LEARNING TO ALLOCATE TOKENS FOR VISION TRANSFORMERS

Anonymous authors
Paper under double-blind review

ABSTRACT

Vision Transformers (ViTs) are a critical backbone of modern computer vision frameworks. Standard ViT architectures work by representing uniformly sized patches as single tokens in a sequence. However, this design is poorly aligned with the true nature of images, where information density is typically not spatially uniform. This results in uninformative image patches receiving the same token allocation as highly informative image patches. To address this limitation, we propose the APE (Adaptive Patch Encoder) architecture, an extension of the traditional ViT that estimates patch importances and dynamically allocates and merges tokens to maximize the information carried in the ViT input sequence. We show that APE is highly efficient while achieving state-of-the-art performance on benchmarks across classification, detection, segmentation, and VQA. Quantitative results across a diverse range of tasks validate that APE's learned saliency metric maximizes the compute-performance tradeoff without domain-specific degradation. Checkpoints and code will be released upon acceptance.

1 INTRODUCTION

Transformer architectures were originally proposed as a backbone to replace RNN and CNN architectures for natural language tasks (Vaswani et al., 2017). The introduction of the Transformer architecture to the visual domain (Dosovitskiy et al., 2020) began a paradigm shift in computer vision, quickly establishing state-of-the-art performance across the visual perception spectrum, from classification (Liu et al., 2021) to 3D geometry estimation (Zhao et al., 2021; Wang et al., 2025). These models work by tokenizing images into uniform, non-overlapping patches, and represent these patches as tokens in a sequence given as input to some Transformer backbone. The Transformer backbone then uses attention to understand inter-patch relationships. However, the standard practice of splitting an image into fixed patches and giving them equal weight implicitly assumes that information is distributed evenly across the spatial domain. This assumption is intuitively false: natural scenes consist of regions with very different information densities, where low-saliency background regions (e.g., sky, roads, out-of-focus areas) coexist with important subjects. Additionally, the importance of different regions may be *task-dependent*. For example, a fine-grained classifier for species identification may benefit from attending to subtle texture and color cues (e.g., fur patterns or leaf venation) that are largely uninformative for coarse recognition tasks.

In a standard ViT backbone, the equal distribution of tokens across patches results in a massive allocation of compute to redundant tokens. Recent analyses (Nikzad et al., 2025; Darcet et al., 2023) reveal that a significant portion of tokens given to modern ViT architectures encode little patch-specific information, with many low-saliency patches being repurposed by the network as scratchpads for global context rather than contributing distinctive local features. These tokens nonetheless consume quadratic resources in the self-attention mechanism. This inefficiency limits the accuracy/compute trade-off in the natural image domain and creates a prohibitive barrier to scaling ViTs to high-resolution and dense-prediction tasks (Xie et al., 2025; Astruc et al., 2025).

Prior efforts have attempted to mitigate the poor inductive bias of uniform token allocation by pruning entire tokens within network layers (Rao et al., 2021) or by imposing simple, static functions to estimate patch importance from pixels (Choudhury et al., 2026). Pruning destroys the spatial structure required for fine-grained and dense prediction tasks (e.g., segmentation), as discarded regions leave "holes" in the feature map, while static functions operating on pixels are often poorly aligned with

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.

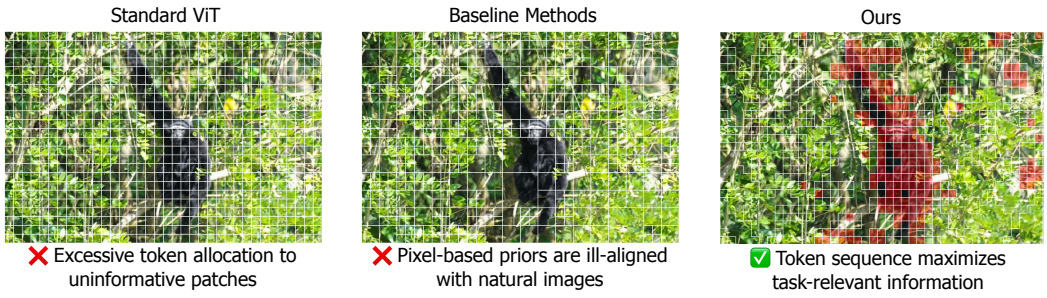


Figure 1: A comparison of how different ViT-based architectures allocate tokens. Red shading indicates allocation of more than one token per patch.

actual patch saliency in many domains. We demonstrate an example of this misalignment in Fig. 1. Traditional methods expend significant representational capacity on uninformative image regions, while other baselines impose inductive biases on patch importance that are ill-aligned with natural images. Our method achieves stronger performance and efficiency by estimating feature importance from information-theoretic embedding properties.

