REPSCAN: REPRESENTATION-ADAPTIVE SCANNING FOR VISUAL STATE SPACE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

High-resolution visual representation learning is hindered by the quadratic complexity of Vision Transformers and the rigid scanning strategies used by existing efficient architectures, which employ uniform token processing or fixed scanning orders that fail to adapt to the highly non-uniform distribution of semantic information in real-world images. To address this, we propose RepScan, a representation-adaptive visual scanning framework that dynamically constructs input-dependent serialization orders. RepScan first introduces Representation Relation Modeling to characterize directed token relations by jointly considering representation affinity and predictive discrepancy. Based on these relations, Adaptive Scan Initialization selects a content-dependent starting token with strong outgoing relational support, while Adaptive Scan Ordering determines the subsequent traversal according to the remaining representation structure. A differentiable permutation relaxation further enables end-to-end learning of the discrete scanning process. RepScan preserves the efficient state-space propagation of the underlying visual SSM while introducing a sparse, input-dependent scan-construction process for adaptive serialization. Extensive experiments across classification, detection, and segmentation demonstrate that RepScan achieves a favorable accuracy–efficiency trade-off across diverse visual recognition tasks.

1 INTRODUCTION

Visual representation learning, which aims to extract structured and task-relevant information from raw pixels, remains a cornerstone of modern computer vision. Vision Transformers (ViTs) (Dosovitskiy, 2020; Vaswani et al., 2017) have achieved remarkable performance across image classification, object detection, and semantic segmentation by modeling long-range dependencies through self-attention. However, the computational complexity of self-attention grows quadratically with the number of image tokens (Dong et al., 2022; Liu et al., 2021; 2024; Wu et al., 2024), making high-resolution visual modeling computationally expensive. More recently, visual state-space models (VSSMs), particularly Mamba-based architectures (Zubic et al., 2024; Zhu et al., 2024; Gu & Dao, 2024), have emerged as an efficient alternative by modeling visual tokens with linear complexity. Despite their efficiency, existing VSSMs typically rely on predefined scanning patterns, such as raster, bidirectional, or cross scanning, to serialize 2D visual representations into 1D sequences. Such designs impose the same scanning topology on different inputs and therefore cannot adapt to content-dependent representation structures.

In practice, semantic information in real-world images is highly non-uniform and input-dependent. Informative regions, such as object parts, structural boundaries, and lesions, often occupy only a subset of spatial locations. Moreover, semantic dependencies between tokens do not necessarily follow spatial proximity: spatially distant tokens may exhibit strong semantic correlations, while neighboring tokens may contain redundant or weakly relevant information. Consequently, fixed geometric scans may place strongly related tokens far apart in the serialized sequence, leading to a mismatch between the spatial scanning topology and the underlying representation dependency structure.

This observation raises a natural question: *Can the scanning entry point and order be adapted to the representation structure of each input?* From this perspective, visual serialization should not be treated as a fixed architectural constant. Instead, we formulate it as an input-dependent latent sequential topology inferred from visual representations.

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.

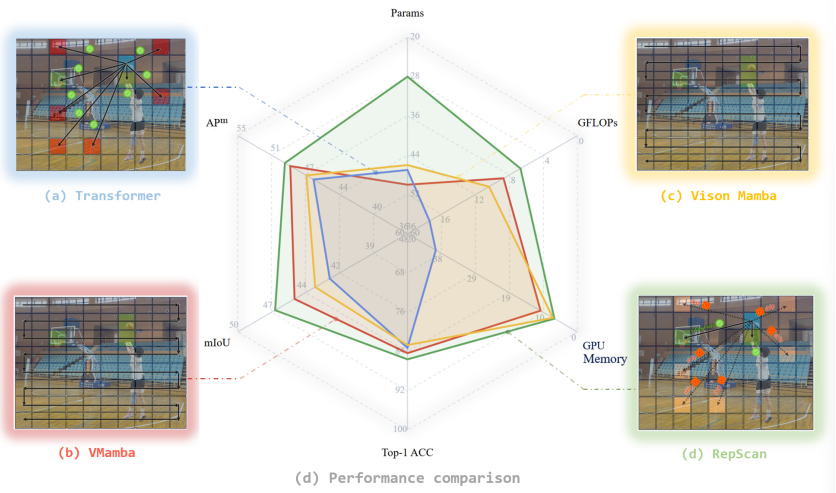


Figure 1: Scanning strategies and performance comparison. Transformer uses global attention, while VMamba and Vision Mamba adopt fixed scanning paths; RepScan dynamically traverses informative patches. The radar chart shows that RepScan achieves a favorable efficiency–accuracy trade-off across multiple metrics.

To this end, we propose RepScan, a representation-adaptive visual scanning framework that dynamically constructs input-dependent serialization orders. We first introduce Representation Relation Modeling (RRM), which jointly considers representation affinity and predictive discrepancy to characterize directed relations between visual tokens. The resulting relation structure captures not only semantic correlations between tokens but also how informative one token is for explaining another. Building upon these relations, we further develop Adaptive Scan Initialization (ASI) and Adaptive Scan Ordering (ASO). ASI selects a content-dependent starting token according to its outgoing connectivity in the representation relation structure, allowing scanning to begin from an informative region. ASO then dynamically determines the subsequent traversal based on the relations among the remaining tokens, enabling the scanning topology to adapt to the representation structure of each input. To make the discrete scanning process learnable, we further employ a differentiable permutation relaxation, allowing the scanning policy to be jointly optimized with the visual backbone in an end-to-end manner.

Unlike the dense token interactions in Transformers or the fixed scanning patterns adopted by existing Mamba-based vision models, RepScan constructs a representation-conditioned sequence for each input. This adaptive sequential topology better organizes structurally meaningful token dependencies while retaining efficient sequential modeling, resulting in a favorable trade-off between accuracy and computational efficiency. Extensive experiments demonstrate the effectiveness of RepScan across diverse visual recognition tasks. On ImageNet-1K, RepScan-T/S/B achieve 83.7%, 84.9%, and 86.8% Top-1 accuracy, consistently outperforming the corresponding InfoScan variants by 0.3, 0.3, and 1.6 percentage points, respectively.

Our main contributions are summarized as follows: ① We introduce a new perspective for visual state-space modeling: instead of treating serialization as a fixed architectural constant, we formulate it as an input-dependent latent sequential topology inferred from visual representations. ② We instantiate this formulation with RepScan, which jointly models directed representation relations and adaptively determines both the scan origin and subsequent traversal through Representation Relation Modeling, Adaptive Scan Initialization, and Adaptive Scan Ordering. For feasible training, we further develop a differentiable local permutation relaxation that enables the discrete adaptive serialization process to be optimized end-to-end while preserving sparse candidate interactions. ③ Extensive experiments across classification, detection, and segmentation show that RepScan consistently improves representation quality while maintaining a favorable accuracy–efficiency trade-off.

2 Related Work

Efficient and Adaptive Computation in Vision. Modern vision transformers rely on self-attention to model long-range dependencies, using dot-product similarity as a proxy for relevance. While ef-

fective, this approach conflates feature similarity with functional relatedness (Touvron et al., 2021; Child, 2019; Bengio et al., 2013; Wang et al., 2022). Efficiency-focused variants–such as sparse attention (Tay et al., 2020), axial attention (Ho et al., 2019), and dynamic token pruning–inherit this assumption (Kim et al., 2024), selecting which connections to retain based primarily on attention scores or learned saliency. However, similarity alone is insufficient to guarantee meaningful dependencies. Patches may appear similar due to shared low-level statistics like texture or color, even when they are semantically unrelated. This leads to spurious correlations that degrade performance under occlusion, distribution shift, or adversarial perturbations. Some recent efforts incorporate reconstruction or information-theoretic objectives to assess dependency strength, but these are typically used as auxiliary losses rather than hard gating criteria. As a result, long-range connections remain largely driven by superficial resemblance, lacking a joint requirement of semantic alignment and statistical sufficiency.

Visual Scanning Strategy. Visual state-space models commonly serialize 2D visual features using predefined geometric orders, such as raster, bidirectional, zigzag, or space-filling scans (Zhu et al., 2024; Pei et al., 2025; Shi et al., 2024; Shaker et al., 2025; Ma et al., 2019; Kamata et al., 1999). While such designs are computationally convenient and often preserve spatial locality, their serialization patterns are largely content-independent and therefore do not explicitly account for the input-dependent semantic structure of visual representations. In real-world images, semantic information is typically distributed non-uniformly: tokens that are strongly related in representation space may be separated along a fixed scan, whereas consecutive positions may correspond to redundant or weakly informative regions. Because state-space propagation is inherently order-sensitive, such a mismatch can affect how contextual information is accumulated along the sequence. Recent approaches have begun to relax fixed serialization through adaptive mechanisms, such as deformable scanning or information-aware traversal (Wu et al., 2026; Liu et al., 2025). However, these methods primarily adapt spatial sampling or resource allocation rather than explicitly constructing the serialization order from directed representation relations. This motivates a complementary question: can both the scan origin and the subsequent traversal be inferred directly from the representation structure of each input, while retaining sparse and efficient sequential computation.

3 METHODOLOGY

3.1 PROBLEM FORMULATION

Given a set of visual representations $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^{N}$, where $\mathbf{x}_i \in \mathbb{R}^d$, a visual state-space model serializes them according to a scanning permutation

$$\pi = (\pi_1, \ldots, \pi_N), \qquad \mathbf{X}_\pi = (\mathbf{x}_{\pi_1}, \ldots, \mathbf{x}_{\pi_N}). \tag{1}$$

Existing methods typically employ a predefined permutation π shared across all inputs, thereby imposing a content-independent sequential topology despite the fact that semantic relationships among visual representations can vary substantially across inputs.

We instead formulate scanning as a representation-conditioned ordering problem:

$$\pi^* = f_\theta(\mathbf{X}) \in \mathbf{\Pi} \tag{2}$$

where the optimized scanning permutation π^* is dynamically determined from the input representations via $f_\theta(\mathbf{X})$, which is an element of the set $\mathbf{\Pi}$ denoting the set of all possible π. We decompose this ordering problem into two complementary decisions: selecting an informative starting token and determining how to traverse the remaining tokens. These two decisions motivate our **Adaptive Scan Initialization and Adaptive Scan Ordering**, respectively, which jointly construct an input-dependent sequential topology based on the relations among visual representations. To control the cost of scan construction, relation modeling is restricted to a bounded number of candidate transitions per token.

