HARNESSING SMOOTHNESS: ASYMMETRIC DIFFERENTIAL LINEAR ATTENTION FOR VISION TRANSFORMERS

Anonymous authors
Paper under double-blind review

ABSTRACT

Linear attention reduces the quadratic cost of softmax attention, but its inherent smoothing effect often causes redundant information to obscure critical visual cues in the fused representation, leading to performance degradation. Rather than merely treating this smoothness as a limitation, we argue that it is inherently well-suited to modeling visual redundancy, as smooth attention weights naturally capture broadly distributed and redundant patterns. Based on this insight, we propose **Asymmetric Differential Linear Attention (ADLA)**, built upon an asymmetric dual-branch architecture. Specifically, the saliency branch extracts discriminative features via group-weighted key-value summarization, while the redundancy branch provides an explicit prior for visual redundancy via vanilla linear attention. Through their differential interaction, ADLA suppresses visual redundancy while reinforcing critical details, sharpening the attention focus. We further incorporate a dynamic kernel routing mechanism to refine feature focusing. Extensive experiments across diverse vision tasks and ablation studies demonstrate that ADLA consistently achieves strong performance while preserving linear complexity.

1 INTRODUCTION

Vision Transformers (ViTs) Dosovitskiy et al. (2020) have become a universal backbone for diverse vision tasks Han et al. (2023); Liu et al. (2021); Pu et al. (2025); Peebles & Xie (2023). The core softmax attention effectively captures long-range dependencies, yet its dense query-key similarity matrix incurs $\mathcal{O}(N^2)$ complexity in sequence length N, severely limiting scalability.

To address the quadratic complexity bottleneck, linear attention Katharopoulos et al. (2020) replaces the softmax operation with kernel-based feature mappings and reorders matrix multiplications, reducing complexity to linear time $\mathcal{O}(N)$. Despite its significant computational advantage, linear attention often lags far behind softmax attention in practical performance. Existing studies suggest that this performance degradation mainly stems from the lack of focusing ability in linear attention Han et al. (2023); Meng et al. (2025b); Qin et al. (2022); Han et al. (2024a). As illustrated in Fig. 1, linear attention produces oversmoothed query-key weight distributions, causing redundant tokens to overwhelm critical visual cues in the fused representation.

Recent efforts attempt to alleviate this issue from different perspectives: some primarily address deficiencies in kernel functions Han et al. (2023); Meng et al. (2025b); Kacham et al. (2023), while others investigate limitations induced by the computation paradigm of linear attention, such as low-rank global KV summarization Fan et al. (2025b); Zhang et al. (2026), magnitude neglect Fan et al. (2025a), and injectivity issues Han et al. (2024a). While striving to restore the model's ability to focus on critical information, these methods exclusively view smoothness as a mere defect to be eliminated.

Yet, this smoothness provides a natural way to capture how visual redundancy manifests. Unlike natural language, natural images exhibit strong spatial redundancy, with many tokens contributing little to visual decisions He et al. (2022); Rao et al. (2021). These redundant tokens are inherently diffuse and broadly distributed across the image. When processed by linear attention, its inherent smoothness indiscriminately aggregates information over broad contexts, which naturally makes it well-suited to capturing these diffuse, redundant patterns. In other words, the smoothness that prior work treats as a defect may be an effective mechanism for modeling visual redundancy.

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.

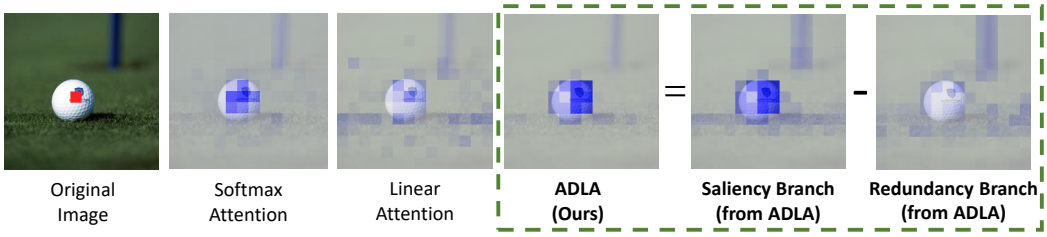


Figure 1: **Attention weight visualization from DeiT-T.** ADLA yields sharper, more focused attention for the query (red block). The green dashed box illustrates our mechanism: subtracting the redundancy branch from the saliency branch suppresses visual redundancy (more examples in Appendix F.1).

This insight suggests that the smoothness can be harnessed as a prior for visual redundancy and explicitly suppressed through a differential mechanism, rather than merely discarded as a flaw. Based on this, we propose **Asymmetric Differential Linear Attention (ADLA)** paradigm consisting of two different branches. The redundancy branch retains the vanilla linear attention form and exploits its smoothing tendency to model redundant responses in visual features. The saliency branch is specifically designed to extract discriminative features. Subtracting the redundancy branch from the saliency branch explicitly suppresses visual redundancy and highlights critical tokens.

To ensure highly discriminative representations, the saliency branch overcomes a core limitation behind the smoothing effect in linear attention: all key–value information is compressed into a single global summary shared by every query Fan et al. (2025b); Zhang et al. (2026). To break this homogeneity, the saliency branch performs Group-Weighted KV Summarization (GWS), where aggregation weights are derived from the semantic similarity between query groups and key groups. We further introduce a Dynamic Kernel Routing mechanism that assigns a dedicated kernel function to each token, thereby enhancing feature focusing.

This innovative asymmetric design enables ADLA to achieve a sharper and more discriminative attention distribution (Fig. 1). Extensive experiments across image classification, object detection, semantic segmentation, and image generation demonstrate that ADLA consistently yields clear performance improvements while maintaining linear complexity.

Our main contributions are summarized as follows: **(1)** We revisit the prevailing view that linear attention's inherent smoothness is merely a defect, proposing the **ADLA** paradigm that leverages this smoothness as an explicit redundancy prior to suppress visual redundancy and highlight critical features via an asymmetric differential mechanism. **(2)** We design Group-Weighted KV Summarization (GWS) to break global homogeneity and introduce Dynamic Kernel Routing for adaptive kernel mapping, enhancing the feature-focusing capability of the saliency branch. **(3)** Extensive experiments across multiple vision tasks demonstrate ADLA's strong performance with $\mathcal{O}(N)$ linear complexity, while comprehensive ablation studies and visual analyses validate the effectiveness of both core components and the dual-branch architecture.

2 Related Work

Vision Transformer. ViT Dosovitskiy et al. (2020) adapted Transformers to vision but incurs quadratic self-attention complexity. To improve computational and architectural efficiency, variants restrict attention within shifted or cross-shaped windows (Swin Liu et al. (2021), CSWin Dong et al. (2022)), downsample spatial keys/values (PVT Wang et al. (2021)), or utilize distillation tokens (DeiT Touvron et al. (2021)). Others introduce dynamic sparsity via deformable, neighborhood, or routing mechanisms (DAT Xia et al. (2022), NAT Hassani et al. (2023), BiFormer Zhu et al. (2023)), or integrate CNN-style local priors Dai et al. (2021); Guo et al. (2022); Si et al. (2022); Hou et al. (2024). Yet they cannot achieve both global interactions and linear complexity in one attention layer.

Linear Attention. Linear attention Katharopoulos et al. (2020) achieves $\mathcal{O}(N)$ complexity via kernelization but suffers from accuracy degradation. Subsequent variants address this in different ways: Efficient Attention Shen et al. (2021) applies softmax to $\mathbf{Q}$ and $\mathbf{K}$ separately; Nyströmformer Xiong et al. (2021) and SOFT Lu et al. (2021) approximate softmax via matrix factorization; FLatten

Transformer Han et al. (2023) sharpens focus with a power function and InLine Han et al. (2024a) improves injectivity; RALA Fan et al. (2025b) raises the rank of KV summaries and output features, while SAGA Cao & Wang (2025) selectively gates KV aggregation; PolaFormer Meng et al. (2025b) models signed $\mathbf{Q}$-$\mathbf{K}$ interactions to recover negative values; MALA Fan et al. (2025a) and NaLaFormer Meng et al. (2025a) recover magnitude and norm information. Agent Attention Han et al. (2024b) and VCA Pu et al. (2025) insert extra token sets as mediators. Notably, VCA introduces a symmetric, softmax-based difference, where the subtraction process may inevitably risk losing more details. MHLA Zhang et al. (2026) constructs distinct KV summaries via learned block-wise weights to mitigate context collapse, providing a foundation for our saliency branch; however, its static design lacks dynamic, query-dependent adaptability to varying content. A detailed qualitative and quantitative comparison with VCA and MHLA is provided in Appendix C. Unlike prior work treating smoothness as a flaw, we harness it to model redundancy for our asymmetric design.

3 PRELIMINARY

In this section, we first introduce softmax attention and linear attention, followed by an analysis of the limitations of linear attention.

3.1 SOFTMAX ATTENTION AND LINEAR ATTENTION

Given an input token sequence $\mathbf{X} \in \mathbb{R}^{N\times d}$, where N is the sequence length and d the feature dimension, we obtain queries, keys, and values via linear projections: $\mathbf{Q} = \mathbf{X}\mathbf{W}_Q, \mathbf{K} = \mathbf{X}\mathbf{W}_K, \mathbf{V} = \mathbf{X}\mathbf{W}_V$, with $\mathbf{W}_Q, \mathbf{W}_K, \mathbf{W}_V \in \mathbb{R}^{d\times d}$. The output of the i-th token is:

$$\mathbf{O}_i = \frac{\sum_{j=1}^{N} \mathrm{Sim}(\mathbf{Q}_i, \mathbf{K}_j)\mathbf{V}_j}{\sum_{m=1}^{N} \mathrm{Sim}(\mathbf{Q}_i, \mathbf{K}_m)}, \tag{1}$$

where $\mathrm{Sim}(\cdot,\cdot)$ denotes a similarity function. Classical softmax attention sets $\mathrm{Sim}(\mathbf{Q}_i, \mathbf{K}_j) = \exp(\mathbf{Q}_i\mathbf{K}_j^\top/\sqrt{d})$, requiring $\mathcal{O}(N^2d)$ computation. Linear attention instead approximates the similarity via a kernel function, $\mathrm{Sim}(\mathbf{Q}_i, \mathbf{K}_j) \approx \phi(\mathbf{Q}_i)\phi(\mathbf{K}_j)^\top$, where $\phi(\cdot)$ is an element-wise non-negative transform (e.g., $\phi(x) = \mathrm{ReLU}(x)$). Substituting this approximation into Eq. (1) yields:

$$\mathbf{O}_i = \frac{\phi(\mathbf{Q}_i)\sum_{j=1}^{N} \phi(\mathbf{K}_j)^\top\mathbf{V}_j}{\phi(\mathbf{Q}_i)\sum_{m=1}^{N} \phi(\mathbf{K}_m)^\top}. \tag{2}$$

Let $\mathbf{S} = \sum_{j=1}^{N} \phi(\mathbf{K}_j)^\top\mathbf{V}_j \in \mathbb{R}^{d\times d}$ as the global key-value summary Zhang et al. (2026), and $\mathbf{z} = \sum_{m=1}^{N} \phi(\mathbf{K}_m)^\top \in \mathbb{R}^d$ as the normalization factor. Since $\mathbf{S}$ and $\mathbf{z}$ can be precomputed once over all keys and values, the complexity reduces to $\mathcal{O}(Nd^2)$, which scales linearly with sequence length.

