HALA: HOUSEHOLDER-ADAPTED LINEAR ATTENTION FOR VISION TRANSFORMERS

Anonymous authors
Paper under double-blind review

ABSTRACT

Linear attention offers an efficient alternative to softmax attention, but its representational gap remains poorly understood. We show that part of this gap arises from a *broken* geometric symmetry. Softmax attention is coordinate-free: rotating queries and keys together preserves their inner products and leaves attention unchanged. ReLU-style linear attention, however, applies a non-negative gate along fixed feature axes, making attention sensitive to the coordinate system. This gate can passively truncate useful semantic directions before aggregation, leading to a failure mode we call coordinate-lazy truncation. To address this, we propose Householder-adapted Linear Attention (HaLA), a geometry-aware linear attention framework that turns passive truncation into *active reorientation*. By leveraging learnable Householder reflections, HaLA rotates the query-key coordinate system into the *non-negative orthant* before non-negative projection to maximize information retention. Built on this principle, HaLA introduces a cohesive multi-scale design: block-wise isometries enhance local discriminability, variance-aware modulation stabilizes long-context activation diversity, and cross-head reflections integrate dispersed subspaces through global covariance mixing. Across various vision tasks including long-sequence super-resolution and image generation, HaLA consistently strengthens linear attention and closes much of the gap to softmax attention under efficient linear-time computation. Source code can be found in the supplementary materials.

1 INTRODUCTION

Transformers have become the dominant architecture for visual and language representation learning, largely due to the ability of self-attention to aggregate global context through softmax-normalized query-key interactions Vaswani et al. (2017); Dosovitskiy et al. (2021). Despite its effectiveness, softmax attention scales quadratically with the sequence length, requiring $O(N^2d)$ computation and $O(N^2)$ attention memory for a sequence of length N and feature dimension d. This quadratic dependence is especially restrictive in *vision*, where high-resolution dense prediction, visual restoration, and large-scale generative modeling can involve thousands to *tens of thousands* of visual tokens, making global attention a dominant bottleneck.

Linear Attention (LA) (Katharopoulos et al., 2020; MiniMax et al., 2025; Meng et al., 2025b) addresses this limitation by reformulating softmax attention using kernel feature maps that enable associative reordering of matrix multiplications. This reformulation shifts the computation order from $\mathrm{Softmax}(\mathbf{Q}\mathbf{K}^\top)\mathbf{V}$ to $\phi(\mathbf{Q})(\phi(\mathbf{K})^\top\mathbf{V})$. Consequently, the complexity is reduced to $O(Nd^2)$, maintaining the benefits of global context modeling while ensuring computational efficiency for resource-intensive tasks. However, despite extensive research, linear attention methods consistently underperform their softmax-based counterparts in practice, especially in vision tasks. Understanding and closing this representational gap remains a central challenge.

Previous approaches primarily attribute this gap to the design of kernel feature maps $\phi(\cdot)$. To ensure numerical stability, linear attention typically requires non-negative query-key similarities, which are commonly enforced by *axis-aligned activations* such as ReLU (Cai et al., 2023; Han et al., 2023), $1 + \mathrm{ELU}$ (Katharopoulos et al., 2020), or exponential mappings (Nahshan et al., 2024). While these mappings stabilize the denominator of linear attention, they also impose a fixed *sign gate* on the feature coordinates. As a result, semantic directions that are meaningful under dot-product attention

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.

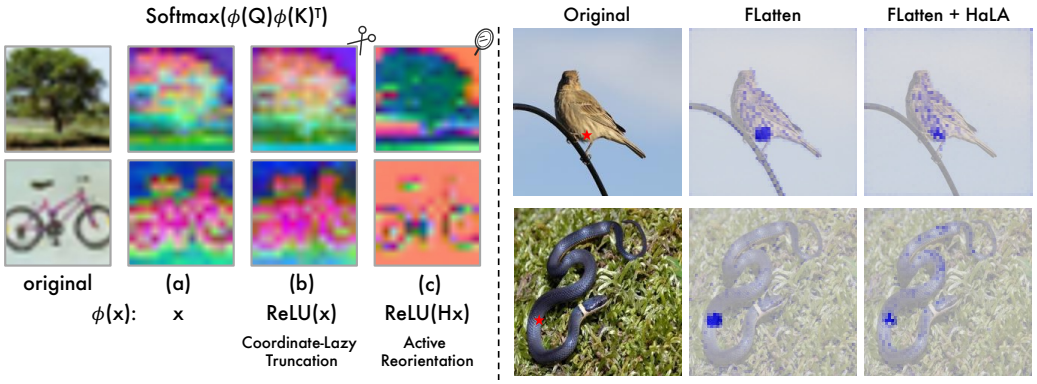

Figure 1: **Left: Feature topology under coordinate-sensitive gating.** We compare three query-key mappings: (a) *Vanilla dot-product geometry*, $\phi(\mathbf{x}) = \mathbf{x}$; (b) *Coordinate-lazy truncation*, $\phi(\mathbf{x}) = \mathrm{ReLU}(\mathbf{x})$; (c) *Active reorientation*, $\phi(\mathbf{x}) = \mathrm{ReLU}(\mathbf{Hx})$. Bound to fixed axes, the coordinate-lazy truncation in (b) causes severe global deactivation, whereas the active reorientation in (c) retains richer active structures. **Right: Attention maps** (from the final Transformer block). Due to truncation-induced global deactivation, FLatten attends only locally, causing misaligned attention. In contrast, FLatten + HaLA retains the baseline's properties while actively capturing global features, yielding accurate, target-aligned distributions.

may become "dead" after non-negative projection. Some efforts attempt to mitigate the information loss caused by strict feature mapping, for instance, Hedgehog (Zhang et al., 2024) employs MLP-exp based mappings, while Polaformer (Meng et al., 2025b) and Nalaformer (Meng et al., 2025a) explore sign-decomposition and cosine similarity. However, they largely redesign the projection after the coordinate system has already been fixed. We instead ask a more primitive question: *before applying a non-negative feature map, is the query-key coordinate system itself appropriate for an axis-aligned gate?*

Softmax attention is invariant to shared orthogonal reparameterizations of queries and keys: for any orthogonal transformation $\mathbf{H}$,

$$(\mathbf{QH})(\mathbf{KH})^\top = \mathbf{QK}^\top. \tag{1}$$

Thus, many coordinate systems are equivalent under dot-product attention. ReLU-style linear attention breaks this equivalence because the non-negative gate is tied to fixed axes:

$$\mathrm{ReLU}(\mathbf{QH})\,\mathrm{ReLU}(\mathbf{KH})^\top \neq \mathrm{ReLU}(\mathbf{Q})\,\mathrm{ReLU}(\mathbf{K})^\top \quad \text{in general.} \tag{2}$$

This coordinate sensitivity exposes a failure mode we call *coordinate-lazy truncation*: a ReLU linear attention may inherit a sign-misaligned basis, so informative directions are suppressed not because they are irrelevant, but because they *fall into inactive coordinates*. The remedy is therefore not only to design a stronger feature map, but to **reorient** the query-key basis before the non-negative gate.

We visualize this effect in Fig. 1 by comparing standard ReLU truncation with a Householder-reoriented ReLU map. The Householder reflection $\mathbf{H}$ preserves pre-gate inner products, i.e., $\langle \mathbf{Hq}, \mathbf{Hk} \rangle = \langle \mathbf{q}, \mathbf{k} \rangle$, but changes how features are exposed to the ReLU gate. Standard ReLU produces a sparse and blurred topology, suggesting that fixed-axis truncation attenuates fine-grained semantic contrast. In contrast, $\phi(\mathbf{x}) = \mathrm{ReLU}(\mathbf{Hx})$ retains richer active responses under the same non-negative constraint. The attention maps show the same trend: adding the mirrored reorientation to FLatten (Han et al., 2023) reduces diffuse responses and spatial shifts, producing more target-aligned localization. These observations suggest that part of the information loss in ReLU-style linear attention can be simply mitigated by learning the coordinate system presented to the non-negative feature map.

Motivated by these observations, we propose **Householder-adapted Linear Attention (HaLA)**, a linear attention framework that shifts the paradigm from *passive clipping* to *active reorientation*. HaLA adopts a hierarchical design based on learnable Householder reflections to optimize feature topology at *local*, *dynamic*, and *global* levels. At the *single-head* level, HaLA enables precise geometric control of full-rank rotation by decomposing high-dimensional feature space into disjoint 2D subspaces. We then apply strictly isometric Householder reflections within each block to align information-bearing directions into the non-negative orthant without altering pairwise distances. To avoid mirroring features into overly narrow subspace and trigger feature collapse in long-context

modeling, we make the reflection *variance-conditioned*: the reflection angle is coupled to the block-wise signal variance, inducing a dynamic perturbation that yields a theoretical lower bound on activation diversity. This mitigates the dead-zone effect, where clustered tokens in low-variance regimes are indiscriminately truncated by static non-negativity boundaries. To overcome the rank fragmentation inherent in independent multi-head processing (Zhou et al., 2025; Wang et al., 2025), we extend HaLA to the global scale via a unified *cross-head reflection*. This global operator couples feature spaces across heads through covariance mixing, allowing dormant subspaces to inherit discriminative structure from informative heads and improving overall expressivity.

We evaluate our method on five vision benchmarks across nine datasets, covering image classification, detection and segmentation, semantic segmentation, diffusion models, and ultra-long sequence super-resolution (SR). HaLA consistently outperforms baselines, with gains of up to 4.4% on classification, 4.7% on detection and segmentation, and a 6.6 mIoU improvement on semantic segmentation under lower FLOPs. Notably, for ultra-long sequence SR, it achieves higher accuracy while reducing memory by up to 81.4% and latency by 78%, alongside lower FID in diffusion tasks.

2 METHOD

The Information Loss of Linear Attention. Given an input sequence of length N and dimension d, Query ($\mathbf{Q}$), Key ($\mathbf{K}$), and Value ($\mathbf{V}$) matrices are in $\mathbb{R}^{N\times d}$. Standard attention computes pairwise similarities via:

$$\mathbf{o}_t = \sum_{i=1}^{N} \frac{\exp(\mathbf{q}_t \mathbf{k}_i^\top / \sqrt{d})}{\sum_{j=1}^{N} \exp(\mathbf{q}_t \mathbf{k}_j^\top / \sqrt{d})} \mathbf{v}_i, \tag{3}$$

which results in $\mathcal{O}(N^2 d)$ complexity. To alleviate this, Linear Attention (LA) re-formulates the attention mechanism by replacing the softmax-normalized dot product with a generalized kernel function $\mathrm{sim}(\mathbf{q}, \mathbf{k}) = \phi_r(\mathbf{q})\phi_r(\mathbf{k})^\top$. By leveraging the associative property of matrix multiplication, the complexity can be reduced to $\mathcal{O}(Nd^2)$ as

$$\mathbf{o}_t = \frac{\sum_{i=1}^{N} \phi_r(\mathbf{q}_t)\phi_r(\mathbf{k}_i)^\top \mathbf{v}_i}{\sum_{j=1}^{N} \phi_r(\mathbf{q}_t)\phi_r(\mathbf{k}_j)^\top} = \frac{\phi_r(\mathbf{q}_t) \sum_{i=1}^{N} \phi_r(\mathbf{k}_i)^\top \mathbf{v}_i}{\phi_r(\mathbf{q}_t) \sum_{j=1}^{N} \phi_r(\mathbf{k}_j)^\top}. \tag{4}$$

However, a critical constraint in this formulation is that the choice of the kernel feature map $\phi_r(\cdot)$ must ensure *non-negativity* $\mathrm{sim}(\mathbf{q}, \mathbf{k}) \geq 0$. Otherwise, the denominator $\phi(\mathbf{q}) \sum \phi(\mathbf{k_j})^\top$ will approach zero, resulting in numerical instability and gradient explosion during training. To satisfy this for arbitrary data distributions of $\mathbf{Q}$ and $\mathbf{K}$, previous methods typically employ *axis-aligned* mappings such as $\mathrm{ReLU}(\cdot)$ or $\exp(\cdot)$ to constrain the feature maps to the non-negative orthant. Despite their simplicity, such projections act as a hard thresholding operator and indiscriminately *suppress* features in the negative domain (as depicted in Fig.2), leading to irreversible information loss that hampers the model's expressive capacity.

Overview of HaLA. To resolve the conflict between the non-negativity constraint and information preservation, we propose HaLA. Drawing on deep geometric isometries (Mhammedi et al., 2017), HaLA introduces a learnable reflection-based mechanism. Rather than passively *truncating* negative features, we optimally *reorient* the feature geometry into the non-negative orthant via Householder transformations prior to activation so that the QK interactions can be steadily preserved. HaLA comprises three components: (1) *Block-wise Householder isometries* for local feature preservation (Sec. 2.1); (2) *Variance-aware modulation* to counteract oversmoothing in long-context modeling (Sec. 2.1); (3) *Cross-head reflection* to enable inter-head communications and global covariance mixing (Sec. 2.3).