3.2 REPRESENTATION RELATION MODELING (RRM)

Representation Relation Modeling constructs sparse, input-dependent relations to guide adaptive scanning. Given visual representations $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^{N}$, $\mathbf{x}_i \in \mathbb{R}^d$, we associate each source token s with a candidate index set $\mathcal{N}(s) \subseteq \{1, \ldots, N\} \setminus \{s\}$, where $|\mathcal{N}(s)| \leq K$ and K is independent of

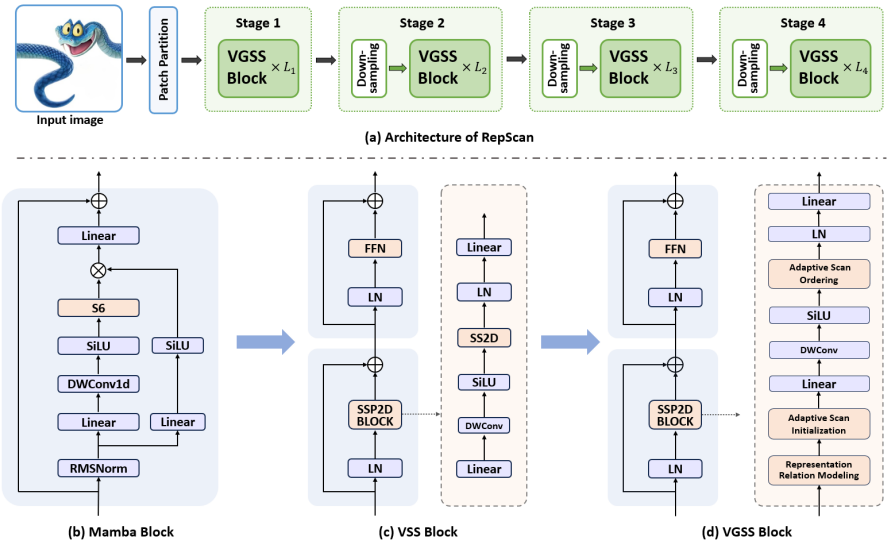


Figure 2: (a) Overall architecture with patch embedding and hierarchical VGSS stages. (b) Mamba block for state-space sequence modeling. (c) VSS block integrating SS2D and FFN. (d) VGSS block incorporating SSP2D for representation-adaptive scanning.

N. Candidates are generated from at most $K-1$ predefined local and dilated spatial offsets, together with the successor of s under a fixed raster scan, when available. Invalid and duplicate indices are discarded. This construction yields a sparse directed edge set

$$\mathcal{E} = \{(s, g) : g \in \mathcal{N}(s)\}, \qquad |\mathcal{E}| \leq NK, \tag{3}$$

without computing all-pairs similarities. While the candidate locations are fixed by spatial rules, their scores depend on the input through two complementary signals: *representation affinity* and *predictive discrepancy*.

3.2.1 Representation Affinity

For each $(s, g) \in \mathcal{E}$, we measure representation compatibility using a learnable bilinear function:

$$\tilde{\mathcal{M}}(\mathbf{x}_s, \mathbf{x}_g) = \phi(\mathbf{x}_s)^\top \mathbf{W} \phi(\mathbf{x}_g), \tag{4}$$

where $\phi : \mathbb{R}^d \to \mathbb{R}^{d'}$ is a learnable feature transformation and $\mathbf{W} \in \mathbb{R}^{d' \times d'}$ is a learnable symmetric matrix. The affinity is obtained by normalizing over the source-specific candidate set:

$$\mathcal{M}(\mathbf{x}_s, \mathbf{x}_g) = \frac{\exp\Big(\tilde{\mathcal{M}}(\mathbf{x}_s, \mathbf{x}_g)/\tau\Big)}{\sum_{g' \in \mathcal{N}(s)} \exp\Big(\tilde{\mathcal{M}}(\mathbf{x}_s, \mathbf{x}_{g'})/\tau\Big)}, \tag{5}$$

where $\tau > 0$ controls the sharpness of the distribution. Although the bilinear compatibility is symmetric, the source-specific normalization generally yields asymmetric affinities when both edge directions are present.

3.2.2 Predictive Discrepancy

We complement affinity with a directional measure of how closely a candidate matches a prediction from the source:

$$\mathcal{D}(\mathbf{x}_s, \mathbf{x}_g) = \|\mathbf{x}_g - g_\psi(\mathbf{x}_s)\|_2^2, \qquad (s, g) \in \mathcal{E}, \tag{6}$$

where $g_\psi : \mathbb{R}^d \to \mathbb{R}^d$ is a learnable predictor. Its output is computed once per source token and reused across outgoing edges. A smaller discrepancy indicates closer agreement with the source-conditioned prediction.

To reduce scale variation across source tokens, we normalize the discrepancy by its mean over the candidate set:

$$\bar{\mathcal{D}}(\mathbf{x}_s, \mathbf{x}_g) = \frac{\mathcal{D}(\mathbf{x}_s, \mathbf{x}_g)}{\frac{1}{|\mathcal{N}(s)|} \sum_{g' \in \mathcal{N}(s)} \mathcal{D}(\mathbf{x}_s, \mathbf{x}_{g'}) + \epsilon}, \tag{7}$$

where $\epsilon > 0$ ensures numerical stability.

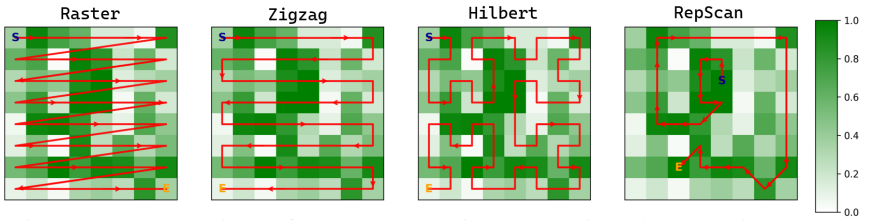


Figure 3: A comparison of RepScan scanning paths with other scanning patterns.

3.2.3 Directed Representation Relation

We combine affinity and predictive discrepancy into a directed relation score:

$$\mathcal{R}_{sg} = \alpha\, \mathcal{M}(\mathbf{x}_s, \mathbf{x}_g) - \lambda\, \bar{\mathcal{D}}(\mathbf{x}_s, \mathbf{x}_g) + \eta, \qquad (s, g) \in \mathcal{E}, \tag{8}$$

where $\alpha, \lambda \geq 0$ control the contributions of the two signals and η is an offset. A higher score indicates a stronger preference for traversing from s to g. The source-conditioned normalization and prediction allow the score to express directional traversal preferences.

The resulting sparse relations are shared by Adaptive Scan Initialization and Adaptive Scan Ordering to determine the scan origin and subsequent traversal order, respectively.

3.3 Adaptive Scan Initialization (ASI)

The scan origin determines which representation first enters sequential state propagation. Adaptive Scan Initialization selects this origin using the sparse directed relations provided by RRM, allowing the starting position to depend on the input.

Outgoing relation score. Let $b = (b_1, \ldots, b_N)$ denote the fixed raster permutation used in candidate construction, and let $\ell(b_t) = b_{t+1}$ for $t < N$. We consider starting candidates $\mathcal{I} = \{b_1, \ldots, b_{N-1}\}$, each of which has a baseline successor included in its candidate set. For each $i \in \mathcal{I}$, we aggregate its outgoing scores using a normalized log-sum-exp:

$$s_i^{\text{start}} = \tau_a \log \left(\frac{1}{|\mathcal{N}(i)|} \sum_{j \in \mathcal{N}(i)} \exp\left(\frac{\mathcal{R}_{ij}}{\tau_a} \right) \right), \tag{9}$$

where $\tau_a > 0$ controls the emphasis on strong outgoing relations. The normalization accounts for differences in candidate-set size. This nonlinear aggregation preserves sensitivity to the distribution of edge scores, which can be lost when averaging the normalized RRM signals. A higher score favors a token with strong candidate transitions for initiating the scan.

Relation-gated start selection. We assess each candidate relative to its baseline successor rather than applying a shared absolute threshold. The reference score is

$$\gamma_i = \mathcal{R}_{i,\ell(i)}, \qquad i \in \mathcal{I}. \tag{10}$$

The valid starting candidates are

$$\mathcal{V} = \left\{ i \in \mathcal{I} \,\middle|\, s_i^{\text{start}} \geq \gamma_i \right\}. \tag{11}$$

Thus, a candidate is retained when its aggregate outgoing score is at least as high as its baseline transition score. We select the scan origin as

$$\pi_1 = \begin{cases} \arg\max_{i \in \mathcal{V}} s_i^{\text{start}}, & \mathcal{V} \neq \varnothing, \\ b_1, & \mathcal{V} = \varnothing. \end{cases} \tag{12}$$

Ties are resolved by the baseline order. For $N = 1$, the single token is selected directly.

3.4 Adaptive Scan Ordering (ASO)

Starting from the token selected by ASI, Adaptive Scan Ordering (ASO) constructs the remaining permutation using the sparse relations provided by RRM. Each decision considers both the immediate transition score and the candidate successor's connectivity to unvisited tokens. This local look-ahead criterion guides traversal without constructing a dense assignment matrix.

3.4.1 CONNECTIVITY-AWARE TRAVERSAL

After selecting π_t, let the unvisited token set and the available neighbors of token i be

$$\mathcal{U}_t = \{1, \dots, N\} \setminus \{\pi_1, \dots, \pi_t\}, \qquad \mathcal{A}_t(i) = \mathcal{N}(i) \cap \mathcal{U}_t. \tag{13}$$

To measure connectivity beyond the baseline relation, we define nonnegative weights on candidate edges:

$$w_{ij} = [\mathcal{R}_{ij} - \gamma_i]_+, \qquad (i,j) \in \mathcal{E}, \tag{14}$$

where $[z]_+ = \max(z, 0)$ and $\gamma_i = \mathcal{R}_{i,\ell(i)}$ reuses the reference defined in ASI. For the last baseline token b_N, which has no baseline successor, we use the mean of its outgoing relation scores as γ_{b_N} when its candidate set is nonempty, and zero otherwise.

