequence global modeling entirely through dynamic parameterization while maintaining linear complexity, effectively replacing explicit attention?* To explore this, we design various dynamic parameter prediction strategies and integrate them into standard network layers. Extensive empirical studies on vision models demonstrate that dynamic parameterization can indeed serve as a highly effective, linear-complexity alternative to explicit attention, offering a promising paradigm for efficient sequence modeling.

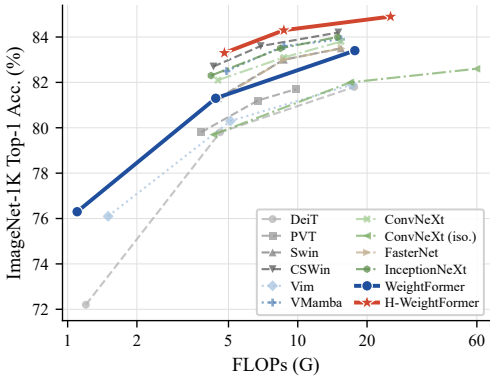

Figure 1: ImageNet accuracy of WeightFormer and baselines.

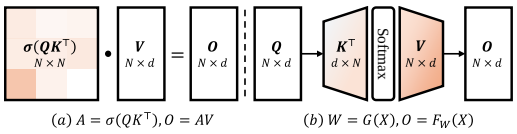

Figure 2: (a) Explicit weighted aggregation, where attention weights are computed and applied to values. (b) Attention as a dynamic MLP with parameters $W = G(X)$, where $K^\top$ and V are dynamic weights and Softmax acts as non-linearity, enabling implicit global modeling via $O = F_W(X)$.

1 INTRODUCTION

The success of Transformers across various domains is primarily rooted in their capacity for global sequence modeling, which is conventionally attributed to the attention mechanism. Under this prevailing perspective, attention is conceptualized as an explicit token-wise weighted aggregation process: attention weights are computed from pairwise token similarities (e.g., $A = \mathrm{Softmax}(QK^\top)$), and then used to explicitly recombine value representations (e.g., $O = AV$). Consequently, efforts to improve efficiency have predominantly focused on approximating or sparsifying this explicit attention matrix (Child, 2019; Beltagy et al., 2020; Zaheer et al., 2020; Wang et al., 2020; Katharopoulos et al., 2020; Choromanski et al., 2020; Han et al., 2025; Yuan et al., 2025; Xiong et al., 2021), fundamentally bounding the design space to explicit feature recombination.

In this work, we rethink this prevailing assumption by introducing a different perspective: attention can be mathematically reframed as a dynamic MLP. As illustrated in Figure 2, the key matrix $K^\top$

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.

and value matrix V can be viewed as the weights of the first and second linear layer in an MLP, while the Softmax operation serves as its activation function. Crucially, these parameters are not static; they are dynamically generated conditioned on the input.

This perspective shift provides a new explanation for how attention achieves global sequence modeling. Instead of relying on explicit token-to-token routing, global modeling emerges implicitly: the dynamically predicted parameters act as a compact representation that compresses the global context of the input. Forwarding the input through this dynamically parameterized network integrates long-range dependencies naturally, without ever needing to compute or apply explicit attention weights.

This insight leads to our central research question:

If global modeling is essentially a byproduct of dynamic parameterization, can we use dynamic parameters to completely replace attention, thereby achieving sequence global modeling while maintaining linear complexity?

To answer this question, we explore whether the implicit global modeling capabilities of attention can be decoupled from its quadratic matrix multiplications. We design and analyze several lightweight mechanisms for dynamically predicting network parameters (e.g., linear and depthwise convolution layers) based on global sequence context. By generating weights dynamically, we bypass the need for an $N \times N$ token interaction matrix, preserving linear computational complexity with respect to the sequence length.

It is important to note that our goal in this paper is not to engineer a new state-of-the-art visual architecture heavily optimized for specific benchmarks. Instead, we aim to systematically explore and validate the feasibility of using dynamic parameters as an alternative for attention. We instantiate our dynamic strategies in an isotropic backbone (WeightFormer) and a hierarchical backbone (H-WeightFormer) to empirically test this hypothesis.

Extensive experiments demonstrate that models relying solely on our dynamic parameterization strategies can achieve competitive global receptive fields and representational power comparable to Vision Transformers, but with significantly lower computational overhead.

In summary, our contributions are:

- We mathematically reframe attention as a dynamic MLP, providing a new explanation for its global sequence modeling capabilities based on implicit parameter generation rather than explicit weight aggregation.
- Inspired by this perspective, we explore the feasibility of replacing attention with dynamic parameterization to achieve global modeling while maintaining linear complexity.
- We introduce and evaluate various dynamic parameter strategies within vision models. Our results validate that dynamic parameters can effectively substitute attention for global sequence modeling, paving the way for fundamentally more efficient architectures.

2 Related Work

Attention and Global Modeling Paradigms. Attention (Vaswani et al., 2017; Devlin et al., 2019; Brown et al., 2020; Dosovitskiy, 2020; Touvron et al., 2021) has become the dominant mechanism for global modeling in Transformers. A substantial line of research has focused on improving the efficiency of attention. Representative approaches include sparse attention (Child, 2019; Beltagy et al., 2020; Zaheer et al., 2020; Yuan et al., 2025), low-rank approximations (Wang et al., 2020; Xiong et al., 2021), structured attention patterns (Han et al., 2025) and kernelized attention (Katharopoulos et al., 2020; Choromanski et al., 2020), which aim to reduce the quadratic complexity of the attention matrix. Despite their success, these meth-

Table 1: Summary of weight prediction strategies.

Layer	Strategy	ΔW
Linear	GAP	$\mathrm{MLP}(\mathrm{GAP}(X))$
	Linear	$W_1(X^\top X)W_2$
	Nonlinear	$\sigma\left(W_1(X^\top X)W_2\right)$
	Deep	$W_1\sigma\left(W_2(X^\top X)W_3\right)W_4$
	Bilateral	$W_1\sigma(W_2X^\top)\sigma(XW_3)\,W_4$
DWC	GAP	$\mathrm{MLP}(\mathrm{GAP}(X))$
	Adaptive	$\mathrm{MLP}(\mathrm{AAP}(X))$
	Amp-Dir	$s(X)\cdot\frac{\mathrm{MLP}(X')}{\lVert\mathrm{MLP}(X')\rVert_F+\epsilon}$
	Conv	$\mathrm{AAP}(f(X))$

ods fundamentally remain within the conventional attention paradigm: global modeling is achieved through explicitly computing attention weights and applying them to aggregate values.

