Foveal-Mamba: Inside-Out Ring Scanning with Recurrent Offset Prediction for Visual State Space Models

Yi-Kuan Hsieh
Department of Artificial Intelligence
National Yang Ming Chiao Tung University
Tainan, Taiwan (R.O.C.)
yikuan0725@gmail.com

Kuan-Chuan Peng
Mitsubishi Electric Research Labs (MERL)
Cambridge, MA, USA
kpeng@merl.com

Xin Li
Department of Computer Science
State University of New York at Albany
Albany, NY, USA
xin.li@mail.wvu.edu

Ming-Ching Chang
Department of Electrical & Computer Engineering
State University of New York at Albany
Albany, NY, USA
mingching@gmail.com

Chi-Chia Sun
Department of Electronic & Computer Engineering
National Taiwan University of Science and Technology
Taipei, Taiwan (R.O.C.)
chichiasun@ntust.edu.tw

Yu-Chee Tseng
Department of Artificial Intelligence
National Yang Ming Chiao Tung University
Tainan, Taiwan (R.O.C.)
yc0405@gmail.com

Jun-Wei Hsieh*
Department of Artificial Intelligence
National Yang Ming Chiao Tung University
Tainan, Taiwan (R.O.C.)
jwhsieh@nycu.edu.tw

Abstract

Visual state space models (SSMs) have emerged as efficient alternatives to attention-based backbones, yet their performance remains sensitive to how visual evidence is sampled and propagated. Deformable visual Mamba methods address fixed scanning by predicting adaptive sampling locations, but offset prediction is still driven primarily by local responses, producing locally plausible samples that are weakly coupled to the SSM's recurrent state evolution, particularly in off-center or multi-object scenes. We present **Foveal-Mamba**, a scan-path-aware deformable visual Mamba framework that reformulates offset prediction as recurrent scan-path prediction. Two components drive this design. First, an **Inside-Out Ring Scan** organizes visual tokens along a center-to-periphery path, providing a stable propagation prior that does not assume object centering. Second, a lightweight Mamba module embedded within the offset prediction network forms the **Recurrent Scan-Path Offset Prediction Network (RS-OPN)**, which predicts offsets from hidden states accumulated along the ring path. RS-OPN tightly couples deformable sampling with recurrent state evolution and produces coherent, object-aware sampling trajectories. Foveal-Mamba consistently outperforms strong baselines across multiple

*Corresponding author: jwhsieh@nycu.edu.tw

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

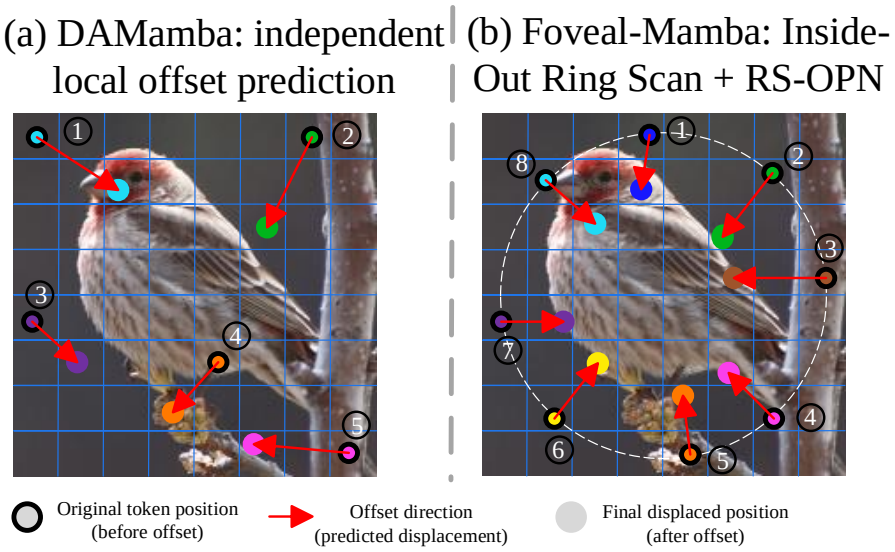


Figure 1: **Motivation of Foveal-Mamba:** (a) DAMamba predicts offsets from local spatial responses, so displaced sampling points are only weakly coupled to the recurrent scan-path context. (b) Foveal-Mamba arranges reference points in an inside-out ring order and uses RS-OPN to predict offsets from accumulated recurrent states. This design encourages the sampling points to move more coherently toward object-centered regions. White circles, colored circles, and red arrows denote original token positions, final displaced sampling positions, and predicted offset directions, respectively.

benchmarks. On ImageNet-1K, Foveal-Mamba-Tiny achieves 84.3% Top-1 accuracy, surpassing DAMamba-Tiny by 0.5% with fewer parameters and FLOPs. On ADE20K semantic segmentation, Tiny/Small variants reach 50.8/51.8 mIoU. On COCO, it achieves 48.9/50.2 box AP and 43.7/44.9 mask AP for object detection and instance segmentation, respectively. Qualitative results further demonstrate earlier object focus and more stable activations in multi-object scenes.

1 Introduction

Extracting geometrically adaptive features is a core goal in visual representation learning. A strong visual module should changes in shape, pose, scale, and spatial layout, while preserving precise spatial information for dense prediction tasks. Deformable Convolutional Networks (DCNs) [4] address this by adding learnable offsets to regular convolutional sampling grids, allowing the receptive field to adapt to geometric variation. The effectiveness of these deformable operators depends heavily on the Offset Prediction Network (OPN), which determines where each sampling point moves. For efficiency, existing OPNs are usually lightweight convolutional predictors that estimate offsets only from local spatial responses. As a result, sampling becomes locally adaptive but remains unaware of how visual evidence will be integrated later in the network.

Recently, State Space Models (SSMs), especially Mamba [7], have emerged as efficient alternatives to self-attention for modeling long-range dependencies with linear complexity. In vision, two-dimensional feature maps are flattened into sequences before recurrent state propagation. Early visual Mamba models, such as Vim [48], VMamba [22], and PlainMamba [41], show that selective SSMs can effectively replace attention-based global modeling. However, these methods rely on predefined scan orders, leaving geometric deformation and recurrent state evolution only loosely connected.

To improve flexibility, recent works has introduced dynamic and deformable scanning. DAMamba [19] adapts sampling locations before state-space modeling, while DefMamba [21] dynamically modifies reference positions and scan behavior through deformable state-space blocks. These methods show that adaptive sampling improves visual Mamba beyond fixed traversal. However, existing methods mainly focus on *where* to sample, while offset prediction itself remains weakly coupled with recurrent SSM dynamics.

This gap is most apparent in object-centric scenes. Natural images often contain off-center objects, partial occlusions, or multiple competing regions. In such cases, deformable sampling should preserve coherent alignment with object structure rathe than simply follow locally strong responses. For example, DAMamba predicts adaptive sampling locations but still arranges sampled patches in raster-scan order before SSM processing. As illustrated in Fig. 1, this improves *where* tokens are sampled, but not *how* the displacement field evolves along an object-aware recurrent path. When multiple objects compete for visual evidence, locally predicted offsets can become scattered instead of forming a coherent object-focused sampling pattern.

In this paper, we argue that offset prediction in visual Mamba should not be treated as independent point-wise regression. Deformable sampling determines what visual evidence is observed, while the SSM determines how that evidence is sequentially integrated. These two processes should therefore be designed jointly. Each offset should depend not only on local visual responses, but also on recurrent context accumulated along a structured scan path. This turns offset prediction from a local displacement problem into a recurrent scan-path prediction problem.

To this end, we propose **Foveal-Mamba**, a scan-path-aware deformable Mamba framework with two tightly coupled components. First, we introduce an **Inside-Out Ring Scan** as a stable geometric prior for recurrent state propagation. The scan begins at central tokens and expands outward toward, providing an ordered path for context accumulation. Second, we propose a **Recurrent Scan-path Offset Prediction Network** (RS-OPN) that embeds a lightweight Mamba module inside the OPN. Local features are organized along the ring path, and a compact selective SSM accumulates scan-path context. The recurrent hidden states are then used to predict offsets. Each offset is therefore conditioned on both local visual evidence and the accumulated recurrent context. Importantly, this structured scan path does not force sampling to remain near the image center. Instead, RS-OPN dynamically adopts the sampling trajectory to input-dependent visual cues, allowing Foveal-Mamba to adapt flexibly toward off-center objects or multiple object regions.

The proposed design differs from existing deformable visual Mamba methods in how offset prediction is coupled with recurrent state evolution. DAMamba [19] changes where each token samples from, but its OPN is still driven by local features and does not explicitly model recurrent state evolution during offset prediction. DefMamba [21] introduces deformable token indexing and dynamic reference-point adjustment, but its offset prediction is not formulated as a recurrent process along a structured scan path. In contrast, Foveal-Mamba jointly redesigns both the scan path and the offset predictor. The Inside-Out Ring Scan provides a structured state-propagation path for state propagation, while RS-OPN predicts offsets conditioned on recurrent context along the same path. This coupling allows deformable sampling to follow sequential state evolution rather than a set of independent local displacement estimates.

