DART: DIFFERENTIABLE ADAPTIVE REGION TOKENIZER FOR UNIFIED VISUAL BACKBONES

Anonymous authors
Paper under double-blind review

ABSTRACT

A 29M-parameter Vim-Small equipped with our tokenizer surpasses a 98M-parameter Vim-Base on ImageNet (82.2% vs. 81.9%) at half the computational cost (10.9G vs. 19.9G FLOPs). We achieve this through a simple allocation principle: each token should cover an equal share of learned importance rather than an equal share of area. We realize this principle through DART, a closed-form construction that inverts a piecewise-linear CDF at uniform quantile points. The resulting tokenizer is differentiable, always outputs a fixed-length regular tensor, and recovers the standard fixed grid exactly when the importance map is uniform. Under the standard 196-token budget, DART consistently improves accuracy by +0.8% to +1.6% across five backbone configurations spanning Transformers and state space models. The consistency of improvements across architectures with fundamentally different sequence-processing mechanisms suggests the equal-importance principle captures a general property of visual information rather than exploiting architecture-specific biases.

1 INTRODUCTION

Before a visual backbone begins inference, the tokenizer has already determined how much representational capacity each spatial region receives. The standard fixed grid divides the image into equal-area patches, letting a bird's eye and blank sky share the same token budget. This strategy is simple and hardware-friendly, but the discriminative contributions of different regions in natural images vary greatly, making equal-area allocation not always reasonable.

High-resolution processing amplifies this problem. Under a fixed grid, increasing resolution causes all regions to simultaneously receive more tokens, including information-sparse backgrounds, so computational growth far outpaces the improvement in effective representation.

Existing methods face three mutually tense design constraints. (i) **Content adaptiveness** requires that token boundaries be determined by image content. Classical vision algorithms (quadtree, graph clustering) satisfy this but are non-differentiable, preventing end-to-end optimization. (ii) **End-to-end differentiability** requires that the partitioning process participate in gradient optimization. Gumbel-Softmax or straight-through estimators achieve differentiability, but often produce irregular tensors, making theoretical FLOPs savings hard to translate into actual speedup. (iii) **Regular tensor output** means a fixed sequence length ensuring standard batching and GPU parallelism. Hierarchical backbones (e.g., Swin (Liu et al., 2021)) satisfy this but their downsampling is content-agnostic and changes the flat-sequence interface. Internal token pruning methods (DynamicViT (Rao et al., 2021), ToMe (Bolya et al., 2023)) preserve the architecture but act as a post-hoc remedy applied after the fixed grid has already determined the initial tokenization. Existing methods typically trade off among these constraints.

This paper proposes **DART** (Differentiable Adaptive Region Tokenizer), a lightweight front-end module that replaces the fixed-grid tokenizer with a content-aware partition governed by a simple allocation principle: *each token should cover an equal share of learned importance*. A lightweight scoring network predicts per-location importance and normalizes it into a probability distribution. Region boundaries are then obtained in closed form via inversion of the piecewise-linear CDF at uniform quantile points, a construction rooted in classical quantile function theory that partitions the importance mass into equal portions. This formulation addresses all three design constraints identified above: it is differentiable (supporting end-to-end training), always outputs a fixed-length regular

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.

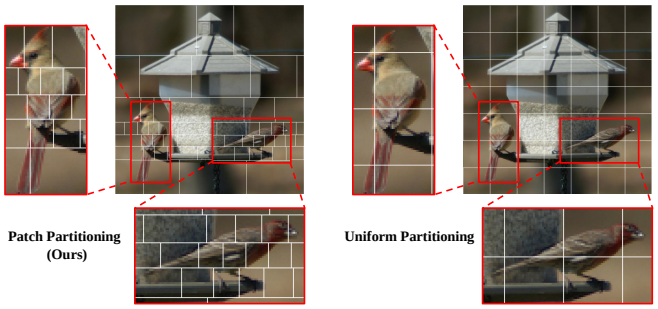


Figure 1: Comparison between our patch partitioning (left) and the uniform partitioning (right). Finer patches are allocated to the bird region, merging low-information background.

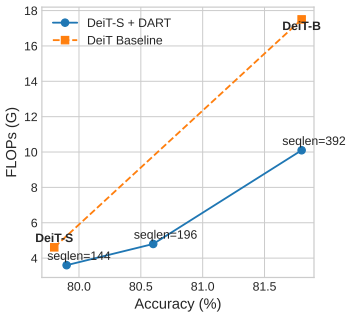


Figure 2: FLOPs vs. Accuracy trade-off curve for DeiT models.

tensor (compatible with standard batching and GPU parallelism), and adapts to image content, concentrating tokens on informative regions while maintaining a complete, non-overlapping partition of the image domain (Figure 1). DART adopts a separable two-stage decomposition: DART-Grid partitions the two axes independently using marginal distributions, while DART-Flow, our primary method, redistributes the token budget globally over the full 2D importance landscape, allowing tokens to flow toward informative regions across the entire image. The scoring network determines only partition boundaries; token content is always sampled from the original image, decoupling the routing decision from the representation.

Under the standard 196-token budget (sequence length unchanged, only spatial support domains change), DART yields consistent improvements: +1.6%/+0.8% on DeiT-Ti/S and +1.1%/+1.0% on Vim-Ti/S. In high-resolution fine-tuning, a DART-equipped DeiT-S matches the fixed-grid baseline's accuracy at only 46% of the computational cost (7.2 vs. 15.5 GFLOPs, 288 vs. 576 tokens). With 392 tokens, both DeiT-S+DART (24M total parameters) and Vim-S+DART (29M) reach or exceed the accuracy of their respective Base counterparts (DeiT-B 86M at 81.8%; Vim-B 98M at 81.9%) at substantially lower computational cost. A controlled sanity check confirms that uniform importance scores cause DART to degenerate exactly to the standard fixed grid, recovering baseline performance, which verifies that the observed improvements require learned non-uniform allocation. The allocation principle generalizes beyond classification: DART-Grid yields +0.5 mIoU on Swin-T/UPerNet for ADE20k segmentation, and in video classification, DART improves SSv2 accuracy by +0.5% while using only 59% of the baseline computation.

Our contributions are as follows:

- We propose the equal-importance allocation principle as a replacement for the standard equal-area assumption in visual tokenization, and show that this single change improves both efficiency and accuracy without altering the backbone architecture or training objective.
- We realize this principle through DART, a closed-form construction based on CDF inversion that is differentiable, always outputs a fixed-length regular tensor, and recovers the standard fixed grid exactly when the importance map is uniform.
- We validate DART across image classification, dense prediction, and video understanding on both Transformer (DeiT (Touvron et al., 2021), Swin (Liu et al., 2021)) and state-space-model (Vim (Zhu et al., 2024), VideoMamba (Li et al., 2024)) backbones, demonstrating consistent improvements and favorable comparisons against dynamic token reduction methods.

2 Related Work

2.1 Solving the Tokenization Bottleneck: Architectural vs. Front-End Solutions

The inefficiency of the uniform tokenizer in ViT (Dosovitskiy et al., 2021) has been widely recognized, leading to two distinct philosophical approaches for resolution.

Architectural Solutions. The first approach is architectural, exemplified by hierarchical models like PVT (Wang et al., 2021) and Swin Transformer (Liu et al., 2021). These models use downsample mechanisms like patch merging to create multi-scale feature pyramids, reducing spatial redundancy in deeper layers. However, this is a heavyweight solution that fundamentally alters ViT's simple, homogenous structure. This incurs architectural costs: complexity in multimodal fusion and a mismatch with the ecosystem of pretrained uniform ViTs. Furthermore, this hierarchical downsampling process is **content-agnostic**; it applies a fixed, structural reduction across the entire feature map instead of dynamically allocating more resolution to critical details.

Front-End Solutions. In contrast, DART is a front-end module that performs content-aware partitioning before the backbone, providing it with an information-dense token sequence while preserving the uniform backbone architecture.

2.2 Dynamic Inference: Dynamic Token Reduction and Adaptive Tokenization

Dynamic Token Reduction and Merging. A prominent line of work focuses on alleviating the computational burden of standard ViTs by progressively reducing the token count *within* the backbone. Methods like DynamicViT (Rao et al., 2021) and A-ViT (Yin et al., 2022) employ learnable scoring modules to dynamically prune less informative tokens at various stages. Similarly, ToMe (Bolya et al., 2023) introduces a parameter-free strategy that progressively merges similar tokens via bipartite soft matching, while IA-RED^2 (Pan et al., 2021) utilizes a multi-level interpreter to discard redundant patches. While effective for efficiency, these methods operate after the initial tokenization has already been determined by a fixed grid. Furthermore, these reduction decisions, whether strictly pruning or merging, can result in variable spatial structures or necessitate masking, complicating downstream processing.

Adaptive Tokenization. Alternative approaches seek to address the representation bottleneck at the source by replacing the fixed grid with content-adaptive partitioning. One category of methods leverages classic computer vision algorithms to group pixels, such as Quad-trees (Ronen et al., 2023), Graph Clustering (Aasan et al., 2024), or Watershed algorithms (Chen et al., 2025). However, these mechanisms are inherently **non-differentiable**, preventing end-to-end optimization with the backbone and often necessitating complex multi-stage training pipelines.

To achieve end-to-end differentiability, a second category of methods introduces approximation techniques. MSViT (Havtorn et al., 2023) employs Gumbel-Softmax, while ∂HT (Aasan et al., 2025) relies on a "mean-injection" trick reminiscent of Straight-Through Estimation (STE). Furthermore, they inherently produce **Ragged Tensors**, which causes poor parallelization efficiency on standard hardware, often negating theoretical FLOPs reductions.