Dynamic Networks and Connections to Attention. Dynamic networks, where parameters adapt based on the input, have long been explored as a means to increase model expressiveness (Ha et al., 2016; Han et al., 2022). In vision, this includes dynamic filter generation (Jia et al., 2016), conditional convolutions (Yang et al., 2019), weight modulation (Perez et al., 2018), and dynamic depthwise operations (Chen et al., 2020). Several recent works have drawn connections between attention and dynamic convolutions. Zhou et al. (2023) interprets Vision Transformers as ConvNets equipped with dynamic convolutions. Han et al. (2021) establishes theoretical and empirical links between local attention and dynamic depthwise convolution. Involution (Li et al., 2021) inverts standard convolution principles by making kernels spatially specific and channel-agnostic, yielding an operation reminiscent of attention. These approaches highlight the dynamic nature of attention weights, effectively treating them as input-dependent kernels that are explicitly generated and applied to aggregate features. However, they still operate within the classic attention paradigm of computing weights and using them for explicit feature recombination.

3 Attention as a Dynamic MLP

3.1 Attention as Explicit Weighting

Given an input feature matrix $X \in \mathbb{R}^{N\times d}$, the query, key, and value are computed as $Q = XW_Q, K = XW_K, V = XW_V$, where $W_Q, W_K, W_V \in \mathbb{R}^{d\times d}$ are learnable projection matrices. The output is given by

$$\mathrm{Attention}(Q,K,V) = \mathrm{Softmax}\left(\frac{QK^\top}{\sqrt{d}}\right)V. \tag{1}$$

The prevailing perspective in the field interprets Equation (1) as a weighted combination of value vectors. By defining an attention matrix $A = \mathrm{Softmax}\left(\frac{QK^\top}{\sqrt{d}}\right) \in \mathbb{R}^{N\times N}$, the operation becomes $O = AV$. Here, A_{ij} explicitly represents the pairwise similarity of token j to token i. This view frames global modeling as a two-step process: (1) explicitly computing the dense affinity matrix A, and (2) using A to recombine representations from V.

3.2 Reformulating Attention as a Dynamic MLP

We now introduce an alternative perspective that departs from the explicit weighting paradigm. Rather than viewing attention as computing and applying an attention matrix, we reinterpret it as a *dynamic parameterized MLP* whose parameters are predicted from the input (illustrated in Figure 2). Consider the output corresponding to a single query vector $q_i \in \mathbb{R}^d$ (the i-th row of Q). From Equation (1), we have $o_i = \mathrm{Softmax}\left(\frac{q_iK^\top}{\sqrt{d}}\right)V$. This expression reveals a structure directly analogous to a two-layer MLP applied to q_i. Specifically, $\frac{q_iK^\top}{\sqrt{d}}$ corresponds to a linear transformation with weight matrix $K^\top$, followed by a Softmax non-linearity, and finally a second linear transformation with weight matrix V. Crucially, $K^\top$ and V are not static parameters. They are dynamically generated from the input: $K = XW_K, V = XW_V$. Therefore, for each input sequence, the model instantiates a unique MLP whose parameters are conditioned on the entire input. In other words, attention implements a dynamic MLP: $W = G(X), O = F_W(X)$, where $G(\cdot)$ predicts the dynamic parameters $W = \{K^\top, V\}$ from X, and $F_W(\cdot)$ denotes the forward pass of the resulting MLP.

From this viewpoint, global context is *compressed into the dynamically predicted parameters*. This compression can be interpreted as reducing the representation from the full channel dimension to the individual head dimension, as we conceptualize each head as an independent MLP operating on the global sequence. By forwarding the input through these parameters, the model implicitly integrates global dependencies, without explicitly constructing or applying an attention weight matrix.

Table 2: Comparison of different dynamic weight prediction strategies for linear and depthwise convolution layers. The baseline employs static weights. "Dynamic Linear 1/2" indicates that dynamic weights are applied to the first and/or second linear layer within the MLP. Throughput unit: img/s.

Method	Prediction Strategy	#Param	FLOPs	Throughput	Acc ↑
Baseline	Static CNN	5.7M	1.1G	4172	73.3
Dynamic Linear 1	GAP 5	10.1M	1.1G	3759	74.7
	Linear w/o Pool 6	6.8M	1.5G	2951	75.8
	Linear w/ Pool 6	6.8M	1.2G	3278	76.2
	Nonlinear 7	6.8M	1.2G	3198	76.0
	Deep 8	6.9M	1.2G	3267	76.2
	Bilateral 9	6.9M	1.2G	3213	76.4
Dynamic Linear 2	Bilateral 9	6.9M	1.2G	3219	75.5
Dynamic Linear 1 + 2	Bilateral 9	8.2M	1.3G	2659	76.7
Dynamic DWC	GAP 11	7.9M	1.1G	4119	74.1
	Adaptive 12	6.2M	1.1G	4051	74.6
	Amp-Dir 13	6.4M	1.1G	4028	74.8
	Conv 14	6.7M	1.3G	3765	74.8
Dynamic Linear 1 + DWC	Bilateral + Adaptive 9 12	7.4M	1.2G	3201	76.8

3.3 Complexity and Design Implications

Explaining Quadratic Complexity. In this formulation, the effective width of the dynamic MLP scales with the number of tokens N, since $K^\top \in \mathbb{R}^{d\times N}$ and $V \in \mathbb{R}^{N\times d}$. As a result, the dynamic network grows with sequence length, naturally leading to the quadratic computational and memory complexity of attention.

Beyond Attention. More importantly, this view suggests that explicit attention weights are not essential for global modeling. Instead, global modeling can be achieved through dynamic weight prediction. This insight opens a new design space for efficient architectures.

Proving the Feasibility of Replacing Attention with Dynamic Parameters. Inspired by this perspective, we aim to validate the feasibility of using dynamic parameterization as a complete replacement for explicit attention in achieving Transformer-level global sequence modeling. To this end, we extend dynamic parameterization to convolutional networks by generating the weights of linear and depthwise convolution layers conditioned on the global input context. This enables standard CNN to implicitly capture global dependencies.

4 Dynamic Weight Prediction: A Design Space

Motivated by the dynamic-MLP interpretation of attention, we now instantiate *implicit global modeling* within convolutional networks. In Transformers, global context is compressed into K and V, and global interactions emerge implicitly by forward pass. Our goal is to replicate this mechanism in CNNs: we aim to compress global information into input-conditioned linear and depthwise convolution weights, achieving global modeling through forward passes without explicit token-to-token interactions. However, unlike attention which achieves this compression via head-wise dimensionality reduction, our approach introduces spatial compression paradigms to map the global context into a fixed-size parameter space.