The main contributions of this work are as follows:

- **Recurrent scan-path offset prediction:** We show that existing deformable visual Mamba methods adapt sampling locations without explicitly coupling offset prediction with recurrent scan-path context. We reformulate offset prediction as a recurrent scan-path prediction problem, where each offset is conditioned on both local visual cues and accumulated recurrent state.
- **Inside-Out Ring Scan with content-adaptive deformation:** We propose an inside-out ring traversal that provides a stable centric propagation prior without assuming that objects are centered in the image. Combined with RS-OPN, the sampling process can adapt more effectively to off-center objects and multi-object scenes.
- **Recurrent Scan-path Offset Prediction Network (RS-OPN):** We introduce a lightweight Mamba module into the OPN to predict offsets from recurrent hidden states accumulated along the ring path. Compared with convolutional OPNs, RS-OPN explicitly couples deformable sampling with recurrent state evolution, leading to more coherent and structure-aware sampling.
- **Extensive evaluation:** Experiments on ImageNet-1K, ADE20K, and MS COCO show that Foveal-Mamba consistently outperforms strong Mamba baselines, including DAMamba and DefMamba, validating the benefit of jointly designing spatial traversal and recurrent offset prediction.

2 Method

Overview We propose **Foveal-Mamba**, a deformable visual Mamba framework that couples offset prediction with recurrent state evolution. Existing deformable visual Mamba methods improve

fixed feature traversal by adapting sampling locations or reference positions. However, their offset predictors remain largely local: offsets are predicted from local spatial responses and then passed to a state-space module. This creates a mismatch between *where* features are sampled and *how* visual evidence is sequentially accumulated by the SSM. Our key observation is that deformable sampling and state-space propagation should be designed jointly rather than treated as separate operations. Deformable sampling determines which visual evidence enters the model, while the scan path determines how that evidence is accumulated, retained, and propagated. Therefore, offset prediction should depend not only on local visual features, but also on recurrent context accumulated along a structured scan path.

Foveal-Mamba addresses this issue with two coupled designs. First, we introduce an **Inside-Out Ring Scan**, which organizes visual tokens from the geometric center outward to peripheral rings. This traversal provides a stable centric spatial prior and gives the recurrent state a structured path for progressive context accumulation. Second, we propose a **Recurrent Scan-path Offset Prediction Network (RS-OPN)**, which embeds a lightweight selective SSM inside the OPN. Instead of predicting each offset independently from local spatial responses, RS-OPN predicts offsets from hidden states accumulated along the ring path. This turns offset prediction from local displacement regression into a recurrent scan-path prediction.

Preliminaries A selective State Space Model (SSM) processes a sequence by recurrently updating a hidden state. Given a token sequence $\{x_t\}_{t=1}^{L}$, where $x_t \in \mathbb{R}^C$ is the input token at step t, let $h_t \in \mathbb{R}^N$ denote the hidden state, $y_t \in \mathbb{R}^C$ the output token, where N is the state dimension and $\bar{\mathbf{A}}_t$, $\bar{\mathbf{B}}_t$, $\mathbf{C}_t$, and $\mathbf{D}$ the state-space parameters. The recurrent update is:

$$h_t = \bar{\mathbf{A}}_t h_{t-1} + \bar{\mathbf{B}}_t x_t, \qquad y_t = \mathbf{C}_t h_t + \mathbf{D} x_t. \tag{1}$$

Mamba makes these dynamics input-dependent and implements them efficiently, enabling linear-complexity sequence modeling.

For visual inputs, a feature map $\mathbf{X} \in \mathbb{R}^{B\times C\times H\times W}$ is serialized into a token sequence before SSM propagation, where B is the batch size, C is the channel dimension, and H, W denote spatial resolution. In Foveal-Mamba, the same ring path is used for both offset prediction and main SSM feature mixing, so that deformable sampling and recurrent aggregation follow a shared scan path.

Deformable sampling augments a regular sampling grid with learnable offsets, allowing the model to adapt to geometric variations. This principle follows the core idea of Deformable Convolutional Networks, where offsets are learned from downstream tasks without extra supervision. In our framework, offsets are predicted differently: instead of relying only on local convolutional responses, RS-OPN predicts offsets from recurrent states accumulated along the scan path.

Inside-Out Ring Scan The first design choice of Foveal-Mamba is to replace unstructured flattening with a deterministic center-to-periphery traversal. This design does not assume radial symmetric, but provides a stable geometric anchor: central tokens are processed first to build compact context, and outer regions are gradually incorporated. Given a feature map of size $H \times W$ with center $\mathbf{c} = ((H-1)/2, (W-1)/2)$, we compute the distance of each position $p_i = (y_i, x_i)$ to the center and assign a ring level using the L_∞ distance. Tokens with smaller ring levels are processed earlier. Within each ring, tokens are ordered clockwise by angle. Let $r(p_i)$ denote the L_∞ ring distance, $\theta(p_i)$ the clockwise angular order. The final ordering is defined by:

$$\kappa(p_i) = \alpha r(p_i) + \theta(p_i), \tag{2}$$

where $\alpha > 2\pi$ ensures that ring level has higher priority than angular order. Sorting by $\kappa(p_i)$ yields the ring sequence $\mathcal{I}_{ring}$, and its inverse $\mathcal{I}_{inv}$ restores the original spatial layout. Since $\mathcal{I}_{ring}$ and $\mathcal{I}_{inv}$ depend only on resolution, they are precomputed as lookup tables and reused during training and inference, avoiding repeated construction of the scan path.

Recurrent Scan-path Offset Prediction Network A key limitation of existing deformable visual Mamba methods is that offsets are predicted via local spatial regression. This is suboptimal for SSM-based models, where sampled features are later aggregated through sequential state propagation. Without scan-path context, offsets may be locally reasonable but globally inconsistent.

As shown in Fig. 2, the proposed **Recurrent Scan-path Offset Prediction Network (RS-OPN)** addresses this by using the inside-out ring path for both token ordering and context modeling.

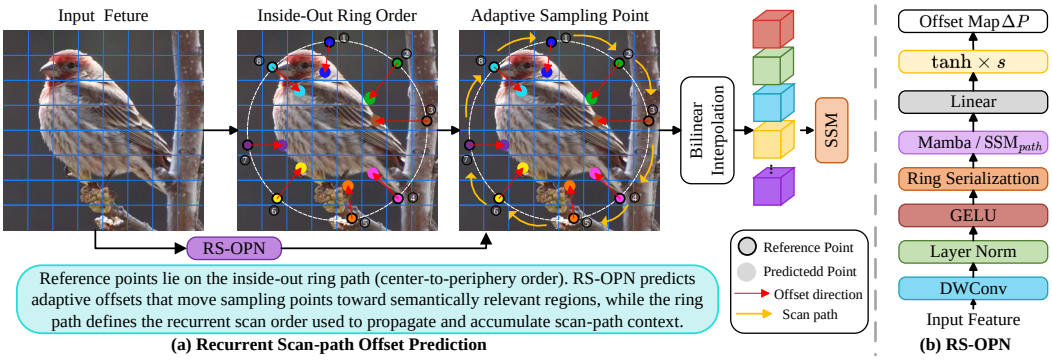


Figure 2: **Recurrent scan-path offset prediction in Foveal-Mamba:** (a) Reference points are organized in an inside-out ring order, and RS-OPN predicts adaptive offsets from recurrent scan-path states. The ring path defines the state propagation order, while the predicted offsets shift sampling locations toward semantically relevant regions before bilinear interpolation and SSM processing. (b) RS-OPN extracts local features, serializes them along the ring path, applies a lightweight Mamba/SSM module, and predicts the offset map ΔP from the resulting recurrent hidden states.

Given an input feature map, RS-OPN first extracts local features using a lightweight depth-wise convolutional encoder, then serializes them in ring order. A compact selective SSM is applied to aggregate scan-path context, which is used to predict offsets.

For the t-th ring token, let h_t denote the recurrent hidden state, $\mathrm{LN}(\cdot)$ denote layer normalization, ϕ_{off} denote the linear offset head, s denote the offset scale, and $\Delta p_t \in \mathbb{R}^{2GK_o}$ denote the predicted offset vector, where G is the number of deformable groups and K_o is the number of sampling points per group. Before offset regression, the hidden state h_t is first normalized by layer normalization and then passed to the offset head. The offset is computed as:

$$\Delta p_t = s \cdot \tanh\left(\phi_{off}(\mathrm{LN}(h_t))\right), \tag{3}$$

where $\tanh(\cdot)$ bounds the offset magnitude for stable deformable sampling.