A third category, exemplified by SPFormer (Mei et al., 2025), opts for a heavyweight architectural redesign employing a dual-branch structure with iterative cross-attention to refine superpixels. This approach fundamentally alters the ViT architecture, and its reliance on serialized computation for iterative refinement is unfriendly to GPU parallelism, incurring significant latency overhead.

In contrast, by leveraging a quantile-based inverse transform, DART achieves differentiability and outputs **Regular Tensors** (fixed sequence length), ensuring seamless compatibility within standard ViT/Mamba pipelines.

2.3 Emerging Uniform Backbones and the Need for Adaptive Tokenization

The sequence modeling landscape is evolving, with State Space Models (SSMs) like Mamba (Gu & Dao, 2024) emerging as powerful alternatives to self-attention. Vision Mamba (Vim) adapts this for vision, yet as a new member of the uniform backbone family, it inherits ViT's standard fixed-patch tokenizer. This makes it a natural test case for DART's applicability beyond attention-based architectures. We observe that a front-end that converts an image into an information-dense token sequence is useful regardless of the backbone's processing mechanism (attention, SSMs, or future ones).



(a) 1D differentiable quantile algorithm.

(b) `DART-Grid` partitioning.

Figure 3: Illustration of our core differentiable partitioning mechanism. **(a)** Boundaries for a 1D distribution are found by inverting its piecewise-linear CDF at uniform quantiles (e.g., 1/3, 2/3). **(b)** The `DART-Grid` is formed by independently applying this 1D algorithm to the horizontal and vertical marginal distributions of the 2D score map, where the score map produced by the lightweight network is overlaid on the image.

3 METHOD

3.1 OVERALL PIPELINE

Our dynamic, content-aware tokenizer replaces the conventional fixed-grid patcher via a three-stage process. First, a lightweight scoring network predicts a map of information density. Second, a differentiable partitioning module uses this map to compute adaptive patch boundaries. Third, the resulting non-uniform patches are resampled to a fixed size and projected into tokens. For a detailed visualization of the architecture and data flow, please refer to Figure 12 in the Appendix. This partitioning module has two variants: a grid-preserving method and a global-reallocation method.

Differentiable Quantile Computation. Given a 1D score distribution $S = \{S_i\}_{i=0}^{\text{seqlen}-1}$, we construct a continuous, strictly increasing CDF by treating each score as a uniform density over its unit interval (Figure 3a). Because the CDF is continuous and strictly increasing, its inverse is also continuous, mapping any target cumulative probability to a unique spatial position. Partition boundaries are computed by evaluating this inverse at uniformly-spaced quantiles $q_k = k/N$, yielding boundary positions that vary continuously with the input scores. (The piecewise-linear CDF is the simplest member of a family of continuous approximations; higher-order alternatives such as spline interpolation would yield a globally smooth inverse, but the first-order construction already provides stable gradient-based optimization.) The construction is differentiable with respect to S, enabling end-to-end training (a detailed mathematical formalism is provided in Appendix E.2).

3.2 SCORE PREDICTION NETWORK

Given an input image $X \in \mathbb{R}^{H\times W\times 3}$, a lightweight CNN extracts a feature map $F \in \mathbb{R}^{H'\times W'\times C}$, from which a trainable MLP (scoring head) predicts a single-channel score map $\{s_{i,j}\}$. To produce a stable 2D probability distribution $\{\tilde{s}_{i,j}\}$, these raw scores undergo sequential normalization: a sigmoid constrains values to $[0, 1]$, and a per-sample normalization ensures they sum to 1. This final distribution quantifies each location's relative importance.

3.3 DART-GRID: GRID-PRESERVING PARTITIONING

The most intuitive adaptive partitioning approach creates a non-uniform grid that preserves the original patch topology. We call this method **DART-Grid**. It independently partitions the horizontal and vertical axes based on their marginal probability distributions:

1. The marginal distributions for the y-axis (P_Y) and x-axis (P_X) are computed by summing the 2D probability distribution $\{\tilde{s}_{i,j}\}$ along each axis.

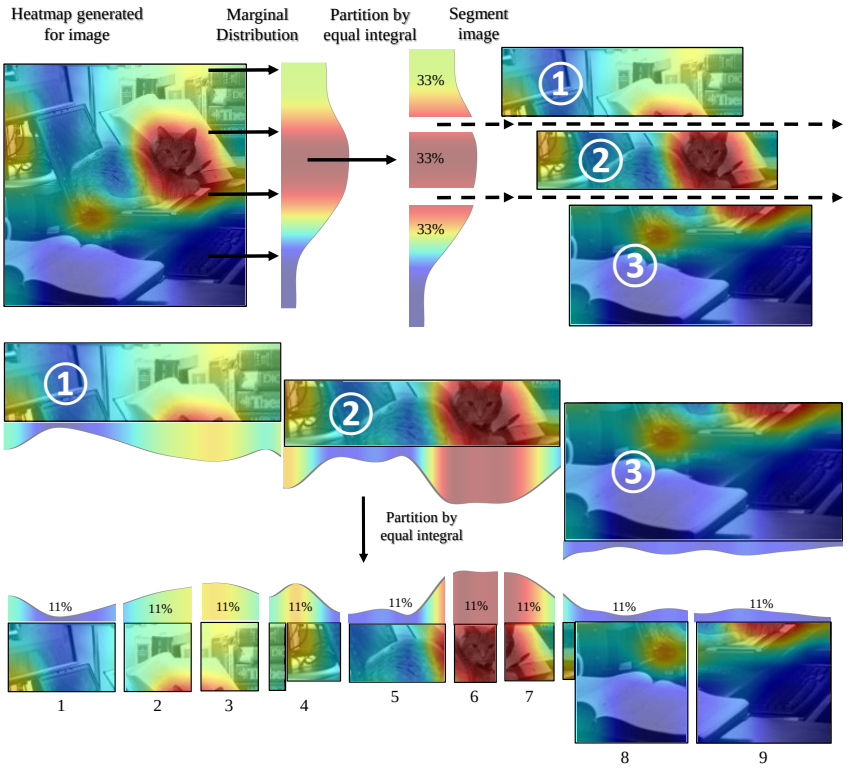


Figure 4: The **DART-Flow** process. The partitioning algorithm is applied sequentially: first to create adaptive horizontal rows, and then to globally allocate tokens across all rows.

2. Our 1D differentiable quantile algorithm is applied to P_Y to find $N_h - 1$ horizontal boundaries, defining N_h rows of varying heights.

3. The same algorithm is similarly applied to P_X to find $N_w - 1$ vertical boundaries, defining N_w columns of varying widths.

The result is a content-aware $N_h \times N_w$ grid where each patch's area is inversely proportional to its region's information density, while the overall grid structure remains intact, as shown in Figure 3b.

3.4 DART-Flow: Global Token Reallocation

While DART-Grid adapts patch sizes, it constrains the token budget within a Cartesian-product structure where all rows share identical column boundaries. To relax this constraint, we propose **DART-Flow**, which allows each row to have independent column boundaries. As our primary method, it is referred to simply as DART throughout this paper and operates in two sequential stages:

1. **Adaptive Row Partitioning:** First, we partition the image horizontally into N_h rows of varying heights, identical to the y-axis partitioning step in DART-Grid. This initial step allocates vertical space based on horizontal information density.

2. **Global Token Allocation via Virtual Flattening:** Next, we conceptually concatenate these N_h adaptive rows into a single, long 1D sequence. The probability distribution over this flattened sequence is derived from the original 2D score map. We then apply our 1D differentiable quantile algorithm once to this long sequence to find all $N_{\text{total}} - 1$ boundaries for the final patches.

This design relaxes the Cartesian-product constraint of DART-Grid. It allows the total token budget (N_{total}) to be distributed according to the full 2D importance landscape rather than independent marginals, as shown in Figure 4.

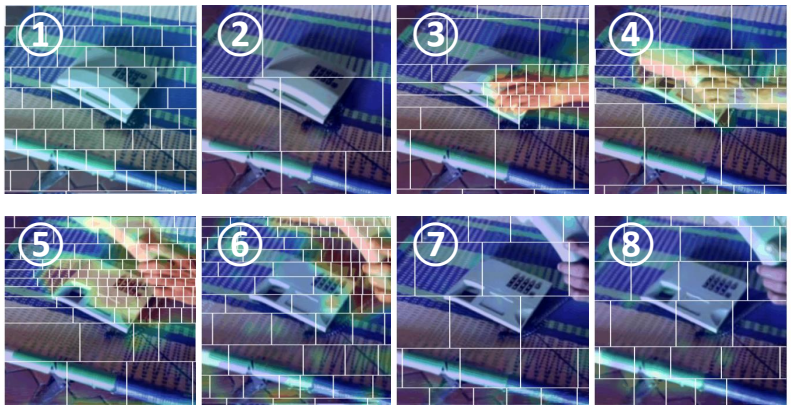


Figure 5: Patch partitioning on an example from the SSv2 dataset (Goyal et al., 2017).

3.5 Application to Video

Our partitioning approach extends naturally to video by vertically concatenating all frames into a composite image and applying our dynamic partitioning. This enables the uneven distribution of a fixed token budget across space and time, concentrating resources on key frames or critical regions.

To capture temporal importance, the scoring network processes the feature difference between consecutive frames, learning to focus on motion. In practice, a stationary object is encoded densely only when it first appears. As it remains unchanged in subsequent frames, it receives few computational resources, achieving efficient temporal redundancy compression as shown in Figure 5.