For a proposed successor $j \in \mathcal{A}_t(\pi_t)$, we measure its connectivity to its unvisited candidates:

$$c_t(j) = \frac{\sum_{k \in \mathcal{A}_t(j)} w_{jk}}{|\mathcal{A}_t(j)| + \epsilon}, \tag{15}$$

where $\epsilon > 0$ and an empty candidate set gives $c_t(j) = 0$. Since self-edges are excluded in RRM, $j \notin \mathcal{N}(j)$; thus, this quantity describes the candidates that remain available after selecting j. We combine immediate preference and forward connectivity:

$$a_t(j) = \mathcal{R}_{\pi_t j} + \mu c_t(j), \qquad j \in \mathcal{A}_t(\pi_t), \tag{16}$$

where $\mu \geq 0$ controls the contribution of look-ahead.

Let r_t be the first unvisited token in the baseline permutation b. The next token is selected as

$$\pi_{t+1} = \begin{cases} \arg\max\limits_{j \in \mathcal{A}_t(\pi_t)} a_t(j), & \mathcal{A}_t(\pi_t) \neq \varnothing, \\ r_t, & \mathcal{A}_t(\pi_t) = \varnothing, \end{cases} \tag{17}$$

for $t = 1, \dots, N-1$, with ties resolved by the baseline order. The fallback starts a new traversal segment without requiring a scored edge from the current token. It changes the next token in the sequence but does not reset the state-space operator. Every step selects an unvisited token, so the procedure produces a valid permutation containing all tokens exactly once.

3.4.2 TRAINING AND STATE-SPACE PROCESSING

The traversal decisions are discrete. During training, we use a local straight-through estimator to provide task-loss gradients to the relation scores. For a nonempty $\mathcal{A}_t(\pi_t)$, we define

$$p_t(j) = \frac{\exp(a_t(j)/\tau_o)}{\sum_{k \in \mathcal{A}_t(\pi_t)} \exp(a_t(k)/\tau_o)}, \tag{18}$$

where $\tau_o > 0$. Let $z_t(j) = \mathbb{I}[j = \pi_{t+1}]$ denote the hard selection. The surrogate weights and the next sequence input are

$$\begin{aligned} \widetilde{z}_t(j) &= \mathrm{sg}(z_t(j) - p_t(j)) + p_t(j), \\ \widehat{\mathbf{x}}_{t+1} &= \sum_{j \in \mathcal{A}_t(\pi_t)} \widetilde{z}_t(j) \mathbf{x}_j, \end{aligned} \tag{19}$$

where sg stops gradients. The forward pass satisfies $\widehat{\mathbf{x}}_{t+1} = \mathbf{x}_{\pi_{t+1}}$, while the backward pass uses the local soft probabilities as a surrogate for the hard selection. On fallback steps, we directly use $\widehat{\mathbf{x}}_{t+1} = \mathbf{x}_{r_t}$.

For ASI, the same estimator is applied over the valid starting set $\mathcal{V}$, using s_i^{start} as logits; an empty valid set uses $\widehat{\mathbf{x}}_1 = \mathbf{x}_{b_1}$. These estimators are biased local surrogates. Candidate eligibility, visited-set updates, fallback decisions, and index assignments are treated as discrete operations during back-propagation.

The resulting sequence is processed by the state-space operator, and the outputs are restored to their original spatial positions:

$$\mathbf{H} = \mathrm{SSM}\left(\widehat{\mathbf{x}}_1, \dots, \widehat{\mathbf{x}}_N\right), \qquad \mathbf{y}_{\pi_t} = \mathbf{h}_t. \tag{20}$$

The downstream task loss trains the representation, relation, and state-space parameters through this surrogate computation. At inference, ASI and ASO retain the same hard selection rules, and tokens are gathered directly according to the resulting permutation.

4 EXPERIMENTS

In this section, we conduct experiments to evaluate RepScan across three mainstream vision tasks, and validate the effectiveness of each component. More details can be found in Appendix B.1.

Table 1: Comparison with representative visual backbones on ImageNet-1K classification, MSCOCO2017 object detection and instance segmentation, and ADE20K semantic segmentation. P denotes backbone parameters (M) for the task-specific classification and segmentation configurations, and total detector parameters (M) for COCO; F denotes FLOPs (G). COCO results are obtained using Mask R-CNN with the $1\times$ schedule. AP^b and AP^m denote box AP and mask AP.

(a) ImageNet-1K Classification

Type	Backbone	Venue	P	F	Test Thr.	Train Thr.	Top-1 (%)
CNN	ConvNeXt-T	CVPR'22	29	4.5	73.6	36.5	82.1
Trans.	Swin-T	ICCV'21	28	4.6	68.6	59.8	81.6
	CSWin-T	CVPR'22	23	4.3	79.3	41.2	82.7
SSM	VMamba-T	NeurIPS'24	31	4.9	77.2	24.8	82.5
	VMamba-S	NeurIPS'24	50	8.7	47.1	17.0	83.2
	MSVMamba-T	NeurIPS'24	32	5.1	69.6	37.4	83.0
Adaptive SSM	GroupMamba-T	CVPR'25	23	4.7	58.5	35.1	83.3
	DefMamba	CVPR'25	49	5.7	82.2	44.9	84.3
Adaptive SSM	InfoScan-T	ICLR'26	10	2.5	91.7	46.5	83.4
	InfoScan-S	ICLR'26	24	4.8	63.8	33.6	84.6
	InfoScan-B	ICLR'26	38	8.4	54.0	26.2	85.2
Ours	**RepScan-T**	–	28	2.5	93.8	48.4	83.7
	RepScan-S	–	47	6.7	59.3	39.9	84.9
	RepScan-B	–	51	10.3	44.1	27.5	**86.8**

(b) MSCOCO2017 Object Detection & Instance Segmentation

Type	Backbone	Venue	P	AP^b	AP^m
Trans.	Swin-T	ICCV'21	48	42.9	38.1
CNN	ConvNeXt-T	CVPR'22	48	44.3	39.6
SSM	VMamba-T	NeurIPS'24	50	46.8	42.9
	VMamba-S	NeurIPS'24	70	48.4	43.3
	MSVMamba-T	NeurIPS'24	53	46.9	42.5
Adaptive SSM	DefMamba	CVPR'25	49	46.9	42.8
	InfoScan-T	ICLR'26	42	47.1	41.2
Adaptive SSM	InfoScan-S	ICLR'26	59	48.8	43.4
	InfoScan-B	ICLR'26	78	49.8	44.7
Ours	**RepScan-T**	–	37	48.4	41.7
	RepScan-S	–	56	49.3	44.5
	RepScan-B	–	75	**51.3**	**45.5**

(c) ADE20K Semantic Segmentation

Type	Backbone	Venue	P	mIoU (%)
CNN	ConvNeXt-T	CVPR'22	29	46.0
Trans.	Swin-T	ICCV'21	28	44.5
SSM	Vim-S	ICML'24	46	45.3
	MSVMamba-T	NeurIPS'24	43	46.7
Adaptive SSM	DefMamba	CVPR'25	49	46.3
Adaptive SSM	InfoScan-T	ICLR'26	11	40.9
	InfoScan-S	ICLR'26	38	45.8
Ours	**RepScan-T**	–	11	41.9
	RepScan-S	–	39	**46.9**

4.1 RESULTS OF IMAGE CLASSIFICATION

Setup: We evaluate image classification on ImageNet-1K (Deng et al., 2009) (1.28M training and 50K validation images) at 224×224 resolution. Following VMamba, we use AdamW with cosine learning-rate decay (initial LR 1×10^{-3}) and standard data augmentations.

Results: As shown in Table 1(a), RepScan achieves Top-1 accuracies of 83.7%, 84.9%, and 86.8% on ImageNet-1K for Tiny, Small, and Base variants. RepScan-T outperforms InfoScan-T by 0.3 percentage points at the same computational cost of 2.5G FLOPs. RepScan-S improves accuracy

over InfoScan-S by 0.3 percentage points, with FLOPs increasing from 4.8G to 6.7G. RepScan-B achieves the highest Top-1 accuracy among the compared models at 86.8%, exceeding InfoScan-B by 1.6 percentage points with 10.3G versus 8.4G FLOPs. These results demonstrate consistent accuracy improvements over the corresponding InfoScan variants across all three model scales.

4.2 Results of Object Detection

Setup: We evaluate object detection and instance segmentation on MSCOCO2017 (Lin et al., 2014) using Mask R-CNN with the $1\times$ schedule. Detailed settings are provided in Appendix B.2.

Results: Table 1(b) summarizes the results on MSCOCO2017. RepScan consistently improves detection and instance segmentation performance over the corresponding InfoScan variants. Specifically, RepScan-B achieves the highest box AP and mask AP among the compared models, reaching **51.3** and **45.5**, respectively. It outperforms InfoScan-B by 1.5 box AP and 0.8 mask AP while using fewer parameters (75M vs. 78M). RepScan-S achieves 49.3 box AP and 44.5 mask AP with 56M parameters, improving over InfoScan-S by 0.5 and 1.1 AP, respectively, with 3M fewer parameters. RepScan-T reaches 48.4 box AP and 41.7 mask AP with 37M parameters, surpassing InfoScan-T by 1.3 and 0.5 AP, respectively, while using 5M fewer parameters. These results support the transferability of RepScan to object detection and instance segmentation across model scales.

4.3 Results of Semantic Segmentation

Setup: We evaluate semantic segmentation on ADE20K (Zhou et al., 2017) using UPerNet (Xiao et al., 2018) at 512×512 resolution. We follow the standard training protocol and train all models on the ADE20K training set. Detailed experimental settings are provided in Appendix B.3.

Results: As shown in Table 1(c), RepScan achieves competitive semantic segmentation performance across different model scales. Specifically, RepScan-T obtains 41.9% mIoU, improving over InfoScan-T by 1.0 point at the same model scale. RepScan-S further achieves the best mIoU of **46.9%**, outperforming InfoScan-S by 1.1 points and Vim-S by 1.6 points. It also surpasses MSVMamba-T and DefMamba, which achieve 46.7% and 46.3% mIoU, respectively. These results demonstrate that RepScan effectively transfers the learned visual representations to dense prediction tasks and consistently improves segmentation performance over existing SSM-based backbones.