The predicted ring-ordered offsets are first mapped back to the original spatial layout using $\mathcal{I}_{inv}$ and then applied in DCNv3-style deformable sampling. As the modulation mask is fixed to ones in our implementation, RS-OPN focuses solely on offset prediction. Compared with a convolutional OPN, RS-OPN replaces local feature-driven prediction with scan-path-aware hidden states. As a result, offsets are generated from accumulated contextual information along the ring path rather than from local responses alone. Thus, the ring path defines the recurrent propagation order, while the predicted offsets determine the final adaptive sampling locations.

Path Regularization Since RS-OPN predicts offsets along a structured sequence, we optionally apply regularization to the ring-ordered offsets. Let $\Delta p_l \in \mathbb{R}^{D_o}$ denote the offset vector of the l-th ring token, where $D_o = 2GK_o$ is the offset dimension. The first-order smoothness loss encourages smooth changes between adjacent tokens along the scan path:

$$\mathcal{L}_{path} = \frac{1}{L-1} \sum_{l=2}^{L} \|\Delta p_l - \Delta p_{l-1}\|_1 . \tag{4}$$

We further define a curvature loss:

$$\mathcal{L}_{curv} = \frac{1}{L-2} \sum_{l=3}^{L} \|\Delta p_l - 2\Delta p_{l-1} + \Delta p_{l-2}\|_1 . \tag{5}$$

The first two terms regularize the *variation* of the offset trajectory. We use ℓ_1 penalties for these terms because they suppress abrupt local changes while being less sensitive to occasional large variations.

In addition, we introduce a magnitude regularization term:

$$\mathcal{L}_{mag} = \frac{1}{L} \sum_{l=1}^{L} \frac{1}{D_o} \|\Delta p_l\|_2^2 . \tag{6}$$

Table 1: Performance comparison on ImageNet-1K. All images are resized to 224×224. Throughput (TP) is measured on an Nvidia A100 GPU.

Model	Params	GFlops	TP	Top-1
Vim $_{\text{ICML'24}}$ [48]	26 M	5.3 G	811	80.5
VMamba $_{\text{NeurIPS'24}}$ [22]	30 M	5.6 G	1686	82.6
SiMBA $_{\text{arXiv'24}}$ [28]	15.3 M	5.0 G	-	81.7
Zigma $_{\text{ECCV'24}}$ [15]	31 M	5.1 G	-	82.4
QuadMamba $_{\text{NeurIPS'24}}$ [40]	31 M	5.5 G	1252	81.4
LocalMamba $_{\text{ECCV'24}}$ [16]	26 M	5.7 G	-	82.7
FractalMamba $_{\text{AAAI'25}}$ [38]	31 M	4.8 G	-	83.0
Adventurer $_{\text{CVPR'25}}$ [35]	12 M	4.2 G	2757	78.2
SparX-Mamba $_{\text{AAAI'25}}$ [27]	27 M	5.2 G	1370	83.5
EfficientVMamba $_{\text{AAAI'25}}$ [29]	33 M	4.0 G	-	81.8
PlainMamba $_{\text{BMVC'24}}$ [41]	25 M	8.1 G	-	81.6
GroupMamba $_{\text{CVPR'25}}$ [31]	23 M	7.0 G	803	83.3
VSSD $_{\text{ICCV'25}}$ [32]	14 M	4.5 G	-	82.5
DefMamba $_{\text{CVPR'25}}$ [21]	26 M	4.8 G	-	83.5
MaIR $_{\text{CVPR'25}}$ [18]	26 M	5.4 G	-	83.1
DAMamba-Tiny $_{\text{NeurIPS'25}}$ [19]	26 M	4.8 G	1018	83.8*
Foveal-Mamba-Tiny	25 M	4.6 G	872	84.3 (+0.5)

Model	Params	GFlops	TP	Top-1
VMamba $_{\text{NeurIPS'24}}$ [22]	50 M	11.2 G	877	83.6
SiMBA $_{\text{arXiv'24}}$ [28]	40 M	9.0 G	-	84.7
QuadMamba $_{\text{NeurIPS'24}}$ [40]	50 M	9.3 G	582	83.8
LocalVMamba $_{\text{ECCV'24}}$ [16]	50 M	11.4 G	-	83.7
FractalMamba $_{\text{AAAI'25}}$ [38]	54 M	9.6 G	-	83.6
Adventurer $_{\text{CVPR'25}}$ [35]	44 M	8.3 G	1405	81.8
SparX-Mamba $_{\text{AAAI'25}}$ [27]	47 M	9.3 G	-	84.2
PlainMamba $_{\text{BMVC'24}}$ [41]	50 M	14.4 G	-	82.3
GroupMamba $_{\text{CVPR'25}}$ [31]	57 M	14.0 G	-	84.5
MambaOut $_{\text{CVPR'25}}$ [42]	48 M	9.0 G	-	84.1
QFormer$_h$ $_{\text{TPAMI'23}}$ [44]	51 M	8.9 G	-	84.0
PartialFormer $_{\text{ACL'24}}$ [45]	64 M	6.8	-	83.9
BiFormer $_{\text{CVPR'23}}$ [47]	57 M	9.8 G	-	84.3
VSSD $_{\text{ICCV'25}}$ [32]	40 M	7.4 G	-	84.1
DefMamba $_{\text{CVPR'25}}$ [21]	50 M	8.5 G	-	84.2
DAMamba-Small $_{\text{NeurIPS'25}}$ [19]	45 M	10.3 G	595	84.8*
Foveal-Mamba-Small	44 M	10.2 G	482	85.2 (+0.4)

Note: The symbol * indicates that the reported result could not be reproduced under our training setting.

Unlike $\mathcal{L}_{path}$ and $\mathcal{L}_{curv}$, this term regularizes the *absolute magnitude* of the predicted offsets. We use an ℓ_2 penalty here because its quadratic growth more strongly discourages excessively large displacements, preventing sampling points from drifting too far from their reference positions. Although ℓ_1 and ℓ_2 penalties have different numerical scales, their contributions are balanced by the weighting factors λ_p, λ_c, and λ_m.

The overall objective is:

$$\mathcal{L} = \mathcal{L}_{task} + \lambda_p \mathcal{L}_{path} + \lambda_c \mathcal{L}_{curv} + \lambda_m \mathcal{L}_{mag}, \tag{7}$$

where $\mathcal{L}_{task}$ is the task loss, and λ_p, λ_c, and λ_m control the relative strengths of path smoothness, curvature, and magnitude regularization, respectively. This regularization is optional and encourages smoother scan-path deformation.

Ring-Ordered SSM Mixer The same ring path is also used in the main token mixer. Given the sampled feature map $\tilde{\mathbf{X}}$, we flatten and reorder it using $\mathcal{I}_{ring}$: $\tilde{\mathbf{S}}_{ring} = \text{Flatten}(\tilde{\mathbf{X}})[\mathcal{I}_{ring}]$ and $\tilde{\mathbf{S}}_{ring} \in \mathbb{R}^{B\times L\times C}$. The sequence is then processed by the main SSM as $\mathbf{Y}_{ring} = \text{SSM}_{main}\left(\tilde{\mathbf{S}}_{ring}\right)$ and $\mathbf{Y}_{ring} \in \mathbb{R}^{B\times L\times C}$. Finally, the output is mapped back to the spatial layout as $\mathbf{Y} = \text{Reshape}\left(\mathbf{Y}_{ring}[\mathcal{I}_{inv}]\right)$ and $\mathbf{Y} \in \mathbb{R}^{B\times C\times H\times W}$. This design aligns the offset prediction path with the main feature mixing path, allowing deformable sampling and recurrent aggregation to follow a shared structured traversal.

3 Experiment Results

Implementation details: Our models are trained from scratch using the AdamW optimizer [26] over 300 epochs with batch size of 128. We apply a 5-epoch linear warm-up, followed by a cosine learning rate schedule [25] with an initial learning rate of 5×10^{-4}. The optimizer uses a momentum of 0.9. For fair comparison, we follow prior work [34] and adopt standard data augmentations, including mixup [43], random erasing [46], and AutoAugment [3]. Throughput is measured on a single Nvidia A100 GPU. We compare our method against 16 Vision Mamba variants listed in Tab. 1.

3.1 Image Classification on ImageNet-1K

Experimental settings: We conducted image classification experiments on ImageNet-1K [30], which contains 1,281,167 training images and 50,000 validation images across 1,000 categories. Our implementation is based on PyTorch and Timm libraries. For fair comparison, we follow the standard

Table 2: Comparison of semantic segmentation on ADE20K using UPerNet. FLOPs are computed at an input resolution of 512×2048. SS and MS denote single-scale and multi-scale testing, respectively.