3.6 Differentiable Resampling and Positional Transformation

Once patch boundaries are determined, a differentiable sampling module converts the non-uniform regions into fixed-size tokens. For image content, a regular 16×16 target grid is defined for each output token. DART's dynamic boundaries define an affine transformation mapping this grid to locations on the input image, from which a standardized 16×16 patch is sampled via bilinear interpolation. These patches, created through a differentiable process, are then linearly projected into tokens. To maintain spatial awareness, positional embeddings (PE) are transformed similarly. The PE is treated as a learnable, low-resolution map ($H_{pe} \times W_{pe} \times D$), and each token's PE is derived by sampling from this map at the token's center coordinate using bilinear interpolation. This critical step preserves spatial relationships by informing the model of each token's origin. The system is trained with a standard cross-entropy loss, with gradients flowing through the boundary computation and resampling steps.

4 Experiments

4.1 Experimental Setup

Our main image classification experiments are conducted on the ImageNet-1K ILSVRC-2012 dataset (Deng et al., 2009), using Top-1 accuracy as the primary metric. The evaluation focuses on the uniform backbone family. DeiT (Touvron et al., 2021) represents the canonical uniform ViT, the same flat-sequence architecture that underpins CLIP (Radford et al., 2021), LLaVA (Liu et al., 2023), and DiT-based generative models (Peebles & Xie, 2023). Vision Mamba (Zhu et al., 2024) extends the evaluation to the emerging SSM paradigm. For a fair comparison, all training hyperparameters strictly follow the official configurations of the baseline models, with details provided in Appendix C.2. All experiments were performed on a single machine with eight NVIDIA A100 GPUs.

Table 1: Efficiency on high-resolution inputs. † denotes long-sequence fine-tuning.

Backbone	Tokenizer	Params	Patches	FLOPs	Top-1 (%)
DeiT-S†	Baseline	22M	576	15.5G	81.6
DeiT-S†	**DART**	24M	**288**	**7.2G**	**81.5**
VideoMamba-Ti†	Baseline	7M	1296	7.11G	79.6
VideoMamba-Ti†	**DART**	8M	**392**	**2.24G**	**79.7**

Table 2: Reaching Base-model accuracy with smaller architectures. † denotes long-sequence fine-tuning.

Backbone	Params	Patches	FLOPs	Top-1 (%)
DeiT Family				
DeiT-B (Target)	86M	196	17.5G	81.8
DeiT-S†	22M	576	15.5G	81.6
DeiT-S† + **DART**	24M	**392**	**10.1G**	**81.8**
Vim Family				
Vim-B (Target)	98M	196	19.9G	81.9
Vim-S†	26M	784	19.6G	81.6
Vim-S† + **DART**	29M	**392**	**10.9G**	**82.2**

4.2 High-Resolution Scaling

A primary method for enhancing Vision Transformer performance is long-sequence fine-tuning, which increases input resolution. However, this approach uniformly increases token density across all regions, so computational cost grows quadratically while much of the additional budget is spent on low-information areas. DART's content-aware allocation addresses this by concentrating tokens on informative regions regardless of input resolution, yielding two practical benefits: substantial computational savings at matched accuracy, and the ability for smaller models to reach performance levels previously requiring larger architectures.

First, we demonstrate DART's value in high-resolution fine-tuning. As shown in Table 1 and Figure 2, a DART-equipped DeiT-S matches a conventional baseline's performance (81.5%) with only 46% of the computational cost (7.2 vs. 15.5 GFLOPs). This efficiency principle is broadly applicable and sometimes even more dramatic; on VideoMamba-Ti (Li et al., 2024), for example, DART delivered similar accuracy using just 31% of the computational cost.

DART also provides an alternative scaling path, as shown in Table 2. While standard high-resolution fine-tuning is costly and fails to elevate small models (DeiT-S, Vim-S) to match their larger Base counterparts, DART-equipped models succeed. A DART-powered DeiT-S matches the accuracy of DeiT-Base, and a Vim-S with DART surpasses its Vim-Base counterpart. This suggests that improving token allocation quality with a smaller model can serve as an alternative to increasing model capacity.

The presented results are from a single model checkpoint fine-tuned on longer sequences. Thanks to DART's dynamic partitioning, this single model is exceptionally flexible, able to seamlessly process various sequence lengths at inference time, including non-square numbers like 288 or 392. Crucially, when this fine-tuned model is fed the original 196-token input, its performance drops by only 0.2% from its pre-fine-tuning level. This capability allows users to select their desired accuracy-compute trade-off at test time without retraining.

These results suggest that improving token allocation quality can be complementary to increasing model capacity as a path to higher performance.

4.3 Fixed-Budget Evaluation

We replace the standard tokenizer with DART while keeping the total token count fixed at 196. As shown in Table 3, DART yields consistent improvements across all five tested configurations

Table 3: Accuracy at the standard 196-token budget.

Backbone	Tokenizer	Params	FLOPs	Top-1 (%)
DeiT-Ti	Baseline	6M	1.26G	72.2
DeiT-Ti	**DART**	7M	1.32G	73.8 **(+1.6)**
DeiT-S	Baseline	22M	4.61G	79.8
DeiT-S	**DART**	24M	4.84G	80.6 **(+0.8)**
Vim-Ti	Baseline	7M	1.60G	76.1
Vim-Ti	**DART**	8M	1.68G	77.2 **(+1.1)**
Vim-S	Baseline	26M	5.30G	80.5
Vim-S	**DART**	29M	5.55G	81.5 **(+1.0)**
VideoMamba-Ti	Baseline	7M	1.08G	76.9
VideoMamba-Ti	**DART**	8M	1.15G	78.2 **(+1.3)**

Table 4: Comparison with dynamic inference methods (Yin et al., 2022; Havtorn et al., 2023; Rao et al., 2021; Pan et al., 2021).

Method	Backbone	Patches	FLOPs	Top-1 (%)
A-ViT	DeiT-Ti	dyn.	0.8G	71.0
DART	DeiT-Ti	121	0.8G	**71.8**
MSViT	DeiT-S	dyn.	3.3G	78.8
A-ViT	DeiT-S	dyn.	3.6G	78.6
DART	DeiT-S	144	3.6G	**79.9**
DynamicViT	DeiT-S	dyn.	7.0G	80.3
DART	DeiT-S	288	7.2G	**81.5**
DynamicViT	DeiT-B	dyn.	11.2G	81.3
IA-RED^2	DeiT-B	dyn.	11.8G	80.3
DART	DeiT-S	392	10.1G	**81.8**

Table 5: Semantic segmentation on the ADE20k validation set.

Backbone	mIoU (%)
Swin-T (Baseline)	44.5
+ **DART-Grid**	45.0 **(+0.5)**

Table 6: Video classification results on SSv2 and Kinetics-400.

Dataset	Method	Patches	Top-1 (%)
SSv2	VideoMamba-Ti	1568	63.2
SSv2	+ **DART**	784	**63.7** **(+0.5)**
K-400	VideoMamba-Ti	1568	76.9
K-400	+ **DART**	1568	**77.3** **(+0.4)**

spanning both Transformer and SSM backbones, with all backbone parameters, training data, and training recipe unchanged. The improvement is particularly notable on DeiT-Ti, which sees a +1.6% increase in Top-1 accuracy. One potential reason is that improving upon a lower-performance baseline can result in a larger gain in the absolute accuracy metric as we observe that DeiT-Ti's baseline performance (72.2%) is considerably lower than other similarly-scaled models such as Vim-Ti (76.1%) and VideoMamba-Ti (76.9%). Notably, the improvement is consistent across both architectural families (attention-based and SSM-based), suggesting the benefit is not tied to a specific backbone mechanism.

4.4 Generalization to Dense Prediction and Video

4.4.1 Case Study: Generalization to Dense Prediction

To validate DART on dense prediction, we integrate it into a UPerNet (Xiao et al., 2018) with a Swin-T backbone on the ADE20k dataset (Zhou et al., 2017). Swin Transformer presents a unique case, as its design already addresses the detail-redundancy trade-off by merging small early-layer tokens into larger ones in deeper layers. However, this merging process is structural and content-agnostic.

Since Swin Transformer's mechanisms (e.g., windowed attention and patch merging) require a regular grid, we used our grid-preserving DART-Grid variant. As shown in Table 5, adding DART-Grid yields a **+0.5 mIoU** improvement on this already strong multi-scale baseline.

4.4.2 Case Study: Extension to the Spatiotemporal Domain

DART's effectiveness in the spatiotemporal domain is highlighted by the results in Table 6. On the motion-reliant Something-Something-V2 (SSv2) dataset, ours improves accuracy by +0.5% while cutting GFLOPs by 41% by allocating tokens to the most informative moments. On the more scene-centric Kinetics-400 (Carreira & Zisserman, 2017), DART still provides a robust +0.4% accuracy gain. The scoring network uses the same weights as in the image experiments, with no video-specific training.

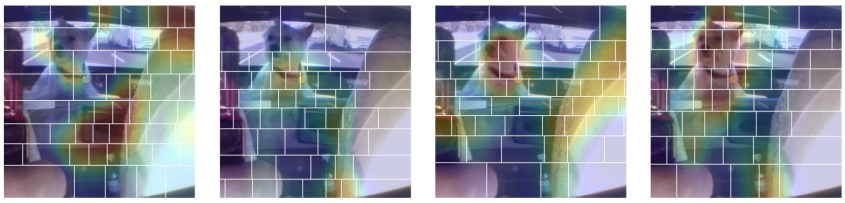


Figure 6: Visualization of the learned score map and patch boundaries at different training epochs, showing the model progressively learning to focus on the salient object.