This spatial compression is crucial: by projecting the context into a fixed size, the parameter generation process becomes independent of the input resolution, thereby preserving linear complexity. Two global statistics already present in attention suggest how to form this projection. Removing the softmax from Equation (1) recovers linear attention, whose mixing is exactly the correlation

$$\sum_j (q_i k_j^\top)\, v_j = x_i\, W_Q W_K^\top (X^\top X)\, W_V, \tag{2}$$

while the high-temperature limit of softmax attention degenerates to a mean over values, $\frac{1}{N}\sum_j v_j$. These identities are design starting points rather than equivalences to softmax attention. They mo-

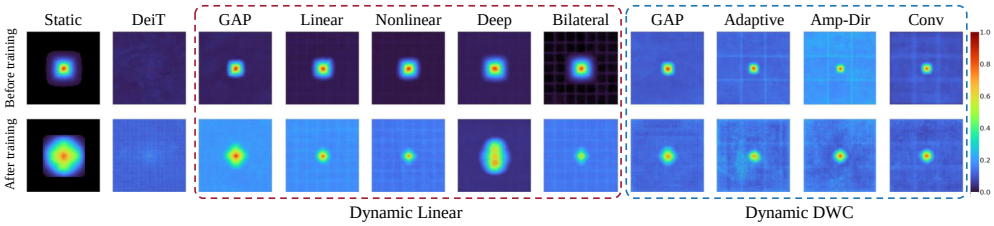

Figure 3: Comparison of ERF between DeiT, static and dynamic weight prediction strategies. Pixels with higher intensity indicate larger responses related to the central pixel.

tivate two spatial compression paradigms. Formally, given an input feature matrix $X \in \mathbb{R}^{N\times d}$, we construct a compact global representation $\phi(X)$ independent of N and predict dynamic parameters from it:

$$\text{Pooling:} \quad \phi_{\text{pool}}(X) = \text{Pool}(X) \in \mathbb{R}^{M\times d}, \quad \text{Correlation:} \quad \phi_{\text{corr}}(X) = X^\top X \in \mathbb{R}^{d\times d}. \tag{3}$$

Unlike channel-wise compression in attention ($C \to d$), our dynamic prediction strategies inherently focus on spatial compression. For instance, the pooling paradigm compresses the sequence into a fixed-size ($M \times d$) descriptor, while the correlation paradigm distills global second-order statistics. Based on these principles, we design dynamic parameter prediction strategies for linear and depthwise convolution layers, summarized in Table 1.

4.1 Dynamic Linear Layers

Linear layers operate on channel dimensions and do not perform token-wise mixing. However, when their parameters are conditioned on global input, they can implicitly integrate global context into channel transformations. For an input $X \in \mathbb{R}^{N\times d}$, we generate a dynamic update $\Delta W(X) \in \mathbb{R}^{d\times d}$ to modulate a learnable static weight W_0:

$$W(X) = W_0 + \Delta W(X). \tag{4}$$

Pooling-Based Strategies. The simplest approach leverages Global Average Pooling (GAP) to compress the sequence into a vector $z = \text{GAP}(X)$, which is then mapped to parameters via a lightweight MLP:

$$\Delta W(X) = \text{Reshape}(\text{MLP}(z)). \tag{5}$$

While efficient, compressing entire sequence into a single vector risks losing fine-grained features.

Correlation-Based Strategies. To capture higher-order feature interactions, we leverage the correlation matrix $X^\top X \in \mathbb{R}^{d\times d}$. Recognizing the limitations of simple linear mappings, we investigate a progression of predictors inspired by the success of deep non-linear architectures:

$$\text{Linear:} \quad \Delta W = W_1(X^\top X)W_2, \tag{6}$$

$$\text{Nonlinear:} \quad \Delta W = \sigma\left(W_1(X^\top X)W_2\right), \tag{7}$$

$$\text{Deep:} \quad \Delta W = W_1\sigma\left(W_2(X^\top X)W_3\right)W_4. \tag{8}$$

Specifically, the Nonlinear variant introduces an activation function σ (e.g., SiLU). The Deep strategy further factorizes the prediction into a low-rank transition to reduce FLOPs.

Beyond explicitly processing the second-order correlation matrix, we further explore an alternative architectural hypothesis termed *Bilateral Activation*. We attempt to factorize the weight prediction process into two complementary, non-linear branches that act independently on the input X and $X^\top$:

$$\Delta W(X) = W_1\sigma(W_2X^\top)\sigma(XW_3)W_4. \tag{9}$$

To further reduce computation, we apply Adaptive Average Pooling (AAP) to downsample X by a factor of 2 along the spatial dimensions before computing $X^\top X$. Unless otherwise specified, all experiments adopt this pooling strategy.

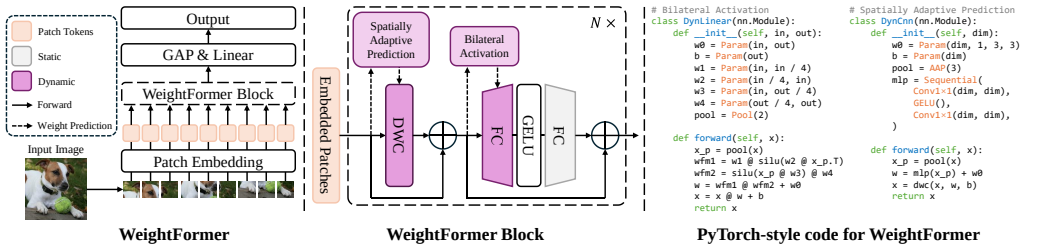

Figure 4: Illustration of the WeightFormer block shared by the isotropic WeightFormer and hierarchical H-WeightFormer. LayerNorm is omitted for simplicity.

Experimental Results. Table 2 reports the performance of different linear dynamic weight prediction strategies on ImageNet-1K. Introducing pooling before computing $X^\top X$ not only lowers FLOPs but also slightly improves accuracy, likely by suppressing high-frequency noise while retaining essential global context. Among all strategies, the Bilateral Activation applied to the first linear layer strikes the best accuracy-efficiency trade-off, achieving 76.4% top-1 accuracy with moderate parameter and FLOPs overhead. Extending it to both linear layers yields further gains (76.7%), but with higher costs and reduced throughput, suggesting diminishing returns in deeper layers.