Method	SS	MS	Params	FLOPs
UniRepLKNet-T $_{\text{CVPR'24}}$ [5]	48.6	49.1	61 M	946 G
ConvNeXt-T $_{\text{CVPR'22}}$ [24]	46.0	46.7	60 M	939 G
Swin-T $_{\text{ICCV'21}}$ [23]	44.4	45.8	60 M	945 G
Agent-Swin-T $_{\text{ECCV'24}}$ [12]	46.7	-	61 M	954 G
NAT-T $_{\text{CVPR'23}}$ [13]	47.1	48.4	58 M	934 G
QFormer$_h$-T $_{\text{TPAMI'23}}$ [44]	46.9	48.1	61 M	-
PartialFormer-B3 $_{\text{ACL'24}}$ [45]	47.0	-	65 M	923 G
BiFormer-S $_{\text{CVPR'23}}$ [47]	49.8	50.8	-	-
MambaOut-T $_{\text{CVPR'25}}$ [42]	47.4	48.6	54 M	938 G
VMamba-T $_{\text{NeurIPS'24}}$ [22]	48.0	48.8	62 M	949 G
LocalVMamba-T $_{\text{ECCVW'24}}$ [17]	47.9	49.1	57 M	970 G
DAMamba-Tiny $_{\text{NeurIPS'25}}$ [19]	50.3	51.2	55 M	937 G
Foveal-Mamba-Tiny	**50.8**	**51.6**	**52 M**	910 G

Method	SS	MS	Params	FLOPs
UniRepLKNet-Small $_{\text{CVPR'24}}$ [5]	50.5	51.0	86 M	1036 G
ConvNeXt-Small $_{\text{CVPR'22}}$ [24]	48.7	49.6	82 M	1027 G
Swin-Small $_{\text{ICCV'21}}$ [23]	47.6	49.5	81 M	1039 G
Agent-Swin-Small $_{\text{ECCV'24}}$ [12]	48.1	-	81 M	1043 G
NAT-Small $_{\text{CVPR'23}}$ [13]	48.0	49.5	82 M	1010 G
QFormer$_h$-Small $_{\text{TPAMI'23}}$ [44]	48.9	50.3	82 M	-
PartialFormer-B3 $_{\text{ACL'24}}$ [45]	48.3	-	95 M	1005 G
BiFormer-Big $_{\text{CVPR'23}}$ [47]	51.0	51.7	-	-
MambaOut-Small $_{\text{CVPR'25}}$ [42]	49.5	50.6	76 M	1032 G
VMamba-Small $_{\text{NeurIPS'24}}$ [22]	50.6	51.2	82 M	1028 G
LocalVMamba-Small $_{\text{ECCVW'24}}$ [17]	50.0	51.0	81 M	1095 G
DAMamba-Small $_{\text{NeurIPS'25}}$ [19]	51.2	52.0	75 M	1050 G
Foveal-Mamba-Small	**51.8**	**52.5**	**71 M**	1013 G

training recipe of DeiT. We use AdamW with cosine learning rate decay and a 20-epoch linear warm-up. All models are trained for 300 epochs at 224×224 resolution. We adopt common data augmentation and regularization strategies, including RandAugment, repeated augmentation, Mixup, CutMix, random erasing, weight decay, label smoothing, and stochastic depth. During evaluation, we apply center cropping to obtain 224×224 inputs. Training is performed on 16 RTX 3090 GPUs. We do not use exponential moving average (EMA) since it does not provide consistent performance gains in our setting.

Result: As shown in Tab. 1, the proposed **Foveal-Mamba-Tiny** consistently outperforms strong CNNs, Vision Transformers, and existing vision SSM baselines across scales. Specifically, **Foveal-Mamba-Tiny** achieves a Top-1 accuracy of **84.3%**, improving over VMamba (82.6%) by **+1.7%** and surpassing the previous leading SSM backbone, DAMamba-Tiny (83.8%), by **+0.5%**. Results for the Small and Big variants are provided in the supplementary material.

Comparison with DAMamba. The most significant improvement is observed compared with DAMamba, which uses a data-driven Dynamic Adaptive Scan (DAS) for deformable sampling. Although DAMamba improves modeling flexibility through its Offset Prediction Network (OPN), its offsets are still predicted mainly from local spatial responses, making the sampling process less aware of global structure in complex scenes. In contrast, **Foveal-Mamba-Tiny** achieves higher accuracy with comparable parameters and FLOPs. This demonstrates the effectiveness of our scan-path-aware design, where recurrent context is propagated along the ring path before predicting offsets. By conditioning offsets on accumulated scan-path states, Foveal-Mamba produces more coherent and object-aware sampling behavior than isolated local regression in DAMamba.

3.2 Semantic Segmentation on ADE20K

Experimental settings: We evaluate semantic segmentation on ADE20K using the UPerNet [39] framework, a widely used segmentation setting for assessing the transferability of visual backbones. Following common practice, the backbone is initialized with ImageNet-1K pretrained weights, while the segmentation head and other newly added layers are randomly initialized. All models are trained for 160K iterations with the AdamW optimizer and a batch size of 12. To ensure a fair comparison, we keep the UPerNet training recipe, data processing pipeline, and evaluation protocol consistent across different backbones. All experiments are implemented with MMSegmentation [2].

Result: As shown in Tab. 2, we evaluate Foveal-Mamba on ADE20K semantic segmentation using the UperNet framework. The proposed method consistently improves segmentation performance across different model scales. At the Tiny scale, **Foveal-Mamba-Tiny** achieves 50.8 mIoU, outperforming VMamba-Tiny (48.0 mIoU) and DAMamba-Tiny (50.3 mIoU). At the Small scale, **Foveal-Mamba-Small** further improves to 51.8 mIoU, surpassing DAMamba-Small (51.2 mIoU). These results show that the proposed scan-path-aware design remains effective as model capacity increases.

Table 3: Comparison of object detection and instance segmentation on COCO using Mask R-CNN. FLOPs are computed at an input resolution of 1280×800.

Backbone	AP^{b}	AP^{b}_{50}	AP^{b}_{75}	AP^{m}	AP^{m}_{50}	AP^{m}_{75}	Params	FLOPs
Swin-Tiny $_{\text{ICCV'21}}$ [23]	42.7	65.2	46.8	39.3	62.2	42.2	48 M	267 G
DAT-Tiny $_{\text{CVPR'22}}$ [37]	44.7	67.6	48.7	42.4	66.1	45.5	42 M	278 G
CSWin-Tiny $_{\text{CVPR'22}}$ [6]	46.7	68.6	51.3	42.2	65.6	45.4	42 M	279 G
ConvNeXt-Tiny $_{\text{CVPR'22}}$ [24]	44.2	66.6	48.8	40.1	63.3	42.8	48 M	262 G
PVTv2-B2 $_{\text{CVMJ'22}}$ [36]	45.3	66.1	49.6	41.1	64.2	44.5	39 M	309 G
QFormer$_h$-Tiny $_{\text{TPAMI'23}}$ [44]	45.9	68.5	50.3	41.5	65.2	44.6	49 M	–
PartialFormer-B3 $_{\text{ACL'24}}$ [45]	45.0	–	–	40.9	–	–	54 M	248 G
BiFormer-Tiny $_{\text{CVPR'23}}$ [47]	47.8	69.8	52.2	43.0	66.8	46.5	–	–
MambaOut-Tiny $_{\text{CVPR'25}}$ [42]	45.1	67.3	49.3	41.0	64.1	44.1	43 M	262 G
VMamba-Tiny $_{\text{NeurIPS'24}}$ [22]	47.3	69.3	52.0	42.7	66.4	45.9	50 M	271 G
LocalVMamba-Tiny $_{\text{ECCVW'24}}$ [17]	46.7	68.7	51.2	42.2	65.9	45.5	41 M	296 G
FractalMamba-Tiny $_{\text{AAAI'25}}$ [38]	46.8	68.7	50.8	42.4	65.9	45.8	45 M	266 G
DAMamba-Tiny $_{\text{NeurIPS'25}}$ [19]	48.5	70.3	53.3	43.4	67.2	46.7	45 M	284 G
Foveal-Mamba-Tiny	**48.9**	**70.8**	**53.5**	**43.7**	**67.7**	**47.1**	43 M	258 G
Swin-Small $_{\text{ICCV'21}}$ [23]	44.8	68.6	49.4	40.9	65.3	44.2	69 M	354 G
Agent-Swin-Small $_{\text{ECCV'24}}$ [12]	47.2	69.6	52.3	42.7	66.6	45.8	–	364 G
DAT-Small $_{\text{CVPR'22}}$ [37]	47.1	69.0	51.5	42.5	67.7	45.7	69 M	372 G
CSWin-Small $_{\text{CVPR'22}}$ [6]	47.9	70.1	52.6	43.2	67.4	46.2	54 M	342 G
ConvNeXt-Small $_{\text{CVPR'22}}$ [24]	45.4	67.1	50.7	41.8	65.2	45.1	70 M	348 G
PVTv2-B3 $_{\text{CVMJ'22}}$ [36]	45.7	67.8	50.1	41.7	65.0	44.5	70 M	397 G
BiFormer-Big $_{\text{CVPR'23}}$ [47]	48.6	70.5	53.8	43.7	67.5	47.1	–	–
MambaOut-Small $_{\text{CVPR'25}}$ [42]	48.0	70.1	52.4	43.7	66.4	46.2	65 M	354 G
VMamba-Small $_{\text{NeurIPS'24}}$ [22]	48.7	70.7	53.1	44.3	67.1	47.0	70 M	349 G
LocalVMamba-Small $_{\text{ECCVW'24}}$ [17]	48.4	69.9	52.7	43.3	66.7	46.5	69 M	414 G
DAMamba-Small $_{\text{NeurIPS'25}}$ [19]	49.8	71.2	54.7	44.5	68.4	48.2	65 M	395 G
Foveal-Mamba-Small	**50.2**	**71.7**	**55.2**	**44.9**	**68.7**	**48.8**	62 M	371 G