Table 7: Ablation on partitioning strategies using DeiT-Ti as backbone.

Method	Top-1 (%)
DeiT-Ti	72.2
+ `DART-Grid`	73.1
+ **`DART-Flow`**	**73.8**

Table 8: Inference speed comparison on a 3090 GPU (batch size 512).

Model	Patches	Latency	Img/s
DeiT-B	196	1951 ms	262
DeiT-S	576	1575 ms	325
+DART	392	1142 ms	448
+DART	288	814 ms	629

4.5 Comparison with Dynamic Inference Methods

While motivated by the shared goal of improving efficiency, DART's approach of optimizing tokens at their source is fundamentally different from methods that adapt the sequence internally by pruning or merging tokens. As detailed in Table 4, DART achieves favorable accuracy at matched or lower computational budgets compared to internal token reduction methods.

4.6 Analysis and Ablation Studies

Ablation on Partitioning Strategy. We compare the two DART variants: DART-Flow and the grid-preserving DART-Grid. While DART-Grid improves over the baseline, DART-Flow is significantly better (Table 7), confirming the benefit of relaxing the Cartesian-product constraint. As a crucial sanity check, we verified experimentally that a uniform score map degenerates our mechanism to the standard ViT tokenizer, matching its baseline performance and confirming that the improvements require learned non-uniform allocation. The consistency of gains across architectures with fundamentally different sequence-processing mechanisms (self-attention in DeiT vs. selective state spaces in Vim/VideoMamba) suggests the equal-importance principle is architecture-agnostic, a direct consequence of operating at the tokenization stage, before any backbone-specific processing begins.

Visualization of the Learning Process. Figure 6 visualizes the evolution of DART's learned score map. This smooth, gradual refinement is a consequence of the differentiable boundary computation, which allows the model to learn where to allocate its token budget via end-to-end optimization. The visualization clearly shows the model learning to concentrate tokens on the foreground object over time, validating our approach.

5 Conclusion

We introduced DART, a content-aware tokenizer that replaces the equal-area assumption in visual tokenization with equal-importance allocation. The construction is differentiable, always outputs a fixed-length tensor, and recovers the standard fixed grid when the importance map is uniform. Empirically, DART consistently improves accuracy across Transformer and SSM backbones at the standard token budget, and substantially reduces the computational cost of high-resolution processing by concentrating tokens on informative regions. These results suggest that how a fixed token budget is spatially allocated is a meaningful design axis complementary to model scaling. We discuss directions for future work in Appendix A.

AI USE STATEMENT

In the preparation of this manuscript, we utilized Large Language Models (LLMs) as a writing assistant. The role of the LLM was strictly limited to improving the clarity, flow, and grammatical correctness of the text. Specific tasks included rephrasing sentences for better readability, correcting spelling and grammatical errors, and ensuring a consistent and professional tone. The authors have carefully reviewed and edited all text and take full responsibility for the final content of the manuscript.

ETHICS STATEMENT

This work proposes DART, a content-aware tokenizer that improves the computational efficiency of visual backbone models. Our method operates as a general-purpose front-end module and does not introduce new capabilities beyond what the underlying backbone models already possess. We do not foresee direct negative societal impacts from this contribution. By reducing computational costs for visual recognition tasks, our method may contribute to lowering the energy consumption and environmental footprint of vision model deployment.

REPRODUCIBILITY STATEMENT

To ensure the reproducibility of our research, we have made a comprehensive effort to provide all necessary components. Our approach and resources are organized as follows:

- **Source Code:** The complete source code for our DART module is provided as a ZIP file in the supplementary materials. The code includes a detailed README file with instructions for setting up the environment and running the experiments. We will release the code publicly upon acceptance.
- **Methodology and Implementation:** The core methodology of DART is described in detail in Section 3. To further clarify the implementation, we provide pseudocode for our main method (`DART-Flow`) in Figure 13, and for the differentiable quantile computation in Figure 14. A complete mathematical formalism is presented in Appendix E.2. Specific architectural details for the scoring networks used with different backbones are listed in Appendix Table 19.
- **Training Configurations:** Our main experimental setup is outlined in Section 4. We strictly followed the official training configurations of the baseline models to ensure fair comparisons. Any modifications, specifically the hyperparameters for our long-sequence fine-tuning experiments, are explicitly documented in Appendix C.2.
- **Datasets and Baselines:** All experiments were conducted on standard, publicly available datasets, including ImageNet-1K, ADE20k, Something-Something-V2, and Kinetics-400. We utilized well-established open-source implementations for all baseline models (e.g., DeiT, Vision Mamba), and followed their standard data preprocessing pipelines.
- **Computational Environment:** Our primary experiments were conducted on a server with eight NVIDIA A100 GPUs. Details on the inference benchmarks, including latency and throughput measurements, which were performed on a single NVIDIA RTX 3090 GPU, are provided in Table 8 and Appendix Table 10.

We believe these resources provide a clear and complete roadmap for replicating our findings.

AUTHOR CONTRIBUTIONS

ACKNOWLEDGMENTS

REFERENCES

Marius Aasan, Odd Kolbjørnsen, Anne Schistad Solberg, and Adín Ramírez Rivera. A spitting image: Modular superpixel tokenization in vision transformers. In *CVF/ECCV Computer Vision – ECCVW 2024 – MELEX*, 2024. doi: https://doi.org/10.1007/978-3-031-93806-1_11.

Marius Aasan, Martine Hjelkrem-Tan, Nico Catalano, Changkyu Choi, and Adín Ramírez Rivera. Differentiable hierarchical visual tokenization. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=y8VWYf5cVI.

Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token merging: Your ViT but faster. In *International Conference on Learning Representations*, 2023.

João Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4724–4733, 2017. doi: 10.1109/CVPR.2017.502.

Delong Chen, Samuel Cahyawijaya, Jianfeng Liu, Baoyuan Wang, and Pascale Fung. Subobject-level image tokenization. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=imkFoKwFwd.

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

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=YicbFdNTTy.

Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzynska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fründ, Peter N. Yianilos, Moritz Mueller-Freitag, Florian Hoppe, Christian Thurau, Ingo Bax, and Roland Memisevic. The something something video database for learning and evaluating visual common sense. *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 5843–5851, 2017. URL https://api.semanticscholar.org/CorpusID:834612.

Benjamin Graham, Alaaeldin El-Nouby, Hugo Touvron, Pierre Stock, Armand Joulin, Herve Jegou, and Matthijs Douze. Levit: A vision transformer in convnet's clothing for faster inference. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 12259–12269, October 2021.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. In *First conference on language modeling*, 2024.

Jakob Drachmann Havtorn, Amelie Royer, Tijmen Blankevoort, and Babak Ehteshami Bejnordi. Msvit: Dynamic mixed-scale tokenization for vision transformers. In *ICCV Workshop on New Ideas in Vision Transformers*, 2023.

Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, et al. Searching for mobilenetv3. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 1314–1324, 2019.

Forrest N. Iandola, Song Han, Matthew W. Moskewicz, Khalid Ashraf, William J. Dally, and Kurt Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and <0.5mb model size, 2016. URL https://arxiv.org/abs/1602.07360.

Andrew Jaegle, Felix Gimeno, Andy Brock, Oriol Vinyals, Andrew Zisserman, and Joao Carreira. Perceiver: General perception with iterative attention. In *International conference on machine learning*, pp. 4651–4664. PMLR, 2021.

Kunchang Li, Xinhao Li, Yi Wang, Yinan He, Yali Wang, Limin Wang, and Yu Qiao. Videomamba: State space model fornbsp;efficient video understanding. In *Computer Vision ECCV 2024: 18th European Conference, Milan, Italy, September 29October 4, 2024, Proceedings, Part XXVI*, pp. 237255, Berlin, Heidelberg, 2024. Springer-Verlag. ISBN 978-3-031-73346-8. doi: 10.1007/978-3-031-73347-5_14. URL https://doi.org/10.1007/978-3-031-73347-5_14.

Yanyu Li, Geng Yuan, Yang Wen, Ju Hu, Georgios Evangelidis, Sergey Tulyakov, Yanzhi Wang, and Jian Ren. Efficientformer: Vision transformers at mobilenet speed. *Advances in Neural Information Processing Systems*, 35:12934–12949, 2022.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning, 2023.

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 (ICCV)*, 2021.

Sachin Mehta and Mohammad Rastegari. Mobilevit: Light-weight, general-purpose, and mobile-friendly vision transformer. In *International Conference on Learning Representations*, 2022.

Jieru Mei, Liang-Chieh Chen, Alan Yuille, and Cihang Xie. SPFormer: Enhancing vision transformer with superpixel representation. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856. URL https://openreview.net/forum?id=nu1SjVgSuy.

Bowen Pan, Rameswar Panda, Yifan Jiang, Zhangyang Wang, Rogerio Feris, and Aude Oliva. Ia-redˆ2: Interpretability-aware redundancy reduction for vision transformers. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 24898–24911. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/d072677d210ac4c03ba046120f0802ec-Paper.pdf.

Junting Pan, Adrian Bulat, Fuwen Tan, Xiatian Zhu, Lukasz Dudziak, Hongsheng Li, Georgios Tzimiropoulos, and Brais Martinez. Edgevits: Competing light-weight cnns on mobile devices with vision transformers. In *European Conference on Computer Vision*, 2022.

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

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8748–8763. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/radford21a.html.

Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. Dynamicvit: Efficient vision transformers with dynamic token sparsification. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=jB0Nlbwlybm.