In this work, we demonstrate that ViT architectures can be made increasingly performant and efficient by estimating token saliency *directly from latent patch representations*. We propose the APE framework, a new ViT architecture that abandons uniform embedding of patches into tokens. Unlike previous works, APE prunes the token set directly after patch embedding and estimates token importance from latent patch representations instead of pixels, resulting in stronger performance and efficiency. By realigning the computational allocation with the intrinsic information structure of the visual signal, APE consistently improves the accuracy/compute trade-off across a wide suite of computer vision tasks. We summarize our contributions as follows:

1. We propose APE, a ViT-based architecture that estimates patch importances *from their representations*. APE dynamically picks the most information-dense token sequence by selecting and merging candidate tokens.
2. We experimentally validate the increased accuracy and efficiency of APE, and show emergent properties of APE's token allocation mechanism.
3. We demonstrate that APE generalizes strongly across domains, learning task-dependent metrics of patch saliency and improving accuracy/compute trade-offs relative to strong uniform ViTs and adaptive patching baselines.

2 Related Work

Vision Transformers. The introduction of the Vision Transformer (ViT) (Dosovitskiy et al., 2020) catalyzed a massive paradigm shift from convolutional inductive biases to global self-attention mechanisms. While ViTs demonstrate immense scalability and representational capacity, their $\mathcal{O}(N^2)$ computational complexity severely restricts their application to high-resolution dense tasks such as Earth observation (Astruc et al., 2025) and 3D geometry estimation (Wang et al., 2025). Hierarchical adaptations like the Swin Transformer (Liu et al., 2021) mitigate this via local windows, yet foundational models still largely rely on deterministic, uniform grid patchification, operating under the assumption that information is homogeneously distributed across natural images.

Efficient Transformer architectures. To alleviate the quadratic bottleneck, token sparsification methods such as DynamicViT (Rao et al., 2021), A-ViT (Yin et al., 2022), and ATS (Fayyaz et al., 2022) employ prediction modules to permanently prune redundant tokens. However, pruning destroys spatial contiguity, generating feature-map "holes" that are highly detrimental to dense prediction tasks. Conversely, token merging strategies like ToMe (Bolya et al., 2023) and PatchMerger (Renggli et al., 2022) maintain global information by fusing similar tokens with comparatively low overhead, but operate after full patch embedding and risk blurring foreground/background semantic boundaries. Furthermore, recent analyses reveal that standard ViTs repurpose uninformative patches to store global context, inducing high-norm artifacts unless explicit register tokens are utilized (Darcet et al.,

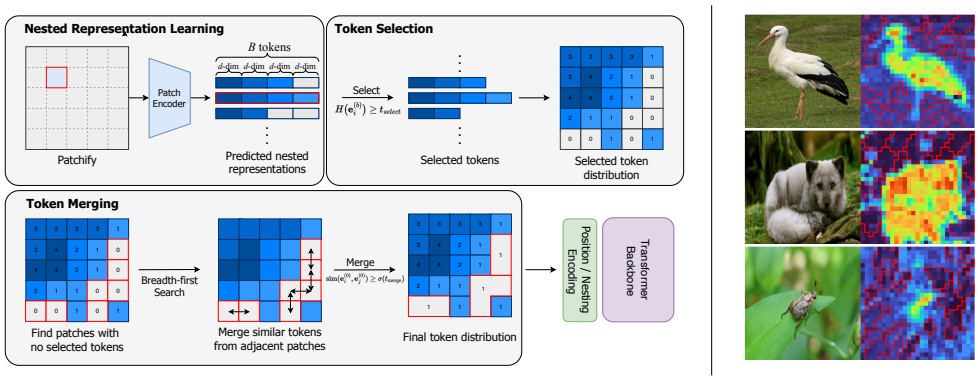


Figure 2: Left: Overview of the APE architecture. We divide the pipeline into nested representation learning, token selection, and token merging. Right: Examples of learned patch importance and merged regions. Merged tokens are shown with red borders.