4.4 Ablation Study

We ablate the two complementary components of RRM: representation affinity (Q1) and predictive discrepancy (Q2). As shown in Table 2, either component alone improves the baseline across all three datasets. Combining Q1 and Q2 yields the largest gains, improving ImageNet-1K accuracy from 83.5% to 86.8%, ADE20K mIoU from 42.9% to 46.9%, and BraTS2021 mIoU from 15.3% to 23.4%. These results indicate that representation affinity and predictive discrepancy provide complementary relational cues for adaptive scanning. More ablations are provided in Appendix C.1.

Table 2: Ablation of Representation Relation Modeling on RepScan-B. Q1 and Q2 denote representation affinity and predictive discrepancy, respectively. The "None" setting removes both relation signals while keeping the remaining architecture unchanged.

Q1	Q2	ImageNet-1K Acc. (%)	ADE20K mIoU (%)	BraTS2021 mIoU (%)
✗	✗	83.5	42.9	15.3
✓	✗	84.6	43.6	16.4
✗	✓	85.2	44.4	17.4
✓	✓	**86.8**	**46.9**	**23.4**

4.5 Comparison and Analysis of Visual Scanning

Random initialization improves performance across tasks. As shown in Table 3, changing the starting point of Single Hilbert Scan from fixed to random (No. 4 vs. No. 3) consistently improves performance by 0.9, 0.6, and 1.2 points on ImageNet-1K, ADE20K, and BraTS2021, respectively. This suggests that random initialization reduces the dependence on a predefined starting position and improves robustness across different tasks. RepScan further employs a representation-guided

Table 3: Comparison of predefined and adaptive scanning strategies on RepScan-B. "Fixed" and "Random" denote predefined and stochastic starting points, respectively.

No.	Method	Start	ImageNet-1K Top-1 (%)	ADE20K mIoU (%)	BraTS2021 mIoU (%)
1	Single Raster Scan	Fixed	83.5	42.9	15.3
2	Triple Raster Scan	Fixed	83.6	41.8	16.5
3	Single Hilbert Scan	Fixed	83.6	42.9	17.1
4	Single Hilbert Scan	Random	84.5	43.5	18.3
5	Triple Hilbert Scan	Fixed	84.8	44.2	16.8
6	RepScan	Adaptive	**86.8**	**46.9**	**23.4**

adaptive scanning policy and achieves the best results on all three benchmarks, reaching 86.8% Top-1 accuracy and 46.9%/23.4% mIoU on ImageNet-1K, ADE20K, and BraTS2021, respectively.

Repeated static scanning provides inconsistent gains. Increasing the number of fixed scanning passes does not consistently improve performance. Triple Raster Scan (No. 2 vs. No. 1) yields only a 0.1-point gain on ImageNet-1K while decreasing ADE20K mIoU by 1.1 points. Similarly, Triple Hilbert Scan (No. 5 vs. No. 3) improves ImageNet-1K and ADE20K by 1.2 and 1.3 points, respectively, but decreases BraTS2021 mIoU by 0.3 points. These results suggest that simply repeating predefined scanning paths is insufficient for robust feature acquisition. In contrast, RepScan dynamically adapts its scanning trajectory to input representations, yielding consistent improvements across all three benchmarks.

4.6 Visualization Analysis

Figure 4 compares scanning trajectories in natural and blurry scenes. Raster, Bidirectional, and Hilbert scans follow predefined geometric patterns, while Random Scan generates orders independently of visual content. RepScan instead uses learned representation relations to select the starting token and construct the subsequent traversal. The visualized examples illustrate how its scan origin and trajectory vary across inputs, highlighting representation-conditioned serialization under both visual conditions.

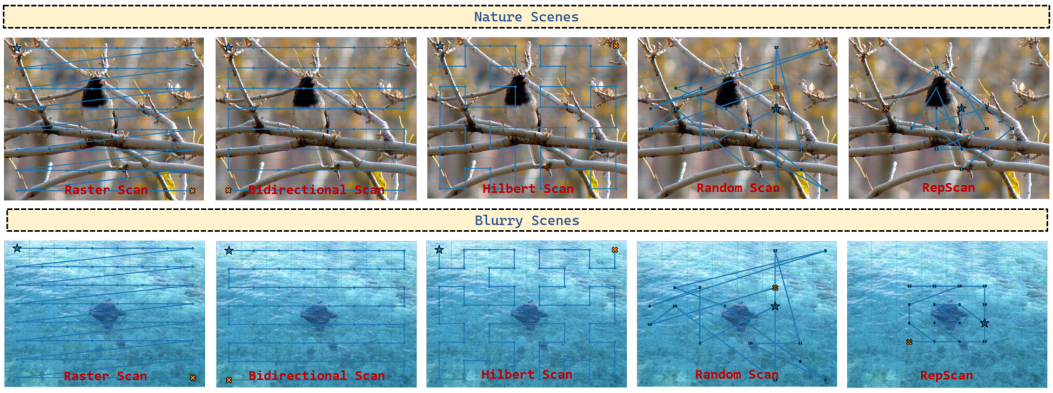


Figure 4: Visualization comparison of different scanning strategies across diverse visual scenarios. Raster, bidirectional, and Hilbert scans follow predefined traversal patterns, while random scanning is content-agnostic. In contrast, RepScan dynamically adapts its scanning trajectory to the input.

Additional results, including the in-depth analysis and visualization of the scanning trajectories, computational efficiency and impact of experimental settings, ***are provided in Appendix C***.

5 Conclusion

We presented RepScan, a representation-adaptive visual backbone that replaces fixed scanning with information-driven dynamic serialization. By dynamically prioritizing informative tokens and meaningful non-local dependencies, RepScan achieves a favorable accuracy–efficiency trade-off rather than eliminating the additional cost of adaptive scan construction. Moreover, its input-dependent scanning mechanism provides an interpretable view of how token traversal adapts to learned representation relations. Future work will extend RepScan to video and multimodal learning.

AI USE STATEMENT

In this work, we used generative AI tools for code completion, unit test generation, and language editing. We have not used generative AI tools for formulating research hypotheses or drafting core scientific arguments. All generated code snippets were independently executed, unit-tested, and verified for correctness by the authors. We take full responsibility for the final content and technical artifacts presented in this work.

ETHICS STATEMENT

This work focuses on foundational algorithms and is evaluated entirely on standard, publicly available benchmark datasets. To the best of our knowledge, this work does not involve human participants, proprietary personal data, or direct applications with potential negative societal impacts. All scientific sources and baseline works have been properly cited.

REPRODUCIBILITY STATEMENT

To ensure the reproducibility of our work, we provide detailed descriptions of the algorithmic procedures, mathematical formulations, and experimental configurations used in RepScan. Specifically, the core components of RepScan, including Representation Relation Modeling, Adaptive Scan Initialization, Adaptive Scan Ordering, and the differentiable permutation relaxation, are described in Section 3, with additional derivations and theoretical justifications provided in Appendix A. Detailed experimental settings, including datasets, optimization protocols, training schedules, and data augmentation strategies, are provided in Appendix B, while additional ablation studies, visualization analyses, and computational efficiency evaluations are reported in Appendix C. We evaluate RepScan on standard vision benchmarks, including ImageNet-1K for image classification, MSCOCO2017 for object detection and instance segmentation, ADE20K for semantic segmentation. All comparative experiments follow the corresponding standard evaluation protocols and are conducted under consistent experimental settings whenever applicable. The complete source code and configuration files will be released upon publication to facilitate reproducibility and further research.

REFERENCES

Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*, 2013.

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

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

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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12124–12134, 2022.

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

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

Jonathan Ho, Nal Kalchbrenner, Dirk Weissenborn, and Tim Salimans. Axial attention in multidimensional transformers. *arXiv preprint arXiv:1912.12180*, 2019.

S-I Kamata, Richard O Eason, and Yukihiro Bandou. A new algorithm for n-dimensional hilbert scanning. *IEEE Transactions on Image Processing*, 8(7):964–973, 1999.

Minchul Kim, Shangqian Gao, Yen-Chang Hsu, Yilin Shen, and Hongxia Jin. Token fusion: Bridging the gap between token pruning and token merging. In *IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 1383–1392, 2024.

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

Leiye Liu, Miao Zhang, Jihao Yin, Tingwei Liu, Wei Ji, Yongri Piao, and Huchuan Lu. Defmamba: Deformable visual state space model. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8838–8847. IEEE, 2025.

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

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *IEEE/CVF International Conference on Computer Vision*, pp. 10012–10022, 2021.

Haiyu Ma, Aijun Sang, Cheng Zhou, Xiaofeng An, and Lijun Song. A zigzag scanning ordering of four-dimensional walsh basis for single-pixel imaging. *Optics Communications*, 443:69–75, 2019.

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, pp. 6443–6451, 2025.

Abdelrahman Shaker, Syed Talal Wasim, Salman Khan, Juergen Gall, and Fahad Shahbaz Khan. Groupmamba: Efficient group-based visual state space model. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14912–14922. IEEE, 2025.

Yuheng Shi, Minjing Dong, and Chang Xu. Multi-scale vmamba: Hierarchy in hierarchy visual state space model. *Advances in Neural Information Processing Systems*, 37:25687–25708, 2024.

Yi Tay, Dara Bahri, Liu Yang, Donald Metzler, and Da-Cheng Juan. Sparse sinkhorn attention. In *International Conference on Machine Learning*, pp. 9438–9447. PMLR, 2020.

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

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

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):415–424, 2022.

Hongtao Wu, Yijun Yang, Huihui Xu, Weiming Wang, Jinni Zhou, and Lei Zhu. Rainmamba: Enhanced locality learning with state space models for video deraining. In *ACM International Conference on Multimedia*, pp. 7881–7890, 2024.

Yifeng Wu, Huimin Huang, Shangjie Zhou, Yawen Huang, Hao Zheng, Yun Chen, Xian Wu, Ruize Han, and Guanhua Chen. Infoscan: Information-efficient visual scanning via resource-adaptive walks. In *International Conference on Learning Representations*, volume 2026, pp. 58743–58760, 2026.