3.2 LIMITATIONS OF LINEAR ATTENTION

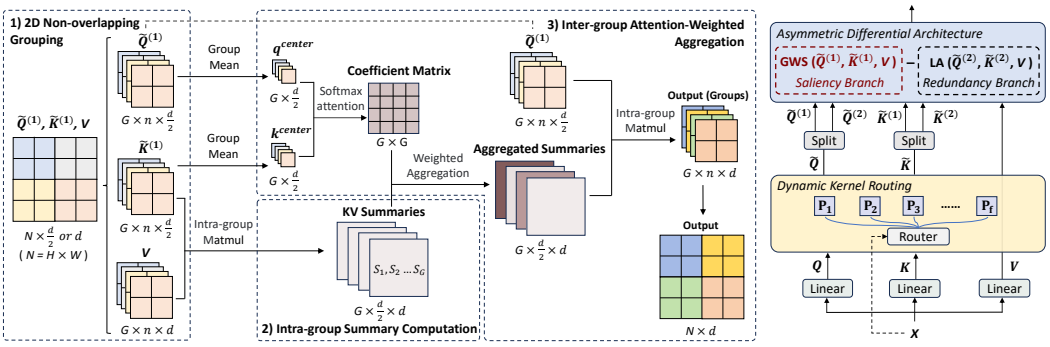
Linear attention reduces complexity to linear time but still lags behind softmax attention on vision tasks. By replacing softmax with kernel mappings $\phi(\cdot)$, it reorders matrix multiplications into a static global key–value summary $\mathbf{S} = \sum_{j=1}^{N} \phi(\mathbf{K}_j)^\top\mathbf{V}_j$. Each query then interacts only with this static summary, yielding two properties (quantitative evidence in Fig. 3):

1) *Low-rank bottleneck.* The attention matrix $\mathbf{A} \in \mathbb{R}^{N\times N}$ satisfies:

$$\mathrm{rank}(\mathbf{A}) = \mathrm{rank}\big(\phi(\mathbf{Q})\,\phi(\mathbf{K})^\top\big) \le \min\big(\mathrm{rank}(\phi(\mathbf{Q})),\, \mathrm{rank}(\phi(\mathbf{K}))\big) \le d, \tag{3}$$

where d is usually smaller than N in common vision transformer designs. This restricts the rank and hence the diversity of output features. Han et al. (2023).

2) *Smooth, high-entropy attention.* Without the exponential sharpening of softmax, linear attention relies on the same static summary for all queries, losing the ability to dynamically reweight key tokens according to query-specific relevance. This lack of query adaptivity forces attention weights into diffuse, high-entropy distributions Zhang et al. (2024; 2026).



(a) (b)

Figure 2: (a) Illustration of **Group-Weighted KV Summarization (GWS)**; (b) **The overall framework of ADLA.** LA represents vanilla Linear Attention.

Natural images are highly spatially redundant, with many tokens carrying limited task-relevant information He et al. (2022); Rao et al. (2021). Since these redundant tokens are numerous and spatially distributed, the smoothness of linear attention is naturally suited to aggregating their broadly distributed responses. As shown in Fig. 1, when querying a particular region, linear attention tends to activate many unrelated redundant tokens nearly as strongly as the target, introducing noise that hurts downstream tasks. Appendix B quantifies this vulnerability, showing that visual redundancy progressively amplifies the performance disadvantage of linear attention. These observations motivate our asymmetric differential design: an enhanced saliency branch sharpens discriminative responses, while a vanilla linear-attention branch provides a smooth prior for diffuse redundancy. Subtracting the latter suppresses weakly informative responses and improves the focus on critical visual evidence.

4 METHOD

The overall architecture of **Asymmetric Differential Linear Attention (ADLA)** is illustrated in Fig. 2b. Given a token sequence, we project it into queries, keys, and values. **Dynamic Kernel Routing** then adaptively assigns a dedicated kernel to each query and key to sharpen similarity distributions. Subsequently, the mapped features pass through an **Asymmetric Differential Architecture**, where a saliency branch with **Group-Weighted KV Summarization (GWS)** and a redundancy branch with vanilla linear attention interact differentially. We detail each component below.

4.1 DYNAMIC KERNEL ROUTING

To adaptively modulate the focusing degree for each token based on its semantic property, we introduce a dynamic kernel routing mechanism. Prior works Meng et al. (2025b); Han et al. (2023) have shown that power functions with exponents larger than 1 can sharpen similarity distributions. We therefore define a kernel function with exponent p for a vector $\mathbf{u} \in \mathbb{R}^{1\times d}$:

$$\phi_p(\mathbf{u}) = \frac{\mathrm{ReLU}(\mathbf{u})^p}{\|\mathrm{ReLU}(\mathbf{u})^p\|_2} \cdot \|\mathrm{ReLU}(\mathbf{u})\|_2, \quad \text{where } p = 1 + \gamma \cdot \sigma(w), \tag{4}$$

where exponentiation is element-wise, w is a learnable scalar, $\sigma(\cdot)$ denotes the sigmoid function, and $\gamma > 0$ is a scaling hyperparameter.

To allow each token to dynamically select its exponent, we introduce f learnable factors $\{w_1, \ldots, w_f\}$ that yield candidate exponents $\{p_1, \ldots, p_f\}$ via Eq. (4). For the i-th token with feature $\mathbf{X}_i \in \mathbb{R}^{1\times d}$, its exponent index t_i is dynamically predicted via a lightweight linear projection $\mathbf{W}_F \in \mathbb{R}^{d\times f}$:

$$\tilde{\mathbf{Q}}_i = \phi_{p_{t_i}}(\mathbf{Q}_i), \quad \tilde{\mathbf{K}}_i = \phi_{p_{t_i}}(\mathbf{K}_i), \quad \text{where } t_i = \underset{t\in\{1,\ldots,f\}}{\mathrm{argmax}} (\mathbf{X}_i \mathbf{W}_F)_t. \tag{5}$$

During training, this discrete routing is optimized end-to-end via Gumbel-Softmax Jang et al. (2016). The resulting kernel-mapped queries and keys $\tilde{\mathbf{Q}}, \tilde{\mathbf{K}} \in \mathbb{R}^{N\times d}$, along with $\mathbf{V} \in \mathbb{R}^{N\times d}$, serve as inputs to the subsequent asymmetric differential architecture.

4.2 Asymmetric Differential Architecture

To exploit the inherent smoothing of linear attention as a visual redundancy prior while enhancing focal discriminability, we propose an asymmetric differential architecture with a dual-branch design. We split $\tilde{\mathbf{Q}}$ and $\tilde{\mathbf{K}}$ along the channel dimension into two halves, one dedicated to saliency enhancement and the other to redundancy modeling: $\tilde{\mathbf{Q}}^{(1)}, \tilde{\mathbf{Q}}^{(2)}, \tilde{\mathbf{K}}^{(1)}, \tilde{\mathbf{K}}^{(2)} \in \mathbb{R}^{N\times d/2}$.

Saliency Branch. The saliency branch employs Group-Weighted KV Summarization (GWS) on $\tilde{\mathbf{Q}}^{(1)}$, $\tilde{\mathbf{K}}^{(1)}$, and $\mathbf{V}$ to produce a query-adaptive, discriminative representation $\mathbf{O}^{\mathrm{s}} \in \mathbb{R}^{N\times d}$. The detailed mechanism is deferred to Sec. 4.3.

Redundancy Branch. The redundancy branch applies vanilla linear attention to $\tilde{\mathbf{Q}}^{(2)}$, $\tilde{\mathbf{K}}^{(2)}$ and $\mathbf{V}$:

$$\mathbf{O}_i^{\mathrm{r}} = \frac{\tilde{\mathbf{Q}}_i^{(2)} \sum_{j=1}^{N} (\tilde{\mathbf{K}}_j^{(2)})^\top \mathbf{V}_j}{\tilde{\mathbf{Q}}_i^{(2)} \sum_{m=1}^{N} (\tilde{\mathbf{K}}_m^{(2)})^\top}, \tag{6}$$

where i, j, m all index tokens. The smoothness of linear attention makes $\mathbf{O}^{\mathrm{r}} \in \mathbb{R}^{N\times d}$ suitable for modeling redundant visual information, serving as an explicit redundancy prior.

The final attention output is obtained by subtracting the redundancy branch from the saliency branch:

$$\mathbf{O} = \mathbf{O}^{\mathrm{s}} - \lambda \mathbf{O}^{\mathrm{r}}. \tag{7}$$

Following the initialization strategy in Ye et al. (2024), λ is a learnable scalar that is initialized to λ_{init}. More details can be found in Appendix A.2.

4.3 Group-Weighted KV Summarization

To break the query-homogeneity of static summaries and produce more discriminative features within the saliency branch, we propose **Group-Weighted KV Summarization (GWS)**. GWS dynamically computes inter-group attention weights based on the semantic similarity between query groups and key groups, producing query-adaptive aggregation that enables each query group to selectively focus on informative regions. Meanwhile, tokens within each group are further distinguished by kernel inner products. As illustrated in Fig. 2a, GWS proceeds in three steps with inputs $\tilde{\mathbf{Q}}^{(1)}$, $\tilde{\mathbf{K}}^{(1)}$, and $\mathbf{V}$:

2D Non-overlapping Grouping. We reshape $\tilde{\mathbf{Q}}^{(1)}, \tilde{\mathbf{K}}^{(1)}$, and $\mathbf{V}$ into 2D spatial grids of size $H \times W = N$. Each grid is partitioned into $G = \left\lfloor \alpha\sqrt{N} \right\rfloor$ non-overlapping rectangular groups along the spatial dimensions ($\alpha > 0$ is a grouping multiplier, zero-padded along boundaries if unaligned). Each group $g \in \{1, \ldots, G\}$ contains $n = N/G$ tokens. In the rest of this subsection, we drop the superscript (1) and write $\tilde{\mathbf{Q}}, \tilde{\mathbf{K}}$ in place of $\tilde{\mathbf{Q}}^{(1)}, \tilde{\mathbf{K}}^{(1)}$ for simplicity.