Tomer Ronen, Omer Levy, and Avram Golbert. Vision transformers with mixed-resolution tokenization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4613–4622, 2023.

Michael Ryoo, AJ Piergiovanni, Anurag Arnab, Mostafa Dehghani, and Anelia Angelova. Tokenlearner: Adaptive space-time tokenization for videos. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 12786–12797. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/6a30e32e56fce5cf381895dfe6ca7b6f-Paper.pdf.

Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, Francisco Massa, Daniel Haziza, Luca Wehrstedt, Jianyuan Wang, Timothée Darcet, Théo Moutakanni, Leonel Sentana, Claire Roberts, Andrea Vedaldi, Jamie Tolan, John Brandt, Camille Couprie, Julien Mairal, Hervé Jégou, Patrick Labatut, and Piotr Bojanowski. DINOv3, 2025. URL https://arxiv.org/abs/2508.10104.

Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pp. 6105–6114. PMLR, 2019.

Mingxing Tan, Bo Chen, Ruoming Pang, Vijay Vasudevan, Mark Sandler, Andrew Howard, and Quoc V Le. Mnasnet: Platform-aware neural architecture search for mobile. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2820–2828, 2019.

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Herve Jegou. Training data-efficient image transformers amp; distillation through attention. In *International Conference on Machine Learning*, volume 139, pp. 10347–10357, July 2021.

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

Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *European Conference on Computer Vision*. Springer, 2018.

Hongxu Yin, Arash Vahdat, Jose M. Alvarez, Arun Mallya, Jan Kautz, and Pavlo Molchanov. A-ViT: Adaptive tokens for efficient vision transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10809–10818, June 2022.

Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *Proceedings of the 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. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

A Discussion and Future Work

First, a natural next step is to integrate DART as the vision front-end for larger-scale systems such as Large Multimodal Models and generative models, where visual inputs are complex and token efficiency directly affects inference cost.

Second, the scoring module uses a lightweight CNN as its feature backbone, with the scoring head trained from scratch alongside the main backbone. For specialized applications such as medical or satellite imagery, where the definition of "information-rich" regions is domain-specific, adapting the scoring module on task-relevant data presents a natural opportunity to further boost performance.

Third, the current framework optimizes the token budget within each sample. This naturally opens the door to a more advanced paradigm: inter-sample dynamic allocation, where the total token budget could vary based on sample complexity. Crucially, DART's proven ability to handle arbitrary, non-square sequence lengths provides the essential technical foundation for this next step. Future models could leverage this capability to allocate more computation to "hard" examples and less to "easy" ones, further pushing the boundaries of model efficiency.

Fourth, while our primary method, DART-Flow, is tailored for the uniform backbone paradigm, the core principle of content-aware tokenization remains broadly applicable. We demonstrate this with our compatible DART-Grid variant, which successfully enhances hierarchical models like Swin Transformer, as shown in our dense prediction experiments. This not only confirms the versatility of our approach but also points to a promising research avenue: co-designing novel adaptive tokenizers that are even more deeply integrated with the architectural priors of hierarchical models.

Fifth, we consider the potential applicability of DART to low-level vision and generative tasks, specifically Super-Resolution (SR). Standard Transformer-based SR models often face computational bottlenecks when processing ultra-high-resolution inputs (e.g., 4K/8K). However, the requirement for detail refinement is typically spatially sparse, with perceptual quality hinging largely on high-frequency structures rather than smooth background regions. DART offers a mechanism to decouple input resolution from the computational budget. By adapting the scoring network to identify high-frequency artifacts, DART provides a feasible pathway to process high-resolution inputs using

a fixed, tractable token count, thereby extending the efficiency benefits of adaptive tokenization to generative domains.

Finally, extending DART to larger-scale backbones and integrating it with large-scale pretraining pipelines is a natural next step.

B Additional Experiments and Analysis

B.1 Impact of the Scoring Network

DART is robust to the choice of scoring network, and a stronger scorer further improves performance (Table 9). For instance, on DeiT-Ti, using EfficientNet-B0 instead of MobileNetV3-S increases the accuracy gain from +1.6% to +2.9%.

Table 9: Impact of the scoring network architecture on DeiT-Ti.

Scoring Network	**FLOPs (G)**	**Top-1 (%)**
w/o (Baseline)	1.26	72.2
MobileNetV3-S (Howard et al., 2019)	1.32	73.8 **(+1.6)**
MnasNet (Tan et al., 2019)	1.37	74.0 **(+1.8)**
SqueezeNet (Iandola et al., 2016)	1.54	74.3 **(+2.1)**
EfficientNet-B0 (Tan & Le, 2019)	2.41	75.1 **(+2.9)**

B.2 Inference Speed

Table 10 reports inference throughput and latency for a standard DeiT-S and a DART-equipped variant under identical configuration (196 tokens, NVIDIA RTX 3090, batch size 128).

Table 10: DART module overhead on an NVIDIA RTX 3090 (196 tokens, batch size 128).

Model	**Throughput (img/s)**	**Latency (ms)**
DeiT-S (Baseline)	1110.5	115.3
DeiT-S + DART	1053.2 **(-5.2%)**	121.5 **(+6.2ms)**

B.3 Generalization to Fine-Grained Classification

To validate DART's ability to capture subtle discriminative details, we evaluated it on the fine-grained CUB-200-2011 dataset using linear probing. As shown in Table 11, DART-equipped DeiT-S achieves 76.2% accuracy, surpassing the baseline (75.4%) by +0.8%. This confirms that DART effectively concentrates the token budget on critical local features (e.g., beaks, feathers) rather than the background.

Table 11: Linear probing accuracy on CUB-200-2011.

Method	**Backbone**	**Top-1 (%)**	**Gain**
Baseline	DeiT-S	75.4	-
DART (Ours)	DeiT-S	**76.2**	**+0.8**

B.4 Ablation on Training Strategy

We compared two training configurations for the scoring module: updating only the scoring head, versus updating both the feature extractor and the scoring head jointly. As shown in Table 12, joint training yields only a marginal additional gain (+0.1%), indicating that the scoring head alone has sufficient capacity to learn effective allocation policies.

Table 12: Training strategy ablation on DeiT-Ti.

Strategy	**Top-1 (%)**
Baseline	72.2
DART (scoring head trained)	73.8 (+1.6)
DART (scoring head + extractor trained)	**73.9 (+1.7)**

B.5 Ablation on Input Resolution

We ablated the source image resolution while keeping the token count fixed at 196. Each token is resampled to a fixed 16×16 patch regardless of source resolution, so the total pixel count entering the backbone is always $196 \times 256 = 50{,}176$. A higher source resolution provides finer spatial precision for the adaptive boundary computation. As shown in Figure 7, performance improves up to 448×448 and saturates beyond.

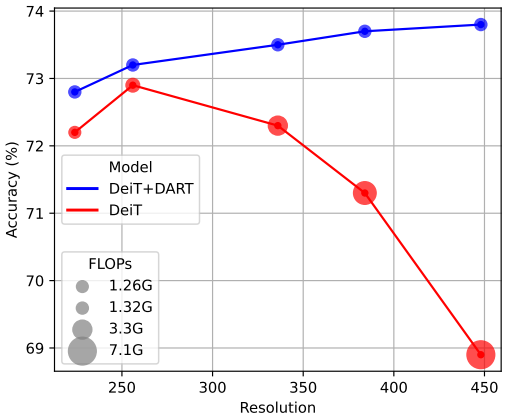


Figure 7: Ablation on input resolution.

B.6 Generalization to Self-Supervised Foundation Models

To validate DART beyond supervised classification and across model scales, we evaluate on DINOv3 Siméoni et al. (2025), a purely self-supervised vision foundation model trained with DINO+iBOT objectives and employing RoPE positional encoding. We freeze the scoring network learned during DeiT supervised training and transfer it directly to DINOv3, fine-tuning the backbone with LoRA (rank 8) at 448 input resolution with 196 tokens (AdamW, lr $= 10^{-3}$, weight decay 0.01, batch size 128, 10 epochs).

Table 13: DART on self-supervised DINOv3 models (ImageNet-1K, 448 input, 196 tokens, LoRA rank 8).

Model	**Params**	**Uniform**	**+ DART**	**Gain**
DINOv3 ViT-S	21M	80.22%	80.60%	+0.38%
DINOv3 ViT-L	300M	85.4%	85.9%	+0.5%

The gains persist across a $50\times$ parameter range (from the original 6M in our main experiments to 300M here), confirming scalability. The frozen scorer's effectiveness on a self-supervised model suggests that spatial importance patterns are properties of image content rather than artifacts of a specific training objective. DART directly substitutes dynamically computed token center coordinates into the RoPE formula without modifying any frequency parameters, confirming out-of-the-box compatibility.

B.7 Isolating the Benefit of Adaptive Allocation

A natural question is whether DART's classification gains stem from (a) the adaptive spatial allocation itself, (b) access to higher-resolution source pixels, or (c) extra parameters in the scoring network. We design controlled experiments to isolate each factor.

Resolution $\times$ Allocation Factorial. We evaluate DeiT-Ti under four configurations: two source resolutions (224, 448) crossed with two allocation strategies (uniform, adaptive). All configurations use the full DART pipeline with bilinear resampling to 16×16, producing 196 tokens via patch16 projection.

Table 14: Factorial ablation: source resolution vs. allocation strategy (DeiT-Ti, 196 tokens).