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

Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 633–641, 2017.

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.

Nikola Zubic, Mathias Gehrig, and Davide Scaramuzza. State space models for event cameras. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5819–5828, 2024.

A Derivations and Theoretical Justifications

In this section, we provide additional derivations and theoretical justifications for Representation Relation Modeling (RRM), Adaptive Scan Initialization (ASI), and Adaptive Scan Ordering (ASO). We first provide a probabilistic interpretation of the directional predictive discrepancy and discuss its source-conditioned normalization. We then explain the local-reference principle used by ASI and ASO, and finally derive the local straight-through relaxation used to optimize the discrete traversal decisions.

A.1 Probabilistic Interpretation of Predictive Discrepancy

Recall that RRM defines the directional predictive discrepancy from a source token $\mathbf{x}_s$ to a candidate token $\mathbf{x}_g$ as

$$\mathcal{D}(\mathbf{x}_s, \mathbf{x}_g) = \|\mathbf{x}_g - g_\psi(\mathbf{x}_s)\|_2^2 . \tag{21}$$

This quantity admits a probabilistic interpretation. Assume that the candidate representation conditioned on the source follows an isotropic Gaussian model,

$$p(\mathbf{x}_g \mid \mathbf{x}_s; \psi) = \mathcal{N}\left(g_\psi(\mathbf{x}_s), \sigma^2 \mathbf{I}\right). \tag{22}$$

The corresponding negative log-likelihood is

$$\begin{aligned} -\log p(\mathbf{x}_g \mid \mathbf{x}_s; \psi) &= \frac{1}{2\sigma^2} \|\mathbf{x}_g - g_\psi(\mathbf{x}_s)\|_2^2 + \frac{d}{2}\log(2\pi\sigma^2) \\ &= \frac{1}{2\sigma^2}\mathcal{D}(\mathbf{x}_s, \mathbf{x}_g) + \text{const}. \end{aligned} \tag{23}$$

Therefore, for fixed σ, minimizing $\mathcal{D}(\mathbf{x}_s, \mathbf{x}_g)$ is equivalent to maximizing the conditional likelihood of $\mathbf{x}_g$ given $\mathbf{x}_s$.

This interpretation also explains the directionality of the discrepancy. In general,

$$p(\mathbf{x}_g \mid \mathbf{x}_s) \neq p(\mathbf{x}_s \mid \mathbf{x}_g), \tag{24}$$

and consequently

$$\mathcal{D}(\mathbf{x}_s, \mathbf{x}_g) \neq \mathcal{D}(\mathbf{x}_g, \mathbf{x}_s). \tag{25}$$

Such asymmetry is desirable for adaptive scanning because the suitability of traversing from $\mathbf{x}_s$ to $\mathbf{x}_g$ need not equal that of the reverse transition.

A.2 Source-Conditioned Normalization

The affinity $\mathcal{M}(\mathbf{x}_s, \mathbf{x}_g)$ is normalized over the candidate set associated with source $\mathbf{x}_s$. In contrast, the raw predictive discrepancy depends on the scale of the feature representations. Directly combining the two quantities may therefore introduce an undesirable scale imbalance.

For each source token, define its mean candidate discrepancy as

$$\mu_s^{\mathcal{D}} = \frac{1}{|\mathcal{N}(s)|} \sum_{g' \in \mathcal{N}(s)} \mathcal{D}(\mathbf{x}_s, \mathbf{x}_{g'}). \tag{26}$$

The normalized discrepancy is

$$\bar{\mathcal{D}}(\mathbf{x}_s, \mathbf{x}_g) = \frac{\mathcal{D}(\mathbf{x}_s, \mathbf{x}_g)}{\mu_s^{\mathcal{D}} + \epsilon}. \tag{27}$$

Hence, $\bar{\mathcal{D}}$ measures the predictive inconsistency of a candidate relative to the typical discrepancy observed for the same source token.

Combining affinity and normalized discrepancy yields the directed representation relation

$$\mathcal{R}_{sg} = \alpha\, \mathcal{M}(\mathbf{x}_s, \mathbf{x}_g) - \lambda\, \bar{\mathcal{D}}(\mathbf{x}_s, \mathbf{x}_g) + \eta. \tag{28}$$

A larger $\mathcal{R}_{sg}$ therefore indicates stronger semantic compatibility together with stronger directional predictive consistency.

A.3 Unified Local-Reference Principle

A central principle of RepScan is that adaptive scanning decisions are evaluated relative to the local transition provided by a predefined baseline scan. This avoids introducing a globally shared absolute threshold whose appropriate scale may vary across tokens and inputs.

Let $\ell(i)$ denote the successor of token i under the predefined baseline scanning rule. We define the source-conditioned local reference as

$$r_i^{\mathrm{loc}} = \mathcal{R}_{i,\ell(i)}. \tag{29}$$

Using Eq. (28), this reference can be written as

$$r_i^{\mathrm{loc}} = \alpha\, \mathcal{M}\left(\mathbf{x}_i, \mathbf{x}_{\ell(i)}\right) - \lambda\, \bar{\mathcal{D}}\left(\mathbf{x}_i, \mathbf{x}_{\ell(i)}\right) + \eta. \tag{30}$$

The local reference is both input- and source-dependent. It provides a relative criterion for determining whether the learned relational structure supports an adaptive decision beyond the transition already available from the baseline scan. ASI uses this reference to determine where to start, while ASO uses it to evaluate candidate transitions.

A.4 Adaptive Scan Initialization

ASI evaluates the suitability of token i as a scan origin by aggregating its outgoing relations. Consistent with the main text, we use a normalized log-sum-exp aggregation:

$$s_i^{\mathrm{start}} = \tau_a \log\left(\frac{1}{|\mathcal{N}(i)|}\sum_{j\in\mathcal{N}(i)} \exp\left(\frac{\mathcal{R}_{ij}}{\tau_a}\right)\right), \tag{31}$$

where $\tau_a > 0$ controls the concentration of the aggregation.

The normalized form provides a smooth interpolation between average-like aggregation and emphasis on strong outgoing relations. In particular, as τ_a increases, the score approaches the average outgoing relation up to higher-order terms, whereas a smaller τ_a places greater emphasis on strongly related candidates. The normalization by $|\mathcal{N}(i)|$ prevents the score from being trivially affected by the number of available candidates.

Let

$$\mathcal{I} = \{b_1, \ldots, b_{N-1}\} \tag{32}$$

denote the set of candidate starting tokens, where each token has a well-defined successor under the baseline raster permutation. For each $i \in \mathcal{I}$, the local reference used by ASI is

$$\gamma_i = r_i^{\mathrm{loc}} = \mathcal{R}_{i,\ell(i)}. \tag{33}$$

Thus, the validity condition

$$s_i^{\mathrm{start}} \geq \gamma_i \tag{34}$$

is equivalently

$$s_i^{\mathrm{start}} - r_i^{\mathrm{loc}} \geq 0. \tag{35}$$

The valid starting set is therefore

$$\mathcal{V} = \left\{ i \in \mathcal{I} \,\middle|\, s_i^{\mathrm{start}} - r_i^{\mathrm{loc}} \geq 0 \right\}. \tag{36}$$

When $\mathcal{V} \neq \varnothing$, ASI selects

$$\pi_1 = \arg\max_{i\in\mathcal{V}} s_i^{\mathrm{start}}. \tag{37}$$

If $\mathcal{V} = \varnothing$, ASI falls back to the first token b_1 of the predefined baseline permutation. ASI therefore changes the scan origin only when the learned representation relations provide sufficient evidence for doing so.

A.5 Relative Relational Gain for Adaptive Scan Ordering

For a candidate transition $i \to j$, ASO evaluates its relation relative to a source-conditioned local reference. For all tokens except the last token b_N in the baseline permutation, this reference is given by the relation to the baseline successor. For b_N, which has no baseline successor, we follow the definition in Sec. 3.4.1 and use the mean of its outgoing relation scores when its candidate set is nonempty. Specifically,

$$\gamma_i^{\mathrm{rel}} = \begin{cases} \mathcal{R}_{i,\ell(i)}, & i \neq b_N, \\ \dfrac{1}{|\mathcal{N}(i)|} \displaystyle\sum_{k \in \mathcal{N}(i)} \mathcal{R}_{ik}, & i = b_N,\ \mathcal{N}(i) \neq \varnothing, \\ 0, & i = b_N,\ \mathcal{N}(i) = \varnothing. \end{cases} \tag{38}$$

Thus, all candidate successors of the same source token are evaluated against the same source-conditioned reference.

The corresponding gated relational gain is

$$w_{ij} = \left[\mathcal{R}_{ij} - \gamma_i^{\mathrm{rel}}\right]_+, \qquad [z]_+ = \max(z, 0). \tag{39}$$

Accordingly,

$$w_{ij} > 0 \quad \Longleftrightarrow \quad \mathcal{R}_{ij} > \gamma_i^{\mathrm{rel}}. \tag{40}$$

The quantity w_{ij} therefore measures the positive relational gain of candidate transition $i \to j$ relative to the local reference of source token i.

Offset invariance. The additive offset η in the representation relation does not affect the relative relational gain. For $i \neq b_N$, substituting Eq. (28) into Eq. (39) gives

$$\begin{aligned} w_{ij} = \Big[& \alpha\big(\mathcal{M}(\mathbf{x}_i, \mathbf{x}_j) - \mathcal{M}(\mathbf{x}_i, \mathbf{x}_{\ell(i)})\big) \\ & - \lambda\big(\bar{\mathcal{D}}(\mathbf{x}_i, \mathbf{x}_j) - \bar{\mathcal{D}}(\mathbf{x}_i, \mathbf{x}_{\ell(i)})\big)\Big]_+ . \end{aligned} \tag{41}$$

The common offset η cancels exactly.

For $i = b_N$ with $\mathcal{N}(i) \neq \varnothing$, the same cancellation holds because γ_i^{rel} is the mean of the outgoing relation scores, each of which contains the same additive offset η. Therefore, ASO depends on relative changes in affinity and predictive discrepancy rather than on the absolute offset of the relation scores.