4.2 Dynamic Depthwise Convolutional Layers

Depthwise convolutions provide spatial inductive bias but are inherently local. By dynamically predicting depthwise kernels from global context, we enable convolutions to achieve Transformer-level global receptive fields. We extend the principle of dynamic parameterization to depthwise convolution layers by predicting input-conditioned depthwise kernels. For an input feature map $X \in \mathbb{R}^{d\times H\times W}$, we generate a kernel update $\Delta W(X) \in \mathbb{R}^{d\times K\times K}$ (where $K = 3$) to modulate a learnable static kernel W_0:

$$W(X) = W_0 + \Delta W(X). \tag{10}$$

The resulting kernels are applied via depthwise convolution, which naturally maintains linear complexity relative to the spatial resolution. Similar to linear layers, we explore several paradigms to eliminate the dependency on input size.

Global Pooling-Based Strategies. A common baseline involves collapsing all spatial dimensions via GAP to obtain a channel descriptor $z = \mathrm{GAP}(X) \in \mathbb{R}^d$. An MLP then maps z to the kernel:

$$\Delta W(X) = \mathrm{Reshape}\left(\mathrm{MLP}(z)\right). \tag{11}$$

While widely adopted in early dynamic networks (Yang et al., 2019; Wu et al., 2019), this strategy ignores the spatial structure of the input, potentially limiting the adaptivity of the generated filters.

Spatially Adaptive Strategies. To preserve structural information, we investigate a *Spatially Adaptive* approach. Instead of a global scalar, we reduce the input to a fixed $K \times K$ resolution that is grid-aligned with the target kernel:

$$X' = \mathrm{AAP}(X, (K, K)) \in \mathbb{R}^{d\times K\times K}, \quad \Delta W(X) = \mathrm{MLP}(X'). \tag{12}$$

Inspired by the stability of weight normalization (Salimans & Kingma, 2016), we further explore a *Decoupled Amplitude-Direction (Amp-Dir)* strategy:

$$s(X) = \mathrm{Sigmoid}(\mathrm{GAP}(X)W), \quad \Delta W(X) = s(X) \cdot \frac{\mathrm{MLP}(X')}{\|\mathrm{MLP}(X')\|_F + \epsilon}. \tag{13}$$

We also explore predicting depthwise convolution kernels using a convolutional network. The input feature map is first processed by a network composed of two 3×3 convolutions with a channel bottleneck and GELU. The resulting feature map is then pooled to the target kernel size:

$$\Delta W(X) = \mathrm{AAP}(f(X), (K, K)), \tag{14}$$

where $f(\cdot)$ denotes the convolutional network.

Table 3: Comparison with Transformer (Touvron et al., 2021; Wang et al., 2021; Liu et al., 2021; Dong et al., 2022), Mamba (Zhu et al., 2024; Liu et al., 2024), and ConvNet (Liu et al., 2022; Chen et al., 2023; Yu et al., 2024) on ImageNet-1K.

Method	#Params	FLOPs	Acc@1
DeiT-T	6M	1.2G	72.2
DeiT-S	22M	4.6G	79.8
DeiT-B	87M	17.6G	81.8
PVT-S	25M	3.8G	79.8
PVT-M	44M	6.7G	81.2
PVT-L	61M	9.8G	81.7
Swin-T	29M	4.5G	81.3
Swin-S	50M	8.7G	83.0
Swin-B	88M	15.4G	83.5
CSWin-T	23M	4.3G	82.7
CSWin-S	35M	6.9G	83.6
CSWin-B	78M	15.0G	84.2
Vim-T	7M	1.5G	76.1
Vim-S	26M	5.1G	80.3
Vim-B	98M	17.2G	81.9
VMamba-T	31M	4.9G	82.5
VMamba-S	50M	8.7G	83.6
VMamba-B	89M	15.4G	83.9

Method	#Params	FLOPs	Acc@1
ConvNeXt-T	29M	4.5G	82.1
ConvNeXt-S	50M	8.7G	83.1
ConvNeXt-B	89M	15.4G	83.8
ConvNeXt-S (iso.)	22M	4.3G	79.7
ConvNeXt-B (iso.)	87M	16.9G	82.0
ConvNeXt-L (iso.)	306M	59.7G	82.6
FasterNet-S	31M	4.6G	81.3
FasterNet-B	54M	8.7G	83.0
FasterNet-L	93M	15.5G	83.5
InceptionNeXt-T	28M	4.2G	82.3
InceptionNeXt-S	49M	8.4G	83.5
InceptionNeXt-B	87M	14.9G	84.0
WeightFormer-T	7M	1.1G	76.3
WeightFormer-S	27M	4.4G	81.3
WeightFormer-B	27M	17.7G	83.4
H-WeightFormer-T	29M	4.8G	83.3
H-WeightFormer-S	58M	8.7G	84.3
H-WeightFormer-B	58M	25.3G	84.9

Experimental Results. Table 2 reports the performance of different depthwise convolution dynamic weight prediction strategies on ImageNet-1K. Although Amp-Dir and Conv achieve marginal gains (0.2%), they incur higher parameters, FLOPs, and lower throughput. In contrast, the Spatially Adaptive method preserves spatial structure during kernel prediction while maintaining computational efficiency, offering the best practical trade-off. We therefore adopt it as our depthwise convolution dynamic weight strategy and combine it with Bilateral Activation equation 9 for the first linear layer, yielding our default design termed "Dynamic Linear 1 + DWC" in Table 2, which achieves the highest accuracy (76.8%) while remaining efficient.

Validating Global Modeling. We verify that our dynamic networks achieve genuine global modeling by analyzing their Effective Receptive Fields (ERF) (Luo et al., 2016). Figure 3 shows that the static baseline remains localized, while all dynamic variants develop expansive receptive fields covering the entire input. The evolution from localized responses before training to dense global patterns after training further demonstrates that dynamic weights serve as an efficient alternative for global context aggregation.

5 Experiments

Based on the exploration in Section 4, we instantiate two complementary backbones that share the same dynamic operators and differ only in macro design: an isotropic model, **WeightFormer**, and a hierarchical model, **H-WeightFormer**. The shared building block is illustrated in Figure 4; architectural specifications of both families are deferred to Section A. **WeightFormer** is an isotropic architecture that inserts dynamic parameterization sparsely among static blocks. This sparse placement, validated in Section 5.5, balances modeling capacity and computational cost while keeping a controlled comparison in parameters and FLOPs. We also introduce **H-WeightFormer**, a four-stage pyramid that produces multi-scale features while reusing the same dynamic operators.

Table 4: Results of image generation.