2.1 BLOCK-WISE HOUSEHOLDER ISOMETRIES

Householder Transformation. To preserve feature magnitude while satisfying non-negativity, we employ Householder reflections. The Householder matrix associated with a unit mirror vector $\mathbf{u} \in \mathbb{R}^d$ defines a reflection across the hyperplane orthogonal to $\mathbf{u}$:

$$\mathbf{H_u} = \mathbf{I} - 2\frac{\mathbf{u}\mathbf{u}^\top}{||\mathbf{u}||_2^2}. \tag{5}$$

This transformation is orthogonal ($\mathbf{H_u^\top H_u} = \mathbf{I}$), ensuring it is an isometry that preserves Euclidean norms. Accordingly, the reflection of an input vector $\mathbf{x}$ with respect to this hyperplane is obtained as $\mathbf{H_u x}$. We first show a simplified 2D case to build a geometric interpretation:

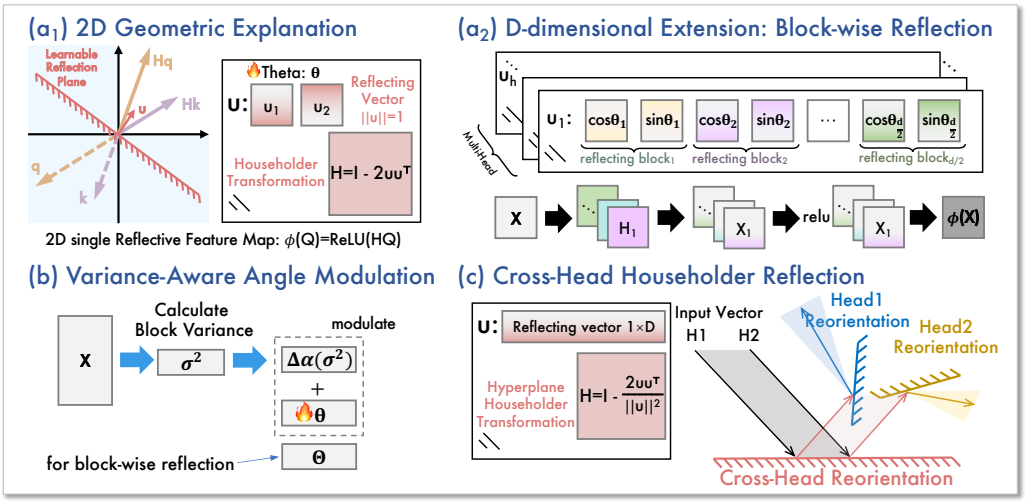

Figure 2: **Overview of the HaLA framework**. (a_1) In 2D, a learnable Householder operator $\mathbf{H}$ reorients features $\mathbf{Hq}$ $\mathbf{Hk}$ before applying an axis-aligned non-negativity map, converting passive truncation into active alignment while preserving inner products by isometry. (a_2) Extension to D-dim via block-wise and reflection with low overhead. (b) Adaptively adjusting reflection angles based on block variance σ^2 to enhance feature diversity. (c) A global transformation before head-wise decomposition to encourage inter-head communication.

2D Case: Intuition of Mirrored Feature Map. Let the mirror vector $\mathbf{u} = [u_1, u_2]^\top \in \mathbb{R}^2$ be a unit mirror vector and the input be $\mathbf{x} \in \mathbb{R}^2$. The corresponding Householder reflection is:

$$\tilde{\mathbf{x}} = \mathbf{H_u}\mathbf{x}, \quad \mathbf{H_u} = \mathbf{I} - 2\mathbf{u}\mathbf{u}^\top, \tag{6}$$

where $\|\mathbf{u}\|_2 = 1$. Parameterizing $\mathbf{u}$ by an angle θ as $\mathbf{u} = [\cos\theta, \sin\theta]$, we obtain the closed form 2D Householder formula as

$$\mathbf{H_u} = \begin{bmatrix} 1 - 2u_1^2 & -2u_1u_2 \\ -2u_1u_2 & 1 - 2u_2^2 \end{bmatrix} = \begin{bmatrix} -\cos 2\theta & -\sin 2\theta \\ -\sin 2\theta & \cos 2\theta \end{bmatrix}. \tag{7}$$

Thus, in two dimensions, the Householder transformation can be interpreted as a reflection across a line with angle θ, providing an intuitive view of how the input is mirrored before further non-negative mapping such as ReLU, which is defined as,

$$\phi_r(\mathbf{x}) = \mathrm{ReLU}(\mathbf{H_u}\mathbf{x}) \tag{8}$$

As shown in Fig. 2 (left), a learnable unit vector $\mathbf{u}$ parameterized by an angle θ allows the model to optimally rotate the mirror geometry relative to the ReLU activation boundary. Instead of passive truncation, the elements are reflected into the active region, preserving structural information that would otherwise be discarded. Consequently, the reflected features retain richer information and display more balanced statistics, which enhances the expressiveness of the feature map and yields more *stable* linear attention estimation.

Extension to High-Dimension: Block-wise Reflection. To scale this to arbitrary dimensions D efficiently, we partition the query or key feature vector $\mathbf{x} \in \mathbb{R}^D$ into $M = D/2$ disjoint two-dimensional blocks: $\mathbf{x} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_M]$, $\mathbf{x}_m \in \mathbb{R}^2$.

For each block m, we associate a learnable mirror vector $\mathbf{u}_m = [\cos\theta_m \ \sin\theta_m]$ and reflect each block independently:

$$\mathbf{H}_m = \mathbf{I}_2 - 2\mathbf{u}_m(\mathbf{u}_m)^\top, \ \tilde{\mathbf{x}}_m = \mathbf{H}_m\mathbf{x}_m, \tag{9}$$

Stacking the reflected blocks gives the full transformed vector, which is followed by ReLU to enforce non-negativity:

$$\phi_r(\mathbf{x}) = \mathrm{ReLU}\left(\tilde{\mathbf{x}}\right), \ \tilde{\mathbf{x}} = \left[\tilde{\mathbf{x}}_1, \ldots, \tilde{\mathbf{x}}_M\right] \in \mathbb{R}^D. \tag{10}$$

Given a query $\mathbf{q}$ and key $\mathbf{k}$, their linear attention kernel is computed in a blockwise *additive* manner. For each head h, we evaluate the similarity by summing contributions from all 2D blocks:

$$\langle \phi_r(\mathbf{q}), \phi_r(\mathbf{k}) \rangle = \sum_{m=1}^{M} \langle \phi_r(\mathbf{q}_m), \phi_r(\mathbf{k}_m) \rangle \,, \tag{11}$$

which is equivalent to the standard inner product over $\mathbb{R}^D$ but explicitly decomposed into 2D reflective components. This design imposes a local mixing bias, reorienting features within small coordinate groups instead of globally entangling all dimensions. It achieves stable geometry adaptation with $\mathcal{O}(d)$ complexity and bounded operator norm inherited from orthogonal reflection.

2.2 Rotating the Mirror: Variance-Aware Angle Modulation

While block-wise reflections optimize static geometry, long-context modeling suffers from *feature oversmoothing*, where token embeddings concentrate in a narrow subspace. If the reflection angle θ_m is static, these aligned tokens may all map to the same sign pattern, causing the subsequent ReLU to zero-out entire sequence segments identically.

To mitigate this, we introduce a *variance-aware angle modulation* mechanism that *adaptively* perturbs the reflection angle based on the block variance σ_m^2 across the sequence length L, encouraging activation-pattern diversity in low-variance subspaces. Let $\mathbf{x}_{t,m} \in \mathbb{R}^2$ denote the feature vector of block m at token t, and let $\bar{\mathbf{x}}_m = \frac{1}{L}\sum_{t=1}^{L} \mathbf{x}_{t,m}$ be the empirical mean. We define the block-level variance as

$$\sigma_m^2 = \frac{1}{2L} \sum_{t=1}^{L} \|\mathbf{x}_{t,m} - \bar{\mathbf{x}}_m\|^2. \tag{12}$$

Intuitively, a small σ_m^2 indicates that token representations within this block carry weak token-specific information, and thus benefit from a stronger reorientation relative to the activation boundary. We accordingly modulate the reflection angle by:

$$\Theta_m = \theta_m + \Delta\alpha(\sigma^2), \quad \Delta\alpha(\sigma^2) = \text{sigmoid}\left(\frac{\lambda}{\sigma_m^2 + \varepsilon}\right) \cdot \alpha_{\max}, \tag{13}$$

where θ_m denotes the learnable base reflection angle for block m, $\alpha_{\max} \in \{\pi/8, \pi/4, \pi/2\}$ is a scaling hyperparameter controlling the maximum perturbation, λ controls the variant sensitivity, and ε is a small constant for numerical stability. We then replace θ_m with the variance-aware angle Θ_m in the Householder reflection, while keeping the per-block computation linear in D.

Theoretical Motivation. The following theorem formalizes why activation diversity and retained active energy are governed by the boundary proximity between the reflected token cluster and the ReLU gate. Importantly, the modulation does not create token-specific variation from collapsed features; instead, it changes the relative position between the token cluster and the activation boundary, reducing the chance that an entire low-variance block is mapped to the same inactive pattern.

Theorem 1 (Collapse-aware Activation Diversification). *Let $\mathcal{X} = \{\mathbf{x}_{t,m}\}_{t=1}^{L} \subset \mathbb{R}^2$ be the token features of a 2D block with empirical variance σ^2 computed over the token dimension. Consider the modulated mirror map*

$$\phi(x; \Theta) \;=\; \text{ReLU}\big(\mathbf{H}_m(\Theta_m)\mathbf{x}_{t,m}\big), \tag{14}$$

where $\Delta\alpha(\sigma^2) = \text{sigmoid}(1/(\sigma^2+\varepsilon)) \cdot \alpha$ and $\mathbf{H}(\Theta_m)$ denotes a 2D Householder reflection parameterized by angle Θ_m (Eq. 13). Define the binary activation mask:

$$s(\mathbf{x}_{t,m}; \Theta_m) \;=\; \mathbb{I}[\mathbf{H}(\Theta_m)\mathbf{x}_{t,m} > \mathbf{0}] \in \{0,1\}^2. \tag{15}$$

Then, in the low-variance regime $\sigma^2 \to 0$, the modulation saturates $\Theta_m \to \theta + \alpha_{\max}$ and induces a non-degenerate reorientation of the reflection hyperplane. In particular, for perturbative tokens $\mathbf{x}_{t,m} = \bar{\mathbf{x}}_m + \delta_t$, the expected mask disagreement satisfies

$$\mathbb{E}_{t \neq t'}[\|s(\mathbf{x}_{t,m}; \Theta) - s(\mathbf{x}_{t',m}; \Theta)\|_0] \geq c(\bar{\mathbf{x}}_m, \Theta) \cdot \mathbb{E}_{t \neq t'}[\|\mathbf{x}_{t,m} - \mathbf{x}_{t',m}\|_2] \,,$$

for some $c(\bar{\mathbf{x}}, \Theta_m) > 0$ whenever $\mathbf{H}(\Theta_m)\bar{\mathbf{x}}$ lies within a bounded margin of the ReLU boundary.

Proof sketch. As $\sigma^2 \to 0$, we have $1/(\sigma^2+\varepsilon) \to \infty$, hence $\Delta\alpha(\sigma^2)$ saturates near $\alpha_{\max}$, yielding the maximal angular shift. Mask disagreement arises when tokens fall on different sides of the coordinate

hyperplanes after reflection, which is controlled by the distance (margin) of $\mathbf{H}(\Theta_m)\bar{\mathbf{x}}$ to the ReLU boundary. By rotating the reflection hyperplane in the low-variance regime, the proposed modulation prevents an entire oversmoothed token cluster from sharing an identical activation pattern, thereby mitigating collapse under long-context accumulation. □

2.3 Cross-Head Householder Reflection

While multi-head factorization improves expressivity, purely head-wise processing implicitly limits interactions across heads and can lead to fragmented representations, especially under non-negative feature maps. Recent studies on *inter-head communication* (Zhou et al., 2025; Wang et al., 2025) suggest that lightweight transformations along the head dimension H can substantially enhance attention capacity by allowing heads to exchange complementary information. Motivated by this perspective, we introduce a global cross-head Householder reflection that couples the full feature space *before* head-wise decomposition, enabling global correlation mixing with stable conditioning.

Formally, let $\mathbf{X} \in \mathbb{R}^{L\times HD}$ denote the concatenated multi-head features, where H is the number of heads and D is the per-head dimension. We apply a global Householder reflection

$$\phi_c(\mathbf{X}) = \mathbf{X}\mathbf{H}_c, \quad \mathbf{H}_c = \mathbf{I}_{HD} - 2\frac{\mathbf{u}_c\mathbf{u}_c^\top}{\|\mathbf{u}_c\|_2^2}, \tag{16}$$

where $\mathbf{u}_c \in \mathbb{R}^{HD}$. Importantly, the transformation preserves the Euclidean geometry while introducing controlled cross-head mixing.

Covariance Mixing. Let $\Sigma = \mathrm{Cov}(\mathbf{X}) \in \mathbb{R}^{HD\times HD}$ denote the covariance of the concatenated head features at a token. Since $\mathbf{H}_c$ is orthogonal, the reflected covariance satisfies,

$$\Sigma' \;=\; \mathrm{Cov}(x\mathbf{H}_c) \;=\; \mathbf{H}_c^\top \Sigma \mathbf{H}_c, \tag{17}$$

which preserves the covariance spectrum but generally redistributes correlation mass across head blocks. As a result, $\phi_c(\cdot)$ mitigates the block-diagonal bias induced by independent head processing, allowing low-variance ("dead") heads to recover discriminative directions from high-variance heads prior to the subsequent head-wise non-negative mapping. After the cross-head reflection, $\phi_c(\mathbf{X})$ is reshaped back into H heads and processed with the variance-aware block-wise Householder reflections from §2.2, followed by ReLU per head. The complete HaLA pipeline is summarized in Algorithm A.3.

3 Experiments

In this section, we conduct extensive experiments to evaluate the effectiveness of HaLA including image classification, object detection and instance segmentation, diffusion transformer and high-resolution dense prediction tasks such as semantic segmentation and super-resolution. All experiments are implemented on workstations equipped with NVIDIA H200, RTX 3090, and A6000 GPUs.

3.1 Image Classification on ImageNet-1K

Settings. On ImageNet-1K (Deng et al., 2009), we evaluate HaLA by training from scratch and reporting Top-1 accuracy. To provide a comprehensive evaluation, we compare our method against various State-of-the-Art (SOTA) efficient vision models. For a fair and structured analysis, these baselines are categorized into several tiers based on parameter count, allowing for direct performance comparisons within similar model scales.

Results. As shown in Tab. 1, the results show that HaLA achieves strong, scalable performance across model sizes. HaLA-T/S/B/L reach 82.8%, 84.2%, 85.3%, and 85.7% Top-1 accuracy, surpassing prior methods (e.g., RAVLT and RMT) by up to 0.4%. The gains remain stable at larger scales, with HaLA-L outperforming both RMT-L and RAVLT-L. The complete results are provided in Tab. 23 of the Appendix.

3.2 Object Detection and Instance Segmentation

Settings. We evaluate HaLA on COCO (Lin et al., 2014) dataset for object detection and instance segmentation. By integrating HaLA into RetinaNet (Lin et al., 2017) and Mask R-CNN (He et al.,

Table 1: Comparison of the ImageNet-1K classification with the SOTA efficient vision models. The P" column denotes the number of model parameters, the F" column represents the computational amount, and the "ACC" (%) column indicates the top-1 accuracy.