Consistency between ASI and ASO. ASI and ASO follow the same source-conditioned reference principle. Since ASI considers only starting candidates $i \in \mathcal{I} = \{b_1, \ldots, b_{N-1}\}$, each such token has a well-defined baseline successor, and its gating condition is

$$\text{ASI}: \qquad s_i^{\mathrm{start}} - \mathcal{R}_{i,\ell(i)} \geq 0. \tag{42}$$

ASO applies the analogous relative criterion during traversal:

$$\text{ASO}: \qquad w_{ij} = \left[\mathcal{R}_{ij} - \gamma_i^{\mathrm{rel}}\right]_+ . \tag{43}$$

The former determines whether a token receives sufficient relational support to serve as an adaptive scan origin, whereas the latter measures the relative advantage of a candidate transition during traversal. For all $i \neq b_N$, both mechanisms use the same baseline-successor reference $\mathcal{R}_{i,\ell(i)}$; ASO additionally handles the terminal baseline token b_N using its mean outgoing relation score.

A.6 Connectivity-Aware Local Ordering

Starting from the token selected by ASI, ASO constructs the remaining permutation sequentially. After visiting $\pi_1, \ldots, \pi_t$, the unvisited set is

$$\mathcal{U}_t = \{1, \ldots, N\} \setminus \{\pi_1, \ldots, \pi_t\}. \tag{44}$$

For the current token π_t, only unvisited neighbors are considered:

$$\mathcal{A}_t(\pi_t) = \mathcal{N}(\pi_t) \cap \mathcal{U}_t. \tag{45}$$

Selecting a successor solely according to the immediate relation $\mathcal{R}_{\pi_t j}$ may lead to a locally strong transition whose subsequent connectivity is weak. ASO therefore augments the immediate relation with a one-step look-ahead term. For $j \in \mathcal{A}_t(\pi_t)$, let

$$\mathcal{B}_t(j) = \mathcal{N}(j) \cap (\mathcal{U}_t \setminus \{j\}) \tag{46}$$

denote the remaining candidates accessible from j.

The connectivity score of candidate j is

$$c_t(j) = \begin{cases} \dfrac{1}{|\mathcal{B}_t(j)|} \displaystyle\sum_{k \in \mathcal{B}_t(j)} w_{jk}, & |\mathcal{B}_t(j)| > 0, \\ 0, & |\mathcal{B}_t(j)| = 0. \end{cases} \tag{47}$$

The ordering score is then

$$a_t(j) = \mathcal{R}_{\pi_t j} + \mu c_t(j), \qquad j \in \mathcal{A}_t(\pi_t), \tag{48}$$

where μ balances immediate relational compatibility and future connectivity.

The hard traversal decision is

$$\pi_{t+1} = \arg \max_{j \in \mathcal{A}_t(\pi_t)} a_t(j). \tag{49}$$

This criterion is local and sparse: it evaluates only the candidate relations supplied by RRM and does not require a dense assignment matrix over all token pairs.

If $\mathcal{A}_t(\pi_t) = \varnothing$ while unvisited tokens remain, ASO falls back to the first unvisited token according to the predefined baseline permutation. Since every selected token is removed from $\mathcal{U}_t$, this procedure produces a valid permutation and visits each token exactly once.

A.7 Local Straight-Through Relaxation

The hard decision in Eq. (49) is discrete and therefore does not provide a useful gradient through the selected index. During training, we introduce a local relaxation over the same sparse candidate set.

For $j \in \mathcal{A}_t(\pi_t)$, define

$$p_t(j) = \frac{\exp\left(a_t(j)/\tau_o\right)}{\sum_{k \in \mathcal{A}_t(\pi_t)} \exp\left(a_t(k)/\tau_o\right)}, \tag{50}$$

where $\tau_o > 0$ controls the sharpness of the local distribution. Let

$$z_t(j) = \mathbb{I}[j = \pi_{t+1}] \tag{51}$$

be the one-hot indicator associated with the hard decision. The straight-through weight is

$$\widetilde{z}_t(j) = \mathrm{sg}\left(z_t(j) - p_t(j)\right) + p_t(j), \tag{52}$$

where $\mathrm{sg}(\cdot)$ denotes the stop-gradient operator.

The representation supplied to the sequential state-space operator is

$$\widehat{\mathbf{x}}_{t+1} = \sum_{j \in \mathcal{A}_t(\pi_t)} \widetilde{z}_t(j) \mathbf{x}_j. \tag{53}$$

The relaxation therefore acts only on the local selection decision; the candidate set, visited-state update, fallback rule, and resulting scan permutation remain discrete.

A.8 Forward Consistency and Surrogate Gradient

We next clarify the behavior of the straight-through construction in the forward and backward passes.

Proposition 1. For every step t with $\mathcal{A}_t(\pi_t) \neq \varnothing$, the straight-through weights in Eq. (52) are numerically identical to the hard one-hot selection during forward propagation:

$$\widetilde{z}_t(j) = z_t(j). \tag{54}$$

Consequently,

$$\widehat{\mathbf{x}}_{t+1} = \mathbf{x}_{\pi_{t+1}}. \tag{55}$$

Proof. The stop-gradient operator is numerically the identity in the forward pass. Therefore,

$$\begin{aligned} \widetilde{z}_t(j) &= z_t(j) - p_t(j) + p_t(j) \\ &= z_t(j). \end{aligned} \tag{56}$$

Substituting this result into Eq. (53) yields

$$\begin{aligned} \widehat{\mathbf{x}}_{t+1} &= \sum_{j \in \mathcal{A}_t(\pi_t)} z_t(j) \mathbf{x}_j \\ &= \mathbf{x}_{\pi_{t+1}}, \end{aligned} \tag{57}$$

because z_t is one-hot at the token selected by ASO. □

Hence, the relaxation does not alter the discrete scanning order in the forward pass. In the backward pass, however, the stop-gradient term has zero derivative, giving

$$\frac{\partial \widetilde{z}_t(j)}{\partial a_t(k)} = \frac{\partial p_t(j)}{\partial a_t(k)}. \tag{58}$$

For the local softmax,

$$\frac{\partial p_t(j)}{\partial a_t(k)} = \frac{1}{\tau_o} p_t(j) \left(\mathbb{I}[j = k] - p_t(k) \right). \tag{59}$$

The straight-through estimator therefore preserves the exact hard traversal during forward propagation while providing local surrogate gradients for learning the relation and ordering scores. As with standard straight-through estimation, these gradients are biased surrogates rather than exact derivatives of the discrete $\arg\max$ operation.

A.9 Sparsity and Complexity of Adaptive Ordering

The local relaxation preserves the sparse structure introduced by RRM. Since each source token retains at most K candidate relations,

$$|\mathcal{N}(i)| \leq K, \tag{60}$$

and consequently

$$|\mathcal{A}_t(\pi_t)| \leq K \tag{61}$$

for every traversal step.

The local softmax and straight-through weighting therefore operate on at most K candidates per step. Across $N - 1$ traversal decisions, the selection relaxation requires $\mathcal{O}(NK)$ candidate-level operations. When K is treated as a fixed candidate budget, this part scales linearly with the number of visual tokens.

The connectivity-aware look-ahead in Eq. (47) is also restricted to sparse RRM neighborhoods. A direct implementation evaluates at most K successor candidates and up to K outgoing relations for each candidate, leading to a conservative upper bound of $\mathcal{O}(NK^2)$ for the ordering stage. Since K is bounded and substantially smaller than N, ASO avoids the $\mathcal{O}(N^2)$ storage associated with dense global assignment matrices.

At inference time, the soft probabilities and straight-through weights are unnecessary. ASO directly applies the hard ordering rule in Eq. (49), together with the fallback mechanism, to construct the final input-dependent permutation.

Table 4: Implementation details of RepScan-B. K denotes the maximum number of candidate transitions per source token. ASI and ASO denote Adaptive Scan Initialization and Adaptive Scan Ordering, respectively.

Stage	Resolution	K	Candidate offsets	ASI	ASO
1	56×56	9	$\mathcal{O}_{\text{local/dilated}}$ + raster	Yes	Yes
2	28×28	9	$\mathcal{O}_{\text{local/dilated}}$ + raster	Yes	Yes
3	14×14	9	$\mathcal{O}_{\text{local/dilated}}$ + raster	Yes	Yes
4	7×7	9	$\mathcal{O}_{\text{local/dilated}}$ + raster	Yes	Yes

B Experimental Setup Details

B.1 Implementation Details

We adopt a four-stage hierarchical architecture for RepScan-B, with feature resolutions progressively reduced from 56×56 to 7×7. Representation-adaptive scanning is enabled at all four stages. For each source token, we construct a sparse candidate set containing local and dilated spatial neighbors together with its successor under the raster scan. The maximum candidate budget is fixed to $K = 9$ across all stages. The detailed stage-wise configuration is summarized in Table 4.

The spatial candidate offsets are defined as

$$\mathcal{O}_{\text{local/dilated}} = \{(\pm 1, 0), (0, \pm 1), (\pm 2, 0), (0, \pm 2)\},$$

corresponding to four local and four dilated neighbors. Together with the raster successor, each source token therefore has at most nine candidate transitions. Invalid or duplicated candidates near feature-map boundaries are discarded. Unless otherwise specified, the relation and ordering hyperparameters are shared across all stages, with $\alpha = 1.0$, $\lambda = 0.5$, $\mu = 0.5$, $\tau = 0.1$, $\tau_a = 0.5$, and $\tau_o = 0.5$.

B.2 Settings for Object Detection

We conduct object detection experiments on the MS-COCO 2017 dataset. The dataset contains 118K training images, 5K validation images, and 20K test images. We adopt the standard Mask R-CNN with the 1× schedule as the base framework. For ViT-based backbones, we follow ViTDet and apply additional designs–such as interleaved window and global attention–to handle high-resolution inputs. For SSM-based Vim backbones, we use the original architecture without modifications. All other training and evaluation settings remain consistent across variants. We optimize using AdamW with a weight decay of 0.1 and a total batch size of 64. The learning rate is initialized to 1×10^{-4}.