	Uniform Allocation	**Adaptive Allocation**
224×224 source	72.2%	73.2% **(+1.0)**
448×448 source	72.2%	73.8% **(+1.6)**

With uniform allocation, increasing the source from 224 to 448 provides *zero* improvement, confirming that higher-resolution pixels alone do not contribute. With adaptive allocation at the standard 224 resolution (identical source pixels as the baseline), DART still yields $+1.0\%$, isolating the benefit of the allocation strategy itself. As an additional control, a standard DeiT-Ti with patch size 32 at 448 (maintaining 196 uniform tokens) yields only 72.0%.

Degeneration Test. We verify that the scoring network's parameters contribute exclusively through the allocation strategy by forcing uniform output scores while keeping the scorer active.

Table 15: Degeneration test: scorer parameters contribute zero when allocation is uniform (DeiT-Ti).

Configuration	**Top-1 (%)**
DeiT-Ti baseline (no scorer)	72.2
DeiT-Ti + DART (scorer present, forced uniform)	72.2
DeiT-Ti + DART (scorer present, adaptive)	73.8

Rows 1 and 2 are identical, confirming that the scorer's extra parameters ($\sim$1M frozen CNN extractor plus a lightweight trainable scoring head comprising $<2\%$ of total parameters) provide no benefit when allocation is uniform. The improvement is entirely attributable to learned non-uniform allocation.

Parameter-Matched Control. To rule out parameter-count effects at the DeiT-S scale, we reduce the backbone embedding dimension from 384 to 366 so that the total parameter count of DeiT-S + DART equals the original 22M of DeiT-S alone.

Table 16: Parameter-matched control (DeiT-S, 448 source, 196 tokens).

Configuration	**Total Params**	**Top-1 (%)**
DeiT-S baseline (dim = 384)	22M	79.8
DeiT-S (dim = 366) + DART	22M	80.4
DeiT-S (dim = 384) + DART	24M	80.6

Even with total parameters held constant, adaptive allocation provides $+0.6\%$, confirming the gain is not attributable to increased model capacity.

B.8 Statistical Analysis of Spatial Token Allocation

To quantify DART's allocation behavior, we analyze the ImageNet-1K validation set using official ILSVRC localization bounding-box annotations. For each image, we run a forward pass with DeiT-S + DART-Flow at 448 input resolution and extract the spatial rectangular coordinates of all 196 tokens. Tokens are classified as foreground or background based on whether their center falls inside the annotated bounding box.

DART allocates on average 131 tokens to the foreground versus 119 for the uniform grid ($+12$). The mean foreground token area is 66% of the background token area (density ratio $1.51\times$), meaning the foreground receives 51% higher spatial resolution. On 87.5% of images, DART assigns at least as many foreground tokens as the uniform grid.

Table 17: Foreground token allocation stratified by object size (DeiT-S + DART-Flow, ImageNet-1K val).

Object Size Percentile	**DART**	**Uniform Grid**	**Surplus**
P10 (smallest objects)	40	28	+12
P25	89	69	+20
P50 (median)	145	126	+19
P75	182	182	0
P90 (largest objects)	196	196	0

The surplus grows as the object shrinks: at P25 DART allocates 20 extra tokens, equivalent to roughly 1.5 additional rows of a 14×14 grid. When the object fills most of the image (P75+), both methods converge.

B.9 Extended Comparison with Adaptive Tokenization Methods

We compare DART against three categories of related methods: Token Merging (ToMe) (Bolya et al., 2023) progressively merges similar tokens within the backbone; ∂HT (Aasan et al., 2025) is a differentiable hierarchical tokenizer producing variable-length (ragged) sequences; SPFormer (Mei et al., 2025) employs a dual-branch architecture with iterative cross-attention.

Table 18: Comparison with adaptive tokenization and token reduction methods on DeiT-S.

Method	**Tokens**	**Top-1 (%)**	**FLOPs**	**Rel. Throughput**
DeiT-S (Baseline)	196	79.8	4.61G	$1.00\times$
ToMe r=8	$\sim$164	79.7	3.43G	—
ToMe r=13	$\sim$143	79.4	2.71G	—
ToMe r=16	$\sim$131	79.1	2.30G	—
∂HT	variable	80.0	—	$0.40\times$
DART (ours)	196	**80.6**	4.84G	$0.95\times$
DART (ours)	144	79.9	3.60G	—

At the standard 196-token budget, DART improves accuracy by $+0.8\%$ with only a 5% throughput reduction, whereas ToMe requires reducing tokens to take effect and provides no gain at the standard budget. At comparable FLOPs ($\sim$3.5G), DART (79.9%) still exceeds the baseline, while ToMe at similar FLOPs falls below (79.7%). ∂HT achieves 80.0% but at a 60% throughput reduction due to ragged tensor outputs. At the DeiT-Ti scale, DART (73.8%, 1.32G FLOPs) matches SPFormer-T (Mei et al., 2025) (73.6%, 1.3G FLOPs) while preserving the backbone architecture entirely.

C Experimental Setup

C.1 Architecture Details

Our DART module uses a lightweight CNN to extract spatial features, from which a scoring head predicts per-location importance scores. The specific configurations are summarized in Table 19.

Table 19: Feature extractor configurations for various backbone models.

Model	Feature Extractor
DeiT-Ti	MobileNetV3 Small[:11]
DeiT-S	MobileNetV3 Large[:15]
Vim-Ti	MobileNetV3 Small[:13]
Vim-S	MobileNetV3 Large[:17]
VideoMamba-Ti	MobileNetV3 Small[:13]

C.2 Training Configurations

To ensure a fair and direct comparison, our experimental setup adheres closely to the established training protocols of the baseline models. Unless otherwise specified, all models were trained using the official, publicly available configurations provided by the original authors of DeiT, Vim, VideoMamba, and Swin Transformer.

The only exception is the long-sequence fine-tuning experiments conducted on DeiT-S, as detailed in Section 4.2. For these specific experiments, we used the following fine-tuning hyperparameters, which can be summarized with the command-line arguments: `-epochs 30 -weight-decay 1e-8 -lr 5e-6 -warmup-epochs 0 -sched step -decay-rate 1`.

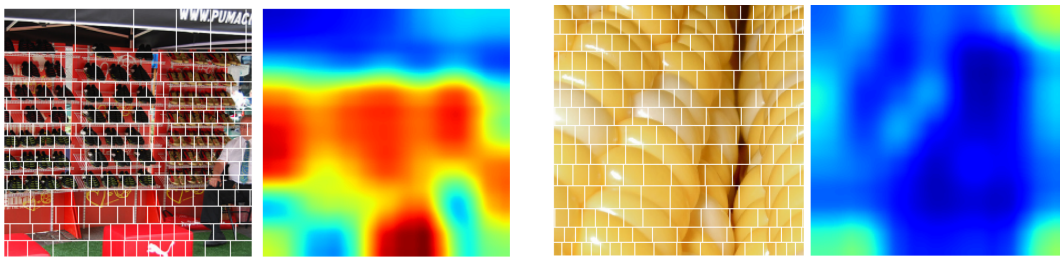
D Qualitative Analysis and Visual Comparison

D.1 Qualitative Analysis of Challenging Cases

To further probe the behavior of DART, we present a qualitative analysis of two challenging cases that reveal both its strengths and inherent limitations. These examples demonstrate how the model dynamically adapts its tokenization strategy in response to vastly different image characteristics.

Case 1: High-Density Scenes. Figure 8a shows a case with many densely packed small objects. Both the baseline and DART fail to classify correctly at 196 tokens, though DART correctly identifies the region of interest.

Case 2: Uniform-Content Images. Figure 8b shows an image with global texture and no distinct foreground. The scoring network produces a near-uniform score map, and DART reverts to a standard grid, consistent with the degeneration property verified in Section 4.6.



(a) High-density small objects: DART concentrates tokens on the region of interest.

(b) Uniform-content image: DART reverts to a near-uniform grid.

Figure 8: Qualitative comparison on two distinct input types.

D.2 Qualitative Analysis of Partitioning

To provide a more intuitive understanding of DART's behavior, we present additional partitioning visualizations on images from the ImageNet-1K validation set in Figures 9 and 10. These examples further demonstrate that DART successfully learns to allocate a denser token budget to information-rich areas, such as object contours and complex textures, while merging low-information background regions into larger patches.

D.3 Partitioning Strategy Comparison

Figure 11 provides a clear visual comparison of the two partitioning strategies. The `DART-Grid` method (middle) adapts patch sizes but is constrained to a fixed row-column structure. In contrast, our main `DART-Flow` method (left) mitigates this issue. Because each row can have independent column boundaries, tokens can concentrate in the most salient regions without the constraint that all rows share identical column partitions.

E Method Details

E.1 Architectural Overview

To provide a comprehensive understanding of the DART framework, we present a detailed architectural diagram in Figure 12. The system operates as a dual-stream pipeline, ensuring that the content-aware partitioning is driven by high-level semantics while the final representation preserves low-level texture details.