2023). Importantly, all of these methods operate *within* the Transformer backbone, after the full token sequence has been embedded; APE instead acts at the patch-encoding stage, and is therefore complementary to and composable with these in-Transformer techniques.

Adaptive discretization and content-aware tokenization. Moving beyond rigid token grids, methods utilizing non-uniform structures (Yin et al., 2025; Liu et al., 2023) or patch pixel entropy (Choudhury et al., 2026) allocate variable patch sizes. However, allocating patches based on pixel-level variance frequently misallocates compute to semantically irrelevant, high-frequency textures (e.g., gravel or foliage). Moreover, because these decisions are driven by low-level image statistics rather than semantic patch relevance, they can allocate patch-encoding compute inefficiently under constrained inference budgets (Fig. 3).

More sophisticated content-aware mechanisms, such as ImagePiece (Yoa et al., 2025) and SATA (Nikzad et al., 2025), group tokens via spatial autocorrelation, while TORE (Olszewski et al., 2025) filters tokens using attention map entropy. APE distinguishes itself by evaluating patch importance directly from latent representations rather than pixels, utilizing an information-theoretic metric robust to low-level noise.

Nested Representation Learning. Matryoshka Representation Learning (MRL) (Kusupati et al., 2022) enforces a coarse-to-fine nested structure within representation vectors, allowing for variable dimensionality during deployment with limited performance loss. Recent extensions like the Matryoshka Query Transformer (Hu et al., 2024) adapt this for visual query compression. APE adapts this paradigm as an internal structural regularizer at the patch-embedding stage, utilizing a nested Shannon entropy loss to strictly concentrate information density. This yields a natural parameter-free metric for token saliency.

3 METHODOLOGY

APE estimates patch importance to dynamically allocate tokens to the Transformer backbone in three stages: nested representation learning, *selection* of informative tokens, and *merging* of unselected tokens. We describe each below.

We define a patch encoder E that produces a nested representation consisting of B constituent embeddings. For an input image I, APE first patchifies I into N non-overlapping square patches. These patches are fed into E, which produces a sequence of shape $N \times (B \cdot d)$, where B controls the maximum number of tokens that may be allocated per patch. We obtain B constituent d-dimensional embeddings from each $(B \cdot d)$-dimensional embedding by splitting it along the channel axis. We define an embedding $\mathbf{e}_i^{(b)}$ as the b-th nested patch embedding from p_i. The set of embeddings $\mathcal{E}^{(b)}$ at nesting level b is given by:

$$\mathcal{E}^{(b)} = \left\{ \mathbf{e}_i^{(b)} \;\middle|\; i = 1, 2, ..., N \right\} \quad , \quad \mathbf{e}_i^{(b)} = E(p_i)[b \cdot d : (b+1) \cdot d], \quad b = 0, 1, \ldots, B-1 \quad (1)$$

3.1 NESTED REPRESENTATION LEARNING

To select the most informative token embeddings from the full $B \cdot d$-dimensional embedding, we impose a nested structure across the B constituent d-dimensional embeddings. This encourages earlier splits to carry denser information than later splits, so that truncated representations remain useful at predefined dimensions.

We encourage this structure with an entropy-ordering loss over the split embeddings:

$$\mathcal{L}_{\text{nest}} = \sum_{i=0}^{B-2} \text{ReLU}\Big(H\big(E[(i+1)\cdot d : (i+2)\cdot d]\big) - H\big(E[i\cdot d : (i+1)\cdot d]\big)\Big), \tag{2}$$

where $H(\cdot)$ denotes the Shannon entropy of a split embedding after channel-wise softmax normalization with fixed temperature τ_H. This loss penalizes cases where a later split has higher entropy than the preceding split, thereby encouraging entropy to decrease monotonically across the nested representation. Since lower entropy corresponds to a more concentrated activation distribution, this ordering acts as a proxy for placing denser, more informative features in the earlier splits.