The improvement is particularly important for dense prediction tasks with large scale variation and cluttered backgrounds, such as ADE20K. Existing deformable visual Mamba methods rely mainly on local offset prediction, which can lead to inconsistent sampling in regions with similar textures or competing semantic cues. In contrast, Foveal-Mamba predicts offsets from accumulated scan-path context, allowing sampling locations to better align with object structure and semantic regions. This leads to more coherent feature aggregation and improved segmentation quality, especially around object boundaries and large-scale regions. These gains are achieved with computational complexity comparable to DAMamba at the corresponding model scales.

3.3 Object Detection and Instance Segmentation on COCO2017

Experimental settings: We conduct object detection and instance segmentation experiments on the COCO 2017 dataset [20], which contains approximately 118K training images and 5K validation images. To evaluate downstream transfer performance, we use Foveal-Mamba as the backbone within the Mask R-CNN [14] framework. The backbone is initialized with ImageNet-1K pretrained weights obtained from 300-epoch pretraining. Following standard settings, models are trained under both the $1\times$ (12 epochs) and $3\times$ (36 epochs) schedules for object detection and instance segmentation. All experiments are implemented using MMDetection [1].

Result: As shown in Tab. 3, we evaluate the transferability of Foveal-Mamba on COCO object detection and instance segmentation using the Mask R-CNN framework. The proposed method consistently improves performance across both Tiny and Small model scales.

Under the standard $1\times$ training schedule, **Foveal-Mamba-Tiny** achieves 48.9 AP^b and 43.7 AP^m, outperforming strong CNN, Vision Transformer, and vision SSM baselines, including DAMamba-Tiny (48.5 AP^b). At the Small scale, **Foveal-Mamba-Small** further improves to 50.2 AP^b and 44.9 AP^m, surpassing DAMamba-Small (49.8 AP^b).

The improvement is particularly evident on the stricter AP^b_{75} metric, where Foveal-Mamba achieves more accurate localization and boundary prediction. Compared with existing deformable visual Mamba methods, the proposed scan-path-aware design predicts offsets from accumulated recurrent

Table 4: Component-wise ablation of Foveal-Mamba on mini-ImageNet. We progressively add Inside-Out Ring Scan, RS-OPN, ring-ordered main SSM mixing, and path regularization. Throughput (TP) is measured on an Nvidia A100 GPU.

Setting	Ring Order in OPN	RS-OPN	Ring-Ordered Main SSM	Path Reg.	TP	Top-1
DAMamba-style OPN	✗	✗	✗	✗	1018	87.48 %
+ Ring order only	✓	✗	✗	✗	1007	87.69 %
+ RS-OPN only	✗	✓	✗	✗	886	87.83 %
+ Ring + RS-OPN	✓	✓	✗	✗	874	88.06 %
+ Ring-ordered main SSM	✓	✓	✓	✗	879	88.21 %
Full Foveal-Mamba	✓	✓	✓	✓	872	88.55 %

context rather than isolated local responses, producing more coherent sampling behavior. This helps reduce sampling drift around complex object boundaries and cluttered regions.

The shared ring-path design further aligns deformable sampling with sequential feature aggregation, leading to more stable object-aware representations for dense prediction tasks. These gains are achieved while maintaining computational complexity comparable to existing deformable visual Mamba backbones.

3.4 Component-wise Ablation of Foveal-Mamba

Table 4 reports the component-wise ablation results on mini-ImageNet. Starting from the DAMamba-style OPN baseline, which achieves 87.48% Top-1 accuracy, adding only the ring order in the OPN improves the result to 87.69%. This moderate gain suggests that the inside-out ring order itself provides a useful structured prior, while introducing little throughput overhead.

Replacing the local offset predictor with RS-OPN brings a larger improvement, increasing Top-1 accuracy to 87.83%. This indicates that predicting offsets from recurrent scan-path states is more effective than relying only on local spatial responses. When the ring order and RS-OPN are combined, the accuracy further increases to 88.06%, showing that the two designs are complementary: the ring order provides a structured recurrent path, while RS-OPN uses this path to accumulate context for offset prediction.

Applying ring-ordered mixing to the main SSM further improves the accuracy to 88.21%. This supports our design choice of aligning the offset prediction path with the main feature propagation path. Finally, the full model with path regularization achieves 88.55%, outperforming the DAMamba-style baseline by 1.07 percentage points. Since path regularization is used only during training, the gain mainly reflects the benefit of encouraging smoother offset trajectories rather than adding extra inference modules. Overall, the ablation confirms that the improvement of Foveal-Mamba comes from the joint design of inside-out scan ordering, recurrent offset prediction, and path-aligned SSM mixing, rather than from a single isolated component.

4 Conclusion

In this paper, we presented **Foveal-Mamba**, a deformable visual Mamba framework that couples deformable sampling with recurrent state evolution through a shared scan-path design. Unlike existing deformable visual Mamba methods that predict offsets mainly from local spatial responses, Foveal-Mamba reformulates offset prediction as a recurrent scan-path prediction problem. The proposed Inside-Out Ring Scan provides a stable center-to-periphery traversal for sequential context propagation, while the Recurrent Scan-path Offset Prediction Network (RS-OPN) predicts offsets from accumulated recurrent states rather than isolated local features. This unified design enables more coherent and object-aware sampling behavior, especially in complex scenes with cluttered backgrounds, off-center objects, and competing semantic regions.

Extensive experiments on ImageNet-1K, ADE20K, and COCO demonstrate the effectiveness of the proposed framework across image classification, semantic segmentation, object detection, and instance segmentation tasks. Foveal-Mamba consistently outperforms strong CNN, Vision Transformer, and recent vision SSM baselines, including DAMamba, while maintaining comparable computa-

tional complexity. These results show that jointly designing scan-path traversal and recurrent offset prediction provides an effective direction for deformable sequential visual modeling.

5 Limitations.

Although Foveal-Mamba improves deformable visual scanning by coupling offset prediction with recurrent scan-path context, several limitations remain. First, the Inside-Out Ring Scan is a fixed geometric prior defined by the feature-map layout rather than explicit object structure, which may be suboptimal for scenes with many small, scattered objects or irregular spatial layouts. Second, the additional sequence reordering and the lightweight SSM inside RS-OPN introduce extra indexing and recurrent computation, which can slightly impact inference efficiency compared to simpler local OPN designs despite similar FLOPs. Third, the path regularization terms rely on fixed weighting coefficients that may not be optimal across different datasets or model scales.

6 Future work

will explore more adaptive scan-path designs that are conditioned on image content or object instances, rather than relying on a fixed center-to-periphery structure. A promising direction is to learn dynamic or multi-center traversal strategies that better reflect complex scene layouts. In addition, improving the efficiency of recurrent offset prediction and developing adaptive regularization schemes could further enhance both the scalability and robustness of deformable visual SSMs.

Acknowledgments and Disclosure of Funding

Kuan-Chuan Peng was exclusively supported by MERL. We thank Xin Li and Ming-Ching Chang for their collaboration. The other authors gratefully acknowledge the assistance provided by the Advanced Technology Diversified Integration Development Project for Unmanned Vehicles at NCSIST.

References

[1] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, et al. Mmdetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019.

[2] MMSegmentation Contributors. MMSegmentation: Openmmlab semantic segmentation toolbox and benchmark. https://github.com/open-mmlab/mmsegmentation, 2020.

[3] Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation strategies from data. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 113–123, 2019.

[4] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 764–773, 2017.