MODEL	P	F	ACC
C2Former-N(Hou et al., 2024)(TPAMI24)	15M	2.2G	81.5
RMT-T(Fan et al., 2024)(CVPR24)	14M	2.5G	82.4
Agent-PVT-T(Han et al., 2024c) (ECCV24)	12M	2.0G	78.4
EfficientViT-B2(Cai et al., 2023)(ICCV23)	24M	1.6G	82.1
RAVLT-T(Fan et al., 2025b)(CVPR25)	15M	2.6G	82.6
NaLaFormer-T(Meng et al., 2025a)(ICML26)	16M	2.7G	82.6
HaLA-T	15M	2.6G	**82.8**
C2Former-T(Hou et al., 2024)(TPAMI24)	27M	4.4G	83.2
MambaOut-T(Yu & Wang, 2025)(CVPR25)	27M	4.5G	82.7
VMamba-T(Liu et al., 2024)(NeurIPS24)	30M	4.9G	82.6
InternImage-T(Wang et al., 2023)(CVPR23)	30M	5.0G	83.5
FL-Swin-S(Han et al., 2023)(ICCV23)	51M	8.7G	83.5
Agent-Swin-T(Han et al., 2024c)(ECCV24)	29M	4.5G	82.6
Vim-S(Zhu et al., 2024)(ICML24)	26M	3.7G	80.6
LVMamba-T(Huang et al., 2024)(ECCV24)	26M	5.7G	82.7
MOAT-0(Yang et al., 2023)(ICLR23)	28M	5.7G	83.3
RMT-S(Fan et al., 2024)(CVPR24)	27M	4.5G	84.1
Pola-Swin-T(Meng et al., 2025b)(ICLR25)	29M	4.5G	82.6
ViG-H-T(Liao et al., 2025)(AAAI25)	29M	4.5G	82.8
MILA-T(Han et al., 2024b)(NeurIPS24)	25M	4.2G	83.5
HaLA-S	26M	4.9G	**84.2**

MODEL	P	F	ACC
MambaOut-S(Yu & Wang, 2025)(CVPR25)	49M	9.0G	84.1
MogaNet-B (Li et al., 2024) (ICLR24)	44M	10G	84.3
VMamba-S(Liu et al., 2024)(NeurIPS24)	50M	8.7G	83.6
SOFT-L (Lu et al., 2024) (IJCV24)	64M	11G	83.1
Agent-Swin-S(Han et al., 2024c)(ECCV24)	50M	8.7G	83.7
Pola-Swin-S(Meng et al., 2025b)(ICLR25)	50M	8.7G	83.6
ViG-H-S(Liao et al., 2025)(AAAI25)	50M	8.8G	83.8
RMT-B(Fan et al., 2024)(CVPR24)	54M	9.7G	85.0
MILA-S(Han et al., 2024b)(NeurIPS24)	43M	7.3G	84.4
RAVLT-B(Fan et al., 2025b)(CVPR25)	48M	10.6G	85.1
HaLA-B	48M	10.6G	**85.3**
MambaOut-B(Yu & Wang, 2025)(CVPR25)	85M	15.8G	84.2
VMamba-B(Liu et al., 2024)(NeurIPS24)	89M	15.4G	83.9
FL-Swin-B(Han et al., 2023)(ICCV23)	89M	15.4G	83.8
VRWKV-B(Duan et al., 2025)(ICLR25)	94M	18.2G	82.0
ViG-H-B(Liao et al., 2025)(AAAI25)	89M	15.5G	84.2
GC-ViT-B(Hatamizadeh et al., 2023)(ICML23)	90M	14.8G	85.0
RMT-L(Fan et al., 2024)(CVPR24)	95M	18.2G	85.5
InLine-Swin-B(Han et al., 2024a)(NeurIPS24)	88M	15.4G	82.0
MILA-B(Han et al., 2024b)(NeurIPS24)	96M	16.2G	85.3
RAVLT-L(Fan et al., 2025b)(CVPR25)	95M	17.3G	85.5
HaLA-L	95M	17.3G	**85.7**

2017) under both 1× and 3× training schedules. For fair comparison, we adopt the evaluation protocol of FL-Transformer (Han et al., 2023).

Results. As shown in Tab. 2, HaLA consistently improves detection and instance segmentation performance under comparable computational budgets. Specifically, HaLA yields up to 1.8 AP^b gains for RetinaNet, while for Mask R-CNN it improves AP^b from 47.2 to 49.9 under the 1× schedule and from 50.7 to 51.3 under the 3× schedule. Detailed results for RetinaNet 1×, Mask R-CNN 1×, and Mask R-CNN 3×, evaluating HaLA across all four model variants, are provided in Tabs. 24, 25, and 26 of the Appendix.

Table 2: **Object detection and instance segmentation results on the COCO dataset.** The "P" column represents the number of parameters, and "F" for FLOPs.

MODEL	P (M)	F (G)	RETINANET 1× AP^b	AP^b_{50}	AP^b_{75}	AP^b_S	AP^b_M	AP^b_L
PVT-T(Wang et al., 2021)	23	221	39.4	59.8	42.0	25.5	42.0	52.1
MPViT-XS(Lee et al., 2022)	20	211	43.8	65.0	47.1	28.1	47.6	56.5
SOFT++ T(Lu et al., 2024)	23	200	41.9	62.7	44.7	27.8	45.4	55.6
RMT-T(Fan et al., 2024)	23	199	45.1	66.2	48.1	28.8	48.9	61.1
RAVLT-T(Fan et al., 2025b)	24	201	44.1	64.3	47.4	26.4	48.1	59.3
HaLA-T	24	200	**46.1**	**67.4**	**49.4**	**28.8**	**50.4**	**61.7**
Swin-T(Liu et al., 2021)	38	248	41.7	63.1	44.3	27.0	45.3	54.7
MPViT-S(Lee et al., 2022)	32	248	45.7	57.3	48.8	28.7	49.7	59.2
PVTv2-B2(Wang et al., 2022)	35	290	44.6	65.6	47.6	27.4	48.8	58.6
CMT-S(Guo et al., 2022a)	44	231	44.3	65.5	47.5	27.1	48.3	59.1
RAVLT-S(Fan et al., 2025b)	44	262	46.7	67.2	50.4	31.4	51.1	62.3
HaLA-S	34	243	**48.5**	**69.8**	**52.2**	**32.2**	**52.9**	**63.7**
PVT-L(Wang et al., 2021)	71	345	42.6	63.7	45.4	25.8	46.0	58.4
SOFT++ M(Lu et al., 2024)	59	322	44.3	64.7	47.4	29.0	48.2	59.9
PVTv2-B3(Wang et al., 2022)	55	-	45.9	66.8	49.3	28.6	49.8	61.4
RAVLT-B(Fan et al., 2025b)	57	353	49.8	71.2	54.0	34.0	54.3	64.9
MAViT-B(Fan et al., 2025a)	57	353	49.9	71.1	53.8	33.7	54.5	**65.5**
HaLA-B	57	352	**50.2**	**71.7**	**54.3**	**34.0**	**54.8**	65.4

MODEL	P (M)	F (G)	MASK R-CNN 1× AP^b	AP^b_{50}	AP^b_{75}	AP^m	AP^m_{50}	AP^m_{75}
MPViT-XS(Lee et al., 2022)	30	231	47.3	69.1	51.9	42.7	66.2	46.0
CA-Swin-T(Han et al., 2026)	-	269	44.5	67.3	48.9	40.5	64.2	43.5
SMamba-T(Xiao et al., 2025)	46	261	47.6	69.6	52.3	42.9	66.5	46.2
PVMamba(Xie et al., 2025)	44	266	47.1	69.3	51.9	42.6	66.5	45.7
MAViT-T(Fan et al., 2025a)	33	226	47.5	69.0	52.3	42.8	66.3	46.3
MambaTreeV-T(Xiao et al., 2024)	-	265	47.0	69.4	51.5	42.7	66.4	46.0
MSVMamba-T(Shi et al., 2024b)	53	252	46.9	68.8	51.4	42.2	65.6	45.4
VSSD-T(Shi et al., 2024a)	44	265	46.9	69.4	51.4	42.6	66.4	45.9
HaLA-T	33	219	**47.7**	**69.6**	**52.6**	**43.0**	**66.6**	**46.2**
MambaOut-T(Yu & Wang, 2025)	43	262	45.1	67.3	49.6	41.0	64.1	44.1
MPViT-S(Lee et al., 2022)	43	268	46.4	68.6	51.2	42.4	65.6	45.7
CMT-S(Guo et al., 2022a)	45	249	44.6	66.8	48.9	40.7	63.9	43.4
MILA-T(Han et al., 2024b)	44	262	46.8	69.5	51.5	42.1	66.4	45.0
HaLA-S	44	262	**49.9**	**71.6**	**54.7**	**44.5**	**68.4**	**48.1**
InternImage-S(Wang et al., 2023)	69	340	47.8	69.8	52.8	43.3	67.1	46.7
STViT-B(Huang et al., 2022)	70	359	49.7	71.7	54.7	44.8	68.9	48.7
SOFT++ M(Lu et al., 2024)	69	342	46.6	67.8	51.2	42.0	64.8	45.2
MILA-S(Han et al., 2024b)	63	319	49.2	71.5	53.9	44.2	68.5	47.2
RAVLT-B(Fan et al., 2025b)	67	372	51.2	72.7	56.4	45.7	69.9	49.5
HaLA-B	67	371	**51.5**	**73.2**	**56.6**	**45.8**	**70.3**	**49.7**

3.3 SEMANTIC SEGMENTATION

Settings. We integrate HaLA into semantic segmentation on ADE20K (Zhou et al., 2019) and Cityscapes (Cordts et al., 2016), using mIoU as the evaluation metric. We initialize the backbone with ImageNet-1K pretrained weights and train the segmentation models using the *mmcv-segmentation* framework (Contributors, 2018), following common practice in recent works (Guo et al., 2022b; Cai et al., 2023). Additionally, we adopt an FPN decoder with other linear attention methods in Appendix Tab. 13.

Results. HaLA consistently improves mIoU under similar computational budgets for semantic segmentation. As shown in Tab. 3, HaLA delivers stronger accuracy while maintaining competitive complexity. Specifically on ADE20K, HaLA-SN-S and HaLA-SM-S achieve 50.9 and 51.5 mIoU, respectively, significantly outperforming EfficientViT-B3 (49.0) and EfficientViT-L1 (49.2) under a smaller parameter budget. Visually, as illustrated in Fig. 3 (left), HaLA produces noticeably cleaner and more precise semantic boundaries. It better separates adjacent objects, accurately captures small details such as traffic signs, and significantly reduces misclassified or missing regions, leading to an improved overall segmentation quality. For more detailed experimental results on the Cityscapes dataset, along with further qualitative visualizations, are provided in Appendix A.8.

Table 3: **Semantic segmentation results** compared with SOTA methods on ADE20K with both SegMAN and SegNeXt. FLOPs (F) is calculated with input size 512×512 for ADE20K.

METHOD	PARA	ADE20K MIOU	ADE20K F
VRWKV-S(Duan et al., 2025)	29M	47.2	46G
MambaOut-T(Yu & Wang, 2025)	54M	47.4	-
VWFormer-B2(Yan et al., 2024)	27M	48.1	47G
EfficientViT-B3(Cai et al., 2023)	40M	49.0	22G
EfficientViT-L1(Cai et al., 2023)	40M	49.2	36G
SegFormer-B2(Xie et al., 2021)	28M	46.5	62G
SegNeXt-B(Guo et al., 2022b)	28M	48.5	35G
HaLA-SN-S	24M	**50.9**	25G
SegMan-S(Fu et al., 2025)	29M	51.3	25G
HaLA-SM-S	29M	**51.5**	27G
MambaOut-S(Yu & Wang, 2025)	76M	49.5	-
VWFormer-B4(Yan et al., 2024)	64M	50.8	80G
VMamba-T(Liu et al., 2024)	62M	47.9	-
EfficientViT-L2(Cai et al., 2023)	51M	50.7	45G
SegFormer-B3(Xie et al., 2021)	47M	49.4	45G
SegFormer-B4(Xie et al., 2021)	64M	50.3	96G
SegNeXt-L(Guo et al., 2022b)	49M	51.0	70G
HaLA-SN-B	48M	**52.6**	57G
SegMAN-B(Fu et al., 2025)	52M	52.6	58G
HaLA-SM-B	55M	**53.0**	57G
ViT-B(Dosovitskiy et al., 2021)	100M	48.8	158G
VRWKV-B(Duan et al., 2025)	107M	49.2	146G
MambaOut-B(Yu & Wang, 2025)	112M	49.6	-
VMamba-B(Liu et al., 2024)	122M	51.0	-
Swin-B(Liu et al., 2021)	121M	48.1	261G
VWFormer-B5(Yan et al., 2024)	85M	52.0	96G
SegFormer-B5(Xie et al., 2021)	85M	51.0	110G
HaLA-SN-L	98M	**53.1**	103G
SegMAN-L(Fu et al., 2025)	92M	53.2	97G
HaLA-SM-L	107M	**53.4**	93G

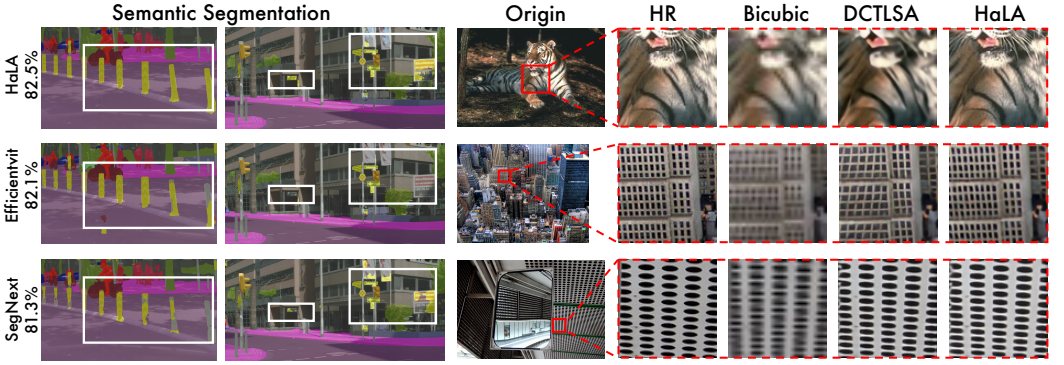

Figure 3: **Visualization of Semantic Segmentation and Super-Resolution tasks.** Left: Comparative results on the Cityscapes dataset, where HaLA achieves superior segmentation integrity, whereas competing methods suffer from incomplete masks. Right: Comparison of SR performance. While DCTLSA introduces noticeable structural distortions, HaLA reconstructs better details.