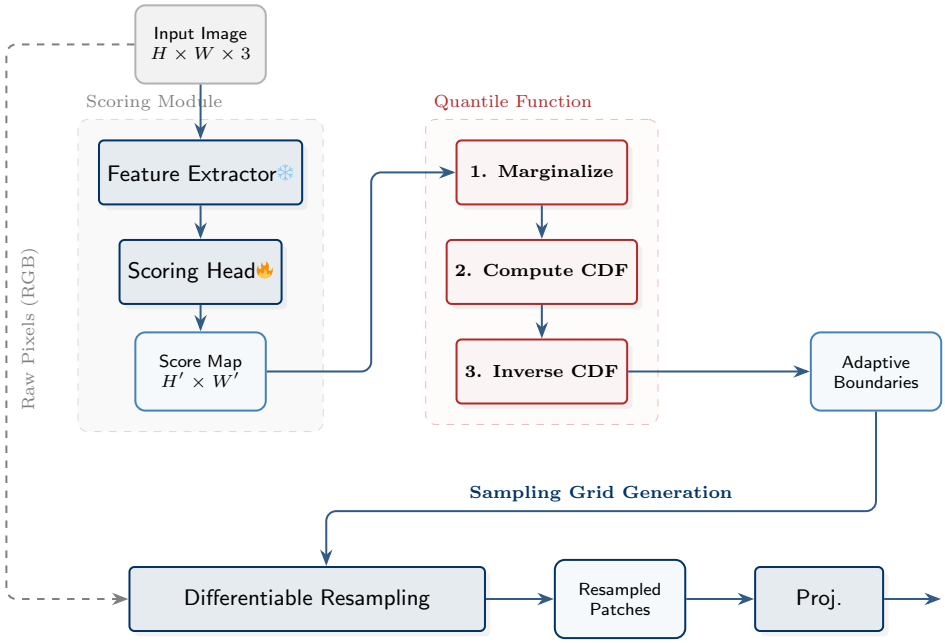


Figure 12: **Detailed Architectural Overview of DART.** The pipeline consists of a control stream (top) for boundary computation and a data stream (bottom/left bypass) for feature preservation.

The data flow proceeds through three key stages:

1. **Scoring Module (Control Stream):** The input image is processed by a lightweight CNN feature extractor followed by a trainable scoring head. This module generates a low-resolution Score Map representing the spatial information density.

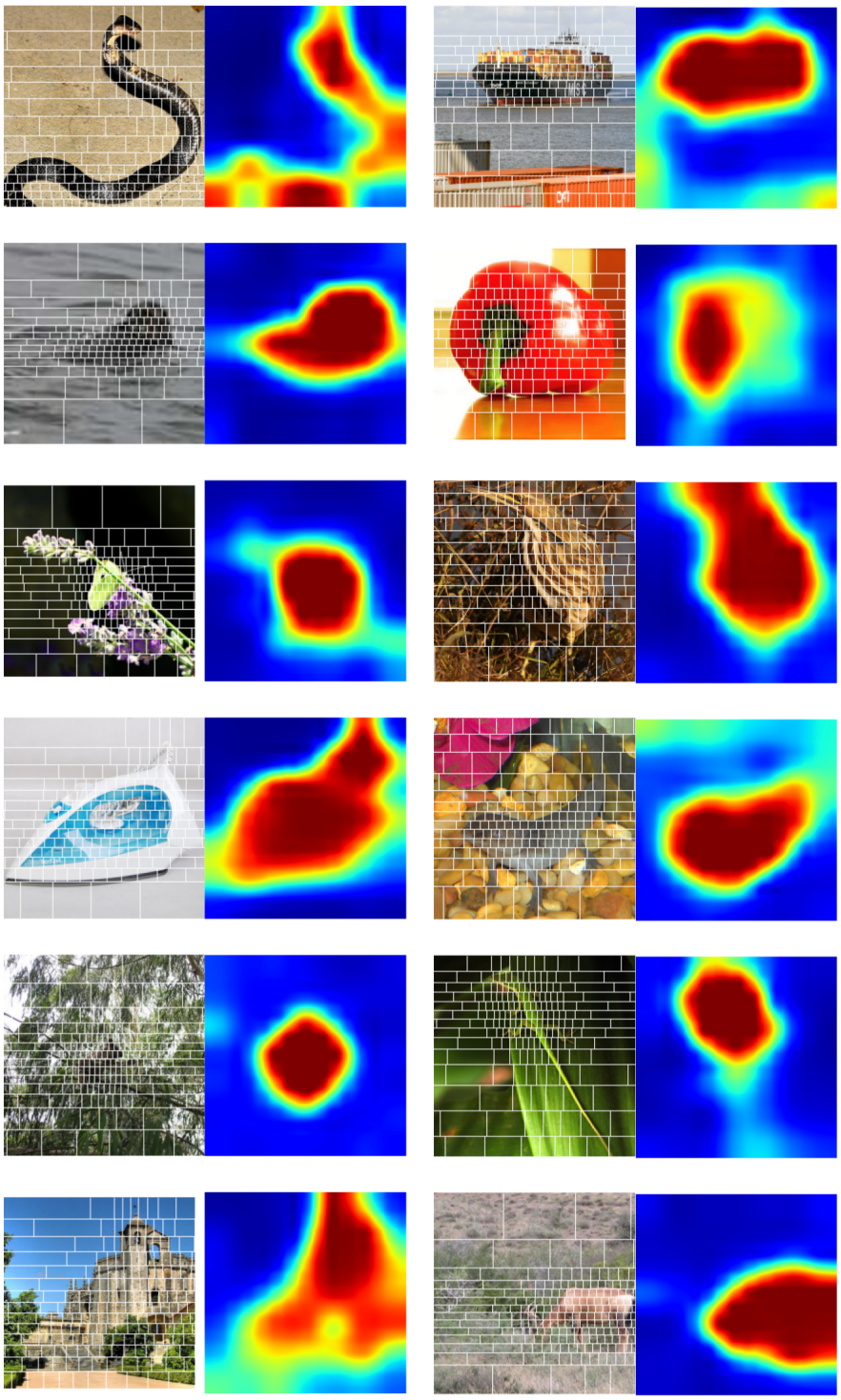


Figure 9: Partition examples produced by our DART model (Part 1).

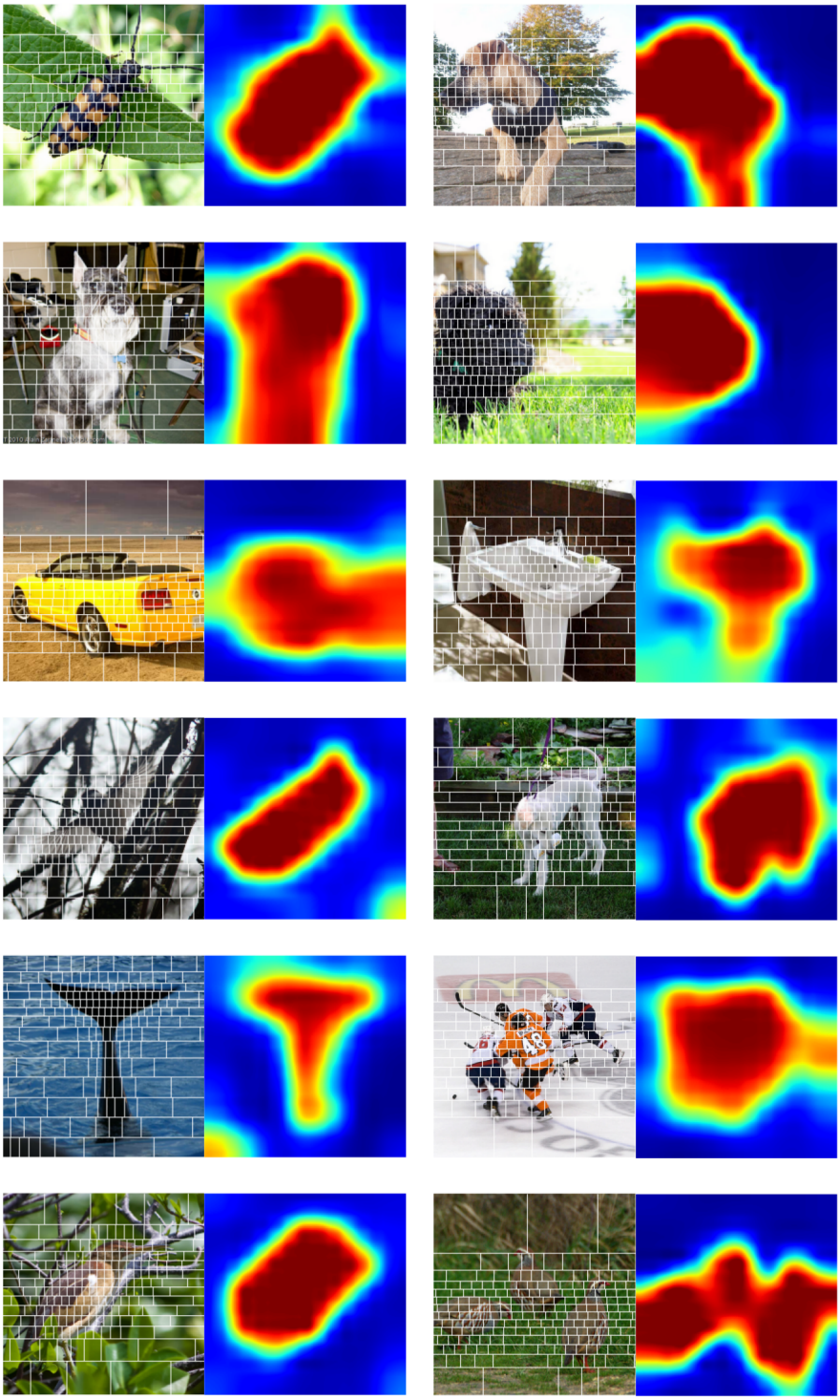


Figure 10: Partition examples produced by our DART model (Part 2).