[5] Xiaohan Ding, Yiyuan Zhang, Yixiao Ge, Sijie Zhao, Lin Song, Xiangyu Yue, and Ying Shan. UniRepLKNet: A universal perception large-kernel ConvNet for audio, video, point cloud, time-series and image recognition. In *CVPR*, 2024.

[6] Xiaoyi Dong, Jianmin Bao, Dongdong Chen, Weiming Zhang, Nenghai Yu, Lu Yuan, Dong Chen, and Baining Guo. Cswin transformer: A general vision transformer backbone with cross-shaped windows. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12124–12134, 2022.

[7] Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. In *First Conference on Language Modeling*, 2024.

[8] Albert Gu, Tri Dao, Stefano Ermon, Atri Rudra, and Christopher Ré. Hippo: Recurrent memory with optimal polynomial projections. *Advances in neural information processing systems*, 33:1474–1487, 2020.

[9] Albert Gu, Karan Goel, Ankit Gupta, and Christopher Ré. On the parameterization and initialization of diagonal state space models. *Advances in Neural Information Processing Systems*, 35:35971–35983, 2022.

[10] Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured state spaces. In *ICLR*, 2022.

[11] Ankit Gupta, Albert Gu, and Jonathan Berant. Diagonal state spaces are as effective as structured state spaces. *Advances in neural information processing systems*, 35:22982–22994, 2022.

[12] 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 *ECCV*, 2024.

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

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

[15] Vincent Tao Hu, Stefan Andreas Baumann, Ming Gui, Olga Grebenkova, Pingchuan Ma, Johannes Fischer, and Björn Ommer. Zigma: A dit-style zigzag mamba diffusion model. In *European conference on computer vision*, pages 148–166. Springer, 2024.

[16] Tao Huang, Xiaohuan Pei, Shan You, Fei Wang, Chen Qian, and Chang Xu. Localmamba: Visual state space model with windowed selective scan. In *European Conference on Computer Vision*, pages 12–22. Springer, 2024.

[17] Tao Huang, Xiaohuan Pei, Shan You, Fei Wang, Chen Qian, and Chang Xu. LocalMamba: Visual state space model with windowed selective scan. In *ECCV Workshop*, 2024.

[18] Boyun Li, Haiyu Zhao, Wenxin Wang, Peng Hu, Yuanbiao Gou, and Xi Peng. MaIR: A locality- and continuity-preserving mamba for image restoration. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 7491–7501, 2025.

[19] Tanzhe Li, Caoshuo Li, Jiayi Lyu, Hongjuan Pei, Baochang Zhang, Taisong Jin, and Rongrong Ji. DAMamba: Vision state space model with dynamic adaptive scan. *Advances in neural information processing systems*, 2025.

[20] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755, 2014.

[21] Leiye Liu, Miao Zhang, Jihao Yin, Tingwei Liu, Wei Ji, Yongri Piao, and Huchuan Lu. Defmamba: Deformable visual state space model. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 8838–8847, 2025.

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

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

[24] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *CVPR*, 2022.

[25] Ilya Loshchilov and Frank Hutter. SGDR: Stochastic gradient descent with warm restarts. In *ICLR*, 2017.

[26] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019.

[27] Meng Lou, Yunxiang Fu, and Yizhou Yu. Sparx: A sparse cross-layer connection mechanism for hierarchical vision mamba and transformer networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 19104–19114, 2025.

[28] Badri N Patro and Vijay S Agneeswaran. Simba: Simplified mamba-based architecture for vision and multivariate time series. *arXiv preprint arXiv:2403.15360*, 2024.

[29] Xiaohuan Pei, Tao Huang, and Chang Xu. Efficientvmamba: Atrous selective scan for light weight visual mamba. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 6443–6451, 2025.

[30] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

[31] Abdelrahman Shaker, Syed Talal Wasim, Salman Khan, Juergen Gall, and Fahad Shahbaz Khan. Groupmamba: Efficient group-based visual state space model. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 14912–14922, 2025.

[32] Yuheng Shi, Mingjia Li, Minjing Dong, and Chang Xu. Vssd: Vision mamba with non-causal state space duality. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10819–10829, 2025.

[33] Jimmy TH Smith, Andrew Warrington, and Scott W Linderman. Simplified state space layers for sequence modeling. In *ICLR*, 2023.

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

[35] Feng Wang, Timing Yang, Yaodong Yu, Sucheng Ren, Guoyizhe Wei, Angtian Wang, Wei Shao, Yuyin Zhou, Alan Yuille, and Cihang Xie. Adventurer: Optimizing vision mamba architecture designs for efficiency. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 30157–30166, 2025.

[36] 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(3):1–10, 2022.

[37] Zhuofan Xia, Xuran Pan, Shiji Song, Li Erran Li, and Gao Huang. Vision transformer with deformable attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4794–4803, June 2022.

[38] Haoke Xiao, Lv Tang, Peng-tao Jiang, Hao Zhang, Jinwei Chen, and Bo Li. Boosting vision state space model with fractal scanning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 8646–8654, 2025.

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

[40] Fei Xie, Weijia Zhang, Zhongdao Wang, and Chao Ma. Quadmamba: Learning quadtree-based selective scan for visual state space model. *Advances in Neural Information Processing Systems*, 37:117682–117707, 2024.

[41] Chenhongyi Yang, Zehui Chen, Miguel Espinosa, Linus Ericsson, Zhenyu Wang, Jiaming Liu, and Elliot J Crowley. PlainMamba: Improving non-hierarchical mamba in visual recognition. In *BMVC*, 2024.

[42] Weihao Yu and Xinchao Wang. Mambaout: Do we really need mamba for vision? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4484–4496, 2025.

[43] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *ICLR*, 2018.

[44] Qiming Zhang, Jing Zhang, Yufei Xu, and Dacheng Tao. Vision transformer with quadrangle attention. *TPAMI*, 2023.

[45] Tong Zheng, Bei Li, Huiwen Bao, Jiale Wang, Weiqiao Shan, Tong Xiao, and Jingbo Zhu. Partialformer: Modeling part instead of whole for machine translation. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 7280–7294, 2024.

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

[47] Lei Zhu, Xinjiang Wang, Zhanghan Ke, Wayne Zhang, and Rynson WH Lau. Biformer: Vision transformer with bi-level routing attention. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10323–10333, 2023.

[48] Lianghui Zhu, Bencheng Liao, Qian Zhang, Xinlong Wang, Wenyu Liu, and Xinggang Wang. Vision mamba: Efficient visual representation learning with bidirectional state space model. *arXiv preprint arXiv:2401.09417*, 2024.

[49] Xizhou Zhu, Han Hu, Stephen Lin, and Jifeng Dai. Deformable convnets v2: More deformable, better results. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9308–9316, 2019.

[50] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*, 2020.

A Related Work

A.1 State Space Models and Visual Mamba

State Space Models (SSMs) have recently attracted increasing attention as efficient alternatives to attention-based architectures for long-sequence modeling. Early structured SSMs, such as HiPPO [8] and S4 [10], provide principled mechanisms for modeling long-range dependencies with favorable computational complexity. Subsequent variants further improve the parameterization and efficiency of SSMs through diagonal or structured state matrices, simplified state-space layers, and improved recurrent computation [9, 11, 33]. More recently, Mamba [7] introduces input-dependent selective state-space modeling and hardware-aware recurrent computation, enabling strong sequence modeling performance with linear scaling in sequence length. Its efficiency makes it particularly attractive for high-resolution visual recognition, where quadratic self-attention can become computationally expensive. Mamba is designed to selectively propagate or forget information along the sequence dimension, and its linear sequence-length scaling has made it a strong candidate for replacing attention in long-context modeling.

Inspired by the success of Mamba in sequence modeling, recent works have adapted SSMs to vision backbones. Vim [48] treats image patches as sequential tokens and applies bidirectional state-space modeling for visual recognition. VMamba [22] introduces a 2D Selective Scan (SS2D) module, which unfolds visual features along multiple traversal routes, applies selective scanning, and merges the scanned sequences back into two-dimensional feature maps. PlainMamba [41] further investigates simple visual Mamba backbones for image recognition. These methods demonstrate that SSMs can serve as efficient visual backbones for classification and dense prediction tasks. However, visual SSMs still require converting two-dimensional feature maps into one-dimensional sequences before state propagation. As a result, the interaction between spatial structure, token traversal, and recurrent state evolution becomes a crucial design issue in visual Mamba architectures. VMamba's SS2D is specifically proposed to bridge one-dimensional selective scanning and two-dimensional visual representation by using cross-scan and cross-merge operations.