3.4 SUPER RESOLUTION

Settings. We evaluate HaLA on lightweight single-image super-resolution by integrating it into an efficient SR baseline, DCTLSA (Zeng et al., 2023). Following common practice, we consider $\times 3$ and $\times 4$ upscaling and report PSNR/SSIM on standard SR benchmarks. We additionally measure inference latency on an RTX 3090 to assess efficiency.

Results. Tab. 4 shows that integrating HaLA into DCTLSA consistently improves PSNR/SSIM for both $\times 3$ and $\times 4$ upscaling. For example, at $\times 4$, Set5 improves from 32.44/0.8973 to 32.53/0.8983, and Urban100 from 26.41/0.7944 to 26.65/0.8007. In addition, DCTLSA-HaLA significantly reduces latency and memory: on Manga109 $\times 4$, latency is reduced by over 75% and memory usage is cut by over 80%. As visualized in Fig. 3 (right), HaLA produces super-resolved images closer to the ground truth, with clear details and better structural preservation.

3.5 DIFFUSION TRANSFORMER

Settings. Diffusion transformers provide a suitable testbed for evaluating linear attention. Following DiT (Peebles & Xie, 2023) and SiT (Ma et al., 2024), we replace the attention module with HaLA and conduct experiments on ImageNet-1K (Deng et al., 2009) under the same training and evaluation protocols as the corresponding baselines.

Results. Tab. 5 reports that integrating HaLA consistently improves diffusion transformers across both DiT and SiT architectures. On the DiT branch, HaLADiT yields a sizable improvement over DiG

(Zhu et al., 2025). Specifically, the FID decreases from 62.06 to 58.07 and the sFID decreases from 11.77 to 11.21, while the IS increases from 22.81 to 24.70. On the SiT branch, HaLASiT achieves the best overall performance. It attains lower FID and sFID values of 52.29 and 8.41, respectively, together with a higher IS of 28.51, and demonstrates strong precision and recall of 0.43 and 0.61. This performance surpasses both SiT (Ma et al., 2024) and EfficientSiT (Pu et al., 2024).

Table 4: **Quantitative comparisons between HaLA and other lightweight image super-resolution methods.** MEM. denotes GPU memory consumption, and LAT. denotes latency.

MODEL	SCALE	SET5		SET14		BSDS100		URBAN100		MANGA109	
		PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Bicubic	×4	28.42	0.8104	26.00	0.7027	25.96	0.6675	23.14	0.6577	24.89	0.7866
SwinIR(Liang et al., 2021)	×4	32.44	0.8976	28.77	0.7858	27.69	0.7406	26.47	0.7980	30.92	0.9151
DCTLSA(Zeng et al., 2023)	×4	32.44	0.8973	28.77	0.7846	27.67	0.7386	26.41	0.7944	31.04	0.9138
DCTLSA-HaLA	×4	**32.53**	**0.8983**	**28.84**	**0.7867**	**27.74**	**0.7409**	**26.65**	**0.8007**	**31.30**	**0.9170**
Bicubic	×3	30.39	0.8682	27.55	0.7742	27.21	0.7385	24.46	0.7349	26.95	0.8556
SwinIR(Liang et al., 2021)	×3	34.62	0.9289	30.54	0.8463	29.20	0.8082	28.66	0.8624	33.98	0.9478
DCTLSA(Zeng et al., 2023)	×3	34.76	0.9296	30.64	0.8474	29.27	0.8093	28.81	0.8674	34.42	0.9492
DCTLSA-HaLA	×3	**34.84**	**0.9301**	**30.70**	**0.8488**	**29.33**	**0.8106**	**29.03**	**0.8688**	**34.61**	**0.9505**
EFFICIENCY@3090	SCALE	LAT.	MEM.	LAT.	MEM.	LAT.	MEM.	LAT.	MEM.	LAT.	MEM.
DCTLSA-Softmax	×4	57.70ms	1.05GB	109.04ms	1.47GB	72.06ms	0.59GB	349.86ms	4.28GB	456.71ms	3.44GB
DCTLSA-HaLA	×4	55.28ms	0.18GB	55.29ms	0.30GB	49.62ms	0.11GB	79.03ms	0.85GB	100.69ms	0.67GB
- SAVE	×4	**4.2%**	**82.9%**	**49.3%**	**79.6%**	**31.1%**	**81.4%**	**77.4%**	**80.1%**	**78.0%**	**80.5%**
DCTLSA-Softmax	×3	57.90ms	1.66GB	113.04ms	2.27GB	76.74ms	0.93GB	355.81ms	7.19GB	571.86ms	5.49GB
DCTLSA-HaLA	×3	46.11ms	0.33GB	51.82ms	0.51GB	42.70ms	0.19GB	146.27ms	1.52GB	183.47ms	1.20GB
- SAVE	×3	**20.4%**	**80.1%**	**54.2%**	**77.5%**	**44.3%**	**79.6%**	**58.9%**	**78.9%**	**67.9%**	**78.1%**

Table 5: **Diffusion transformer results.**

MODEL	FID ↓	SFID ↓	IS ↑	PREC ↑	RECALL ↑
DiT(Peebles & Xie, 2023)	68.40	-	-	-	-
Linear Attn	89.72	21.87	15.24	0.28	0.41
DiG(Zhu et al., 2025)	62.06	11.77	22.81	0.39	0.56
MHLA(Zhang et al., 2026)	59.80	**10.16**	23.49	0.39	0.56
NaLaDiT(Meng et al., 2025a)	61.64	12.40	23.24	0.40	0.58
HaLADiT	**58.07**	11.21	**24.70**	**0.41**	**0.59**
SiT (Ma et al., 2024)	58.61	9.25	24.31	0.41	0.59
EfficientSiT(Pu et al., 2024)	53.57	9.01	27.26	0.43	0.61
HaLASiT	**52.29**	**8.41**	**28.51**	**0.43**	**0.61**

Table 6: **Comparison with LA models.**

SWIN	ACC.	PVT	ACC.
Swin-T(Liu et al., 2021)	81.2	PVT(Wang et al., 2021)	75.1
Efficient (Shen et al., 2021)	81.0	Agent(Han et al., 2024c)	78.4
DiffLA(Pu et al., 2025a)	82.3	InLine(Han et al., 2024a)	78.2
FLatten(Han et al., 2023)	82.1	DiffLA(Pu et al., 2025a)	78.2
Agent(Han et al., 2024c)	82.6	CA(Han et al., 2026)	78.1
InLine(Han et al., 2024a)	82.4	Pola(Meng et al., 2025b)	78.8
Pola(Meng et al., 2025b)	82.6	FLatten(Han et al., 2023)	77.8
LA-ReLU	81.8	HaLA-ReLU	**78.6**
HaLA-ReLU	**82.4**	HaLA-FLatten	**79.2**

3.6 ABLATION STUDY

Comparison with other Linear Attentions. We evaluate HaLA on the Swin (Liu et al., 2021) and PVT (Wang et al., 2021) architectures for image classification, following the protocol of FLatten-Transformer (Han et al., 2023), and further assess it on the LRA (Tay et al., 2021) benchmark for long-sequence tasks. Compared with existing linear attention mechanisms, our method demonstrates superior performance in image classification (Tab. 6) and long-sequence modeling (Tab. 14).

Hyperparameter Analysis. On HaLA-XT, removing cross-head or block-wise reflection lowers Top-1 accuracy by 0.5 or 0.3 percentage points, respectively (Tab. 8). Appendix A.4 provides the full hyperparameter study and additional ablations: it isolates variance-aware modulation, compares Householder reflection with alternative cross-head operators, extends the component study to HaLA-T and DiT, and reports seed stability and a $1 + \mathrm{ELU}$ feature-map control.

Further details are in the Appendix: limitations (A.1), related work (A.2), the algorithm (A.3), ablations (A.4), additional benchmarks (A.5), mechanism analysis (A.6), efficiency analysis (A.7), visualizations (A.8), experimental settings (A.9), and full quantitative tables (A.10).

4 CONCLUSION

In this work, we introduce **HaLA**, a linear attention framework that replaces passive non-negativity clipping with active geometric reorientation via learnable Householder reflections. At the single-head level, we generalize 2D reflections to high-dimensional feature spaces through block-wise decomposition, and further propose Variance-Aware Angle Modulation to enable data-dependent mirror transformations. At the global scale, we design a unified cross-head reflection to enhance inter-head communication. Extensive experiments demonstrate that HaLA consistently achieves superior performance while substantially reducing memory footprint and time.

AI USE STATEMENT

Generative AI tools were used to improve the grammar, clarity, and presentation of the manuscript and to assist in drafting and organizing descriptions of author-provided experimental results. They were not used to generate the research ideas, design the methodology, or produce experimental data. The authors reviewed and revised the AI-assisted text and take full responsibility for the final content of this work.

ETHICS STATEMENT

This work studies an efficient attention mechanism for computer vision. It does not involve a new human-subject study or the collection of new personal data. The experiments use established research datasets and benchmarks for image classification, object detection, segmentation, super-resolution, and image generation. Like other vision models, the resulting systems may reflect biases or privacy limitations of their underlying datasets; applications involving people or consequential decisions require additional, context-specific evaluation.

REPRODUCIBILITY STATEMENT

The method formulation, complexity analysis, and theoretical motivation are described in Sections 2.1–2.3, with pseudocode in Appendix A.3. Model configurations, datasets, and training and evaluation settings are documented in Appendix A.9. Additional ablations, mechanism and efficiency analyses, and complete quantitative results appear in Appendices A.4, A.6, A.7, and A.10. Source code is included in the supplementary materials to facilitate reproduction.

REFERENCES

Eirikur Agustsson and Radu Timofte. NTIRE 2017 challenge on single image super-resolution: Dataset and study. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1122–1131, 2017.

Alaaeldin Ali, Hugo Touvron, Mathilde Caron, Piotr Bojanowski, Matthijs Douze, Armand Joulin, Ivan Laptev, Natalia Neverova, Gabriel Synnaeve, Jakob Verbeek, and Hervé Jégou. Xcit: Cross-covariance image transformers. In *Proc. Annual Conference on Neural Information Processing Systems (NeurIPS)*, pp. 20014–20027, 2021.

Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, and Judy Hoffman. Hydra attention: Efficient attention with many heads. In *Proc. European Conference on Computer Vision (ECCV)*, pp. 35–49, 2022.

Han Cai, Junyan Li, Muyan Hu, Chuang Gan, and Song Han. Efficientvit: Lightweight multi-scale attention for high-resolution dense prediction. In *Proc. International Conference on Computer Vision (ICCV)*, pp. 17256–17267, 2023.

Yuan Cao and Dong Wang. SAGA: selective adaptive gating for efficient and expressive linear attention. *CoRR*, abs/2509.12817, 2025.

Krzysztof Marcin Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamás Sarlós, Peter Hawkins, Jared Quincy Davis, Afroz Mohiuddin, Lukasz Kaiser, David Benjamin Belanger, Lucy J. Colwell, and Adrian Weller. Rethinking attention with performers. In *Proc. International Conference on Learning Representations (ICLR)*, 2021.

MMCV Contributors. MMCV: OpenMMLab computer vision foundation. `https://github.com/open-mmlab/mmcv`, 2018.

Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3213–3223, 2016.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 248–255, 2009.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *Proc. International Conference on Learning Representations (ICLR)*, 2021.

Yuchen Duan, Weiyun Wang, Zhe Chen, Xizhou Zhu, Lewei Lu, Tong Lu, Yu Qiao, Hongsheng Li, Jifeng Dai, and Wenhai Wang. Vision-rwkv: Efficient and scalable visual perception with rwkv-like architectures. In *Proc. International Conference on Learning Representations (ICLR)*, 2025.

Qihang Fan, Huaibo Huang, Mingrui Chen, Hongmin Liu, and Ran He. RMT: retentive networks meet vision transformers. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5641–5651, 2024.

Qihang Fan, Huaibo Huang, Yuang Ai, and Ran He. Rectifying magnitude neglect in linear attention. *CoRR*, abs/2507.00698, 2025a.

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

Yunxiang Fu, Meng Lou, and Yizhou Yu. Segman: Omni-scale context modeling with state space models and local attention for semantic segmentation. In *Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR)*, pp. 19077–19087, 2025.

Jianyuan Guo, Kai Han, Han Wu, Yehui Tang, Xinghao Chen, Yunhe Wang, and Chang Xu. CMT: convolutional neural networks meet vision transformers. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12165–12175, 2022a.

Meng-Hao Guo, Cheng-Ze Lu, Qibin Hou, Zhengning Liu, Ming-Ming Cheng, and Shi-Min Hu. Segnext: Rethinking convolutional attention design for semantic segmentation. In *Proc. Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2022b.

Meng-Hao Guo, Chengze Lu, Zheng-Ning Liu, Ming-Ming Cheng, and Shimin Hu. Visual attention network. *CoRR*, abs/2202.09741, 2022c.

Dongchen Han, Xuran Pan, Yizeng Han, Shiji Song, and Gao Huang. Flatten transformer: Vision transformer using focused linear attention. In *Proc. International Conference on Computer Vision (ICCV)*, pp. 5938–5948, 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. In *Proc. Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2024a.

Dongchen Han, Ziyi Wang, Zhuofan Xia, Yizeng Han, Yifan Pu, Chunjiang Ge, Jun Song, Shiji Song, Bo Zheng, and Gao Huang. Demystify mamba in vision: A linear attention perspective. In *Proc. Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2024b.

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 *Proc. European Conference on Computer Vision (ECCV)*, pp. 124–140, 2024c.

Dongchen Han, Tianyu Li, Ziyi Wang, and Gao Huang. Vision transformers are circulant attention learners. In *Proc. Association for the Advancement of Artificial Intelligence (AAAI)*, pp. 21549–21557, 2026.