B.3 Settings for Semantic Segmentation

We conduct semantic segmentation experiments on the ADE20K and BraTS2021 datasets. ADE20K contains 150 fine-grained semantic categories, with 20K images for training, 2K for validation, and 3K for testing. BraTS2021 includes three semantic classes (tumor sub-regions), and we use T1-weighted MRI scans as input. The training, validation, and test sets contain 21K, 3K, and 6K slices, respectively. We adopt UPerNet as the base framework. During training, we use the AdamW optimizer with a weight decay of 0.01 and a total batch size of 24. The learning rate is initialized to 8×10^{-5}, decayed linearly, and warmed up over the first 2,000 iterations. We train for a total of 180K iterations. Standard data augmentations are applied: random horizontal flipping, random rescaling within the range $[0.5, 2.0]$, and random photometric distortion. At evaluation, input images are resized such that the shorter side is 512 pixels.

C Analysis Study

C.1 Ablation on Adaptive Scan Components

We further investigate the **individual contributions** of Adaptive Scan Initialization (ASI) and Adaptive Scan Ordering (ASO). As shown in Table 5, all variants retain the complete Representation

Table 5: Component-wise ablation of Adaptive Scan Initialization (ASI) and Adaptive Scan Ordering (ASO). All variants use the RepScan-B configuration with the complete RRM.

Variant	Initialization	Ordering	IN-1K Top-1 (%)	ADE20K mIoU (%)	BraTS2021 mIoU (%)
RRM only	Fixed	Local	83.2	43.2	18.7
+ ASI	Adaptive	Local	83.9	44.7	19.6
+ ASO	Fixed	Adaptive	84.4	45.3	21.9
Full RepScan	Adaptive	Adaptive	**86.8**	**46.9**	**23.4**

Relation Modeling (RRM), while selectively replacing adaptive initialization or ordering with its non-adaptive counterpart. This controlled setting isolates the contribution of each component while keeping the underlying representation relation structure unchanged.

Using RRM with a fixed starting point and local ordering serves as the reference configuration, achieving 83.2% Top-1 accuracy on ImageNet-1K and 43.2%/18.7% mIoU on ADE20K/BraTS2021. Introducing ASI while retaining local ordering improves the three metrics to 83.9%, 44.7%, and 19.6%, corresponding to gains of 0.7, 1.5, and 0.9 points, respectively. This indicates that selecting the scanning origin according to input-dependent representation relations provides a more informative initialization than using a predefined starting position.

Enabling ASO with a fixed initialization yields larger improvements, reaching 84.4% Top-1 accuracy and 45.3%/21.9% mIoU. Compared with the reference configuration, this corresponds to gains of 1.2, 2.1, and 3.2 points, respectively. The particularly pronounced improvement on BraTS2021 suggests that adaptively reorganizing the token sequence is beneficial for inputs with complex representation structures, where a predefined local traversal may not adequately preserve informative dependencies.

Finally, combining ASI and ASO yields the best overall performance, achieving 86.8% Top-1 accuracy on ImageNet-1K and 46.9%/23.4% mIoU on ADE20K/BraTS2021. Relative to the RRM-only reference, the complete RepScan improves the three benchmarks by 3.6, 3.7, and 4.7 points, respectively. These results demonstrate that ASI and ASO provide complementary benefits: ASI determines *where to start*, whereas ASO determines *how to traverse* the remaining representations. Their combination enables RepScan to construct a fully input-dependent serialization topology rather than adapting only a single aspect of the scanning process.

C.2 Semantic Analysis of Scanning Trajectories

C.2.1 Qualitative Visualization

To better understand how different serialization strategies organize visual tokens, we visualize their scanning trajectories on representative ADE20K validation images in Fig. 5. Each row corresponds to a different scene, while the columns show the input image, the corresponding semantic foreground mask, and the trajectories produced by Raster Scan, Bidirectional Scan, Hilbert Scan, Random Scan, and RepScan, respectively. The star and cross markers indicate the starting and ending positions of the visualized trajectory.

As shown in Fig. 5, conventional geometric scanning strategies follow predefined spatial patterns that are independent of image content. Raster Scan traverses tokens row by row, Bidirectional Scan follows an alternating horizontal pattern, and Hilbert Scan preserves local spatial continuity through a fixed space-filling curve. Random Scan, in contrast, does not impose spatial continuity and therefore produces irregular long-range transitions. Since these serialization rules are content-independent, their trajectories remain structurally similar across images despite substantial differences in scene layout and semantic composition.

RepScan exhibits a qualitatively different behavior. Its starting position and subsequent transitions vary across images, and the trajectory is more concentrated within the mask-associated semantic regions instead of uniformly traversing the entire spatial grid. In scenes with different spatial layouts, RepScan correspondingly changes both its starting location and traversal pattern. This observation is consistent with the intended role of representation-adaptive serialization: the ordering is condi-

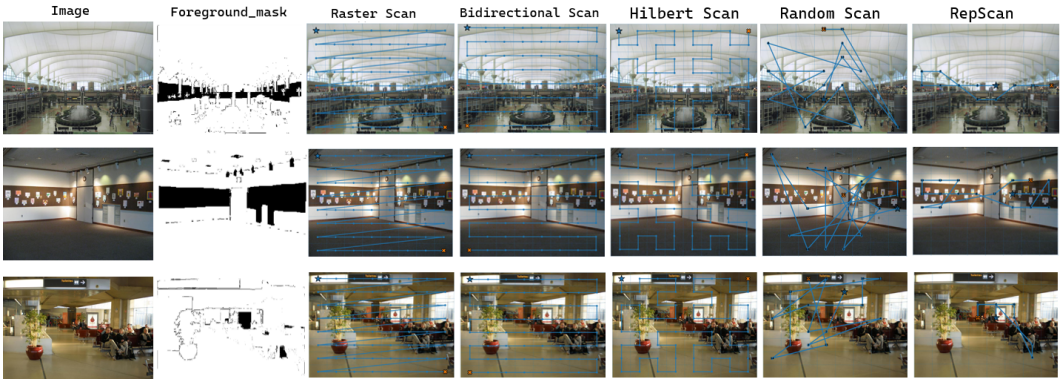


Figure 5: Qualitative comparison of scanning trajectories on representative ADE20K scenes. From left to right, we show the input image, its semantic foreground mask, and the scanning trajectories generated by Raster Scan, Bidirectional Scan, Hilbert Scan, Random Scan, and RepScan. Fixed geometric scans follow predetermined spatial patterns regardless of scene content, while Random Scan produces irregular transitions without explicit semantic structure. In contrast, RepScan exhibits image-dependent starting positions and adaptive traversal patterns, with the visualized trajectory concentrated around mask-associated semantic regions. The star and cross denote the starting and ending positions, respectively. These examples provide a qualitative view of how adaptive serialization changes the order in which visual regions are visited.

tioned on the current visual representation rather than being determined solely by token coordinates. The qualitative results therefore illustrate the input-dependent nature of RepScan and motivate a quantitative analysis of whether semantically relevant regions are indeed encountered earlier in the serialization process.

C.2.2 Early Foreground Ratio

To quantitatively examine whether different scanning strategies prioritize semantically informative regions at different stages of serialization, we introduce the *Early Foreground Ratio* (EFR). Let $\mathcal{F}$ denote a predefined set of foreground semantic categories in ADE20K. The category set $\mathcal{F}$ is fixed before evaluation and shared across all scanning strategies, thereby avoiding method-specific category selection.

For the i-th visual token, let Ω_i denote its corresponding spatial region in the input image, and let $\Omega_i^{\text{valid}} \subseteq \Omega_i$ denote the subset of pixels with valid semantic annotations, excluding ignore/void regions. We define the foreground coverage of token i as

$$r_i^{\text{fg}} = \frac{\sum_{p \in \Omega_i^{\text{valid}}} \mathbf{1}[Y_p \in \mathcal{F}]}{|\Omega_i^{\text{valid}}|}, \tag{62}$$

where Y_p denotes the ground-truth semantic label of pixel p.

Given a scanning permutation $\pi = (\pi_1, \ldots, \pi_N)$, we measure the average foreground coverage among the tokens visited within the first $q\%$ of the scanning sequence as

$$\text{EFR@}q = \frac{1}{K_q} \sum_{t=1}^{K_q} r_{\pi_t}^{\text{fg}}, \qquad K_q = \left\lfloor \frac{q}{100} N \right\rfloor, \tag{63}$$

where $q \in (0, 100]$ specifies the percentage of the scanning sequence considered. A higher EFR indicates that the early portion of the scanning sequence contains tokens with greater spatial overlap with foreground semantic regions. Therefore, EFR measures the semantic alignment of the early scanning order rather than the overall foreground coverage of an image.

Importantly, ground-truth semantic annotations are used *only* for post-hoc computation of EFR and are never provided to any scanning strategy when constructing its traversal order. In particular, the RepScan permutation π is generated solely from input-dependent visual representations through the

learned scanning mechanism. EFR therefore serves as an external diagnostic metric for assessing whether the learned serialization order naturally prioritizes semantically relevant foreground regions, without introducing semantic annotations into the scan-construction process.

As shown in Table 6, RepScan exhibits substantially stronger alignment with foreground semantic regions during the early stage of serialization than all compared scanning strategies. Specifically, RepScan achieves EFR values of 63.7%, 72.6%, and 86.4% within the first 10%, 20%, and 30% of the scanning sequence, respectively. The strongest baseline reaches 34.5%, 36.4%, and 42.8% at the corresponding ratios, resulting in absolute improvements of 29.2, 36.2, and 43.6 percentage points for RepScan.

Table 6: Early Foreground Ratio (EFR) on the ADE20K validation set. EFR measures the average foreground coverage of tokens visited within the first $q\%$ of each scanning sequence. Ground-truth semantic annotations are used only for post-hoc evaluation and are not accessible to any scanning strategy. Higher values indicate stronger alignment between early-scanned tokens and foreground semantic regions. Best results are highlighted in bold.

Scanning Strategy	**EFR@10% ↑**	**EFR@20% ↑**	**EFR@30% ↑**
Raster Scan	23.4%	26.7%	36.9%
Bidirectional Scan	12.3%	19.6%	39.2%
Hilbert Scan	34.5%	36.4%	41.1%
Random Scan	26.3%	29.3%	42.8%
RepScan	**63.7%**	**72.6%**	**86.4%**

Fixed geometric scans exhibit substantially lower EFR because their serialization orders are predetermined by spatial coordinates and cannot adapt to the semantic organization of individual inputs. Random Scan removes the deterministic geometric constraint but likewise lacks an input-conditioned mechanism for consistently organizing semantically informative tokens toward the beginning of the sequence. In contrast, RepScan determines its traversal from learned representation relations, resulting in considerably greater foreground coverage during early serialization.