Intra-group Summary Computation. For each group g, we compute a local KV summary matrix $\mathbf{S}_g \in \mathbb{R}^{d/2\times d}$ and a normalization vector $\mathbf{z}_g \in \mathbb{R}^{d/2}$:

$$\mathbf{S}_g = \sum_{j\in \text{group } g} \tilde{\mathbf{K}}_j^\top \mathbf{V}_j, \quad \mathbf{z}_g = \sum_{j\in \text{group } g} \tilde{\mathbf{K}}_j^\top, \tag{8}$$

Inter-group Attention-Weighted Aggregation. We construct group-level representations by averaging token features within each group and adding learnable spatial positional embeddings. For group g, the center query and center key are defined as:

$$\mathbf{q}_g^{\mathrm{center}} = \frac{1}{n} \sum_{j\in \text{group } g} \tilde{\mathbf{Q}}_j + \mathbf{p}_g^Q, \quad \mathbf{k}_g^{\mathrm{center}} = \frac{1}{n} \sum_{j\in \text{group } g} \tilde{\mathbf{K}}_j + \mathbf{p}_g^K, \tag{9}$$

where $\mathbf{p}_g^Q, \mathbf{p}_g^K \in \mathbb{R}^{1\times d/2}$ are learnable positional embeddings. For a query group $i \in \{1, \ldots, G\}$, the inter-group attention weights are computed as:

$$a_{i,g} = \frac{\exp\left(\mathbf{q}_i^{\mathrm{center}} (\mathbf{k}_g^{\mathrm{center}})^\top / \sqrt{d/2}\right)}{\sum_{g'=1}^{G} \exp\left(\mathbf{q}_i^{\mathrm{center}} (\mathbf{k}_{g'}^{\mathrm{center}})^\top / \sqrt{d/2}\right)}, \tag{10}$$

$a_{i,g}$ represents the attention weight between query group i and key group g, forming the coefficient matrix $\mathbf{A}_c \in \mathbb{R}^{G\times G}$. Using these weights, we aggregate the local KV summaries into a global representation for query group i:

$$\tilde{\mathbf{S}}_i = \sum_{g=1}^{G} a_{i,g}\mathbf{S}_g, \quad \tilde{\mathbf{z}}_i = \sum_{g=1}^{G} a_{i,g}\mathbf{z}_g. \tag{11}$$

Given a query token $\tilde{\mathbf{q}} \in \mathbb{R}^{1\times d/2}$ from group i, its output is:

$$\mathbf{o}(\tilde{\mathbf{q}}) = \frac{\tilde{\mathbf{q}}\tilde{\mathbf{S}}_i}{\tilde{\mathbf{q}}\tilde{\mathbf{z}}_i} = \frac{\sum_{g=1}^{G} a_{i,g}\tilde{\mathbf{q}}\mathbf{S}_g}{\sum_{g=1}^{G} a_{i,g}\tilde{\mathbf{q}}\mathbf{z}_g}. \tag{12}$$

Substituting Eq. (8) into Eq. (12) yields the token-level form, where $g(j)$ denotes token j's group:

$$\mathbf{o}(\tilde{\mathbf{q}}) = \frac{\sum_{j=1}^{N} a_{i,g(j)}(\tilde{\mathbf{q}}\tilde{\mathbf{K}}_j^\top)\mathbf{V}_j}{\sum_{j=1}^{N} a_{i,g(j)}(\tilde{\mathbf{q}}\tilde{\mathbf{K}}_j^\top)}. \tag{13}$$

Eq. (13) shows that the coefficient $a_{i,g(j)}$ scales token contributions based on the semantic relevance between query group i and key group $g(j)$, while the kernel inner product $\tilde{\mathbf{q}}\tilde{\mathbf{K}}_j^\top$ further differentiates token importance within each group.

4.4 Analysis of ADLA

Complexity Analysis. We analyze the additional complexity introduced by ADLA over vanilla linear attention. In ADLA, both branches use query-key pairs in $\mathbb{R}^{N\times d/2}$ and share $\mathbf{V} \in \mathbb{R}^{N\times d}$. Therefore, the extra cost only comes from Dynamic Kernel Routing, the group-level weighting in GWS, and a depthwise convolution (DWC) widely adopted in recent works Han et al. (2023; 2024b); Meng et al. (2025b); Pu et al. (2025). Given f candidate exponents, $G = \lfloor \alpha\sqrt{N} \rfloor$ groups, and DWC kernel size k, the additional complexity is:

$$\Delta\Omega_{\text{ADLA}} = \underbrace{Nfd}_{\text{Routing}} + \underbrace{G^2d/2 + G^2d^2/2}_{\text{GWS}} + \underbrace{k^2Nd}_{\text{DWC}}. \tag{14}$$

In practice, f is fixed to 8, and α is chosen such that $G^2 \approx N$ in the high-resolution stages (see Appendix G.1 for hyperparameter selection). Under these settings, the additional overhead is small and $\Delta\Omega_{\text{ADLA}}$ scales as $O(Nd + Nd^2)$, preserving the linear complexity of vanilla linear attention.

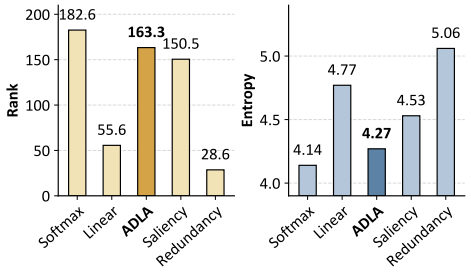


Figure 3: Average rank and entropy of attention scores from DeiT-T on 1,000 ImageNet-1K samples.

Asymmetric Branch Analysis. ADLA addresses the low-rank degeneracy and oversmoothing (high entropy) of linear attention through its asymmetric dual-branch design. Visually (Fig. 1), the saliency branch concentrates on critical structures, while the redundancy branch produces smooth patterns capturing broadly distributed redundant information. Their differential subtraction suppresses irrelevant tokens. Quantitatively (Fig. 3), the redundancy branch yields the lowest rank (28.6) and highest entropy (5.06). The saliency branch maintains a significantly higher rank (150.5) and lower entropy (4.53). Crucially, their differential fusion further elevates rank to 163.3 and drops entropy to 4.27, substantially outperforming vanilla linear attention (55.6, 4.77) and approaching softmax attention (182.6, 4.14). Conceptually, subtracting the smooth baseline suppresses diffuse noise to sharpen focus (reducing entropy), while asymmetric subspace differencing enriches feature diversity (boosting rank), confirming that linear attention's smoothness can serve as an effective prior to restore representational capacity.

5 Experiments

We benchmark ADLA across classification, detection, segmentation, and generation, followed by ablation studies. All experiments are conducted on NVIDIA A800 GPUs (details in Appendix G.2).

Table 1: ImageNet-1K classification results. Left: Results of our method and baselines. Right: Comparison with SOTA linear attention methods.

Method	Params	FLOPs	Acc(%)
DeiT-T	5.7M	1.2G	72.2
DeiT-T-ADLA	5.8M	1.2G	**75.9**(+3.7)
DeiT-S	22.1M	4.6G	79.8
DeiT-S-ADLA	22.2M	4.5G	**80.8**(+1.0)
PVT-T	13.2M	1.9G	75.1
PVT-T-ADLA	11.4M	2.0G	**78.8**(+3.7)
PVT-S	24.5M	3.8G	79.8
PVT-S-ADLA	20.3M	4.1G	**82.2**(+2.4)
PVT-M	35.9M	7.0G	81.2
PVT-M-ADLA	35.5M	7.2G	**83.6**(+2.4)
Swin-T	28.9M	4.5G	81.3
Swin-T-ADLA	28.5M	4.5G	**82.5**(+1.2)
Swin-S	49.7M	8.7G	83.0
Swin-S-ADLA	49.7M	8.8G	**83.5**(+0.5)
Swin-B	88.1M	15.4G	83.5
Swin-B-ADLA	87.9M	15.5G	**84.0**(+0.5)
CSwin-T	20.5M	4.4G	82.7
CSwin-T-ADLA	20.4M	4.4G	**83.2**(+0.5)
CSwin-S	32.8M	6.8G	83.6
CSwin-S-ADLA	32.7M	6.8G	**84.1**(+0.5)

Linear Attention	Params	FLOPs	Acc(%)
Comparison on DeiT-T Setting			
InLine Attn Han et al. (2024a)	6.5M	1.1G	74.5
MALA Fan et al. (2025a)	6.3M	1.1G	75.1
PolaFormer Meng et al. (2025b)	6.1M	1.2G	74.6
VCA Pu et al. (2025)	6.0M	1.2G	75.6
CA Han et al. (2026)	6.1M	1.2G	75.0
ADLA(Ours)	**5.8M**	1.2G	**75.9**
Comparison on DeiT-S Setting			
InLine Attn Han et al. (2024a)	17M	5.0G	80.2
RALA Fan et al. (2025b)	24M	4.6G	80.4
MALA Fan et al. (2025a)	24M	4.6G	80.3
VCA Pu et al. (2025)	23M	4.6G	80.7
ADLA(Ours)	22M	**4.5G**	**80.8**
Comparison on PVT-T Setting			
FLatten Attn Han et al. (2023)	12.2M	2.0G	77.8
InLine Attn Han et al. (2024a)	12.0M	2.0G	78.2
VCA Pu et al. (2025)	11.6M	2.0G	78.2
CA Han et al. (2026)	12.2M	2.0G	78.1
ADLA(Ours)	**11.4M**	2.0G	**78.8**

5.1 Image Classification

We evaluate ADLA on ImageNet-1K Deng et al. (2009) by replacing self-attention in DeiT Touvron et al. (2021), PVT Wang et al. (2021), Swin Liu et al. (2021), and CSwin Dong et al. (2022) with our module. All models are trained from scratch at 224×224 under standard training protocols.

Results. As shown in Tab. 1, ADLA consistently brings top-1 accuracy gains across various architectures and scales. Notably, PVT-S-ADLA surpasses PVT-L (81.7% top-1 accuracy) by $+0.5\%$ accuracy with only 33% of its parameters (61.4M) and 42% of its FLOPs (9.8G). Furthermore, fair comparisons with state-of-the-art linear attention mechanisms (right part of Tab. 1) demonstrate the superior performance of ADLA. More comparisons are provided in Appendix C.

Efficiency. We further evaluate practical deployment efficiency on an NVIDIA A800 GPU. As shown in the Accuracy-Runtime curves (Fig. 4(a)), our models achieve $1.43\times$ and $1.34\times$ faster inference speeds on PVT and Swin backbones, respectively.

5.2 Object Detection

To verify the downstream transferability of ADLA, experiments are conducted on COCO Lin et al. (2014) using Mask R-CNN He et al. (2017) and Cascade Mask R-CNN Cai & Vasconcelos (2018) under $1\times$ and $3\times$ training schedules. Backbones are initialized with ImageNet-pretrained weights.

Results. As summarized in Tab. 2, ADLA backbones consistently surpass their original counterparts across all metrics. For instance, PVT-ADLA improves box AP over the original PVT by $+3.5\%$ to $+5.0\%$. All ADLA variants require fewer FLOPs than their corresponding baselines.

Efficiency. Measurements on NVIDIA A800 GPUs (Fig. 4(b) and (c)) show that ADLA effectively reduces inference latency and increases throughput while significantly improving accuracy, highlighting its efficiency. Additional results and implementation details are provided in Appendix D.

5.3 Semantic Segmentation

Semantic segmentation experiments are conducted on ADE20K Zhou et al. (2019) using SemanticFPN Kirillov et al. (2019) and UperNet Xiao et al. (2018) frameworks. As shown in Tab. 3,

Table 2: Object detection and instance segmentation results. The FLOPs are computed over backbone, FPN and detection head with an input resolution of 1280×800.