Method	#Params	FLOPs	FID ↓
DiT-S/8	33M	0.4G	153.60
DiT-S/4	33M	1.4G	100.41
DiT-S/2	33M	6.1G	68.40
DiT-B/8	131M	1.4G	122.74
DiT-B/4	130M	5.6G	68.38
DiT-B/2	130M	23.0G	43.47
DiG-S/2	33M	4.3G	62.06
DiG-B/2	131M	17.1G	39.50
WeightFormer-S/8	38M	0.3G	149.49
WeightFormer-S/4	38M	1.3G	93.05
WeightFormer-S/2	38M	5.0G	61.39
WeightFormer-B/8	150M	1.4G	122.66
WeightFormer-B/4	149M	5.1G	65.89
WeightFormer-B/2	149M	20.0G	38.21

Table 5: Results of object detection and instance segmentation. FLOPs$^{t/b}$: total/backbone FLOPs.

Backbone	FLOPs$^{t/b}$	AP^b ↑	AP^b_{50} ↑	AP^b_{75} ↑	AP^m ↑	AP^m_{50} ↑	AP^m_{75} ↑
DeiT-T	594G / 106G	44.4	63.0	47.8	38.1	59.9	40.5
WeightFormer-T	566G / 77G	45.0	63.2	48.4	38.3	60.2	40.5

Table 6: Results of semantic segmentation. FLOPs$^{t/b}$: total/backbone FLOPs.

Method	Backbone	#Params	FLOPs$^{t/b}$	mIoU ↑
DeepLab v3+ (Chen et al., 2018)	ResNet-101 (He et al., 2016)	63M	–	44.1
UperNet (Xiao et al., 2018)	ResNet-50 (He et al., 2016)	67M	–	41.2
UperNet (Xiao et al., 2018)	ResNet-101 (He et al., 2016)	86M	–	44.9
UperNet (Xiao et al., 2018)	DeiT-T (Touvron et al., 2021)	11M	42G / 11G	39.2
UperNet (Xiao et al., 2018)	DeiT-S (Touvron et al., 2021)	43M	157G / 35G	44.0
UperNet (Xiao et al., 2018)	WeightFormer-T	12M	38G / 7G	40.7
UperNet (Xiao et al., 2018)	WeightFormer-S	47M	148G / 27G	45.6

5.1 Image Classification

The results are presented in Table 3. We observe that WeightFormer and H-WeightFormer remain competitive with strong Transformer, ConvNet and Mamba baselines. For example, the isotropic WeightFormer surpasses DeiT and isotropic ConvNeXt at matched scale, and H-WeightFormer-T outperforms Swin-T, ConvNeXt-T and VMamba-T under a comparable budget. H-WeightFormer-S further matches a larger CSWin variant while using fewer parameters. These results validate that dynamic parameterization can replace explicit attention for global visual modeling in both isotropic and hierarchical settings.

5.2 Object Detection and Instance Segmentation

The results are presented in Table 5. With high-resolution inputs, token sequences are substantially longer than in classification, so the quadratic cost of explicit attention becomes more pronounced. Under this regime, WeightFormer improves both box and mask quality over DeiT while reducing total and backbone compute. These findings indicate that input-conditioned weights provide a more efficient way to inject global context when the sequence is long.

5.3 Semantic Segmentation

As shown in Table 6, WeightFormer outperforms the corresponding DeiT backbones with lower computational cost, and remains competitive with heavier CNN-based UperNet and DeepLab v3+ baselines. These gains suggest that dynamic parameterization strengthens multi-scale semantic modeling without relying on explicit attention.

5.4 Image Generation

We further evaluate class-conditional generation on ImageNet-1K by replacing Softmax attention in a DiT (Peebles & Xie, 2023) backbone with our dynamic blocks. Following the standard protocol, we report FID at 256×256 resolution in Table 4. WeightFormer consistently improves DiT across patch sizes and model widths, and remains competitive with DiG (Zhu et al., 2025). This suggests that implicit global modeling via dynamic parameters benefits both discriminative and generative tasks.

5.5 Ablation and Analysis

Efficiency Analysis. Figure 5 compares throughput and GPU memory against DeiT. Owing to its linear time and memory complexity, WeightFormer scales more efficiently as image resolution increases, remaining suitable for high-resolution visual tasks.

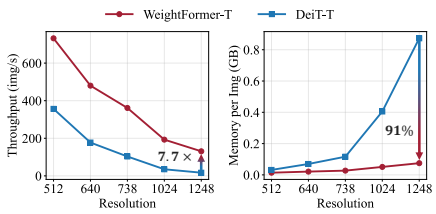

Figure 5: Comparisons between DeiT and WeightFormer in Throughput, and per-image GPU memory usage.

Table 7: Ablation on the frequency of dynamic blocks. N denotes the number of dynamic blocks. * refers to the best accuracy before divergence during training.

N	#Params	FLOPs	Throughput	Loss	Acc
17	10.1M	1.3G	2323	–	70.2*
14	9.3M	1.3G	2460	3.218	76.6
11	8.5M	1.2G	2745	3.191	76.9
8	7.6M	1.2G	3218	3.253	76.2
6	7.1M	1.1G	3515	3.261	76.3
4	6.5M	1.1G	3869	3.343	75.4

Ablation on Dynamic Block Frequency. We study the impact of dynamic block frequency by varying the total number of dynamic blocks. As shown in Table 7, increasing this number enhances the model's theoretical capacity but inevitably incurs higher computational costs. Replacing too many static layers with dynamic ones does not yield monotonic performance gains; rather, it leads to underfitting and optimization challenges. Inserting one dynamic block every third block strikes the best balance between performance and efficiency. Consequently, we adopt this sparse distribution strategy as the default configuration for WeightFormer.

6 Conclusion

We revisit attention as a dynamic MLP with input-conditioned parameters and show that dynamic parameterization can replace explicit attention: WeightFormer and its hierarchical variant H-WeightFormer achieve competitive performance on vision tasks with improved efficiency. This study remains limited in scope: evaluation is restricted to vision tasks, the expressivity and inductive biases of dynamic parameterization are not yet well understood, and optimizing input-conditioned parameters can complicate gradient flow and training stability. Future work includes extending the paradigm to broader domains, improving weight generation, and deepening its theoretical understanding. We hope this work motivates further exploration of dynamic parameterization as an alternative to attention-based architectures.

AI Use Statement

Large language models were used to assist with language polishing during the preparation of this manuscript. All scientific ideas, experimental designs, code implementations, and result analyses were entirely produced by the authors.