Ali Hatamizadeh, Hongxu Yin, Greg Heinrich, Jan Kautz, and Pavlo Molchanov. Global context vision transformers. In *Proc. International Conference on Machine Learning (ICML)*, pp. 12633–12646, 2023.

Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. Mask R-CNN. In *Proc. International Conference on Computer Vision (ICCV)*, pp. 2980–2988, 2017.

Qibin Hou, Cheng-Ze Lu, Ming-Ming Cheng, and Jiashi Feng. Conv2former: A simple transformer-style convnet for visual recognition. *Transactions on Pattern Analysis and Machine Intelligence*, 46(12):8274–8283, 2024.

Huaibo Huang, Xiaoqiang Zhou, Jie Cao, Ran He, and Tieniu Tan. Vision transformer with super token sampling. *CoRR*, abs/2211.11167, 2022.

Tao Huang, Xiaohuan Pei, Shan You, Fei Wang, Chen Qian, and Chang Xu. Localmamba: Visual state space model with windowed selective scan. In *Proc. European Conference on Computer Vision Workshop (ECCV Workshop)*, pp. 12–22, 2024.

Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are rnns: Fast autoregressive transformers with linear attention. In *Proc. International Conference on Machine Learning (ICML)*, pp. 5156–5165, 2020.

Manjin Kim, Paul Hongsuck Seo, Cordelia Schmid, and Minsu Cho. Learning correlation structures for vision transformers. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18941–18951, 2024.

Youngwan Lee, Jonghee Kim, Jeffrey Willette, and Sung Ju Hwang. Mpvit: Multi-path vision transformer for dense prediction. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7277–7286, 2022.

Siyuan Li, Zedong Wang, Zicheng Liu, Cheng Tan, Haitao Lin, Di Wu, Zhiyuan Chen, Jiangbin Zheng, and Stan Z. Li. Moganet: Multi-order gated aggregation network. In *Proc. International Conference on Learning Representations (ICLR)*, 2024.

Jingyun Liang, Jiezhang Cao, Guolei Sun, Kai Zhang, Luc Van Gool, and Radu Timofte. Swinir: Image restoration using swin transformer. In *Proc. International Conference on Computer Vision (ICCV)*, pp. 1833–1844, 2021.

Bencheng Liao, Xinggang Wang, Lianghui Zhu, Qian Zhang, and Chang Huang. Vig: Linear-complexity visual sequence learning with gated linear attention. In *Proc. Association for the Advancement of Artificial Intelligence (AAAI)*, pp. 5182–5190, 2025.

Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. In *Proc. European Conference on Computer Vision (ECCV)*, pp. 740–755, 2014.

Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proc. International Conference on Computer Vision (ICCV)*, pp. 2999–3007, 2017.

Weifeng Lin, Ziheng Wu, Jiayu Chen, Jun Huang, and Lianwen Jin. Scale-aware modulation meet transformer. In *Proc. International Conference on Learning Representations (ICLR)*, pp. 5992–6003, 2023.

Yue Liu, Yunjie Tian, Yuzhong Zhao, Hongtian Yu, Lingxi Xie, Yaowei Wang, Qixiang Ye, Jianbin Jiao, and Yunfan Liu. Vmamba: Visual state space model. In *Proc. Annual Conference on Neural Information Processing Systems (NeurIPS)*, 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 *Proc. International Conference on Computer Vision (ICCV)*, pp. 9992–10002, 2021.

Ze Liu, Han Hu, Yutong Lin, Zhuliang Yao, Zhenda Xie, Yixuan Wei, Jia Ning, Yue Cao, Zheng Zhang, Li Dong, Furu Wei, and Baining Guo. Swin transformer V2: scaling up capacity and resolution. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11999–12009, 2022.

Jiachen Lu, Junge Zhang, Xiatian Zhu, Jianfeng Feng, Tao Xiang, and Li Zhang. Softmax-free linear transformers. *International Journal of Computer Vision*, 132:3355–3374, 2024.

Zhisheng Lu, Juncheng Li, Hong Liu, Chaoyan Huang, Linlin Zhang, and Tieyong Zeng. Transformer for single image super-resolution. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pp. 457–466, June 2022.

Nanye Ma, Mark Goldstein, Michael S. Albergo, Nicholas M. Boffi, Eric Vanden-Eijnden, and Saining Xie. Sit: Exploring flow and diffusion-based generative models with scalable interpolant transformers. In *Proc. European Conference on Computer Vision (ECCV)*, 2024.

Weikang Meng, Yadan Luo, Liangyu Huo, Yaowei Wang, Xin Li, and Zheng Zhang. Nalaformer: Norm-aware linear attention for transformer models. *CoRR*, abs/2506.21137, 2025a.

Weikang Meng, Yadan Luo, Xin Li, Dongmei Jiang, and Zheng Zhang. Polaformer: Polarity-aware linear attention for vision transformers. In *Proc. International Conference on Learning Representations (ICLR)*, 2025b.

Zakaria Mhammedi, Andrew Hellicar, Ashfaqur Rahman, and James Bailey. Efficient orthogonal parametrisation of recurrent neural networks using householder reflections. In *International Conference on Machine Learning*, pp. 2401–2409. PMLR, 2017.

MiniMax, Aonian Li, Bangwei Gong, and et al. Minimax-01: Scaling foundation models with lightning attention. *CoRR*, abs/2501.08313, 2025.

Yury Nahshan, Joseph Kampeas, and Emir Haleva. Linear log-normal attention with unbiased concentration. In *Proc. International Conference on Learning Representations (ICLR)*, 2024.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proc. International Conference on Computer Vision (ICCV)*, pp. 4172–4182, 2023.

Yifan Pu, Zhuofan Xia, Jiayi Guo, Dongchen Han, Qixiu Li, Duo Li, Yuhui Yuan, Ji Li, Yizeng Han, Shiji Song, Gao Huang, and Xiu Li. Efficient diffusion transformer with step-wise dynamic attention mediators. In *Proc. European Conference on Computer Vision (ECCV)*, 2024.

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. *CoRR*, abs/2511.00833, 2025a.

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. *CoRR*, abs/2511.00833, 2025b.

Zhen Qin, Weixuan Sun, Hui Deng, Dongxu Li, Yunshen Wei, Baohong Lv, Junjie Yan, Lingpeng Kong, and Yiran Zhong. Cosformer: Rethinking softmax in attention. In *Proc. International Conference on Learning Representations (ICLR)*, 2022.

Zhen Qin, Weigao Sun, Dong Li, Xuyang Shen, Weixuan Sun, and Yiran Zhong. Various lengths, constant speed: Efficient language modeling with lightning attention. In *Proc. International Conference on Machine Learning (ICML)*, 2024.

Zhuoran Shen, Mingyuan Zhang, Haiyu Zhao, Shuai Yi, and Hongsheng Li. Efficient attention: Attention with linear complexities. In *Proc. Winter Conference on Applications of Computer Vision (WACV)*, pp. 3530–3538, 2021.

Yuheng Shi, Minjing Dong, Mingjia Li, and Chang Xu. VSSD: vision mamba with non-casual state space duality. *CoRR*, abs/2407.18559, 2024a.

Yuheng Shi, Minjing Dong, and Chang Xu. Multi-scale vmamba: Hierarchy in hierarchy visual state space model. In *Proc. Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2024b.

Yi Tay, Mostafa Dehghani, Samira Abnar, Yikang Shen, Dara Bahri, Philip Pham, Jinfeng Rao, Liu Yang, Sebastian Ruder, and Donald Metzler. Long range arena : A benchmark for efficient transformers. In *Proc. International Conference on Learning Representations (ICLR)*, 2021.

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *Proc. International Conference on Machine Learning (ICML)*, pp. 10347–10357, 2021.

Zhengzhong Tu, Hossein Talebi, Han Zhang, Feng Yang, Peyman Milanfar, Alan C. Bovik, and Yinxiao Li. Maxvit: Multi-axis vision transformer. In *Proc. European Conference on Computer Vision (ECCV)*, pp. 459–479, 2022.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proc. Annual Conference on Neural Information Processing Systems (NeurIPS)*, pp. 5998–6008, 2017.

Madhusudan Verma. Revisiting linformer with a modified self-attention with linear complexity. *CoRR*, abs/2101.10277, 2021.

Dustin Wang, Rui-Jie Zhu, Steven Abreu, Yong Shan, Taylor Kergan, Yuqi Pan, Yuhong Chou, Zheng Li, Ge Zhang, Wenhao Huang, and Jason Eshraghian. A systematic analysis of hybrid linear attention. *CoRR*, abs/2507.06457, 2025.

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 *Proc. International Conference on Computer Vision (ICCV)*, pp. 548–558, 2021.

Wenhai Wang, Enze Xie, Xiang Li, Deng Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pvt v2: Improved baselines with pyramid vision transformer. *Computational Visual Media*, 8:415–424, 2022.

Wenhai Wang, Jifeng Dai, Zhe Chen, Zhenhang Huang, Zhiqi Li, Xizhou Zhu, Xiaowei Hu, Tong Lu, Lewei Lu, Hongsheng Li, Xiaogang Wang, and Yu Qiao. Internimage: Exploring large-scale vision foundation models with deformable convolutions. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14408–14419, 2023.

Zhaozhi Wang, Yue Liu, Yunfan Liu, Hongtian Yu, Yaowei Wang, Qixiang Ye, and Yunjie Tian. vheat: Building vision models upon heat conduction. *CoRR*, abs/2405.16555, 2024.

Chaodong Xiao, Minghan Li, Zhengqiang Zhang, Deyu Meng, and Lei Zhang. Spatial-mamba: Effective visual state space models via structure-aware state fusion. In *Proc. International Conference on Learning Representations (ICLR)*, 2025.

Yicheng Xiao, Lin Song, Shaoli Huang, Jiangshan Wang, Siyu Song, Yixiao Ge, Xiu Li, and Ying Shan. Mambatree: Tree topology is all you need in state space model. In *Proc. Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2024.

Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, José M. Álvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. In *Proc. Annual Conference on Neural Information Processing Systems (NeurIPS)*, pp. 12077–12090, 2021.

Fei Xie, Zhongdao Wang, Weijia Zhang, and Chao Ma. Pvmamba: Parallelizing vision mamba via dynamic state aggregation. In *Proc. International Conference on Computer Vision (ICCV)*, 2025.

Haotian Yan, Ming Wu, and Chuang Zhang. Multi-scale representations by varying window attention for semantic segmentation. In *Proc. International Conference on Learning Representations (ICLR)*, 2024.

Chenglin Yang, Siyuan Qiao, Qihang Yu, Xiaoding Yuan, Yukun Zhu, Alan L. Yuille, Hartwig Adam, and Liang-Chieh Chen. MOAT: alternating mobile convolution and attention brings strong vision models. In *Proc. International Conference on Learning Representations (ICLR)*, 2023.

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 *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14431–14442, 2023.

Weihao Yu and Xinchao Wang. Mambaout: Do we really need mamba for vision? In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4484–4496, 2025.

Kun Zeng, Hanjiang Lin, Zhiqiang Yan, and Jinsheng Fang. Densely connected transformer with linear self-attention for lightweight image super-resolution. *IEEE Trans. Instrum. Meas.*, 72:1–12, 2023.

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. *CoRR*, abs/2601.07832, 2026.

Michael Zhang, Kush Bhatia, Hermann Kumbong, and Christopher Ré. The hedgehog & the porcupine: Expressive linear attentions with softmax mimicry. In *Proc. International Conference on Learning Representations (ICLR)*, 2024.

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*, 127:302–321, 2019.

Zhanchao Zhou, Xiaodong Chen, Haoxing Chen, Zhenzhong Lan, and Jianguo Li. Knocking-heads attention. *CoRR*, abs/2510.23052, 2025.

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 *Proc. International Conference on Machine Learning (ICML)*, 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 *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7664–7674, 2025.

A APPENDIX

- A.1 **Limitations.** Practical constraints and opportunities for hardware-aware optimization.
- A.2 **Related Work.** Additional context on efficient vision models and linear attention.
- A.3 **Algorithm.** Pseudocode for the reflecting feature map.
- A.4 **Ablation Study.** Hyperparameter, component, operator, and feature-map comparisons.
- A.5 **Additional Benchmarks.** Results with a Semantic FPN decoder and on Long Range Arena.
- A.6 **Mechanism Analysis.** Boundary proximity, activation diversity, and feature-retention measurements.
- A.7 **Efficiency Analysis.** Runtime overhead, throughput, memory scaling, and super-resolution costs.
- A.8 **Visualizations.** Additional semantic-segmentation and super-resolution examples.
- A.9 **Experimental Settings.** Model configurations, datasets, and training and evaluation protocols.
- A.10 **Full Tables.** Complete classification, detection, and segmentation results.

A.1 LIMITATIONS

While HaLA demonstrates superior representational capacity and theoretical linear complexity, its current implementation is primarily optimized at the functional level. As a result, there remains a natural gap between its theoretical efficiency and the peak practical inference throughput achievable on modern hardware. Specifically, the core geometric modulations (e.g., Householder reflections) have not yet been fully integrated with custom, hardware-aware acceleration libraries like Flash Linear Attention (FLA) via dedicated low-level kernels (e.g., Triton). Consequently, while the model is highly effective and efficient in standard settings, its deployment in ultra-high-dimensional tasks could further benefit from specific engineering efforts, such as custom memory tiling and operator fusion. We view bridging this software-hardware synergy as a promising engineering step for future large-scale deployment.

A.2 RELATED WORK

Efficient Sequential Vision Models. Treating images as token sequences has become a standard approach in modern vision architectures, inspired by the success of Transformer models in natural language processing (Vaswani et al., 2017). Vision Transformer (ViT) (Dosovitskiy et al., 2021) demonstrates that patchified images can be processed by self-attention to capture long-range dependencies, but its reliance on softmax attention incurs quadratic computational and memory costs, which quickly become prohibitive for high-resolution vision tasks. To mitigate this issue, a line of research focuses on improving the efficiency of Transformer-based vision models while largely preserving the attention formulation. DeiT (Touvron et al., 2021) leverages distillation strategies to enhance data efficiency, and Swin Transformer (Liu et al., 2021; 2022) restricts attention to shifted local windows, enabling scalable training while maintaining cross-window information flow. These methods primarily reduce constant factors or constrain attention scope, rather than fundamentally resolving the quadratic complexity of softmax attention. More recent efforts explore alternatives beyond standard attention mechanisms to further improve scalability. State space models (SSMs) have been adapted to vision by sequentially processing patch tokens, with VMamba (Liu et al., 2024; Huang et al., 2024) adopting raster-scan ordering to model spatial dependencies under linear-time complexity. In a different vein, VHeat (Wang et al., 2024) formulates visual representation learning as a diffusion-like process and models patch interactions through heat conduction, achieving sub-quadratic complexity via discrete cosine transform (DCT)-based operations.