Method	FLOPs	Params	AP^b	AP^b_{50}	AP^b_{75}	AP^m	AP^m_{50}	AP^m_{75}
(a) Mask R-CNN ($1\times$ schedule)								
PVT-T	240G	33M	36.7	59.2	39.3	35.1	56.7	37.3
PVT-T-PolaFormer Meng et al. (2025b)	—	—	40.4	62.4	43.9	37.4	59.4	40.3
PVT-T-ADLA	**213G**	**31M**	**40.7**(+4.0)	**63.2**(+4.0)	**44.3**(+5.0)	**38.0**(+2.9)	**60.3**(+3.6)	**40.7**(+3.4)
PVT-S	305G	44M	40.4	62.9	43.8	37.8	60.1	40.3
PVT-S-PolaFormer Meng et al. (2025b)	—	—	43.9	66.1	47.9	40.2	63.1	43.0
PVT-S-ADLA	**254G**	**40M**	**44.1**(+3.7)	**66.4**(+3.5)	**48.4**(+4.6)	**40.5**(+2.7)	**63.6**(+3.5)	**43.4**(+3.1)
Swin-T	267G	48M	43.7	66.6	47.7	39.8	63.3	42.7
Swin-T-Flatten Han et al. (2023)	268G	49M	44.2	67.3	48.5	40.2	63.8	43.0
Swin-T-ADLA	**264G**	48M	**44.6**(+0.9)	**67.5**(+0.9)	**48.7**(+1.0)	**40.5**(+0.7)	**64.1**(+0.8)	**43.3**(+0.6)
(b) Cascade Mask R-CNN ($3\times$ schedule)								
Swin-T	745G	86M	50.4	69.2	54.7	43.7	66.6	47.3
Swin-T-Flatten Han et al. (2023)	747G	87M	50.8	69.6	55.1	44.1	67.0	48.1
Swin-T-ADLA	**743G**	86M	**51.0**(+0.6)	**69.9**(+0.7)	**55.6**(+0.9)	**44.4**(+0.7)	**67.2**(+0.6)	**48.3**(+1.0)
Swin-S	838G	107M	51.9	70.7	56.3	45.0	68.2	48.8
Swin-S-ADLA	**837G**	107M	**52.2**(+0.3)	**71.0**(+0.3)	**57.0**(+0.7)	**45.4**(+0.4)	**68.5**(+0.3)	**49.2**(+0.4)

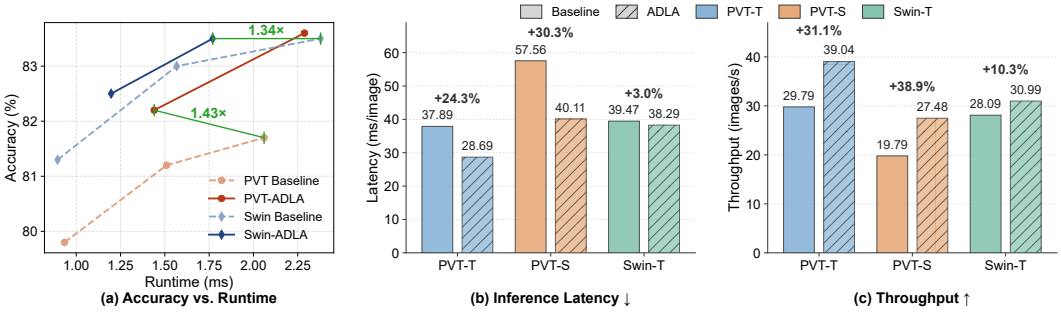


Figure 4: Efficiency comparison. **(a)** Accuracy-Runtime curves for image classification, evaluated at 224×224 resolution. **(b)** Inference latency and **(c)** throughput evaluated on the Mask R-CNN ($1\times$) framework at 1333×800 resolution. All measurements are conducted on NVIDIA A800 GPUs.

our ADLA consistently improves the baselines, yielding mIoU increases of up to 3.44%. Notably, these performance gains require fewer FLOPs (e.g., reducing PVT-S FLOPs from 225G to 171G), validating the efficiency and generalizability of our model in downstream vision tasks.

Table 3: Semantic segmentation results. The FLOPs are computed over encoders and decoders with an input image at the resolution of 512×2048.

Framework: SemanticFPN				
Backbone	FLOPs	Params	mIoU	mAcc
PVT-T	158G	17M	36.57	46.72
PVT-T-ADLA	**129G**	15M	**40.01**(+3.44)	**51.06**
PVT-S	225G	28M	41.95	53.02
PVT-S-ADLA	**171G**	24M	**44.10**(+2.15)	**55.91**

Framework: UperNet				
Backbone	FLOPs	Params	mIoU	mAcc
Swin-T	945G	60M	44.51	55.61
Swin-T-ADLA	**942G**	60M	**46.59**(+2.08)	**58.20**
Swin-S	1038G	81M	47.64	58.78
Swin-S-ADLA	**1037G**	81M	**47.97**(+0.33)	**59.50**

5.4 Image Generation

We further evaluate ADLA on class-conditional image generation using DiT-S/2 and DiT-B/2 Peebles & Xie (2023) on ImageNet-1K at 256×256 resolution, replacing only the self-attention modules. We use the official evaluation code and reference batch statistics released by OpenAI Dhariwal & Nichol (2021) to compute FID Heusel et al. (2017), sFID Nash et al. (2021), IS Salimans et al. (2016), Precision, and Recall Kynkäänniemi et al. (2019). Tab. 4 shows that ADLA outperforms baselines on most metrics and remains competitive with methods specifically designed for diffusion models.

Table 4: Image generation results. "*" denotes results reproduced using the released code. "CFG" indicates classifier-free guidance Ho & Salimans (2022) scale.

Method	CFG=1.0					CFG=4.0				
	FID-50k ↓	sFID ↓	IS ↑	P ↑	R ↑	FID-50k ↓	sFID ↓	IS ↑	P ↑	R ↑
DiT-S* Peebles & Xie (2023)	67.10	11.90	20.37	0.37	0.55	14.59	**10.21**	124.24	0.77	**0.30**
DiG-S Zhu et al. (2025)	62.06	11.77	**22.81**	**0.39**	0.56	—	—	—	—	—
LiT-S Wang et al. (2025a)	63.21	—	22.08	**0.39**	**0.58**	—	—	—	—	—
DiT-S-ADLA	**61.64**(-5.46)	**11.68**	22.72	**0.39**	**0.58**	**13.38**	10.36	**146.01**	**0.81**	0.27
DiT-B* Peebles & Xie (2023)	42.70	**8.23**	33.9	0.49	0.62	13.04	10.87	221.09	0.90	**0.21**
LiT-B Wang et al. (2025a)	40.86	—	35.8	0.50	**0.63**	—	—	—	—	—
DiT-B-ADLA	**39.80**(-2.90)	**8.23**	**36.6**	**0.51**	**0.63**	**12.22**	**10.74**	**242.31**	**0.92**	0.20

Table 5: Ablation of ADLA architectural components on DeiT-T (without DWC for a controlled assessment of attention gains; see Appendix E.3 for DWC's role). "Diff." denotes differential fusion. Static† kernel follows Meng et al. (2025b).

Framework	Focusing Kernel	GWS Application	Params	FLOPs	Acc%
Vanilla LA	—	—	5.7M	1.1G	70.2
Diff. LA	—	—	5.7M	1.1G	70.8
Diff. LA	Static†	—	5.7M	1.1G	71.2
Diff. LA	Dynamic	—	5.7M	1.1G	72.0
Diff. LA	Dynamic	Both	5.8M	1.3G	73.2
Diff. LA	Dynamic	Saliency	5.8M	1.2G	**74.0**

5.5 Ablation Study

We conduct systematic ablations to validate ADLA's core architectural designs and mechanisms.

Effectiveness of Dynamic Kernel Routing. Starting from the vanilla linear attention (Tab. 5), introducing a basic differential linear attention yields a +0.6% improvement (70.8%). While enhancing it with a static channel-wise learnable kernel Meng et al. (2025b) provides a modest +0.4% gain, our dynamic kernel routing delivers a +1.2% improvement, boosting accuracy to 72.0%. This indicates that dynamically adapting the focusing degree to token semantics is more effective than applying static kernel configurations.

Superiority of the Asymmetric Design. Building upon dynamic routing (Tab. 5), we incorporate GWS to validate our structural design. Conventional differential attention typically adopts a symmetric paradigm, where both branches share an identical mathematical formulation Ye et al. (2024). Following this standard by applying GWS to both branches yields a suboptimal 73.2% while increasing the cost to 1.3G FLOPs. Fundamentally, applying GWS to the redundancy branch destroys its inherently smooth prior. It forces the branch to capture certain salient features, leading to the loss of crucial information during subtraction. Our asymmetric design applies GWS exclusively to the saliency branch while preserving vanilla linear attention in the redundancy branch. This decouples discriminative feature extraction from broad redundancy modeling, achieving a higher accuracy (74.0%) with fewer FLOPs (1.2G). These results validate the effectiveness of our asymmetric design.

Additional Ablations. In the Appendix, we further validate the necessity of the differential design against a non-differential baseline (Appendix E.1), examine the sensitivity of key hyperparameters (Appendix E.2), and evaluate model stability across multiple random seeds (Appendix C).

6 Conclusion

We present ADLA, an asymmetric differential linear attention for vision. Rather than merely treating the smooth response of linear attention as a defect to eliminate, ADLA exploits it as a redundancy prior over visual tokens. It introduces a discriminative saliency branch and a smooth redundancy branch, whose differential interaction suppresses visual redundancy, and further incorporates dynamic kernel routing and group-weighted key-value summarization to enhance feature focusing. Extensive experiments on various visual tasks validate the effectiveness and generalization of ADLA.

AI USE STATEMENT

In this work, generative AI tools (such as Large Language Models) were utilized solely for grammar refinement, language polishing, and editorial assistance in preparing the manuscript. No generative AI tools were used to formulate the scientific hypotheses, design the ADLA architecture, produce experimental data, or draw scientific conclusions. The authors have carefully reviewed all manuscript text and assume full responsibility for the final content, claims, and intellectual integrity of this work.

ETHICS STATEMENT

The primary objective of this work is to improve the computational efficiency and representational capacity of vision models. All experiments are conducted on publicly available academic benchmarks (ImageNet-1K, COCO, and ADE20K) under standard ethical licenses, involving no human subjects or personally identifiable information. Our research does not promote discriminatory, privacy-invasive, or malicious uses. A more detailed ethical discussion is presented in Appendix I.

REPRODUCIBILITY STATEMENT

To ensure the reproducibility of our work, we have taken the following measures: (1) Detailed experimental setups, training protocols, and hyperparameter configurations across all benchmarks (ImageNet-1K, COCO, and ADE20K) are documented in Appendix G. (2) We explicitly evaluate run-to-run stability across multiple independent random seeds (reported in Tab. 6). (3) Anonymized source code is submitted in the Supplementary Material and will be made publicly available upon acceptance.

REFERENCES

Fan Bao, Shen Nie, Kaiwen Xue, Yue Cao, Chongxuan Li, Hang Su, and Jun Zhu. All are worth words: A vit backbone for diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 22669–22679, 2023.

Philipp Becker, Abhinav Mehrotra, Ruchika Chavhan, Malcolm Chadwick, Luca Morreale, Mehdi Noroozi, Alberto Gil CP Ramos, and Sourav Bhattacharya. Edit: Efficient diffusion transformers with linear compressed attention. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 19608–19616, 2025.

Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, and Judy Hoffman. Hydra attention: Efficient attention with many heads. In *European conference on computer vision*, pp. 35–49. Springer, 2022.

Han Cai, Chuang Gan, and Song Han. Efficientvit: Enhanced linear attention for high-resolution low-computation visual recognition. *arXiv preprint arXiv:2205.14756*, 3(1), 2022.

Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6154–6162, 2018.

Boyuan Cao, Xingbo Yao, Chenhui Wang, Jiaxin Ye, Yujie Wei, and Hongming Shan. Dynamic differential linear attention: Enhancing linear diffusion transformer for high-quality image generation. *arXiv preprint arXiv:2601.13683*, 2026.

Yuan Cao and Dong Wang. Saga: Selective adaptive gating for efficient and expressive linear attention. *arXiv preprint arXiv:2509.12817*, 2025.