A.2 Deformable Spatial Modeling

Modeling geometric variations is a long-standing challenge in computer vision. Deformable Convolutional Networks (DCNs) [4] address this problem by augmenting regular convolutional grids with learnable offsets, allowing convolutional sampling locations to adapt to object shapes and spatial transformations. DCNv2 [49] further improves deformable modeling by increasing the use of deformable layers and introducing modulation mechanisms, thereby enhancing the ability of learned sampling supports to focus on relevant image regions. These works establish learnable offset prediction as an effective mechanism for geometric adaptation. Nevertheless, conventional deformable convolutions mainly predict offsets from local convolutional responses, which can limit their ability to coordinate deformation globally or sequentially.

The idea of deformable sampling has also been extended beyond convolution. Deformable DETR [50] replaces dense attention over all spatial locations with sparse deformable attention around reference points. Instead of attending to the entire feature map, each query attends to a small set of learned sampling points, reducing computational cost and improving convergence for object detection. This line of work shows that learnable sampling offsets are useful not only for convolutional operators but also for global modeling modules. However, in both deformable convolution and deformable attention, offset prediction is typically formulated as spatial displacement estimation from query or local features. The offset predictor itself does not explicitly model how displacements should evolve along a recurrent state-propagation path.

A.3 Dynamic and Deformable Scanning in Visual SSMs

As visual Mamba models rely on sequential state propagation, recent works have begun to incorporate deformable mechanisms into SSM-based vision backbones. DefMamba [21] proposes deformable visual state-space blocks that dynamically adjust reference positions and scanning behavior through learned offsets, enabling the model to better focus on informative visual regions. DAMamba [19] introduces Dynamic Adaptive Scan (DAS), where an Offset Prediction Network predicts coordinate offsets and samples features from adaptive locations before feeding them into the SSM. These methods show that deformable scanning can improve the flexibility of visual SSMs while preserving the efficiency benefits of Mamba-based architectures. DAMamba reports strong performance as a general visual backbone across image classification, object detection, instance segmentation, and semantic segmentation, while DefMamba explicitly aims to dynamically adjust scanning paths and reference points for more effective visual state-space modeling.

Despite their effectiveness, existing dynamic and deformable scanning methods mainly focus on adapting where features are sampled or how reference positions are shifted. The offset prediction process itself remains largely local or spatially formulated, and is not explicitly designed to model recurrent state evolution along the scan path. In other words, current methods improve the flexibility of visual SSMs by deforming sampling locations, but they do not fully exploit the sequential nature of SSMs inside the offset predictor. This leaves an important gap: in visual Mamba, offset prediction should not only decide spatial displacements, but should also account for the accumulated scan-path context that governs how visual evidence is sequentially integrated.

Our work addresses this gap by introducing a Recurrent Scan-path Offset Prediction Network (RS-OPN). Unlike convolutional OPNs that predict offsets from local responses, RS-OPN embeds a lightweight Mamba module inside the OPN to predict offsets from recurrent hidden states accumulated along an inside-out ring path. Therefore, deformable sampling is explicitly coupled with state-space propagation, turning offset prediction from local displacement regression into scan-path-conditioned recurrent prediction.

B Architecture Design of Foveal-Mamba

As shown in Fig. 3, we instantiate the proposed scan-path-aware design in a standard four-stage hierarchical architecture. The purpose of this architecture is not to introduce a new backbone family, but to provide a consistent evaluation framework for studying how recurrent scan-path offset prediction affects visual representation learning. Given an input image $\mathbf{I} \in \mathbb{R}^{H\times W\times 3}$, an overlapping patch partition module first converts the image into a two-dimensional feature map with spatial resolution $\frac{H}{4} \times \frac{W}{4}$. The feature map is then processed by four consecutive stages, where the spatial

Table 5: Configurations for different Foveal-Mamba variants.

Model	Channels	Blocks
Foveal-Mamba-Tiny	[80, 160, 320, 512]	[3, 4, 12, 5]
Foveal-Mamba-Small	[96, 192, 384, 512]	[4, 8, 20, 6]

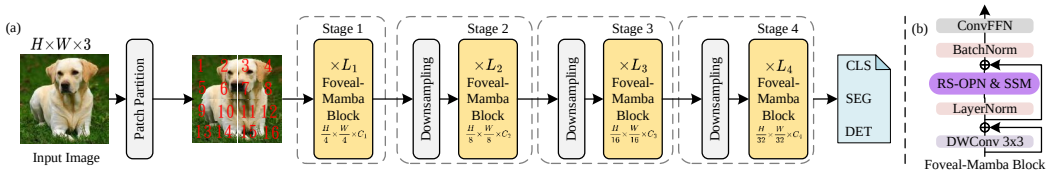


Figure 3: Overview of **Foveal-Mamba**. (a) The backbone adopts a four-stage hierarchical architecture for multi-resolution visual representation learning. (b) The Foveal-Mamba block couples RS-OPN and ring-ordered SSM mixing. RS-OPN serializes locally encoded features with an inside-out ring path and predicts offsets from recurrent scan-path states. The offsets are restored to the spatial layout for deformable sampling, after which the sampled features are mixed by an SSM along the same ring-ordered path.

resolution is progressively reduced by a factor of two and the channel dimension is increased to form a multi-scale representation.

Each stage consists of several stacked Foveal-Mamba blocks. As illustrated in Fig. 3(b), each block follows a simple residual design composed of local convolutional encoding, normalization, the proposed scan-path-aware SSM module, and a convolutional feed-forward network. The local depth-wise convolution provides lightweight spatial encoding before sequential modeling, while the RS-OPN predicts deformable offsets from recurrent states accumulated along the inside-out ring path. The sampled features are then mixed by the main SSM using the same ring-ordered traversal, aligning the offset prediction path with the feature propagation path.

For classification, the final stage output is normalized and aggregated by global average pooling before being passed to a linear classification head. For dense prediction tasks such as semantic segmentation, object detection, and instance segmentation, the stage-wise feature maps are used as multi-scale representations and integrated into standard downstream frameworks, such as UPerNet and Mask R-CNN. The model configurations used in our experiments are summarized in Table **??**.

C Ablation study

C.1 Effect of Offset Predictor Design

Table 6 compares different offset predictor designs under the same deformable sampling operator and training setting. The MLP OPN obtains the lowest accuracy, suggesting that removing local spatial encoding weakens offset prediction. Adding a local encoder improves the result, as shown by the Conv OPN and the DWConv + LN + GELU + Linear OPN, which achieve 87.12% and 87.48% Top-1 accuracy, respectively. This confirms that local spatial cues remain important for predicting meaningful offsets.

More importantly, introducing recurrent modeling inside the OPN brings a clear improvement. Raster Mamba-in-OPN increases the accuracy to 88.09%, showing that conditioning offsets on sequential hidden states is more effective than using local responses alone. When the recurrent path is changed from raster order to the proposed inside-out ring order, the accuracy further improves to 88.55%. This indicates that the gain does not simply come from adding a Mamba module into the OPN, but also from using a structured scan path that better matches the intended recurrent offset prediction process. Overall, these results support the core design of RS-OPN: offset prediction benefits from both local feature encoding and recurrent scan-path context.

Table 6: Ablation study of different Offset Prediction Network designs. All variants use the same deformable sampling operator and training setting, while only the offset predictor is changed.

OPN Design	Local Encoder	Sequence Path	Recurrent State	Top-1
Conv OPN	✓	–	✗	87.12 %
DWConv + LN + GELU + Linear	✓	–	✗	87.48 %
MLP OPN	✗	–	✗	86.92 %
Raster Mamba-in-OPN	✓	Raster	✓	88.09 %
Ring Mamba-in-OPN / RS-OPN	✓	Inside-Out Ring	✓	88.55 %

Table 7: Ablation study of scan paths used in RS-OPN. All variants use the same Mamba-in-OPN design, while only the recurrent token order for offset prediction is changed. SS denote single-scale testing.

Scan Path in RS-OPN	Description	Top-1	ADE20K SS mIoU	COCO AP^b
Raster	top-left to bottom-right	88.09 %	50.3	48.5
Reverse Raster	bottom-right to top-left	88.04 %	50.2	48.4
Serpentine	alternating row direction	88.15 %	50.4	48.5
Outside-In Ring	periphery to center	88.23 %	50.4	48.5
Inside-Out Ring	center to periphery	88.55 %	50.8	48.9

C.2 Effect of Scan Path in RS-OPN

Table 7 studies how different recurrent token orders affect RS-OPN. All variants use the same Mamba-in-OPN design, and only the scan path for offset prediction is changed. Raster and reverse raster achieve similar results, with 88.09% and 88.04% Top-1 accuracy, respectively, suggesting that simply reversing the traversal direction does not provide a meaningful improvement. Serpentine scanning slightly improves the result to 88.15%, indicating that better local continuity can help offset prediction to some extent.