3.2 TOKEN SELECTION

We aim to select tokens globally such that total information across all patches is maximized. First, we select tokens that are learned to be sufficiently "significant" or information-dense. We treat the union of all sets $\mathcal{E}^{(b)}$ as the initial pool of candidate tokens. Whereas prior adaptive patching methods score saliency using a task-agnostic heuristic (Choudhury et al., 2026), scoring tokens in the learned embedding space makes importance task-adaptive by construction. We use Shannon entropy H as a proxy for information density. H is the canonical information-theoretic measure of the descriptive complexity of a vector, and the nested representation loss of section 3.1 further calibrates H as a monotone proxy for information density by construction. We evaluate each token using H to build the set of selected tokens as follows:

$$\mathcal{E}_{\text{select}} = \left\{ \mathbf{e}_i^{(b)} \in \mathcal{E}^{(b)} \;\middle|\; H\big(\mathbf{e}_i^{(b)}\big) \geq t_{\text{select}}, \quad b = 0, 1, \ldots, B-1 \right\} \tag{3}$$

The threshold t_{select} is learned end-to-end via a straight-through estimator (STE) (Bengio et al., 2013). The hard indicator $m_i = \mathbb{1}[H(\mathbf{e}_i^{(b)}) \geq t_{\text{select}}]$ is used in the forward pass, and replaced with a smooth surrogate in the backward pass:

$$s_i = \sigma\big((H(\mathbf{e}_i^{(b)}) - t_{\text{select}})/\tau_{\text{sel}}\big), \qquad g_i = m_i + (s_i - \text{sg}[s_i]), \tag{4}$$

where σ is the sigmoid, $\text{sg}[\cdot]$ the stop-gradient operator, and each selected token is multiplied by g_i. We discuss this mechanism in more detail in the Appendix.

Because token selection operates on the *learned* embedding space rather than on raw pixels, the notion of "informative" image regions is learned to be highly aligned with the task. However, this selection procedure potentially leaves large image regions with no selected tokens. Next, we describe how low-resolution information about such regions may be described with minimal additional cost.

3.3 MERGING TOKENS

Since the token selection procedure may leave some spatial patches with no descriptor tokens, we introduce a merging mechanism that provides low-resolution coverage for these regions without significantly increasing the number of tokens given to the Transformer.

Let $\mathcal{S}$ be the set of patches where at least one token describing the patch was selected. Because the nesting loss equation 2 encourages $H(\mathbf{e}_i^{(b)})$ to be non-increasing in b, $\mathbf{e}_i^{(0)}$ is selected whenever any of $\mathbf{e}_i^{(b)}$ is, so we can equivalently characterize $\mathcal{S}$ by the level-0 embedding:

$$\mathcal{S} = \left\{ i \in \{1, 2, \ldots, N\} \;\middle|\; \mathbf{e}_i^{(0)} \in \mathcal{E}_{\text{select}} \right\}.$$

Also let $\bar{\mathcal{S}} = \{1, \ldots, N\} \setminus \mathcal{S}$ be the unselected patches. We run a multi-source breadth-first search (BFS) on every patch in $\bar{\mathcal{S}}$ and maintain a union-find data structure over $\bar{\mathcal{S}}$ that assigns each

participating patch p_i to a cluster $\mathcal{C}_i$. The BFS operates on a graph connecting each patch p_i to its 4 adjacent patches, which we denote $\mathcal{N}_4(i)$. For unselected patches $i, j \in \bar{\mathcal{S}}$, we merge their sets $\mathcal{C}_i, \mathcal{C}_j$ into each other if their first nested embeddings are sufficiently similar as follows:

$$A_{i,j} = \mathbb{1}\big[j \in \mathcal{N}_4(i)\big] \cdot \mathbb{1}\Big[\mathrm{sim}(\mathbf{e}_i^{(0)}, \mathbf{e}_j^{(0)}) \geq \sigma(t_{\mathrm{merge}})\Big] \tag{5}$$