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.

Zihang Dai, Hanxiao Liu, Quoc V Le, and Mingxing Tan. Coatnet: Marrying convolution and attention for all data sizes. *Advances in neural information processing systems*, 34:3965–3977, 2021.

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.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 12124–12134, 2022.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Qihang Fan, Huaibo Huang, Yuang Ai, and Ran He. Rectifying magnitude neglect in linear attention. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 21505–21514, 2025a.

Qihang Fan, Huaibo Huang, and Ran He. Breaking the low-rank dilemma of linear attention. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 25271–25280, 2025b.

Jianyuan Guo, Kai Han, Han Wu, Yehui Tang, Xinghao Chen, Yunhe Wang, and Chang Xu. Cmt: Convolutional neural networks meet vision transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 12175–12185, 2022.

Dongchen Han, Xuran Pan, Yizeng Han, Shiji Song, and Gao Huang. Flatten transformer: Vision transformer using focused linear attention. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 5961–5971, 2023.

Dongchen Han, Yifan Pu, Zhuofan Xia, Yizeng Han, Xuran Pan, Xiu Li, Jiwen Lu, Shiji Song, and Gao Huang. Bridging the divide: Reconsidering softmax and linear attention. *Advances in Neural Information Processing Systems*, 37:79221–79245, 2024a.

Dongchen Han, Tianzhu Ye, Yizeng Han, Zhuofan Xia, Siyuan Pan, Pengfei Wan, Shiji Song, and Gao Huang. Agent attention: On the integration of softmax and linear attention. In *European conference on computer vision*, pp. 124–140. Springer, 2024b.

Dongchen Han, Tianyu Li, Ziyi Wang, and Gao Huang. Vision transformers are circulant attention learners. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 21549–21557, 2026.

Ali Hassani, Steven Walton, Jiachen Li, Shen Li, and Humphrey Shi. Neighborhood attention transformer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 6185–6194, 2023.

Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pp. 2961–2969, 2017.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 16000–16009, 2022.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Qibin Hou, Cheng-Ze Lu, Ming-Ming Cheng, and Jiashi Feng. Conv2former: A simple transformer-style convnet for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, 46(12):8274–8283, 2024.

Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016.

Praneeth Kacham, Vahab Mirrokni, and Peilin Zhong. Polysketchformer: Fast transformers via sketching polynomial kernels. *arXiv preprint arXiv:2310.01655*, 2023.

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.

Alexander Kirillov, Ross Girshick, Kaiming He, and Piotr Dollár. Panoptic feature pyramid networks. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6392–6401. IEEE, 2019.

Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Improved precision and recall metric for assessing generative models. *Advances in neural information processing systems*, 32, 2019.

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.

Songhua Liu, Weihao Yu, Zhenxiong Tan, and Xinchao Wang. Linfusion: 1 gpu, 1 minute, 16k image. *arXiv preprint arXiv:2409.02097*, 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.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Jiachen Lu, Jinghan Yao, Junge Zhang, Xiatian Zhu, Hang Xu, Weiguo Gao, Chunjing Xu, Tao Xiang, and Li Zhang. Soft: Softmax-free transformer with linear complexity. *Advances in Neural Information Processing Systems*, 34:21297–21309, 2021.

Weikang Meng, Yadan Luo, Liangyu Huo, Yaowei Wang, Xin Li, and Zheng Zhang. Nalaformer: Norm-aware linear attention for transformer models. *arXiv e-prints*, pp. arXiv–2506, 2025a.

Weikang Meng, Yadan Luo, Xin Li, Dongmei Jiang, and Zheng Zhang. Polaformer: Polarity-aware linear attention for vision transformers. *arXiv preprint arXiv:2501.15061*, 2025b.

Charlie Nash, Jacob Menick, Sander Dieleman, and Peter W Battaglia. Generating images with sparse representations. *arXiv preprint arXiv:2103.03841*, 2021.

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.

Yifan Pu, Jixuan Ying, Qixiu Li, Tianzhu Ye, Dongchen Han, Xiaochen Wang, Ziyi Wang, Xinyu Shao, Gao Huang, and Xiu Li. Linear differential vision transformer: Learning visual contrasts via pairwise differentials. *arXiv preprint arXiv:2511.00833*, 2025.

Zhen Qin, Xiaodong Han, Weixuan Sun, Dongxu Li, Lingpeng Kong, Nick Barnes, and Yiran Zhong. The devil in linear transformer. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 7025–7041, 2022.

Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. Dynamicvit: Efficient vision transformers with dynamic token sparsification. *Advances in neural information processing systems*, 34:13937–13949, 2021.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241. Springer, 2015.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *Advances in neural information processing systems*, 29, 2016.

Zhuoran Shen, Mingyuan Zhang, Haiyu Zhao, Shuai Yi, and Hongsheng Li. Efficient attention: Attention with linear complexities. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pp. 3531–3539, 2021.

Chenyang Si, Weihao Yu, Pan Zhou, Yichen Zhou, Xinchao Wang, and Shuicheng Yan. Inception transformer. *Advances in neural information processing systems*, 35:23495–23509, 2022.

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.

Jiahao Wang, Ning Kang, Lewei Yao, Mengzhao Chen, Chengyue Wu, Songyang Zhang, Shuchen Xue, Yong Liu, Taiqiang Wu, Xihui Liu, et al. Lit: Delving into a simple linear diffusion transformer for image generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 16068–16078, 2025a.

Shuai Wang, Zhi Tian, Weilin Huang, and Limin Wang. Ddt: Decoupled diffusion transformer. *arXiv preprint arXiv:2504.05741*, 2025b.

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.

Zhuofan Xia, Xuran Pan, Shiji Song, Li Erran Li, and Gao Huang. Vision transformer with deformable attention. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4794–4803, 2022.

Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *European conference on computer vision*, pp. 432–448. Springer, 2018.

Enze Xie, Junsong Chen, Junyu Chen, Han Cai, Haotian Tang, Yujun Lin, Zhekai Zhang, Muyang Li, Ligeng Zhu, Yao Lu, et al. Sana: Efficient high-resolution image synthesis with linear diffusion transformers. *arXiv preprint arXiv:2410.10629*, 2024.

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.

Zhiyu Yao, Jian Wang, Haixu Wu, Jingdong Wang, and Mingsheng Long. Mobile attention: mobile-friendly linear-attention for vision transformers. In *Forty-first International Conference on Machine Learning*, 2024.

Tianzhu Ye, Li Dong, Yuqing Xia, Yutao Sun, Yi Zhu, Gao Huang, and Furu Wei. Differential transformer. *arXiv preprint arXiv:2410.05258*, 2024.

Haoran You, Yunyang Xiong, Xiaoliang Dai, Bichen Wu, Peizhao Zhang, Haoqi Fan, Peter Vajda, and Yingyan Celine Lin. Castling-vit: Compressing self-attention via switching towards linear-angular attention at vision transformer inference. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 14431–14442, 2023.

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.

Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.

Kewei Zhang, Ye Huang, Yufan Deng, Jincheng Yu, Junsong Chen, Huan Ling, Enze Xie, and Daquan Zhou. Mhla: Restoring expressivity of linear attention via token-level multi-head. *arXiv preprint arXiv:2601.07832*, 2026.

Michael Zhang, Kush Bhatia, Hermann Kumbong, and Christopher Ré. The hedgehog & the porcupine: Expressive linear attentions with softmax mimicry. *arXiv preprint arXiv:2402.04347*, 2024.

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.

Lei Zhu, Xinjiang Wang, Zhanghan Ke, Wayne Zhang, and Rynson Lau. Biformer: Vision transformer with bi-level routing attention. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

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.

APPENDIX

A MORE ABOUT RELATED WORK

A.1 DIFFUSION TRANSFORMER

Traditional diffusion models rely almost entirely on convolution-based U-Nets Ronneberger et al. (2015), but with the growing demand for large-scale data and long-sequence modeling, Transformers have gradually been introduced into the diffusion framework due to their superior global modeling capabilities. DiT Peebles & Xie (2023) models the diffusion denoising process using the ViT architecture and designs adaptive normalization to inject time and conditioning information; U-ViT Bao et al. (2023) explores a similar direction, integrating U-Net skip connections with Transformer global modeling; DDT Wang et al. (2025b) alleviates the slow training convergence issue in large models by decoupling the low-frequency semantic encoding and high-frequency decoding processes. Some works Liu et al. (2024); Xie et al. (2024); Becker et al. (2025); Wang et al. (2025a); Zhu et al. (2025); Cao et al. (2026) employ linear attention within DiT to improve model efficiency. Among them, DyDiLA Cao et al. (2026) routes Q, K, Q', and K' separately to power kernels before token-level differencing. ADLA instead selects a shared exponent for each token's Q and K from its input feature, then subtracts the output of a linear-attention redundancy branch from that of a GWS-based saliency branch. Sharing the routing decision reduces the number of routing operations per token.

A.2 DIFFERENTIAL ATTENTION

Differential Attention Ye et al. (2024) introduces two groups of query-key projections and suppresses common-mode noise by subtracting two attention maps, thereby enhancing discriminative information. Specifically, two groups of queries and keys, along with one value projection, are generated as:

$$[\mathbf{Q}_1;\mathbf{Q}_2] = \mathbf{X}[\mathbf{W}_{Q_1};\mathbf{W}_{Q_2}], \quad [\mathbf{K}_1;\mathbf{K}_2] = \mathbf{X}[\mathbf{W}_{K_1};\mathbf{W}_{K_2}], \quad \mathbf{V} = \mathbf{X}\mathbf{W}_V, \tag{15}$$

where $\mathbf{W}_{Q_1}, \mathbf{W}_{Q_2}, \mathbf{W}_{K_1}, \mathbf{W}_{K_2} \in \mathbb{R}^{d\times d/2}, \mathbf{W}_V \in \mathbb{R}^{d\times d}$. Two softmax attention maps are then computed separately:

$$\mathbf{A}_1 = \mathrm{Softmax}\left(\frac{\mathbf{Q}_1\mathbf{K}_1^\top}{\sqrt{d/2}}\right), \quad \mathbf{A}_2 = \mathrm{Softmax}\left(\frac{\mathbf{Q}_2\mathbf{K}_2^\top}{\sqrt{d/2}}\right). \tag{16}$$

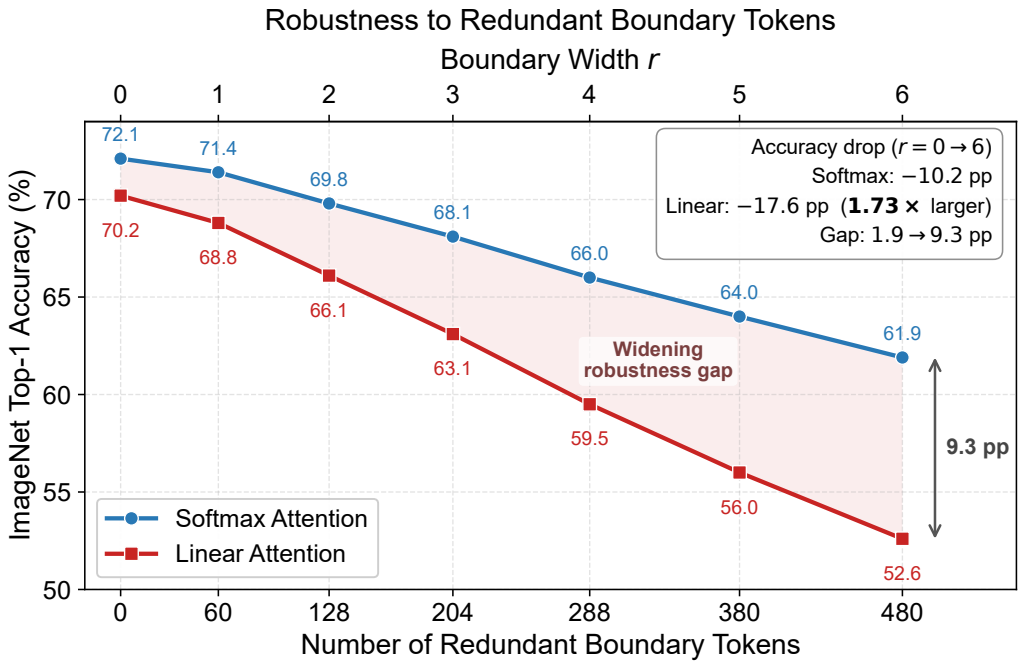


Figure 5: ImageNet Top-1 accuracy under increasing numbers of redundant boundary tokens. The original 196 tokens remain unchanged. Both attention mechanisms are affected by token redundancy, but linear attention exhibits a substantially faster degradation. The shaded region highlights the widening performance gap.