Linear Attention. Linear attention reduces the quadratic complexity of softmax attention by replacing the exponential similarity with kernelized feature maps, enabling associative reordering and linear-time computation (Katharopoulos et al., 2020). Early approaches focus on approximating softmax using separable kernels, with representative designs based on ReLU (Han et al., 2023; Cai et al., 2023), $\exp$ (Nahshan et al., 2024). Performer (Choromanski et al., 2021) utilizes Random Fourier Features (RFF) to provide an unbiased estimate of the softmax kernel, while CosFormer (Qin et al., 2022) employs a linearithmic complexity approach with a ReLU-based kernel and cosine re-weighting to enforce locality. Linformer (Verma, 2021) approximates the self-attention matrix via low-rank decomposition, projecting the key and value sequences into a lower-dimensional subspace to achieve linear complexity. Beyond kernel selection, recent methods further introduce data-dependent modulation or gating mechanisms to stabilize large-scale training (Qin et al., 2024; MiniMax et al., 2025). While effective, such approaches still rely on fixed geometric parameterizations and do not explicitly address the information loss induced by kernelization. From a theoretical perspective, PolaFormer (Meng et al., 2025b) mitigates information loss from non-negative feature maps via sign-aware inner product decomposition and using power function to lower the attention entropy. SAGA (Cao & Wang, 2025) and RALA (Fan et al., 2025b) exploit the property of the Hadamard product to enhance the intermediate matrices, thereby alleviating the low-rank bottleneck in linear attention. In-Line (Han et al., 2024a) shows that softmax attention is injective under mild conditions, which is generally not in linear attention, and MALA (Fan et al., 2025a) and NaLaFormer (Meng et al., 2025a) also link this gap to information loss induced by non-negative feature maps. Consequently, while existing studies strive to preserve non-negativity, they largely overlook the fact that conventional activations as *passive truncation* mechanisms, incurring severe information loss and limiting the representational capacity of linear attention.

A.3 ALGORITHM

Algorithm 1 Reflecting Feature Map

```
# x: input features [B, N, D]
# H: number of heads, d = D / H

# ----- Global reflection -----
u_c = normalize(randn(D))
x = x - 2 * u_c * <x, u_c>
# ----- Head-wise reshape -----
x = reshape(x, [B, H, N, d])
# ----- Block partition -----
x_blk = reshape(x, [B, H, N, d//2, 2])
# ----- Variance-aware angle modulation -----
theta = (randn(H, d//2)
var = Var(x_blk over N and 2D)
delta = sigmoid(lambda / (var + epsilon)) * alpha_max
theta = theta + delta
# ----- Block-wise reflection -----
u = [cos (theta), sin (theta)]
x_blk = x_blk - 2 * u * <x_blk, u>
# ----- Restore shape -----
x = reshape(x_blk, [B, H, N, D])
return ReLU(x)
```

A.4 ABLATION STUDY

Controlled attention replacements. On fixed Swin-T and PVT-T backbones, HaLA-ReLU improves Swin-T accuracy from 81.8% to 82.4%, and HaLA-FLatten improves PVT-T accuracy from 77.8% to 79.2% (Tab. 7). These replacements keep the non-attention architecture and training protocol fixed.

Table 7: **Comparison with other linear attention models on the Swin-T setting.**

SWIN-MODEL	PARAMS	FLOPS	ACC. (%)
Swin-T (Liu et al., 2021)	28M	4.4G	81.2
Hydra (Bolya et al., 2022)	29M	4.5G	80.7
Efficient (Shen et al., 2021)	29M	4.5G	81.0
Linear Angular (You et al., 2023)	29M	4.5G	79.4
FLatten (Han et al., 2023)	29M	4.5G	82.1
Agent (Han et al., 2024c)	29M	4.5G	82.6
InLine (Han et al., 2024a)	30M	4.5G	82.4
PolaFormer (Meng et al., 2025b)	29M	4.5G	82.6
LA-ReLU	28M	4.6G	81.8
HaLA-ReLU	28M	4.6G	**82.4**
PVT-MODEL	**PARAMS**	**FLOPS**	**ACC. (%)**
PVT-T (Wang et al., 2021)	13M	1.9G	75.1
HaLA(ReLU)	12M	2.0G	**78.6**
Agent (Han et al., 2024c)	12M	2.0G	78.4
InLine (Han et al., 2024a)	12M	2.0G	78.2
LinearDiff (Pu et al., 2025b)	12M	2.0G	78.2
CA (Han et al., 2026)	12M	2.0G	78.1
Polaformer Meng et al. (2025b)	12M	2.0G	78.8
FLatten Han et al. (2023)	12M	2.0G	77.8
HaLA-FLatten	12M	2.0G	**79.2**

Hyperparameter analysis. Table 8 reports the original HaLA-XT ImageNet-1K study (6M parameters, 0.6G FLOPs), including sensitivity to λ and $\alpha_{\max}$. The labels "full" and "head" in that table denote cross-head and block-wise reflection, respectively. Removing cross-head reflection reduces accuracy by 0.5 percentage points; removing block-wise reflection reduces it by 0.3 points. Further component isolation and larger-scale controls follow below.

Table 8: **Hyperparameter ablations** for HaLA.

λ	$\alpha_{\max}$	FULL	HEAD	ACC. (%)
1	$\pi/2$	✓	✓	76.0
1	$\pi/2$		✓	75.5 -0.5
1	$\pi/2$	✓		75.7 -0.3
0.5	$\pi/2$	✓	✓	75.7 -0.3
0.25	$\pi/2$	✓	✓	75.8 -0.2
1	$\pi/4$	✓	✓	75.9 -0.1
1	$\pi/8$	✓	✓	75.8 -0.2

Alternative Cross-Head Operators. We replace the cross-head operator in HaLA-T while keeping the model scale approximately matched. Table 9 compares the learned Householder reflection with generic and structured alternatives. The Householder variant achieves the highest accuracy in this controlled comparison. On the smaller HaLA-XT model, a LayerNorm-plus-projection replacement reaches 75.8% versus 76.0% for the original operator.

Table 9: Cross-head operator replacements on HaLA-T. Parameter counts and FLOPs are rounded as reported in the rebuttal.

CROSS-HEAD OPERATOR	PARAMS	FLOPS	TOP-1 (%)
Learned Householder (ours)	15M	2.6G	**82.8**
MLP	15M	2.7G	82.5
Learned rotation	15M	2.6G	82.5
Sign-preserving map	16M	2.7G	82.6
LayerNorm + projection	16M	2.7G	82.4

Component Isolation Across Model Sizes and Tasks. Table 10 separates cross-head reflection, block-wise reflection, and variance-aware modulation on HaLA-XT. The latter is disabled while retaining the two reflection stages, yielding 75.6% accuracy. Table 11 extends the reflection ablations to HaLA-T classification and DiT generation. Both removals reduce classification accuracy by 0.4 points and worsen DiT FID. The DiT columns use the same evaluation protocol as Tab. 5.

Table 10: Component isolation on HaLA-XT (ImageNet-1K). The block-wise branch is absent when block-wise reflection is disabled.

CROSS-HEAD	BLOCK-WISE	VARIANCE MODULATION	TOP-1 (%)
✓	✓	✓	**76.0**
–	✓	✓	75.5
✓	–	–	75.7
✓	✓	–	75.6

Table 11: Reflection ablations on DiT and HaLA-T. Lower FID/sFID and higher IS/precision/recall/accuracy are better.

VARIANT	FID	SFID	IS	PRECISION	RECALL	TOP-1 (%)
HaLA	**58.07**	**11.21**	**24.70**	**0.41**	**0.59**	**82.8**
Without cross-head reflection	58.89	11.62	24.44	0.41	0.58	82.4
Without block-wise reflection	58.83	11.33	24.51	0.40	0.58	82.4

Seed Stability and Alternative Positive Maps. Four independent HaLA-T ImageNet-1K runs give 82.8%, 82.8%, 82.8%, and 82.9% Top-1 accuracy for seeds 0, 42, 84, and 168, respectively. On LRA Retrieval, HaLA improves both the ReLU and $1 + \mathrm{ELU}$ feature maps by 2.6 points (Tab. 12). The latter map smoothly compresses negative inputs instead of setting them exactly to zero; the result tests whether reorientation remains useful beyond a hard ReLU gate.

Table 12: LRA Retrieval accuracy with and without HaLA under two non-negative feature maps.

FEATURE MAP	BASELINE	WITH HALA	GAIN
ReLU	78.9	**81.5**	+2.6
$1 + \mathrm{ELU}$	77.8	**80.4**	+2.6

A.5 ADDITIONAL BENCHMARKS

To further evaluate our approach, we conduct semantic segmentation experiments using the MMSegmentation framework (Contributors, 2018). Following the experimental protocols of PVT (Wang et al., 2021), we train the models equipped with a Semantic FPN decoder. Results demonstrate that HaLA achieves the highest performance among models at a comparable scale. We also evaluate HaLA on the Long Range Arena (LRA) benchmark, as detailed in Tab. 14.

Table 13: **More semantic segmentation results.** Results on Semantic FPN 80K.

METHOD	PARA	FLOPS	MIOU
VAN-B1 (Guo et al., 2022c)	18M	140G	42.9
PVT-v2-B1 Wang et al. (2022)	18M	136G	42.5
RMT-T (Fan et al., 2024)	17M	136G	46.4
FL-PVT-T (Han et al., 2023)	16M	169G	37.2
Agent-PVT-T (Han et al., 2024c)	15M	147G	40.2
RAVLT-T (Fan et al., 2025b)	18M	136G	47.5
HaLA-T	18M	136G	**47.7**

Table 14: **Full table of LRA:** Throughput and Peak Memory of various models. A denotes the accuracy, T denotes the throughput of each model and M denotes the peak memory cost.

MODEL	IMAGE (1K)		PATH (1K)		LIST (2K)		TEXT (4K)		RETRIEVAL (4K)	
	ACC	T	ACC	T	ACC	T	ACC	T	ACC	T
Softmax	39.14	1037	70.39	1014	38.71	535	61.55	306	80.93	152
Kernelized	32.63	1243	69.86	1193	38.46	678	60.02	390	82.11	193
Nystrom	38.94	2206	69.34	1961	37.95	1643	62.36	2708	80.89	1134
Linformer	38.43	3343	65.39	2861	36.44	2073	57.29	2938	77.85	1466
Informer	37.86	1389	56.44	1215	37.05	1067	62.13	1237	79.35	464
Skyformer	40.77	1457	70.73	1361	38.69	1255	64.70	1584	82.06	713
HaLA	42.54	2861	71.31	2631	38.21	1938	73.48	2664	80.42	1285

A.6 Mechanism Analysis

Boundary Proximity and Activation Diversity. For a reflected 2D block mean v, define its normalized distance to the nearest ReLU boundary as

$$\eta(v) = \frac{\min(|v_1|, |v_2|)}{\|v\|_2}. \tag{18}$$

The rebuttal evaluates a trained HaLA-T on 4,096 ImageNet validation images, collecting 1.57 million query/key 2D-block samples per checkpoint. Table 15 reports the fraction of reflected means within three boundary margins at four training checkpoints, with an isotropic-orientation reference. The frequency near the boundary exceeds that reference. This is an empirical observation; the local boundary-margin result in Theorem 1 is conditional on the mean's location and does not guarantee that every block is placed near a boundary.

Table 15: Fraction of reflected block means within normalized ReLU-boundary margins.

Checkpoint (epoch)	$P(\eta < 0.05)$	$P(\eta < 0.1)$	$P(\eta < 0.2)$
97	8.4%	16.8%	33.0%
136	9.1%	17.9%	34.6%
245	10.2%	19.7%	36.3%
247	10.1%	19.5%	36.0%
Isotropic orientation	6.4%	12.7%	25.7%

For each 2D block, an activation pattern is one of the four binary ReLU masks $(0,0)$, $(0,1)$, $(1,0)$, and $(1,1)$. The measured average number of distinct patterns is 3.77 of four over all blocks and 3.62 of four for the lowest-variance 10%. The respective fractions of blocks with only one pattern are 0.26% and 0.09%. These statistics quantify post-gating diversity, including in low-variance blocks.

Energy and Active-Dimension Retention. We compare direct $\mathrm{ReLU}(x)$ with $\mathrm{ReLU}(Hx)$ on the same pre-activation query/key features from a trained HaLA-Tiny checkpoint in inference mode. The measurements use 20 ImageNet-1K validation images and aggregate over all linear-attention layers. Energy is squared ℓ_2 norm: $E = \|x\|_2^2 = \|Hx\|_2^2$, and the reported retained fraction is $\|\mathrm{ReLU}(x)\|_2^2/E$ or $\|\mathrm{ReLU}(Hx)\|_2^2/E$. A feature dimension is active when its post-activation value is positive. Each image contributes 1,204,224 query/key feature dimensions. Table 16 gives the per-image results. Across these images, the direct map discards 13.90–19.74% of energy, compared with 7.33–13.23% after learned reorientation; the active-dimension ratio increases from 61.0–64.4% to 75.1–80.1%.

Table 16: Per-image retention before and after learned reorientation. "Lost" is one minus the squared-energy retention ratio; "active" is the fraction of positive post-activation dimensions.