Ethics Statement

This work proposes a general-purpose visual backbone architecture aimed at improving computational efficiency. We do not foresee any direct negative societal impacts arising from this research.

Reproducibility Statement

All experiments in this paper use publicly available datasets: ImageNet-1K for image classification, COCO 2017 for object detection and instance segmentation, ADE20K for semantic segmentation, and ImageNet-1K for class-conditional image generation. We follow standard training protocols and report all hyperparameters in the appendix. The architectural specifications for both WeightFormer and H-WeightFormer are fully described in Section 5 and the appendix. Code and training logs will be made publicly available to facilitate reproducibility.

References

Iz Beltagy, Matthew E Peters, and Arman Cohan. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*, 2020.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: High quality object detection and instance segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 43(5):1483–1498, 2019.

Jierun Chen, Shiu-hong Kao, Hao He, Weipeng Zhuo, Song Wen, Chul-Ho Lee, and S-H Gary Chan. Run, don't walk: chasing higher flops for faster neural networks. In *2023 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 12021–12031. IEEE, 2023.

Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 801–818, 2018.

Yinpeng Chen, Xiyang Dai, Mengchen Liu, Dongdong Chen, Lu Yuan, and Zicheng Liu. Dynamic convolution: Attention over convolution kernels. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11030–11039, 2020.

Rewon Child. Generating long sequences with sparse transformers. *arXiv preprint arXiv:1904.10509*, 2019.

Krzysztof Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Davis, Afroz Mohiuddin, Lukasz Kaiser, et al. Rethinking attention with performers. *arXiv preprint arXiv:2009.14794*, 2020.

Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pp. 702–703, 2020.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 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.

Xiaoyi Dong, Jianmin Bao, Dongdong Chen, Weiming Zhang, Nenghai Yu, Lu Yuan, Dong Chen, and Baining Guo. Cswin transformer: A general vision transformer backbone with cross-shaped windows. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12114–12124. IEEE, 2022.

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

David Ha, Andrew Dai, and Quoc V Le. Hypernetworks. *arXiv preprint arXiv:1609.09106*, 2016.

Dongchen Han, Tianyu Li, Ziyi Wang, and Gao Huang. Vision transformers are circulant attention learners. *arXiv preprint arXiv:2512.21542*, 2025.

Qi Han, Zejia Fan, Qi Dai, Lei Sun, Ming-Ming Cheng, Jiaying Liu, and Jingdong Wang. On the connection between local attention and dynamic depth-wise convolution. *arXiv preprint arXiv:2106.04263*, 2021.

Yizeng Han, Gao Huang, Shiji Song, Le Yang, Honghui Wang, and Yulin Wang. Dynamic neural networks: A survey. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 44(11): 7436–7456, 2022.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Xu Jia, Bert De Brabandere, Tinne Tuytelaars, and Luc V Gool. Dynamic filter networks. *Advances in neural information processing systems*, 29, 2016.

Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are rnns: Fast autoregressive transformers with linear attention. In *International conference on machine learning*, pp. 5156–5165. PMLR, 2020.

Duo Li, Jie Hu, Changhu Wang, Xiangtai Li, Qi She, Lei Zhu, Tong Zhang, and Qifeng Chen. Involution: Inverting the inherence of convolution for visual recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 12321–12330, 2021.

Yanghao Li, Hanzi Mao, Ross Girshick, and Kaiming He. Exploring plain vision transformer backbones for object detection. In *European conference on computer vision*, pp. 280–296. Springer, 2022.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pp. 740–755. Springer, 2014.

Yue Liu, Yunjie Tian, Yuzhong Zhao, Hongtian Yu, Lingxi Xie, Yaowei Wang, Qixiang Ye, Jianbin Jiao, and Yunfan Liu. Vmamba: Visual state space model. *Advances in neural information processing systems*, 37:103031–103063, 2024.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 10012–10022, 2021.

Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11976–11986, 2022.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Wenjie Luo, Yujia Li, Raquel Urtasun, and Richard Zemel. Understanding the effective receptive field in deep convolutional neural networks. *Advances in neural information processing systems*, 29, 2016.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 4195–4205, 2023.

Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI conference on artificial intelligence*, 2018.

Tim Salimans and Durk P Kingma. Weight normalization: A simple reparameterization to accelerate training of deep neural networks. *Advances in neural information processing systems*, 29, 2016.

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

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

Sinong Wang, Belinda Z Li, Madian Khabsa, Han Fang, and Hao Ma. Linformer: Self-attention with linear complexity. *arXiv preprint arXiv:2006.04768*, 2020.

Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 568–578, 2021.

Felix Wu, Angela Fan, Alexei Baevski, Yann N Dauphin, and Michael Auli. Pay less attention with lightweight and dynamic convolutions. *arXiv preprint arXiv:1901.10430*, 2019.

Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 418–434, 2018.

Yunyang Xiong, Zhanpeng Zeng, Rudrasis Chakraborty, Mingxing Tan, Glenn Fung, Yin Li, and Vikas Singh. Nyströmformer: A nyström-based algorithm for approximating self-attention. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pp. 14138–14148, 2021.

Brandon Yang, Gabriel Bender, Quoc V Le, and Jiquan Ngiam. Condconv: Conditionally parameterized convolutions for efficient inference. *Advances in neural information processing systems*, 32, 2019.

Weihao Yu, Pan Zhou, Shuicheng Yan, and Xinchao Wang. Inceptionnext: When inception meets convnext. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5672–5683. IEEE, 2024.

Jingyang Yuan, Huazuo Gao, Damai Dai, Junyu Luo, Liang Zhao, Zhengyan Zhang, Zhenda Xie, Yuxing Wei, Lean Wang, Zhiping Xiao, et al. Native sparse attention: Hardware-aligned and natively trainable sparse attention. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 23078–23097, 2025.

Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 6023–6032, 2019.

Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. Big bird: Transformers for longer sequences. *Advances in neural information processing systems*, 33:17283–17297, 2020.

Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*, 2018.

Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 13001–13008, 2020.

Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ade20k dataset. *International journal of computer vision*, 127(3):302–321, 2019.

Chong Zhou, Chen Change Loy, and Bo Dai. Interpret vision transformers as convnets with dynamic convolutions. *CoRR*, 2023.

Lianghui Zhu, Bencheng Liao, Qian Zhang, Xinlong Wang, Wenyu Liu, and Xinggang Wang. Vision mamba: efficient visual representation learning with bidirectional state space model. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 62429–62442, 2024.