The final attention weights are defined as their difference, modulated by a learnable scalar λ:

$$\mathbf{A} = \mathbf{A}_1 - \lambda \mathbf{A}_2, \tag{17}$$

where $\lambda = \exp(\boldsymbol{\lambda}_{q_1} \cdot \boldsymbol{\lambda}_{k_1}) - \exp(\boldsymbol{\lambda}_{q_2} \cdot \boldsymbol{\lambda}_{k_2}) + \lambda_{\text{init}}$, and $\boldsymbol{\lambda}_{q_1}, \boldsymbol{\lambda}_{k_1}, \boldsymbol{\lambda}_{q_2}, \boldsymbol{\lambda}_{k_2} \in \mathbb{R}^d$ are learnable vectors. The initialization $\lambda_{\text{init}} = 0.8 - 0.6e^{-0.3 \cdot \text{depth}}$ decays with network depth, where depth is the layer index. The output is computed as:

$$\hat{\mathbf{O}} = \mathbf{A}\mathbf{V}, \quad \mathbf{O} = (1 - \lambda_{\text{init}}) \cdot \text{RMSNorm}(\hat{\mathbf{O}}). \tag{18}$$

Differential attention still maintains a complexity of $\mathcal{O}(N^2 d)$, but through the differential operation it can more effectively focus on important contexts and has demonstrated stronger robustness in language modeling. In our approach, we adopt the learnable scalar λ and RMSNorm from Differential Attention Ye et al. (2024).

B Robustness to Redundant Information

We investigate whether the performance difference between Softmax attention and linear attention is related to their robustness to redundant visual tokens. Starting from the original 14×14 patch grid, which contains 196 tokens, we replicate the boundary tokens outward by r rings while keeping all original foreground tokens unchanged. The resulting number of spatial tokens is

$$N(r) = (14 + 2r)^2, \tag{19}$$

and the number of additionally introduced redundant tokens is

$$N_{\text{rt}}(r) = (14 + 2r)^2 - 196. \tag{20}$$

We evaluate $r \in \{0, 1, \ldots, 6\}$, corresponding to between 0 and 480 redundant tokens. Because the replicated tokens contain no additional semantic information, this setting isolates the influence of token multiplicity from that of additional visual content.

As shown in Fig. 5, increasing the number of redundant tokens causes a monotonic accuracy degradation for both attention mechanisms. Softmax attention decreases from 72.1% to 61.9%, corresponding to a drop of 10.2 percentage points. Linear attention decreases from 70.2% to 52.6%, resulting in a considerably larger drop of 17.6 percentage points. Thus, under the largest redundancy level, linear attention suffers an additional degradation of 7.4 percentage points relative to Softmax attention.

The performance gap also grows consistently with token redundancy. Without additional tokens, the gap between Softmax and linear attention is only 1.9 percentage points. After introducing 480 redundant tokens, this gap increases to 9.3 percentage points. Therefore, redundant tokens do not merely reduce the accuracy of both models by a similar amount; instead, they progressively amplify the disadvantage of linear attention.

These results provide evidence that sensitivity to redundant tokens is an important contributing factor to the lower performance of linear attention, especially when the input sequence contains weakly informative tokens.

C More Experimental Comparison Results

Comparison with Representative Linear Attentions. We report additional comparisons with representative linear attention methods in Tab. 6. To evaluate the robustness of ADLA, we independently trained each model three times using random seeds $\{0, 17, 42\}$. The reported ADLA accuracies are the arithmetic mean and sample standard deviation over the three runs. Across all repeated experiments, the run-to-run accuracy range is no greater than 0.1 percentage points, and the maximum sample standard deviation is 0.06 percentage points, demonstrating stable performance across different random initializations.

Since run-level results are unavailable for methods quoted from previous publications, we do not perform statistical hypothesis tests against these reported values. Instead, we explicitly report the variance of our repeated experiments to provide a transparent assessment of result stability.

ADLA demonstrates consistently strong performance across different backbone families. The repeated experiments exhibit limited variation, with a maximum sample standard deviation of only 0.06 percentage points. These results indicate that the reported performance of ADLA is stable across random initializations and is not sensitive to a particular training run.

Comparison with Closely Related Methods. We further discuss the relationship between ADLA and the most closely related methods, particularly VCA Pu et al. (2025) and MHLA Zhang et al. (2026).

Comparison with VCA. VCA employs spatially pooled contrast tokens as intermediate representations and models visual contrasts through paired positive and negative interaction branches. Its two-stage interaction remains softmax-normalized and adopts a structurally paired differential formulation. In contrast, ADLA investigates differential modeling directly within linear attention and introduces asymmetric differential branches to better separate informative visual correlations from redundant responses. The effectiveness of this asymmetric formulation over its symmetric counterpart is further demonstrated by the ablation experiments in the main paper.

As reported in Tab. 6, ADLA achieves higher accuracy than VCA in most backbone settings. These results suggest that asymmetric differential linear attention can provide consistent benefits across both plain and hierarchical vision architectures. More importantly, ADLA focuses on exploring how differential modeling can improve the ability of linear attention to capture informative visual relationships while reducing the influence of redundant responses.

Comparison with MHLA. ADLA is related to MHLA in that both methods partition tokens into multiple groups, construct key–value summaries within each group, and aggregate information across groups using group-level coefficients. However, the two methods differ fundamentally in how the cross-group interactions are determined.

MHLA employs an input-independent cross-group weight matrix initialized according to spatial distance and subsequently optimized as a static model parameter. Once trained, the same cross-group mixing matrix is applied to all samples. In contrast, our Group-Weighted KV Summarization (GWS) predicts group-level logits from the current visual features and normalizes them using a

Table 6: Extended comparison of different linear attention methods on ImageNet-1K. ADLA accuracies are reported as mean ± sample standard deviation over three independent training runs with random seeds $\{0, 17, 42\}$. The run-to-run range is at most 0.1 percentage points across all repeated settings. The Swin-B result marked by † is obtained from a single training run and is not included in the repeated-run analysis.

Linear Attention	Params	FLOPs	Acc. (%)
Comparison on DeiT-T Setting			
DeiT-T Touvron et al. (2021)	5.7M	1.2G	72.2
Efficient Attn Shen et al. (2021)	5.7M	1.1G	70.2
Hydra Attn Bolya et al. (2022)	5.7M	1.1G	68.3
Enhanced Attn Cai et al. (2022)	5.8M	1.1G	72.9
Angular Attn You et al. (2023)	5.7M	1.1G	70.8
Mobi Attn Yao et al. (2024)	5.7M	1.2G	73.3
FLatten Attn Han et al. (2023)	6.1M	1.1G	74.1
Agent Attn Han et al. (2024b)	6.0M	1.2G	74.9
InLine Attn Han et al. (2024a)	6.5M	1.1G	74.5
VCA Pu et al. (2025)	6.0M	1.2G	75.6
MALA Fan et al. (2025a)	6.3M	1.1G	75.1
PolaFormer Meng et al. (2025b)	6.1M	1.2G	74.6
CA Han et al. (2026)	6.1M	1.2G	75.0
ADLA (Ours)	**5.8M**	1.2G	$\mathbf{75.87 \pm 0.06}$
Comparison on DeiT-S Setting			
DeiT-S Touvron et al. (2021)	22M	4.6G	79.8
InLine Attn Han et al. (2024a)	17M	5.0G	80.2
RALA Fan et al. (2025b)	24M	4.6G	80.4
MALA Fan et al. (2025a)	24M	4.6G	80.3
VCA Pu et al. (2025)	23M	4.6G	80.7
ADLA (Ours)	22M	**4.5G**	$\mathbf{80.80 \pm 0.00}$
Comparison on PVT-T Setting			
PVT-T Wang et al. (2021)	13M	1.9G	75.1
FLatten Attn Han et al. (2023)	12M	2.0G	77.8
InLine Attn Han et al. (2024a)	12M	2.0G	78.2
PolaFormer Meng et al. (2025b)	12M	2.0G	78.8
VCA Pu et al. (2025)	12M	2.0G	78.2
CA Han et al. (2026)	12M	2.0G	78.1
ADLA (Ours)	**11M**	2.0G	$\mathbf{78.77 \pm 0.06}$

Linear Attention	Params	FLOPs	Acc. (%)
Comparison on PVT-S Setting			
PVT-S Wang et al. (2021)	25M	3.8G	79.8
FLatten Attn Han et al. (2023)	22M	4.0G	81.7
PolaFormer Meng et al. (2025b)	21M	4.1G	81.9
InLine Attn Han et al. (2024a)	22M	3.9G	82.0
VCA Pu et al. (2025)	21M	4.1G	82.3
CA Han et al. (2026)	23M	4.0G	81.7
ADLA (Ours)	**20M**	4.1G	82.17 ± 0.06
Comparison on PVT-M Setting			
PVT-M Wang et al. (2021)	35.9M	7.0G	81.2
InLine Attn Han et al. (2024a)	37.6M	6.9G	83.2
VCA Pu et al. (2025)	35.8M	7.2G	83.2
ADLA (Ours)	**35.5M**	7.2G	$\mathbf{83.60 \pm 0.00}$
Comparison on Swin-T Setting			
Swin-T Liu et al. (2021)	29M	4.5G	81.3
Efficient Attn Shen et al. (2021)	29M	4.5G	81.0
Hydra Attn Bolya et al. (2022)	29M	4.5G	80.7
Enhanced Attn Cai et al. (2022)	29M	4.5G	81.8
Angular Attn You et al. (2023)	29M	4.5G	79.4
FLatten Attn Han et al. (2023)	29M	4.5G	82.1
VCA Pu et al. (2025)	29M	4.6G	82.3
CA Han et al. (2026)	28M	4.6G	82.2
ADLA (Ours)	29M	4.5G	$\mathbf{82.47 \pm 0.06}$
Comparison on Swin-B Setting			
Swin-B Liu et al. (2021)	88M	15.4G	83.5
FLatten Attn Han et al. (2023)	89M	15.4G	83.8
PolaFormer Meng et al. (2025b)	88M	15.4G	83.8
ADLA (Ours)	88M	15.5G	$\mathbf{84.0}^{\dagger}$

Table 7: Controlled comparison between MHLA-style static cross-group weighting and the proposed Group-Weighted KV Summarization (GWS).