Ring-based orders perform better than row-based traversals. The outside-in ring order reaches 88.23% Top-1 accuracy, showing that organizing tokens by radial structure is beneficial for recurrent offset prediction. The proposed inside-out ring order achieves the best performance across all evaluated tasks, with 88.55% Top-1 accuracy, 50.8 ADE20K SS mIoU, and 48.9 COCO AP^b. Compared with raster scanning, it improves Top-1 accuracy by 0.46 percentage points, ADE20K SS mIoU by 0.5, and COCO AP^b by 0.4. These results suggest that the inside-out ring order provides a more effective recurrent context path for RS-OPN, allowing offset prediction to benefit from structured scan-path context rather than relying on arbitrary token traversal.

C.3 Effect of Path Regularization

Table 8 examines the effect of the optional path regularization terms. Without regularization, the model achieves 88.21% Top-1 accuracy. Adding the first-order smoothness term improves the result to 88.32%, suggesting that encouraging adjacent ring-ordered offsets to change more smoothly can benefit recurrent offset prediction. Further adding the curvature term increases the accuracy to 88.39%, indicating that penalizing abrupt trajectory changes provides an additional, though moderate, improvement.

The full setting, which combines smoothness, curvature, and magnitude penalties, achieves the best accuracy of 88.55%. This suggests that constraining both the variation and the overall magnitude of predicted offsets can provide a useful training constraint for RS-OPN. Importantly, these regularization terms are applied only during training and do not introduce additional inference-time modules. Therefore, the observed improvement is attributed to a better training constraint for recurrent offset prediction rather than increased inference complexity.

Table 8: Effect of path regularization on mini-ImageNet. We progressively add first-order smoothness, curvature, and magnitude penalties to examine their influence on classification accuracy.

Setting	$\mathcal{L}_{path}$	$\mathcal{L}_{curv}$	$\mathcal{L}_{mag}$	Top-1
w/o regularization	✗	✗	✗	88.21 %
+ path only	✓	✗	✗	88.32 %
+ path + curvature	✓	✓	✗	88.39 %
+ path + curvature + magnitude	✓	✓	✓	88.55 %

Algorithm 1 Recurrent Scan-path Offset Prediction in Foveal-Mamba

Require: Input feature map $\mathbf{X} \in \mathbb{R}^{B\times C\times H\times W}$; feature map to be sampled $\mathbf{V} \in \mathbb{R}^{B\times H\times W\times C}$; cached ring index $\mathcal{I}_{ring} \in \mathbb{N}^L$; cached inverse index $\mathcal{I}_{inv} \in \mathbb{N}^L$, where $L = HW$

Ensure: Sampled feature map $\tilde{\mathbf{X}} \in \mathbb{R}^{B\times C\times H\times W}$

$\mathbf{X}_{local} \leftarrow \mathrm{GELU}(\mathrm{Norm}(\mathrm{DWConv}(\mathbf{X})))$ {Extract local evidence before recurrent offset prediction}

$\mathbf{S}_{row} \leftarrow \mathrm{Flatten}(\mathbf{X}_{local})$ {$\mathbf{S}_{row} \in \mathbb{R}^{B\times L\times C}$}

$\mathbf{S}_{ring} \leftarrow \mathbf{S}_{row}[\mathcal{I}_{ring}]$ {Inside-out ring serialization}

$\mathbf{Z} \leftarrow \mathbf{S}_{ring}\mathbf{W}_{in}$ {$\mathbf{W}_{in} \in \mathbb{R}^{C\times C_p}$ projects features to path dimension C_p}

$\mathbf{H}_{ring} \leftarrow \mathrm{SSM}_{path}(\mathbf{Z})$ {Accumulate recurrent scan-path context}

$\Delta\mathbf{P}_{ring} \leftarrow s \cdot \tanh(\phi_{off}(\mathrm{LN}(\mathbf{H}_{ring})))$ {$\Delta\mathbf{P}_{ring} \in \mathbb{R}^{B\times L\times 2GK_o}$}

$\Delta\mathbf{P}_{row} \leftarrow \Delta\mathbf{P}_{ring}[\mathcal{I}_{inv}]$ {Restore offsets from ring order to row-major order}

$\Delta\mathbf{P} \leftarrow \mathrm{Reshape}(\Delta\mathbf{P}_{row})$ {$\Delta\mathbf{P} \in \mathbb{R}^{B\times H\times W\times 2GK_o}$}

$\mathbf{M} \leftarrow \mathbf{1}^{B\times H\times W\times G}$ {Fixed modulation mask}

$\tilde{\mathbf{X}} \leftarrow \mathrm{DCNv3}(\mathbf{V}, \Delta\mathbf{P}, \mathbf{M})$ {Deformable sampling with recurrently predicted offsets}

return $\tilde{\mathbf{X}}$

C.4 Qualitative Analysis

Fig. 4 compares the stage-wise visual responses of DAMamba and our Foveal-Mamba. A clear difference appears in the early stage. At Stage 1, DAMamba often produces scattered responses over background textures or only partially covers object regions. In contrast, our method already concentrates responses on semantically meaningful object structures, such as the body contours of jellyfish and the clustered shapes of ants. This indicates that Foveal-Mamba can guide feature sampling toward object-related regions earlier in the hierarchy, rather than waiting for deeper stages to gradually suppress background noise.

The advantage becomes more visible in multi-object scenes. When several objects appear simultaneously, DAMamba tends to activate only a subset of instances or distribute responses irregularly across the image. Our method produces more consistent activations over multiple objects, suggesting that the proposed RS-OPN does not simply chase local high-frequency edges, but uses the accumulated scan-path context to form a more object-aware sampling trajectory. This behavior is consistent with our design motivation: the inside-out ring path provides a structured propagation scaffold, while recurrent offset prediction dynamically deforms the sampling locations toward salient object regions. As a result, Foveal-Mamba shows stronger early object localization and better multi-instance focus than DAMamba.

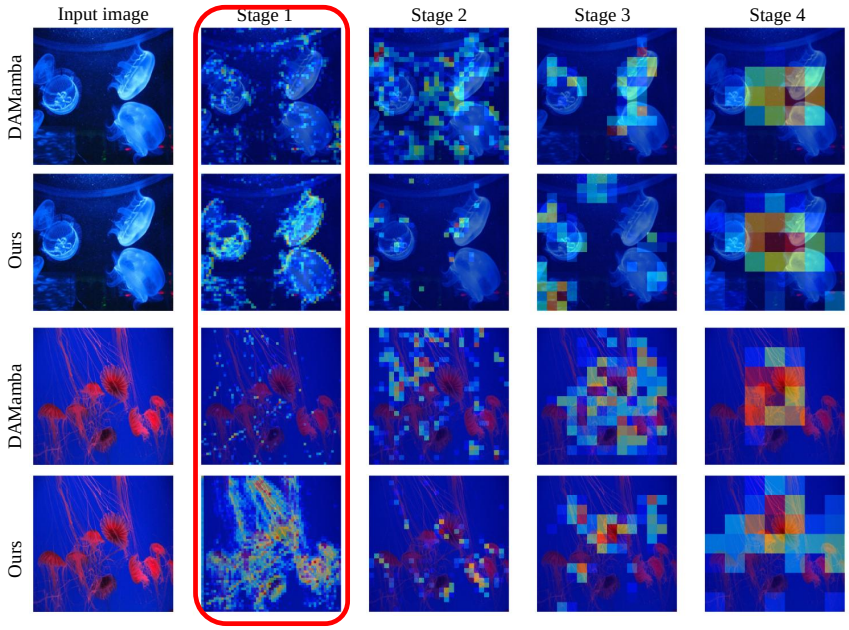


Figure 4: Qualitative visualization of hierarchical object focus across different stages. We compare DAMamba and the proposed Foveal-Mamba from Stage 1 to Stage 4 on several object-centric and multi-object scenes. Compared with DAMamba, our method activates object-related regions at an earlier stage and maintains more consistent responses across multiple instances. In particular, Foveal-Mamba already highlights the main object contours and multi-object regions in Stage 1, while DAMamba tends to produce scattered responses or delayed object concentration. This suggests that the proposed recurrent scan-path offset prediction enables more object-aware feature sampling and more coherent visual evidence aggregation across stages.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes], [No], or [N/A].
- [N/A] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for [N/A]).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will also be asked to include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While [Yes] is generally preferable to [No], it is perfectly acceptable to answer [No] provided a proper justification is given (e.g., error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering [No] or [N/A] is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Please see abstract and introduction.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Please see limitations.

Guidelines:

- The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

   Justification: [N/A]

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Please see methodology and experiments.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.

- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: [N/A]

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).

- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Please see section 3.1, 3.2, 3.3, 3.4 and appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: We use the common result report on ImageNet, COCO datasets and ADE20K datasets, which does not include error bars.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Please see experiments.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.

- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: Our paper did not deviate from the NeurIPS Code of Ethics

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: There is no societal impact of the work performed.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper poses no such risks.

    Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: Please see experiments.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: The paper does not release new assets.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: LLM has little to do with our research.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.