Lianghui Zhu, Zilong Huang, Bencheng Liao, Jun Hao Liew, Hanshu Yan, Jiashi Feng, and Xinggang Wang. Dig: Scalable and efficient diffusion models with gated linear attention. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 7664–7674, 2025.

A Architecture Details

We instantiate the WeightFormer block of Figure 4 into three model families that share the same dynamic operators and differ only in macro design: an isotropic backbone (WeightFormer), a hierarchical four-stage backbone (H-WeightFormer), and a diffusion model for image generation, which follows the macro design of DiT (Peebles & Xie, 2023). The detailed configurations are listed in Table 8, Table 9 and Table 10. Since our dynamic parameterization is decoupled from the sequence length and enjoys $\mathcal{O}(N)$ complexity, all models process the full-resolution token map with a global receptive field and require no window partitioning or token reduction.

Dynamic Operators. Every block contains a depthwise convolution as its token mixer and a two-layer channel MLP with expansion ratio 4. A *dynamic* block replaces exactly two layers with input-conditioned counterparts, namely the depthwise convolution and the first projection of the channel MLP; all remaining layers, as well as all biases, stay static. The dynamic depthwise convolution adopts the Spatially Adaptive strategy of equation 12: the feature map is reduced to 3×3 by adaptive average pooling and mapped to a per-sample kernel $\Delta W(X) \in \mathbb{R}^{d\times3\times3}$ by two 1×1 convolutions with a GELU in between. The generated kernels are applied per sample as a grouped convolution, so the operator remains depthwise. The dynamic linear layer adopts the Bilateral Activation of equation 9: the token map is first downsampled by a factor of 2 along each spatial axis, projected onto two low-rank branches of width d/r and d_{out}/r with $r = 4$, activated by SiLU, expanded back to d and d_{out} by two learnable matrices, and finally contracted into $\Delta W(X) \in \mathbb{R}^{d\times d_{\text{out}}}$. The low-rank factorization keeps the generator cheap, as the cost of predicting a $d\times d_{\text{out}}$ weight is reduced by r^2. In both cases, the update is soft-normalized before being added to the static weight,

$$W(X) = W_0 + \frac{\Delta W(X)}{\|\Delta W(X)\| + 1}, \tag{15}$$

where the norm is taken over the input-channel dimension for linear layers and over the 3×3 spatial support for depthwise kernels.

WeightFormer. The isotropic model patchifies the input with a 16×16 convolution followed by a LayerNorm, and stacks 17 blocks of constant width. Each block applies the depthwise token mixer as a residual branch and then the pre-norm channel MLP; dynamic blocks additionally normalize their input before the token mixer. Following the sparse distribution strategy validated in Section 5.5, one block out of every three is dynamic, which places dynamic blocks at positions $\{2, 5, 8, 11, 14, 17\}$ and yields six dynamic blocks in total. Applying dynamic parameterization to every block would incur substantial cost and hinder a fair comparison with the static baseline; the sparse schedule retains the global-modeling benefit at modest overhead. Each dynamic block uses Spatially Adaptive depthwise convolution equation 12 and Bilateral Activation equation 9 on the first linear layer; the second linear layer and all static blocks remain unchanged. No explicit positional encoding is used: positional information is supplied implicitly by the 3×3 depthwise convolutions inside every block, which act as a conditional positional encoding. Features are aggregated by a final LayerNorm and global average pooling before the linear classifier.

H-WeightFormer. The hierarchical model follows the common four-stage pyramid with strides $4, 8, 16, 32$, producing feature maps at $\frac{H}{4}$, $\frac{H}{8}$, $\frac{H}{16}$ and $\frac{H}{32}$. It reuses the same dynamic operators as WeightFormer: Spatially Adaptive depthwise convolution equation 12 and Bilateral Activation equation 9 on the first linear layer. The stem consists of four 3×3 convolutions (two of stride 2) followed by a 1×1 projection, and each downsampling layer is an inverted bottleneck that expands the channels by $4\times$ with a 1×1 convolution, applies a strided 3×3 depthwise convolution, and projects back with a 1×1 convolution. Each block starts with a 3×3 depthwise convolution added residually as a conditional positional encoding, followed by the pre-norm depthwise token mixer and the pre-norm channel MLP; the MLP additionally contains an inner 3×3 depthwise convolution applied residually between two GELU activations.

WeightFormer-DiT. For class-conditional generation we keep the DiT macro design and replace Softmax attention with our depthwise token mixer. Dynamic blocks reuse the same operators as WeightFormer, namely Spatially Adaptive depthwise convolution equation 12 and Bilateral Activation equation 9 on the first linear layer. The adaLN-Zero conditioning, the fixed 2D sine-cosine

Table 8: Architectures of the WeightFormer model series. Patch "$\downarrow n$" indicates the patch size is n. "$\mathrm{B}(C)$" represents one building block with embedding dimension C. Dynamic block positions are given as 1-indexed block depths.

	WeightFormer-T	WeightFormer-S	WeightFormer-B
Resolution	224×224	224×224	448×448
Backbone	Patch $\downarrow 16$ $\mathrm{B}(192)\times 17$	Patch $\downarrow 16$ $\mathrm{B}(384)\times 17$	Patch $\downarrow 16$ $\mathrm{B}(384)\times 17$
Dynamic blocks	$\{2, 5, 8, 11, 14, 17\}$, one every three blocks		
Classifier	LayerNorm, Global Average Pooling, Linear		

Table 9: Architectures of the H-WeightFormer model series. "$\downarrow n$" indicates the downsampling ratio is n. "$\mathrm{B}(C)$" represents one building block with embedding dimension C.

	Size	H-WeightFormer-T	H-WeightFormer-S	H-WeightFormer-B
Resolution		224×224	224×224	384×384
Stage1	$\frac{H}{4}\times\frac{W}{4}$	Stem $\downarrow 4$ $\mathrm{B}(64)\times 2$	Stem $\downarrow 4$ $\mathrm{B}(64)\times 4$	Stem $\downarrow 4$ $\mathrm{B}(64)\times 4$
Stage2	$\frac{H}{8}\times\frac{W}{8}$	Down $\downarrow 2$ $\mathrm{B}(128)\times 4$	Down $\downarrow 2$ $\mathrm{B}(128)\times 8$	Down $\downarrow 2$ $\mathrm{B}(128)\times 8$
Stage3	$\frac{H}{16}\times\frac{W}{16}$	Down $\downarrow 2$ $\mathrm{B}(320)\times 14$	Down $\downarrow 2$ $\mathrm{B}(320)\times 28$	Down $\downarrow 2$ $\mathrm{B}(320)\times 28$
Stage4	$\frac{H}{32}\times\frac{W}{32}$	Down $\downarrow 2$ $\mathrm{B}(512)\times 3$	Down $\downarrow 2$ $\mathrm{B}(512)\times 6$	Down $\downarrow 2$ $\mathrm{B}(512)\times 6$
Dynamic blocks		One every two blocks, starting from the second		
Classifier		BatchNorm, Global Average Pooling, Linear		

positional embedding, the timestep and label embedders and the final layer remain identical to DiT. Each block modulates the token mixer and the channel MLP with the six adaLN-Zero parameters predicted from the conditioning vector, and both branches are gated. The backbone has 15 blocks, and as in the isotropic model one block out of every three is dynamic, i.e. positions $\{2, 5, 8, 11, 14\}$. We report all three patch sizes $\{8, 4, 2\}$ at two widths, matching the DiT configurations used for comparison in Table 4.