Image	Energy Lost		Active Dims		Image	Energy Lost		Active Dims	
	Direct	HaLA	Direct	HaLA		Direct	HaLA	Direct	HaLA
1	17.85%	10.52%	61.5%	78.8%	11	13.90%	7.33%	63.7%	76.6%
2	17.58%	10.40%	61.4%	79.6%	12	14.77%	8.21%	62.6%	76.6%
3	17.19%	10.10%	61.0%	79.1%	13	14.28%	7.75%	62.6%	77.2%
4	15.81%	8.82%	62.3%	78.5%	14	15.14%	8.63%	62.2%	78.9%
5	17.24%	10.49%	61.0%	77.8%	15	19.74%	13.23%	61.0%	80.1%
6	14.25%	7.51%	63.0%	75.1%	16	15.28%	8.80%	62.3%	76.5%
7	18.07%	11.34%	61.6%	78.7%	17	14.43%	7.97%	64.4%	79.5%
8	14.78%	8.09%	63.0%	79.8%	18	14.94%	8.53%	62.6%	78.0%
9	15.43%	8.79%	63.0%	79.1%	19	15.59%	9.21%	62.3%	78.1%
10	14.80%	8.19%	61.7%	78.6%	20	14.92%	8.56%	63.0%	79.3%

A.7 Efficiency Analysis

Table 17: **Throughput comparison on PVT backbone.**

Attention	Model Size	
	Tiny	Small
Vanilla Attn	1174	724
Agent Attn (Han et al., 2024c)	2667	1430
MALA (Fan et al., 2025a)	1821	967
RAVLT (Fan et al., 2025b)	1764	911
FLatten (Han et al., 2023)	2823	1514
FLatten+HaLA	1730	913
ReLU	2949	1581
RELU+HaLA	1903	1021

Operator Cost and Image Throughput. An operator-level timing decomposition from the rebuttal is shown in Tab. 18. The source reports time values without a unit, so only their relative shares are interpreted here. On one A6000 GPU, the measured ImageNet-1K throughputs of HaLA-T/S/B/L are 1184.2, 668.3, 369.7, and 292.1 images/s, respectively. The throughput of fixed PVT attention replacements is separately reported in Tab. 17; adding the reflection incurs a measurable cost relative to the bare linear-attention feature map.

Table 18: Operator-level runtime decomposition reported in the rebuttal. Time values use the source's unspecified units.

Component	Time	Share
Cross-head reflection	22.7	5.4%
Block-wise reflection	70.0	16.6%
Attention computation	328.6	78.0%
Total	421.3	100%

Long-Sequence Scaling. Tables 19 and 20 reproduce the RTX 3090 end-to-end measurements from the rebuttal. The source does not specify the units for the memory and throughput numbers, so the tables retain its raw values. At 262K tokens, HaLA and the listed linear-attention implementations have similar reported peak memory; the PyTorch softmax implementation runs out of memory beyond 36K in this setup. HaLA's throughput at 262K is 17.5 in the reported units, compared with 2.8 for the Flash softmax implementation and 18.5 for FLatten. These measurements show the practical overhead alongside the shared linear-attention memory trend.

Table 19: Peak memory versus sequence length on RTX 3090 (raw source units). OOM indicates out of memory.

Method	4K	9K	16K	36K	65K	147K	200K	262K
PolaFormer	56.2	97.5	153.7	389.3	1120.9	5372.8	9854.0	OOM
FLatten	41.7	64.6	109.6	396.5	1134.2	5400.1	9894.3	16749.4
Agent	50.4	84.0	128.8	389.6	1121.4	5373.4	9856.7	16702.2
InLine	41.5	64.0	105.9	389.3	1120.9	5372.8	9856.0	16701.4
HaLA	43.1	66.2	105.9	389.3	1120.9	5372.8	9856.8	16701.4
Softmax (PyTorch)	291.4	1346.4	4167.6	20867.9	OOM	OOM	OOM	OOM
Softmax (Flash)	40.9	62.8	105.9	389.3	1120.9	5372.8	9854.0	16701.4

Table 20: Throughput versus sequence length on RTX 3090 (raw source units).

METHOD	4K	9K	16K	36K	65K	147K	200K	262K
PolaFormer	321.1	312.6	279.2	163.0	93.0	33.4	22.8	OOM
FLatten	353.5	336.7	331.7	206.4	116.8	39.9	26.7	18.5
Agent	347.6	349.9	308.4	181.7	103.8	36.2	21.6	17.0
InLine	379.8	379.9	356.7	217.6	123.7	41.8	27.8	19.1
HaLA	296.0	300.7	289.0	186.4	107.5	37.7	25.4	17.5
Softmax (PyTorch)	345.5	174.7	77.0	16.0	OOM	OOM	OOM	OOM
Softmax (Flash)	399.4	359.7	263.9	94.4	37.5	8.3	4.7	2.8

Super-Resolution Efficiency Against a Bare ReLU Map. The main SR table compares HaLA with softmax attention. The additional rebuttal control uses the DCTLSA architecture at $\times 3$ scale on RTX 3090 and replaces attention with a plain ReLU linear map. Table 21 shows that HaLA incurs latency and memory overhead relative to this bare map, while remaining considerably cheaper than softmax attention under the same measurement. The corresponding reconstruction quality of the HaLA and softmax variants is reported in Tab. 4; the rebuttal does not report PSNR/SSIM for this plain ReLU control.

Table 21: DCTLSA $\times 3$ inference latency (ms) and memory (GB) on RTX 3090. Values are taken from the rebuttal's matched ReLU/softmax/HaLA comparison.

	SET5		SET14		B100		URBAN100		MANGA109	
VARIANT	LAT.	MEM.	LAT.	MEM.	LAT.	MEM.	LAT.	MEM.	LAT.	MEM.
ReLU	33.24	0.28	32.05	0.43	25.70	0.16	100.59	1.28	127.38	1.01
Softmax	57.90	1.66	113.04	2.27	76.74	0.93	355.81	7.19	571.85	5.49
HaLA	46.11	0.33	51.82	0.51	42.70	0.19	146.27	1.52	183.47	1.20

A.8 VISUALIZATIONS

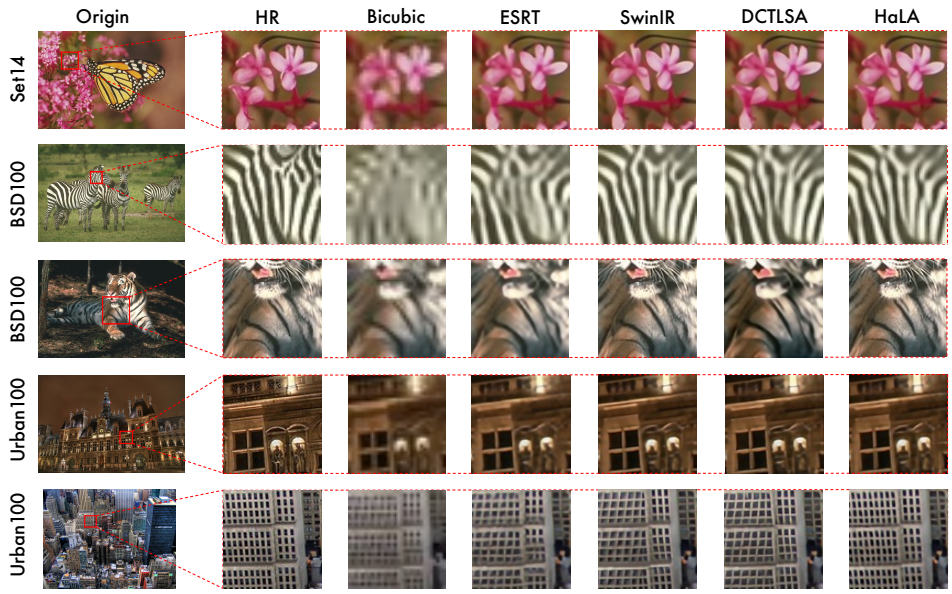

Figure 4: **More visualizations on super-resolution tasks.** Our model produces more clear boundaries, more accurate shapes, and finer-grained textures compared to baselines.

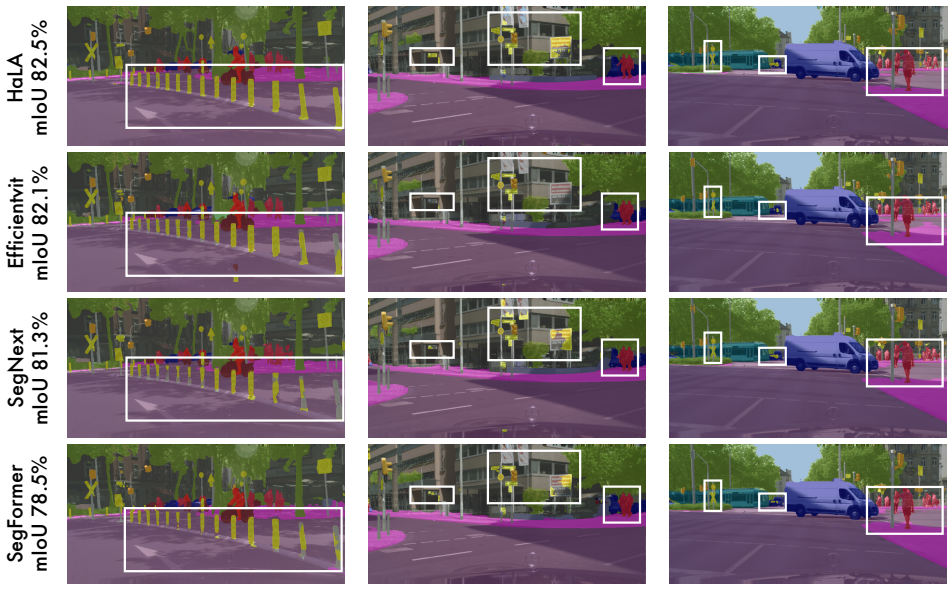

Figure 5: **Visualization results on semantic segmentation.** Compared to other methods, our approach more accurately delineates different objects, effectively avoiding missing predictions and boundary ambiguities.

A.9 Experimental Settings

A.9.1 Model Size

Consistent with established works (Fan et al., 2025b; 2024; Liu et al., 2021), we develope a set of HaLA backbones by only replace the non-negative activations with our reflecting feature map, each with varying configurations of block count and channel dimensions across their respective stages whose the ratio of MLP is set to 3.5. The architecture details are illustrated in the Tab. 22.

Table 22: **Architecture setting details of HaLA.**

Model	Param	Flops	Blocks	Channels	Heads
HaLA-XT	6M	0.6G	[2, 2, 2, 2]	[32, 64, 128, 384]	[1, 2, 4, 8]
HaLA-T	15M	2.6G	[2, 2, 6, 2]	[64, 128, 256, 512]	[1, 2, 4, 8]
HaLA-S	26M	4.9G	[3, 5, 9, 3]	[64, 128, 320, 512]	[1, 2, 5, 8]
HaLA-B	48M	10.6G	[4, 6, 12, 6]	[96, 192, 384, 512]	[1, 2, 6, 8]
HaLA-L	95M	17.3G	[4, 7, 19, 8]	[96, 192, 448, 640]	[1, 2, 7, 10]

A.9.2 Datasets and Experiment Details

Image Classification. For image classification, all models are trained using the AdamW optimizer for 320 epochs, including 20 epochs of linear warm-up. The base learning rate is set to 0.001 with a micro-batch size of 128 and a global batch size of 1024. Our training framework is built upon the official DeiT implementation.

Object Detection and Segmentation. We further conduct extensive experiments on object detection using the COCO dataset (Lin et al., 2014), which consists of 118K training images and 5K validation images annotated with 80 object categories. Our model serves as the backbone, initialized with pretrained weights from ImageNet-1K. All experiments are implemented based on the *mmcv-detection* framework (Contributors, 2018). The models are trained under both $1\times$ (12 epochs) and $3\times$ (36 epochs) schedules. We adopt the AdamW optimizer with a learning rate of 0.0001, a weight decay of 0.0001, and a "step" learning rate policy.

Semantic Segmentation. For semantic segmentation, we conduct experiments on the ADE20K (Zhou et al., 2019) and Cityscapes (Cordts et al., 2016) datasets. ADE20K contains 25,000 densely annotated images depicting complex real-world scenes with rich contextual relationships, while Cityscapes consists of high-resolution urban street scenes with fine-grained annotations. We follow the *mmcv-segmentation* framework (Contributors, 2018) and adopt SegNeXt (Guo et al., 2022b) and SegMAN (Fu et al., 2025) as the decoder heads. All models are trained using the AdamW optimizer with a learning rate of 0.0001 and a weight decay of 0.001.

Super Resolution. To evaluate both the reconstruction performance and computational efficiency of our method in super-resolution tasks, we follow prior efficient SISR works, including ESRT (Lu et al., 2022) and DCTLSA (Zeng et al., 2023). We use DIV2K (Agustsson & Timofte, 2017) as the training dataset. For evaluation, we adopt five widely used benchmark datasets: Set5, Set14, BSDS100, Urban100 and Manga109. Performance is measured using PSNR and SSIM. Additionally, we report memory consumption and inference time to highlight computational efficiency.

Diffusion Transformer. We evaluate the effectiveness of HaLA in generative modeling using diffusion models on ImageNet-1K (Deng et al., 2009). Following DiT (Peebles & Xie, 2023) and SiT (Ma et al., 2024), we adopt the Diffusion Transformer S/2 architecture as the primary experimental setting. HaLA is integrated into both DiT and SiT by replacing the original attention modules, while keeping all other architectural components and training configurations unchanged to ensure fair comparisons. All models are trained and evaluated under the same protocols as prior works. Generative performance is assessed using standard metrics, including FID, sFID, IS, Precision and Recall.

A.10 FULL TABLES

Table 23: **Full Table of the Comparison of the ImageNet-1K classification with the SOTA efficient models.** The "PARAMS" column represents the number of parameters, "ACC" for Top-1 accuracy.