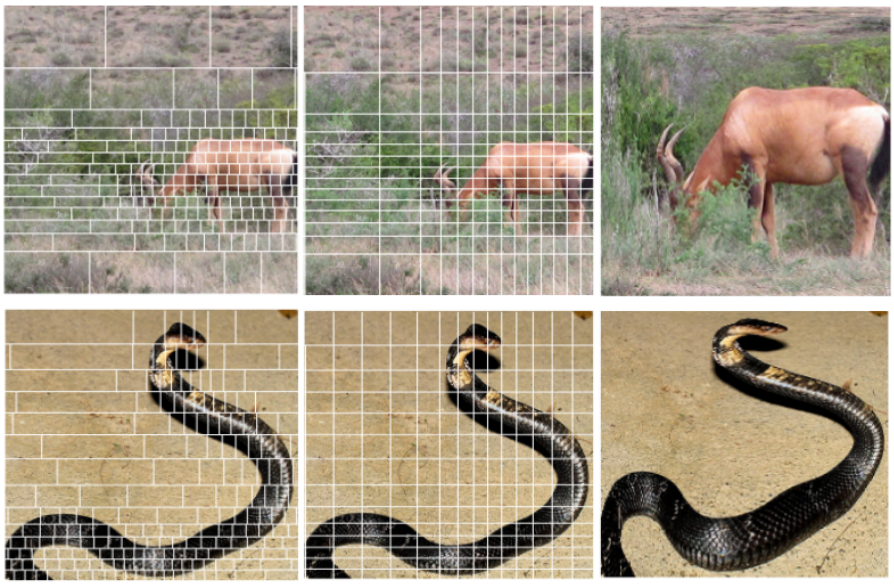


Figure 11: A visual comparison of partitioning strategies. **Left:** Our main `DART-Flow` method. **Middle:** The grid-preserving `DART-Grid` method. **Right:** A visualization simulating the model's input for the `DART-Grid` method.

2. **Differentiable Quantile Function (Algorithm):** The score map is converted into a probability distribution. We sequentially apply marginalization, CDF computation, and the inverse CDF operation. This process is differentiable and outputs the adaptive partition boundaries.
3. **Differentiable Resampling (Execution):** The computed boundaries define a non-uniform grid. The Differentiable Resampling module takes two inputs: the generated grid coordinates and the raw pixel stream (which bypasses the scoring network). It extracts content-aware patches via bilinear interpolation. Finally, these patches are linearly projected and added with positional embeddings to form the final sequence of tokens.

E.2 Mathematical Formalism of the Differentiable Quantile Algorithm

Our objective is to compute quantile boundaries from a discrete probability density function (PDF) within a differentiable framework. Given a 1D discrete probability distribution $S = \{s_0, s_1, \ldots, s_{L-1}\}$, where $\sum_{i=0}^{L-1} s_i = 1$ and $s_i \geq 0$, we aim to find a set of boundary points $\{x_1, x_2, \ldots, x_{K-1}\}$ corresponding to a set of target cumulative probabilities (quantiles) $Q = \{q_1, q_2, \ldots, q_{K-1}\}$.

1. From Discrete PDF to Piecewise-Constant Function. We model the discrete PDF S as a piecewise-constant function $f(x)$ over the continuous interval $[0, L)$:

$$f(x) = s_{\lfloor x \rfloor} \quad \text{for} \quad x \in [0, L) \tag{1}$$

2. Constructing a Piecewise-Linear CDF. The CDF, $C(x) = \int_0^x f(t)\,dt$, is a continuous, piecewise-linear function. The value at any integer point j is:

$$C(j) = \sum_{i=0}^{j-1} s_i \quad (\text{with } C(0) = 0) \tag{2}$$

Within any interval $[j, j+1)$, the CDF increases linearly:

$$C(x) = C(j) + (x - j) \cdot s_j \quad \text{for} \quad x \in [j, j+1) \tag{3}$$

3. Evaluating the Inverse CDF. Because C is continuous, piecewise-linear, and strictly increasing, its inverse C^{-1} is also a continuous piecewise-linear function. For a target quantile q_k, the boundary $x_k = C^{-1}(q_k)$ can be written in closed form. On the j-th linear piece of C (i.e., the segment where $C(j) \leq q_k < C(j+1)$), the expression reduces to:

$$x_k = j + \frac{q_k - C(j)}{s_j} \tag{4}$$

with $C(j)$ being the CDF value at segment start and s_j the segment slope.

4. Summary. The final expression for x_k in Equation 4 is composed of differentiable operations (addition, subtraction, division, and summation). The interval selection step identifies which linear segment of the CDF contains the target quantile. Within each segment, the boundary position is an analytic function of the input scores. Because the CDF is continuous and monotonically increasing, the selected segment is locally stable under small perturbations of the input distribution, so gradients propagate through the boundary computation in the standard way. This piecewise-linear structure is analogous to ReLU activations and is handled natively by automatic differentiation frameworks.

E.3 Algorithm Pseudocode

To enhance clarity and facilitate reproducibility, we present detailed pseudocode for our main DART (DART-Flow) algorithm in Figure 13.

```
def DART_tokenizer(x, scoring_net, pos_embed, N_h=14, N_w=14):
    N_total = N_h * N_w
    # 1. Generate 2D Probability Distribution from scores
    score_map = scoring_net(x)
    prob_map_2d = normalize_scores(score_map)  # sigmoid + sum-to-1
    initial_pdf = prob_map_2d.flatten(1)

    # 2. Stage 1: y-axis marginal -> horizontal boundaries
    pdf_2d_view = initial_pdf.view(-1, H_score, W_score)
    marginal_prob_y = pdf_2d_view.sum(dim=2)
    cdf_y = torch.cumsum(marginal_prob_y, dim=1)
    y_boundaries = inverse_cdf(cdf_y, num_quantiles=N_h)
    row_heights = y_boundaries.diff(prepend=0)

    # 3. Stage 2: Virtually flatten rows -> all token boundaries
    resampled_pdf = resample_1d_by_grid(initial_pdf, grid=row_heights)
    resampled_pdf /= resampled_pdf.sum(dim=-1, keepdim=True)
    cdf_resampled = torch.cumsum(resampled_pdf, dim=1)
    final_edges = inverse_cdf(cdf_resampled, num_quantiles=N_total)

    # 4. Dynamic patch sampling from original image
    patches = dynamic_image_patch_sample(x, row_heights, final_edges, (16,16))

    # 5. Project patches + transformed positional embeddings
    tokens = proj(patches)
    final_pos_embed = transform_pos_embed(pos_embed, row_heights, final_edges)
    return tokens + final_pos_embed
```

Figure 13: Pseudocode for our main DART (DART-Flow) tokenizer.

To complement the mathematical formalism presented in Section E.2, we also provide detailed pseudocode for our differentiable quantile algorithm. Figure 14 describes the practical, highly-optimized implementation used in our work. Rather than an iterative approach, it showcases a parallelized and vectorized solution that processes an entire batch of distributions simultaneously. The algorithm leverages efficient tensor operations like broadcasting to find interval indices and `gather` to retrieve the necessary values for interpolation, completely avoiding slow, sequential loops. This design

is critical for performance, as it fully exploits the parallel architecture of modern GPUs, ensuring the DART tokenizer remains a computationally lightweight component.

```
def inverse_cdf(pdf, p_values, eps=1e-8):
    N, L = pdf.shape; K = p_values.shape[0]
    # 1. Compute CDF
    cumsums = torch.cumsum(pdf, dim=1)
    # 2. Find interval index for each quantile
    mask = (cumsums.unsqueeze(-1) >= p_values.view(1, 1, -1))
    j_indices = torch.argmax(mask.int(), dim=1)
    # 3. Gather interpolation values
    prev_j = torch.clamp(j_indices - 1, min=0)
    prev_area = torch.gather(cumsums, dim=1, index=prev_j)
    prev_area = torch.where(j_indices == 0, 0.0, prev_area)
    pdf_val = torch.gather(pdf, dim=1, index=j_indices)
    edge_val = j_indices.float()
    # 4. Linear interpolation
    quantiles = edge_val + (p_values.unsqueeze(0) - prev_area) / (pdf_val + eps)
    return quantiles
```

Figure 14: Pseudocode for the parallelized, differentiable quantile computation algorithm (`inverse_cdf`).

F Extended Related Work: The Broader Efficiency Landscape

In the main text, we focused on methods directly related to dynamic token reduction and adaptive tokenization. Here, we broaden our scope to discuss the wider landscape of efficient Vision Transformer designs.

Static Efficient Architectures. A significant body of work focuses on designing lightweight, static backbones, often utilizing Neural Architecture Search (NAS) to optimize for specific hardware constraints. Hybrid Architectures like MobileViT (Mehta & Rastegari, 2022) and EdgeViT (Pan et al., 2022) combine CNNs with Transformers, while Pure Efficient ViTs like LeViT (Graham et al., 2021) and EfficientFormer (Li et al., 2022) optimize macro-architectures for latency. However, these specialized designs diverge from the standard uniform ViT architectures that currently dominate the landscape of large-scale foundation models. In contrast, DART is designed as a modular enhancement for the dominant uniform paradigm.

Dynamic Depth and Early Exiting. Another prominent direction for efficiency is reducing the computational depth rather than the spatial width. Early-Exit methods allow "easy" samples to bypass deeper layers of the network once a confidence threshold is met. These methods reduce compute by skipping *layers*, whereas DART reduces compute by processing fewer *tokens* per layer. Methodologically, DART is orthogonal to these approaches.

Global Token Summarization. Methods like TokenLearner (Ryoo et al., 2021) and Perceiver (Jaegle et al., 2021) adopt an attention-based approach to learn a small set of abstract latent tokens that summarize the global context. While highly efficient for classification tasks, these methods typically perform weighted spatial pooling, sacrificing the grid-like spatial topology required for location-sensitive tasks. DART, conversely, preserves the spatial structure through region-based partitioning and coordinate-transformed positional embeddings.