Together with the qualitative trajectories in Fig. 5, these results demonstrate a consistent association between the representation-adaptive ordering learned by RepScan and the semantic structure of the input. Notably, RepScan is not explicitly supervised to identify foreground regions when constructing the scanning permutation; the observed foreground preference emerges from the learned representation-conditioned ordering. We therefore interpret the EFR results as evidence that RepScan organizes semantically relevant visual content earlier in the serialization process, rather than as evidence of an explicit foreground-detection mechanism.

C.2.3 Intervention on Learned Scanning Order

EFR quantifies the semantic alignment of the learned trajectory, but does not directly assess whether the ordering itself contributes to downstream performance. We therefore perform test-time interventions on the scanning permutation while keeping the trained RepScan-B checkpoint fixed. Specifically, we consider four variants: *Fixed Raster*, which replaces the learned permutation with a predefined raster order; *Random Start*, which replaces the ASI-selected origin with a randomly sampled token while retaining the subsequent adaptive traversal; *Shuffle Remainder*, which preserves the learned starting token but randomly permutes the remaining tokens; and *Reverse Remainder*, which preserves the starting token while reversing the remaining learned traversal.

As shown in Table 7, all interventions lead to lower performance than the original RepScan ordering. Replacing the adaptive trajectory with a fixed raster order reduces ImageNet-1K accuracy from 86.8% to 83.1% and ADE20K mIoU from 46.9% to 43.8%, while EFR@20% decreases from 72.6% to 26.7%. Randomizing only the starting token results in a smaller degradation, yielding 85.7% accuracy, 46.1% mIoU, and 54.3% EFR@20%.

Perturbing the subsequent traversal causes larger performance drops. *Shuffle Remainder* achieves 84.3% accuracy and 44.7% mIoU, whereas *Reverse Remainder* obtains 84.9% and 45.2%, respectively. Their EFR@20% values also decrease substantially to 31.8% and 27.9%. These results

indicate that the benefit of RepScan is not determined solely by the selected scan origin. Instead, both adaptive initialization and the representation-conditioned ordering of the remaining tokens contribute to the observed performance gains.

Table 7: Test-time intervention on the learned scanning order of RepScan-B. All variants use the same trained checkpoint and differ only in the serialization order used during inference.

Ordering intervention	IN-1K Acc. (%)	ADE20K mIoU (%)	EFR@20% (%)
Fixed raster	83.1	43.8	26.7
Random start	85.7	46.1	54.3
Shuffle remainder	84.3	44.7	31.8
Reverse remainder	84.9	45.2	27.9
RepScan	**86.8**	**46.9**	**72.6**

C.3 Resolution Scaling and Computational Efficiency

We further investigate the **computational scalability of RepScan across different input resolutions**. Since RepScan augments the linear recurrent propagation of visual state space models with input-dependent relation modeling and adaptive token serialization, it introduces additional computation beyond the underlying state-space propagation. We therefore examine how its practical computational requirements vary with increasing input resolution.

Table 8 compares VMamba-S, InfoScan-S, and RepScan-S at input resolutions ranging from 224^2 to 1024^2. All models are evaluated under the same hardware and software configuration. We report FLOPs, inference latency, throughput, and peak GPU memory to characterize their computational profiles across resolutions.

Table 8: Resolution-scaling comparison of computational efficiency. FLOPs, inference latency, throughput, and peak GPU memory are reported under the same hardware and software configuration across different input resolutions. Latency and throughput are measured empirically and therefore may not scale proportionally with FLOPs.

Method	Resolution	FLOPs (G) ↓	Latency (ms) ↓	Throughput (img/s) ↑	Peak Memory (GB) ↓
VMamba-S	224^2	8.4	13.6	72.2	2.1
InfoScan-S	224^2	4.8	14.2	67.7	2.0
RepScan-S	224^2	**6.7**	16.9	**59.3**	2.6
VMamba-S	384^2	25.3	31.2	31.9	4.2
InfoScan-S	384^2	14.1	34.8	28.7	4.4
RepScan-S	384^2	20.0	39.1	25.6	4.9
VMamba-S	512^2	45.1	52.8	18.9	6.0
InfoScan-S	512^2	25.0	58.1	17.2	6.5
RepScan-S	512^2	36.1	67.4	14.8	7.5
VMamba-S	768^2	101.5	111.7	9.0	11.4
InfoScan-S	768^2	56.2	123.5	8.1	12.1
RepScan-S	768^2	84.7	153.2	6.5	15.2
VMamba-S	1024^2	180.4	194.6	5.1	18.7
InfoScan-S	1024^2	99.9	216.3	4.6	20.0
RepScan-S	1024^2	157.8	195.5	5.1	26.8

At the standard 224^2 resolution, RepScan-S requires 6.7G FLOPs, which is lower than VMamba-S (8.4G) but higher than InfoScan-S (4.8G). It achieves a throughput of 59.3 images/s with a peak memory consumption of 2.6 GB. As the input resolution increases, the computational and memory costs of RepScan-S also increase. Specifically, its FLOPs increase from 6.7G at 224^2 to 20.0G, 36.1G, 84.7G, and 157.8G at 384^2, 512^2, 768^2, and 1024^2, respectively. The corresponding inference latency increases from 16.9 ms to 39.1 ms, 67.4 ms, 153.2 ms, and 195.5 ms, while the peak memory consumption increases from 2.6 GB to 4.9 GB, 7.5 GB, 15.2 GB, and 26.8 GB.

Table 9: Ablation of candidate topology in RepScan-B. All variants use the same RRM, ASI, and ASO modules and differ only in candidate-set construction.

Candidate topology	IN-1K Acc. (%)	ADE20K mIoU (%)	Latency (ms)	EFR@20% (%)
Local	84.4	44.8	15.2	40.6
Local + Dilated	84.9	45.9	17.5	52.3
Random-K	85.3	46.1	17.9	58.7
Feature-KNN	86.4	46.5	24.7	67.8
Current RepScan	**86.8**	**46.9**	17.2	**72.6**

Table 8 further shows that the computational profile of RepScan-S differs from those of VMamba-S and InfoScan-S across input resolutions. RepScan does not modify the linear recurrent propagation mechanism of the underlying VSSM; instead, it introduces additional computation for input-dependent serialization to determine a representation-adaptive token traversal. At commonly used resolutions, RepScan-S maintains moderate computational and memory requirements, whereas the additional cost becomes more noticeable at higher resolutions. These results characterize the practical efficiency trade-off introduced by adaptive scanning without attributing the observed runtime differences solely to FLOPs.

C.4 Ablation on Candidate Topology

We further investigate the effect of candidate-set construction while keeping RRM, ASI, and ASO unchanged. As shown in Table 9, using only local neighbors yields 84.4% Top-1 accuracy on ImageNet-1K and 44.8% mIoU on ADE20K. Introducing dilated neighbors improves the results to 84.9% and 45.9%, respectively, indicating that broader spatial coverage benefits adaptive scanning. Random-K further improves performance to 85.3% Top-1 accuracy and 46.1% mIoU, suggesting that providing non-local transition candidates can be beneficial even without explicit semantic guidance.

Feature-KNN achieves stronger semantic alignment, reaching 86.4% ImageNet-1K accuracy, 46.5% ADE20K mIoU, and 67.8% EFR@20%, but incurs substantially higher latency of 24.7 ms. In contrast, the candidate topology used in RepScan achieves the best overall performance, with 86.8% Top-1 accuracy, 46.9% mIoU, and 72.6% EFR@20%, while maintaining a lower latency of 17.2 ms. These results indicate that the proposed candidate construction provides a favorable balance between semantic coverage, recognition performance, and computational efficiency.

C.5 Sensitivity Analysis of Candidate Budget K

We further evaluate the sensitivity of RepScan to the candidate budget K, while keeping all other settings unchanged. As shown in Table 10, increasing K from 5 to 9 consistently improves performance. In particular, ImageNet-1K Top-1 accuracy increases from 85.6% to 86.8%, while ADE20K mIoU improves from 45.8% to 46.9%. This indicates that a moderate increase in the candidate budget provides broader relational coverage and benefits adaptive scanning.

However, further increasing K does not yield additional gains. When K increases to 11 and 13, ImageNet-1K accuracy decreases to 86.4% and 86.1%, while ADE20K mIoU drops to 45.7% and 45.2%, respectively. Meanwhile, inference latency increases from 17.2 ms at $K = 9$ to 20.4 ms and 22.1 ms at $K = 11$ and $K = 13$, respectively. These results suggest that excessively large candidate sets may introduce less informative or redundant transitions while increasing computational cost. We therefore use $K = 9$ by default, which achieves the best performance while maintaining a favorable accuracy–efficiency trade-off.

D Limitations

Despite the strong empirical performance of RepScan across multiple 2D visual recognition tasks, several limitations remain. First, the representation-adaptive scan construction introduces additional runtime and memory overhead beyond the underlying state-space propagation. Although this over-

Table 10: Sensitivity analysis of the candidate budget K in RepScan-B. All other settings are kept unchanged.

K	IN-1K Acc. (%)	ADE20K mIoU (%)	Latency (ms)
5	85.6	45.8	16.1
7	86.3	46.5	16.9
9	**86.8**	**46.9**	17.2
11	86.4	45.7	20.4
13	86.1	45.2	22.1

head remains moderate at commonly used input resolutions, it becomes increasingly noticeable for very high-resolution inputs, as shown in our resolution-scaling analysis.

Second, RepScan relies on a bounded candidate set to preserve sparse and efficient relation modeling. Our ablation studies indicate that both the candidate topology and candidate budget influence the resulting accuracy–efficiency trade-off. The current implementation therefore adopts a predefined sparse candidate topology, while more efficient learned candidate discovery mechanisms may further improve adaptability without substantially increasing computational cost.

Finally, our current evaluation mainly focuses on static, single-frame 2D inputs. Extending representation-adaptive scanning to video, volumetric data, or multi-view settings would require additional modeling of temporal, cross-slice, or cross-view consistency beyond spatial representation relations. We leave these extensions to future work.