MODEL	PARAMS	FLOPS	ACC.
Conv2Former-N (Hou et al., 2024)	15M	2.2G	81.5
RMT-T (Fan et al., 2024)	14M	2.5G	82.4
Agent-PVT-T (Han et al., 2024c)	12M	2.0G	78.4
RMT-T (Fan et al., 2024)	14M	2.5G	82.4
RAVLT-T (Fan et al., 2025b)	15M	2.6G	82.6
HaLA-T	15M	2.6G	**82.8**
Conv2Former-T (Hou et al., 2024)	27M	4.4G	83.2
MambaOut-T (Yu & Wang, 2025)	27M	4.5G	82.7
VMamba-T (Liu et al., 2024)	30M	4.9G	82.6
Agent-Swin-T (Han et al., 2024c)	29M	4.5G	82.6
Vim-S (Zhu et al., 2024)	26M	3.7G	80.6
VMamba-T (Liu et al., 2024)	30M	4.9G	82.6
LocalVMamba-T (Huang et al., 2024)	26M	5.7G	82.7
MOAT-0 (Yang et al., 2023)	28M	5.7G	83.3
Pola-Swin-T (Meng et al., 2025b)	29M	4.5G	82.6
ViG-H-T (Liao et al., 2025)	29M	4.5G	82.8
MILA-T (Han et al., 2024b)	25M	4.2G	83.5
HaLA-S	26M	4.9G	**84.2**
MambaOut-S (Yu & Wang, 2025)	49M	9.0G	84.1
MogaNet-B (Li et al., 2024)	44M	10G	84.3
VMamba-S (Liu et al., 2024)	50M	8.7G	83.6
StructViT-B-8-1 (Kim et al., 2024)	52M	12G	84.3
SOFT-L (Lu et al., 2024)	64M	11G	83.1
FL-Swin-S (Han et al., 2023)	51M	8.7G	83.5
Agent-Swin-S (Han et al., 2024c)	50M	8.7G	83.7
Pola-Swin-S (Meng et al., 2025b)	50M	8.7G	83.6
ViG-H-S (Liao et al., 2025)	50M	8.8G	83.8
RMT-B (Fan et al., 2024)	54M	9.7G	85.0
MILA-S (Han et al., 2024b)	43M	7.3G	84.4
RAVLT-B (Fan et al., 2025b)	48M	10.6G	85.1
HaLA-B	48M	10.6G	**85.3**
MambaOut-B (Yu & Wang, 2025)	85M	15.8G	84.2
VMamba-B (Liu et al., 2024)	89M	15.4G	83.9
FL-Swin-B (Han et al., 2023)	89M	15.4G	83.8
Agent-Swin-B (Han et al., 2024c)	88M	15.4G	84.0
Pola-Swin-B (Meng et al., 2025b)	88M	15.4G	83.8
SMT-L (Lin et al., 2023)	81M	17.7G	84.6
VRWKV-B (Duan et al., 2025)	94M	18.2G	82.0
ViG-H-B(Liao et al., 2025)	89M	15.5G	84.2
MaxViT-B(Tu et al., 2022)	120M	23.4G	84.9
GC-ViT-B(Hatamizadeh et al., 2023)	90M	14.8G	85.0
RMT-L (Fan et al., 2024)	95M	18.2G	85.5
InLine-Swin-B (Han et al., 2024a)	88M	15.4G	82.0
MILA-B (Han et al., 2024b)	96M	16.2G	85.3
RAVLT-L (Fan et al., 2025b)	95M	17.3G	85.5
HaLA-L	95M	17.3G	**85.7**

Table 24: **Full Table of Object detection and instance segmentation results on the COCO dataset with RetinaNet 1 ×.** The "P" column represents the number of parameters, and "F" for FLOPs.

MODEL	P (M)	F (G)	RETINANET 1× AP^b	AP^b_{50}	AP^b_{75}	AP^b_S	AP^b_M	AP^b_L
PVT-T(Wang et al., 2021)	23	221	39.4	59.8	42.0	25.5	42.0	52.1
MPViT-XS(Lee et al., 2022)	20	211	43.8	65.0	47.1	28.1	47.6	56.5
SOFT++ T(Lu et al., 2024)	23	200	41.9	62.7	44.7	27.8	45.4	55.6
RMT-T(Fan et al., 2024)	23	199	45.1	66.2	48.1	28.8	48.9	61.1
RAVLT-T(Fan et al., 2025b)	24	201	44.1	64.3	47.4	26.4	48.1	59.3
HaLA-T	24	200	**46.1**	**67.4**	**49.4**	**28.8**	**50.4**	**61.7**
Swin-T(Liu et al., 2021)	38	248	41.7	63.1	44.3	27.0	45.3	54.7
MPViT-S(Lee et al., 2022)	32	248	45.7	57.3	48.8	28.7	49.7	59.2
PVTv2-B2(Wang et al., 2022)	35	290	44.6	65.6	47.6	27.4	48.8	58.6
CMT-S(Guo et al., 2022a)	44	231	44.3	65.5	47.5	27.1	48.3	59.1
RAVLT-S(Fan et al., 2025b)	44	262	46.7	67.2	50.4	31.4	51.1	62.3
HaLA-S	34	243	**48.5**	**69.8**	**52.2**	**32.2**	**52.9**	**63.7**
PVT-L(Wang et al., 2021)	71	345	42.6	63.7	45.4	25.8	46.0	58.4
SOFT++ M(Lu et al., 2024)	59	322	44.3	64.7	47.4	29.0	48.2	59.9
PVTv2-B3(Wang et al., 2022)	55	-	45.9	66.8	49.3	28.6	49.8	61.4
RAVLT-B(Fan et al., 2025b)	57	353	49.8	71.2	54.0	34.0	54.3	64.9
MAViT-B(Fan et al., 2025a)	57	353	49.9	71.1	53.8	33.7	54.5	**65.5**
HaLA-B	57	352	**50.2**	**71.7**	**54.3**	**34.0**	**54.8**	65.4
Swin-B(Liu et al., 2021)	98	477	45.0	66.4	48.3	28.4	49.1	60.6
MPViT-B(Lee et al., 2022)	85	482	47.0	68.4	50.8	29.4	51.3	61.5
RMT-L(Fan et al., 2024)	104	537	49.4	70.6	53.1	34.2	53.9	65.2
PVTv2-B5(Wang et al., 2022)	91	538	46.2	67.1	49.5	28.5	50.0	62.5
RAVLT-L(Fan et al., 2025b)	104	482	49.8	71.2	54.0	34.0	54.3	64.9
MAViT-L(Fan et al., 2025a)	104	482	50.6	71.7	54.9	34.1	55.3	65.6
HaLA-L	104	481	**50.9**	**72.2**	**55.0**	**35.4**	**55.3**	**66.7**

Table 25: **Full Table of Object detection and instance segmentation results on the COCO dataset with MR-CNN 1 ×.** The "P" column represents the number of parameters, and "F" for FLOPs.

MODEL	P (M)	F (G)	MASK R-CNN 1×					
			AP^b	AP^b_{50}	AP^b_{75}	AP^m	AP^m_{50}	AP^m_{75}
PVTv2-b1(Wang et al., 2022)	33	243	41.8	54.3	45.9	38.8	61.2	41.6
MPViT-XS(Lee et al., 2022)	30	231	47.3	69.1	51.9	42.7	66.2	46.0
FL-PVT-T(Han et al., 2023)	32	244	38.2	61.6	41.9	37.0	57.6	39.0
RAVLT-T(Fan et al., 2025b)	33	219	47.2	69.1	51.7	42.7	66.0	46.0
CA-Swin-T(Han et al., 2026)	-	269	44.5	67.3	48.9	40.5	64.2	43.5
Spatial-Mamba-T(Xiao et al., 2025)	46	261	47.6	69.6	52.3	42.9	66.5	46.2
PVMamba(Xie et al., 2025)	44	266	47.1	69.3	51.9	42.6	66.5	45.7
MAViT-T(Fan et al., 2025a)	33	226	47.5	69.0	52.3	42.8	66.3	46.3
MambaTreeV-T(Xiao et al., 2024)	-	265	47.0	69.4	51.5	42.7	66.4	46.0
MSVMamba-T(Shi et al., 2024b)	53	252	46.9	68.8	51.4	42.2	65.6	45.4
VSSD-T(Shi et al., 2024a)	44	265	46.9	69.4	51.4	42.6	66.4	45.9
HaLA-T	33	219	**47.7**	**69.6**	**52.6**	**43.0**	**66.6**	**46.2**
InternImage-T(Wang et al., 2023)	49	270	47.2	69.0	52.1	42.5	66.1	45.8
MambaOut-T(Yu & Wang, 2025)	43	262	45.1	67.3	49.6	41.0	64.1	44.1
MPViT-S(Lee et al., 2022)	43	268	46.4	68.6	51.2	42.4	65.6	45.7
CMT-S(Guo et al., 2022a)	45	249	44.6	66.8	48.9	40.7	63.9	43.4
MILA-T(Han et al., 2024b)	44	262	46.8	69.5	51.5	42.1	66.4	45.0
HaLA-S	44	262	**49.9**	**71.6**	**54.7**	**44.5**	**68.4**	**48.1**
InternImage-S(Wang et al., 2023)	69	340	47.8	69.8	52.8	43.3	67.1	46.7
STViT-B(Huang et al., 2022)	70	359	49.7	71.7	54.7	44.8	68.9	48.7
SOFT++ M(Lu et al., 2024)	69	342	46.6	67.8	51.2	42.0	64.8	45.2
MILA-S(Han et al., 2024b)	63	319	49.2	71.5	53.9	44.2	68.5	47.2
RAVLT-B(Fan et al., 2025b)	67	372	51.2	72.7	56.4	45.7	69.9	49.5
HaLA-B	67	371	**51.5**	**73.2**	**56.6**	**45.8**	**70.3**	**49.7**

Table 26: **Full Table of Object detection and instance segmentation results on the COCO dataset with MR-CNN 3 ×.** The "P" column represents the number of parameters, and "F" for FLOPs.

MODEL	P (M)	F (G)	MASK R-CNN 3×					
			AP^b	AP^b_{50}	AP^b_{75}	AP^m	AP^m_{50}	AP^m_{75}
XCiT-T12/8(Ali et al., 2021)	26	266	44.5	66.4	48.8	40.4	63.5	43.3
RMT-T(Fan et al., 2024)	33	218	47.1	68.8	51.7	42.6	65.8	45.9
MPViT-T(Lee et al., 2022)	28	216	44.8	66.9	49.2	41.0	64.2	44.1
HaLA-T	33	129	**49.4**	**70.7**	**54.3**	**44.1**	**67.7**	**47.7**
InternImage-T(Wang et al., 2023)	49	270	49.1	70.4	54.1	43.7	67.3	47.3
RMT-S(Fan et al., 2024)	46	262	50.7	71.9	55.6	44.9	69.1	48.4
FL-Swin-T(Han et al., 2023)	49	268	46.5	68.5	50.8	42.1	65.4	45.1
VMamba-T(Liu et al., 2024)	50	270	48.9	70.6	53.6	43.7	67.7	46.8
MILA-T(Han et al., 2024b)	44	255	48.8	71.0	53.6	43.8	68.0	46.8
HaLA-S	44	262	**51.3**	**72.3**	**56.4**	**45.4**	**69.6**	**49.0**

Table 27: **Full Table of Semantic segmentation results compared with SOTA methods on ADE20K and Cityscapes benchmarks.** FLOPs (F) is calculated with input size 512×512 for ADE20K and 2,048×1,024 for Cityscapes.

METHOD	PARA	ADE20K MIOU	F
VWFormer-B1(Yan et al., 2024)	14M	44.0	13G
EfficientViT-B2(Cai et al., 2023)	15M	45.9	9.1G
SegFormer-B1(Xie et al., 2021)	14M	42.2	16G
SegNeXt-S(Guo et al., 2022b)	15M	44.3	16G
HaLA-SN-T	14M	**48.8**	14G
VRWKV-S(Duan et al., 2025)	29M	47.2	46G
MambaOut-T(Yu & Wang, 2025)	54M	47.4	-
VWFormer-B2(Yan et al., 2024)	27M	48.1	47G
EfficientViT-B3(Cai et al., 2023)	40M	49.0	22G
EfficientViT-L1(Cai et al., 2023)	40M	49.2	36G
SegFormer-B2(Xie et al., 2021)	28M	46.5	62G
SegNeXt-B(Guo et al., 2022b)	28M	48.5	35G
HaLA-SN-S	24M	**50.9**	25G
SegMan-S(Fu et al., 2025)	29M	51.3	25G
HaLA-SM-S	29M	**51.5**	27G

METHOD	PARA	ADE20K MIOU	F
MambaOut-S(Yu & Wang, 2025)	76M	49.5	-
VWFormer-B4(Yan et al., 2024)	64M	50.8	80G
VMamba-T(Liu et al., 2024)	62M	47.9	-
EfficientViT-L2(Cai et al., 2023)	51M	50.7	45G
SegFormer-B3(Xie et al., 2021)	47M	49.4	45G
SegFormer-B4(Xie et al., 2021)	64M	50.3	96G
SegNeXt-L(Guo et al., 2022b)	49M	51.0	70G
HaLA-SN-B	48M	**52.6**	57G
SegMAN-B(Fu et al., 2025)	52M	52.6	58G
HaLA-SM-B	55M	**53.0**	57G
ViT-B(Dosovitskiy et al., 2021)	100M	48.8	158G
VRWKV-B(Duan et al., 2025)	107M	49.2	146G
MambaOut-B(Yu & Wang, 2025)	112M	49.6	-
VMamba-B(Liu et al., 2024)	122M	51.0	-
Swin-B(Liu et al., 2021)	121M	48.1	261G
VWFormer-B5(Yan et al., 2024)	85M	52.0	96G
SegFormer-B5(Xie et al., 2021)	85M	51.0	110G
HaLA-SN-L	98M	**53.1**	103G
SegMAN-L(Fu et al., 2025)	92M	53.2	97G
HaLA-SM-L	107M	**53.4**	93G

METHOD	PARA	CITYSCAPES MIOU	F
VWFormer-B1(Yan et al., 2024)	14M	80.4	-
EfficientViT-B2(Cai et al., 2023)	15M	82.1	74G
SegFormer-B1(Xie et al., 2021)	14M	78.5	244G
SegNeXt-S(Guo et al., 2022b)	15M	81.3	125G
HaLA-T	14M	**82.5**	110G
VWFormer-B2(Yan et al., 2024)	27M	81.7	415G
EfficientViT-B3(Cai et al., 2023)	40M	83.0	179G
EfficientViT-L1(Cai et al., 2023)	40M	82.7	282G
SegFormer-B2(Xie et al., 2021)	28M	81.0	711G
SegNeXt-B(Guo et al., 2022b)	28M	82.6	276G
HaLA-S	24M	**83.5**	205G