B Experimental Details

For each task we specify the dataset, framework, training protocol, and evaluation metric. Throughput and per-image GPU memory in Figure 5 are measured on a single RTX 3090.

B.1 Image Classification

The ImageNet-1K (Deng et al., 2009) dataset contains 1.28M training images and 50K validation images across 1,000 classes. It is the standard benchmark for image classification.

We train WeightFormer and H-WeightFormer from scratch. The same recipe is used for the dynamic-weight prediction study in Section 4 and the ablation in Section 5.5. Training follows the Swin Transformer protocol (Liu et al., 2021); the concrete hyperparameters are listed in Table 11. We report top-1 accuracy on the ImageNet-1K validation set.

Table 10: Architectures of the WeightFormer-DiT model series. Patch "$\downarrow n$" indicates the patch size is n. "$\mathrm{B}(C)$" represents one building block with embedding dimension C and MLP expansion ratio 4. All variants place dynamic blocks at positions $\{2, 5, 8, 11, 14\}$.

	WeightFormer-S/8	WeightFormer-S/4	WeightFormer-S/2
Resolution	256×256		
Backbone	Patch $\downarrow 8$ $\mathrm{B}(384) \times 15$	Patch $\downarrow 4$ $\mathrm{B}(384) \times 15$	Patch $\downarrow 2$ $\mathrm{B}(384) \times 15$
	WeightFormer-B/8	WeightFormer-B/4	WeightFormer-B/2
Resolution	256×256		
Backbone	Patch $\downarrow 8$ $\mathrm{B}(768) \times 15$	Patch $\downarrow 4$ $\mathrm{B}(768) \times 15$	Patch $\downarrow 2$ $\mathrm{B}(768) \times 15$

Table 11: Training hyperparameters for image classification on ImageNet-1K.

Hyperparameter	Value
Input resolution	224×224 (448×448 for WeightFormer-B, 384×384 for H-WeightFormer-B)
Training epochs	300
Optimizer	AdamW (Loshchilov & Hutter, 2017)
Base learning rate	4×10^{-3}
LR schedule	Cosine decay
Warm-up	20 epochs, linear
Weight decay	0.05
Total batch size	2048
Augmentation	RandAugment (Cubuk et al., 2020), Mixup (Zhang et al., 2018), CutMix (Yun et al., 2019), random erasing (Zhong et al., 2020)

B.2 Object Detection and Instance Segmentation

The COCO 2017 dataset (Lin et al., 2014) contains 118K training images and 5K validation images with instance-level box and mask annotations. It is a standard benchmark for object detection and instance segmentation.

We adopt ViTDet (Li et al., 2022) with Cascade Mask R-CNN (Cai & Vasconcelos, 2019), and compare ImageNet-pretrained WeightFormer-T with DeiT-T (Touvron et al., 2021). Backbones are initialized from the corresponding ImageNet-1K checkpoints. Training follows the official ViTDet implementation, including multi-scale training and the default optimization schedule. We report box AP and mask AP on the COCO val2017 split. FLOPs in Table 5 are given as total/backbone under the same detector input size.

B.3 Semantic Segmentation

The ADE20K dataset (Zhou et al., 2019) contains 20K training images and 2K validation images with 150 semantic categories. It is a standard benchmark for semantic segmentation.

We use UperNet (Xiao et al., 2018) with ImageNet-pretrained WeightFormer-T/S and the corresponding DeiT (Touvron et al., 2021) backbones. Training follows the public UperNet optimization recipe, including the default learning-rate schedule and augmentation policy. We report single-scale mIoU on the ADE20K validation set. FLOPs in Table 6 are given as total/backbone under the same input size.

B.4 Image Generation

The ImageNet-1K (Deng et al., 2009) dataset is reused for class-conditional image generation at 256×256 resolution, covering the same 1,000 classes.

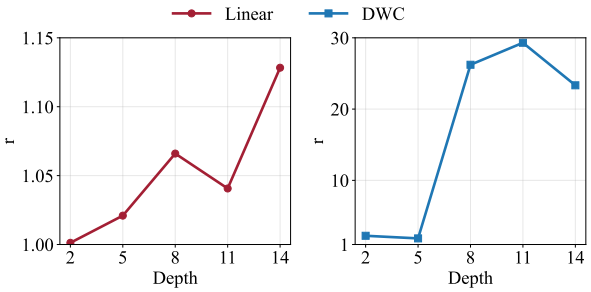

Figure 6: Relative dynamic-update strength $r = \|\Delta W\|_F / \|W_0\|_F$ across depth, measured on ImageNet-1K.

Training and evaluation follow the DiT protocol of the baselines in Table 4, without extra tuning. We report FID on 50K generated samples.

C Relative Strength of Dynamic Updates Across Depth

To analyze how dynamic parameterization contributes across depth, we measure the relative strength of dynamic updates with respect to static parameters on ImageNet-trained WeightFormer. For each dynamic layer, we compute $r = \frac{\|\Delta W\|_F}{\|W_0\|_F}$, where W_0 denotes the static weight and ΔW the predicted dynamic update. This ratio reflects the contribution of input-conditioned parameters. As shown in Figure 6, the relative strength of dynamic parameters exhibits a clear depth-dependent pattern. For dynamic linear layers, the ratio r remains close to 1 across all depths, indicating that channel-mixing transformations are consistently modulated by input-conditioned updates. In contrast, dynamic depthwise convolution exhibits a substantially larger ratio in deeper layers, suggesting increasingly strong spatially adaptive transformations at higher semantic levels. This behavior implies that dynamic depthwise convolutions play a progressively more prominent role in shaping feature representations, while dynamic linear layers provide stable global channel-wise modulation.