Method	**DeiT-T**	**DeiT-S**
MHLA-style weighting Zhang et al. (2026)	75.4	80.5
ADLA with GWS (Ours)	$\mathbf{75.87 \pm 0.06}$	$\mathbf{80.80 \pm 0.00}$

group-level softmax. Therefore, the main distinction of GWS lies in its input-dependent dynamic mixing mechanism. While MHLA learns a dataset-level static interaction pattern, GWS dynamically selects and combines group-wise information according to the content of each input. This allows ADLA to model adaptive cross-group relationships without relying on a fixed spatial mixing pattern.

The original MHLA adaptation to DeiT additionally introduces several architectural modifications, including removing the class token, applying global average pooling over patch tokens, and modifying the feature mappings of queries and keys. Directly comparing the complete MHLA model with ADLA would therefore conflate the effect of cross-group weighting with these additional architectural changes.

To provide a controlled comparison, we incorporate only the MHLA-style input-independent, distance-initialized cross-group weight matrix into our grouped linear attention framework. The backbone structure, group partition, feature mappings, attention formulation, and training recipe are otherwise kept identical to our DeiT setting.

As shown in Tab. 7, GWS improves the ImageNet-1K top-1 accuracy by 0.5 and 0.3 percentage points on DeiT-T and DeiT-S, respectively. This controlled comparison indicates that dynamically predicting cross-group weights from the current input is more effective than applying a shared static mixing matrix to all samples.

D RUNTIME EFFICIENCY ANALYSIS

To evaluate the practical computational efficiency of our Asymmetric Differential Linear Attention (ADLA), we conduct speed and memory measurements based on Mask R-CNN He et al. (2017).

D.1 EXPERIMENTAL SETUP

Our efficiency measurements are conducted on Mask R-CNN models trained with the standard $1\times$ schedule, comparing baseline backbones (PVT-T, PVT-S, Swin-T) with their corresponding ADLA variants (PVT-T-ADLA, PVT-S-ADLA, Swin-T-ADLA). All evaluations are performed on NVIDIA A800-SXM4-80GB GPUs across four image resolutions: 800×480, 1000×600, 1333×800, and 2000×1200. Following standard settings, input images are padded so that their spatial dimensions are divisible by 32.

- **Single-Image Inference Latency (ms):** Measured on a single GPU with a batch size of 1. The timing protocol executes 50 warmup iterations followed by averaging 200 consecutive complete forward passes. The measured scope includes the backbone, FPN, RPN, RoI head, and post-processing steps.
- **Inference Throughput (img/s):** Measured on a single GPU with a batch size of 4 under continuous batch inference.
- **Training Peak Memory (GB):** Measured during multi-GPU training across 4 GPUs with a batch size of 4 per GPU (global batch size of 16). We report the maximum peak memory recorded across all 4 GPUs.

D.2 OBSERVATIONS AND ANALYSIS

As shown in Tab. 8, ADLA consistently improves inference latency and throughput across various resolutions, with speedup advantages becoming increasingly pronounced at higher resolutions on PVT backbones. This is because PVT's attention mechanism still incurs super-linear computational scaling with larger token counts, whereas ADLA's true $\mathcal{O}(N)$ linear complexity drastically cuts latency at extreme resolutions (e.g., reducing PVT-S latency by 61.9% at 2000×1200). In contrast, the speedup on Swin is more moderate because Swin achieves approximate linear complexity via fixed-size local windows; higher resolutions merely increase the number of windows while keeping tokens per window constant, leaving limited theoretical space for further linear attention acceleration. Additionally, ADLA introduces a moderate increase in training memory, while remaining within a practical range across conventional resolutions.

E SUPPLEMENTARY ABLATION STUDIES

To provide deeper insights into ADLA, we present supplementary ablation experiments.

E.1 EFFECTIVENESS OF OUR ASYMMETRIC DIFFERENTIAL MECHANISM

In the main paper, our ablation study primarily focuses on the structural choices within the differential linear attention (Diff. LA) framework. To further validate the necessity of the differential mechanism itself, we provide a supplementary comparison against a non-differential baseline.

In this non-differential setting, the query and key representations are not split along the channel dimension. Instead, we apply both Dynamic Kernel Routing and Group-Weighted KV Summarization (GWS) directly to the entire module (i.e., the full feature dimension d). As shown in Table 9, this configuration achieves an accuracy of 73.1% with 5.9M parameters and 1.3G FLOPs.

Compared to our asymmetric differential design (74.0% Acc, 5.8M Params, 1.2G FLOPs), the non-differential framework exhibits two inherent drawbacks:

- **Suboptimal Redundancy Suppression:** Despite the strong feature extraction capability of GWS, the absence of a redundancy branch means there is no smooth background prior for

Table 8: Efficiency measurement with baseline and ADLA backbones across different resolutions on NVIDIA A800 GPUs.

Model Backbone	Latency (ms/img) ↓	Throughput (img/s) ↑	Peak Mem (GB) ↓
Input Resolution: 800×480			
PVT-T	18.84	70.19	4.08
PVT-T-ADLA (Ours)	**17.29**	**72.38**	4.83
PVT-S	25.40	53.59	6.28
PVT-S-ADLA (Ours)	**22.69**	**55.47**	7.69
Swin-T	21.86	55.56	5.15
Swin-T-ADLA (Ours)	**21.06**	**57.40**	6.45
Input Resolution: 1000×600			
PVT-T	23.69	51.20	6.76
PVT-T-ADLA (Ours)	**20.95**	**56.29**	8.04
PVT-S	33.25	36.10	10.67
PVT-S-ADLA (Ours)	**28.41**	**41.24**	12.80
Swin-T	27.53	41.95	7.65
Swin-T-ADLA (Ours)	**26.00**	**43.46**	9.51
Input Resolution: 1333×800			
PVT-T	37.89	29.79	13.00
PVT-T-ADLA (Ours)	**28.69**	**39.04**	13.41
PVT-S	57.56	19.79	21.16
PVT-S-ADLA (Ours)	**40.11**	**27.48**	21.94
Swin-T	39.47	28.09	12.50
Swin-T-ADLA (Ours)	**38.29**	**30.99**	16.22
Input Resolution: 2000×1200			
PVT-T	98.41	10.21	40.67
PVT-T-ADLA (Ours)	**58.62**	**17.34**	**29.63**
PVT-S	223.28	3.77	69.34
PVT-S-ADLA (Ours)	**85.10**	**8.62**	**48.63**
Swin-T	90.62	11.66	27.18
Swin-T-ADLA (Ours)	**87.07**	**15.52**	36.52

Table 9: Comparison between the non-differential framework and our asymmetric differential design.

Framework	Focusing Kernel	GWS Application	Params	FLOPs	Acc. (%)
Non-Diff.	Dynamic Routing	Entire Module	5.9M	1.3G	73.1
Diff. LA (Ours)	Dynamic Routing	Saliency Branch	**5.8M**	**1.2G**	**74.0**

subtraction. Consequently, the model struggles to effectively filter out broadly distributed visual redundancy, leading to a -0.9% accuracy drop.

- **Higher Computational Overhead:** In our differential design, the GWS module only operates on half of the channels ($d/2$) within the saliency branch. Applying GWS to the full-dimensional features (d) in the non-differential framework inevitably introduces redundant parameters and heavier computational FLOPs.

This comparison confirms that the differential architecture is structurally indispensable: it not only provides a natural subtraction mechanism for visual denoising but also yields better computational efficiency through channel splitting.

E.2 Ablation on Key Hyperparameters

We conduct comprehensive sensitivity analysis on the key hyperparameters introduced in ADLA, including the number of candidate power exponents f and the scaling factor γ. The results are summarized in Tab. 10.

Table 10: Hyperparameter ablation studies on DeiT-T setting on ImageNet-1K.

(a) Candidate exponents f

f	Acc. (%)
4	75.7
6	75.8
8	**75.9**
10	75.8

(b) Scaling factor γ

γ	Acc. (%)
2	75.9
3	75.9
4	75.9
5	75.9

Impact of Candidate Power Exponents f. We vary $f \in \{4, 6, 8, 10\}$ to evaluate the effect of candidate power exponent capacity in Dynamic Kernel Routing. As shown in Tab. 10a, the top-1 accuracy remains stable across different choices. This demonstrates that ADLA is robust to the choice of f. We select $f = 8$ as default to balance dynamic non-linear expressiveness and computational overhead.

Impact of Scaling Factor γ. We evaluate the scaling factor $\gamma \in \{2, 3, 4, 5\}$ in Eq. (4) of the main paper. As shown in Tab. 10b, the top-1 accuracy consistently achieves 75.9% across all evaluated settings. The underlying reason is that γ merely acts as a scaling constant, whereas the power exponents themselves are end-to-end learnable. Consequently, varying γ has negligible impact on final accuracy. We fix $\gamma = 4$ as the default setting.

E.3 Role of Depthwise Convolution

DWC is commonly used for local feature enhancement in vision linear attention Han et al. (2023; 2024b); Meng et al. (2025b). We therefore include it in the final ADLA models, while excluding it from every variant in Tab. 5 to isolate the attention mechanism. Without DWC, ADLA improves vanilla linear attention from 70.2% to 74.0%, demonstrating a substantial gain from the proposed design alone. With DWC, the complete models achieve competitive or higher accuracy than representative DWC-equipped alternatives (Tabs. 1 and 6). Together, these results show that ADLA's effectiveness does not depend on DWC and remains competitive when this commonly used component is included.

F Attention Visualization

F.1 More Attention Weight Visualization Results

Fig. 6 presents extended attention weight visualizations across different images from the ImageNet-1K dataset. While vanilla linear attention consistently produces diffuse and oversmoothed responses that activate irrelevant background tokens, ADLA robustly concentrates attention on semantically critical foreground targets.

F.2 Non-Target Regions

A natural question regarding our asymmetric differential design is: *What happens when the query token itself is located in a redundant region (i.e., a region that makes no prominent contribution to the final visual decision)? Does the differential mechanism interfere with the representation?*

To address this, we visualize the attention weights when querying redundant tokens in Fig. 7. This visualization reveals the inherent robustness and self-consistency of our asymmetric framework. When querying redundant regions, "redundancy removal" is not strictly necessary, as the homogeneous context itself is the target. The detailed mechanics are as follows:

- **Saliency Branch:** Benefiting from the query-adaptive nature of the Group-Weighted KV Summarization, the saliency branch dynamically adjusts to the query's semantics. When querying a redundant token, it successfully matches with other similar tokens, effectively capturing the necessary homogeneous visual context without forcing attention onto irrelevant salient objects.

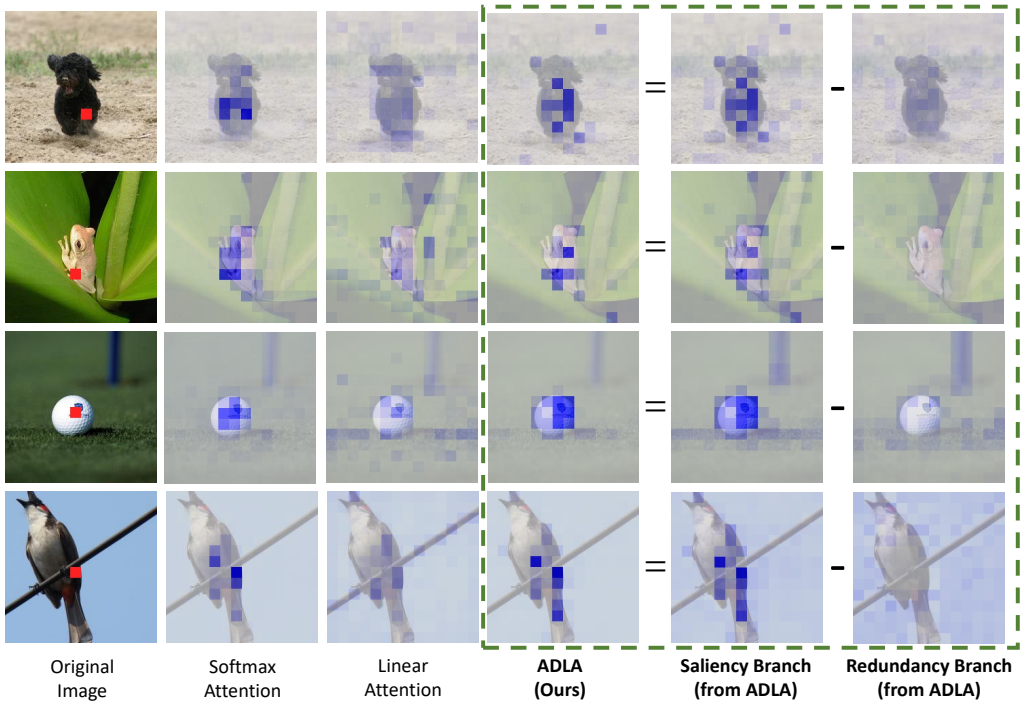


Figure 6: **Extended attention weight visualizations on ImageNet-1K using DeiT-T.** We compare Softmax attention, vanilla linear attention, and the proposed ADLA across different natural images.

- **Redundancy Branch:** In this scenario, vanilla linear attention lacks specific semantic anchors and tends toward a completely uniform, globally averaged distribution across the entire image.
- **The Differential Subtraction:** Since the redundancy branch provides merely a flat, globally averaged baseline, subtracting it from the saliency branch introduces no disruptive interference. The dominant attention weights established by the saliency branch on the redundant tokens are safely preserved.

This analysis demonstrates the dynamic adaptability of ADLA. When querying critical features, the redundancy branch actively subtracts background noise. However, when querying redundant regions, the saliency branch intelligently shifts to capturing the required homogeneous context, and the redundancy branch safely degenerates into a uniform baseline. This ensures that the differential subtraction consistently enhances or preserves the representation without ever degrading it.

G Experiment Settings

G.1 Hyperparameter Settings of ADLA

The selection process can be found in Appendix E.2. We list the key hyperparameters introduced in ADLA and explain their default settings. The number of candidate power exponents f is set to 8. The scaling factor γ in Eq. (4) of the main paper is set to 4, which provides a balance between sharpening strength and training stability.

For grouping, we adopt stage-wise multipliers $\{\alpha_s\}$, where s indexes the network stage. Let N_s be the token count at stage s and define $G_s = \lfloor \alpha_s \sqrt{N_s} \rceil$. In the computationally dominant high-resolution stages, α_s is chosen such that G_s^2 is close to or smaller than N_s (e.g., $\alpha_s = 1$ or 0.6), while larger multipliers are used only in later low-resolution stages and never exceed 3. This yields low overhead in practice while preserving sufficient grouping flexibility in deeper layers. This design provides a favorable trade-off between model accuracy and computational efficiency.

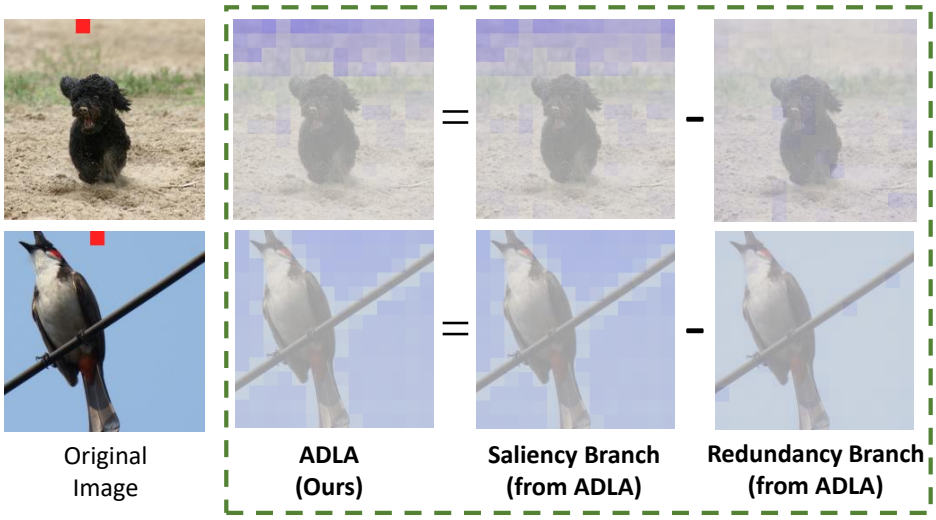


Figure 7: Attention visualization when the query token (red dot) is located in a **redundant region**. The GWS-enhanced saliency branch adaptively captures the homogeneous context, while the vanilla redundancy branch provides a uniform baseline, ensuring that the differential subtraction preserves the accurate capture of contextual information.

G.2 Dataset and Training Setup

All experiments are conducted on 8 NVIDIA A800 GPUs. We detail the training recipes for each task below.

ImageNet-1K Classification. ImageNet Deng et al. (2009) comprises 1000 classes, with 1.2 million training images and 50,000 validation images. To ensure a fair comparison, we train our ADLA model with the same settings as the corresponding baseline model. All models are trained from scratch for 300 epochs with the AdamW Loshchilov & Hutter (2019) optimizer, using a cosine learning-rate decay and 20 epochs of linear warm-up. The base learning rate is set to 1×10^{-3} for a batch size of 1024 and is linearly scaled with the total batch size. Following DeiT Touvron et al. (2021), we apply RandAugment Cubuk et al. (2020), Mixup Zhang et al. (2017), CutMix Yun et al. (2019), and random erasing Zhong et al. (2020) to prevent overfitting. The weight decay is set to 0.05. A more comprehensive comparison is presented in Tab. 6.

COCO Object Detection and Instance Segmentation. We conduct experiments on the COCO Lin et al. (2014) object detection and instance segmentation dataset, which contains 118k training images and 5k validation images. All backbones are initialized with ImageNet-1K pretrained weights and fine-tuned on the detection task using the AdamW optimizer and the configurations specified in the corresponding original papers. Standard data augmentations, including resizing, random horizontal flipping, and normalization, are employed. We train the models with an initial learning rate of 1×10^{-4} following either a $1\times$ or $3\times$ schedule. The $1\times$ schedule spans 12 epochs, with the learning rate decayed by a factor of 10 at epochs 8 and 11. The $3\times$ schedule spans 36 epochs, with the learning rate reduced by a factor of 10 at epochs 27 and 33. All reported mAP results are evaluated at a fixed input resolution of 1333×800.

Semantic Segmentation. We evaluate semantic segmentation performance on the ADE20K Zhou et al. (2019) dataset, which consists of 20K images for training and 2K for validation. Following standard practice, all backbone networks are initialized with ImageNet-1K pretrained weights prior to fine-tuning. We implement our models using two representative frameworks. For UperNet Xiao et al. (2018), models are trained for 160K iterations using the AdamW optimizer. The initial learning rate is set to 6×10^{-5}, accompanied by a linear warmup phase of 1,500 iterations. When employing SemanticFPN Kirillov et al. (2019), the training spans 40K iterations with an initial learning rate of 2×10^{-4}, also optimized via AdamW. During training, standard data augmentations are applied, including random resizing and cropping to a resolution of 512×512. For evaluation, the shorter side of the input images is re-scaled to 512 pixels.

Table 11: Comparison of parameters and FLOPs for DiT and DiT-ADLA models.

Method	Params	FLOPs
DiT-S	33.0M	6.1G
DiT-S-ADLA	33.2M	6.0G
DiT-B	130.5M	23.0G
DiT-B-ADLA	130.9M	23.0G

Image Generation. For image generation, we build ADLA upon the official DiT Peebles & Xie (2023) codebase. Training follows the standard DDPM Ho et al. (2020) noise schedule with 1,000 denoising steps. All models are trained on ImageNet-1K at a resolution of 256×256. We adopt the DiT-S/2 and DiT-B/2 architectures, replacing the original self-attention modules with ADLA blocks while keeping all other structural configurations identical to the baseline. For evaluation, we generate 50,000 images following the standard protocol to compute FID-50K. We use the official evaluation codebase and reference batch statistics released by OpenAI Dhariwal & Nichol (2021) to assess generation quality via FID Heusel et al. (2017), sFID Nash et al. (2021), IS Salimans et al. (2016), and Precision and Recall Kynkäänniemi et al. (2019). The parameters and FLOPs of the DiT and DiT-ADLA models are detailed in Tab. 11.

H Limitations

Despite ADLA effectively suppressing redundancy and enhancing attention focus while maintaining linear complexity, several limitations remain: (i) due to resource constraints, it has not been tested on larger-scale models; (ii) the additional linear overhead may reduce speed improvements on small images; and (iii) evaluations are mainly conducted on 2-D vision tasks, while extensions to 3-D vision, video, language, and speech domains remain unexplored.

I Ethical Considerations

I.1 Intended Use and Potential Misuse

The ADLA attention mechanism proposed in this study is intended for academic research in the field of computer vision. It aims to optimize the computational efficiency and performance of Vision Transformers, supporting model development and performance evaluation for visual tasks such as image classification, object detection, and image generation. We strongly oppose the use of this technology for unauthorized visual surveillance, deepfake image generation, discriminatory visual recognition systems, malicious image analysis, or any applications that violate privacy, ethics, or legal regulations. Users are strongly advised to strictly comply with relevant research ethics and applicable laws and regulations when applying and studying this technology.

I.2 Risk Mitigation Measures

This work advocates that users establish clear compliance guidelines for the use of this technology and strictly prohibit applying ADLA to scenarios involving privacy infringement, content forgery, or malicious monitoring. It is recommended that research institutions and enterprises conduct ethical reviews prior to deploying this technology, in order to prevent safety and ethical risks arising from potential misuse.

I.3 Licensing of Existing Resources

The public datasets used in this study, including ImageNet-1K, COCO and ADE20K, are utilized in accordance with their official open-source licensing agreements and are restricted to academic research purposes. The experiments are based on open-source models such as DeiT, Swin Transformer, and DiT, all of which are properly cited and used in compliance with their respective licenses.

I.4 Environmental Impact

All experiments in this study are conducted on NVIDIA A800 high-performance GPUs, and the operation of computing resources generates certain energy consumption and carbon emissions. The total computational cost was more than 1,800 GPU hours. Efficient training strategies and reasonable scheduling of computing resources are adopted in our experiments to reduce energy consumption. We advocate researchers in the computer vision community to optimize computing resource allocation and adopt efficient algorithm design during model development, training and evaluation, so as to avoid unnecessary computing consumption and reduce the carbon footprint of scientific research activities.