$$\mathcal{C}_i, \mathcal{C}_j \leftarrow \mathcal{C}_i \cup \mathcal{C}_j \quad \text{if} \quad A_{i,j} = 1 \tag{6}$$

where $\mathrm{sim}(\cdot,\cdot)$ denotes cosine similarity in the level-0 embedding space, t_{merge} is a learned scalar parameter, and σ denotes the sigmoid function. Absorbed patches are then added to the BFS queue. For each cluster $\mathcal{C}_k$ that absorbed at least one patch, we construct a single *summary token*:

$$\mathbf{z}_k = \frac{1}{|\mathcal{C}_k|} \sum_{j \in \mathcal{C}_k} g_j \, \mathbf{e}_j^{(0)}, \qquad \mathbf{p}_k = \Big(\overline{x}_{\mathcal{C}_k},\ \overline{y}_{\mathcal{C}_k},\ 0 \Big), \tag{7}$$

where $\overline{x}_{\mathcal{C}_k}$ and $\overline{y}_{\mathcal{C}_k}$ are the mean grid coordinates of the absorbed patches and the nesting index is set to 0. The per-absorption gates g_j implement the same STE construction as Eq. 4 for t_{merge}. We leave precise implementation details to the Appendix.

The summary tokens are appended to the selected tokens and their fractional positions are recorded for the positional encoding, giving the Transformer both a coarse representation and an approximate spatial location for every merged region.

3.4 Encoding Position and Nesting Level

To jointly encode spatial position and nesting level in patch representations, we adapt the RoPE formulation (Su et al., 2024) of positional embeddings. We treat the nesting level as a third dimension alongside the two spatial dimensions, forming a three-dimensional rotary encoding. This allows the model to handle dynamic resolutions and to cleanly represent fractional patch indices produced by merging.

In standard RoPE, the per-head query and key dimension d_h is partitioned into pairs, and each pair (x_{2i}, x_{2i+1}) is rotated by an angle proportional to the token's scalar position p:

$$\begin{pmatrix} x'_{2i} \\ x'_{2i+1} \end{pmatrix} = \begin{pmatrix} \cos(p\,\theta_i) & -\sin(p\,\theta_i) \\ \sin(p\,\theta_i) & \cos(p\,\theta_i) \end{pmatrix} \begin{pmatrix} x_{2i} \\ x_{2i+1} \end{pmatrix}, \qquad \theta_i = \frac{1}{\beta^{\,2i/d_h}}, \tag{8}$$

where $\beta = 10{,}000$ is the frequency base. We extend this to the three-dimensional position tuple $\mathbf{p} = (p_x, p_y, p_s)$, where p_x, p_y are the spatial grid coordinates and $p_s \in \{0, \ldots, B{-}1\}$ is the Matryoshka split index. We partition d_h into three groups of even size:

$$d_h = d_x + d_y + d_s, \qquad d_x \approx d_y \approx d_s \approx \lfloor d_h/3 \rfloor, \tag{9}$$

and assign each group its own set of frequencies and position component. Concretely, for a dimension pair $(2i, 2i{+}1)$ belonging to group $g \in \{x, y, s\}$, we apply

$$\theta_i^{(g)} = \frac{1}{\beta^{\,2i/d_g}}, \qquad \begin{pmatrix} x'_{2i} \\ x'_{2i+1} \end{pmatrix} = \mathbf{R}\Big(p_g \cdot \theta_i^{(g)}\Big) \begin{pmatrix} x_{2i} \\ x_{2i+1} \end{pmatrix}, \tag{10}$$

where $\mathbf{R}(\alpha)$ is the standard $2{\times}2$ rotation matrix. Each group is rotated independently so that no cross-axis interference occurs. The resulting rotary embedding is applied to queries and keys inside every self-attention layer.

Because the position coordinates p_x and p_y are not necessarily on integer image coordinates (as is the case for summary tokens produced by the merging mechanism described in section 3.3), the encoding naturally interpolates between integer grid positions, giving the attention mechanism a smooth spatial signal for merged regions.

3.5 Implementation Details

Following baselines (Choudhury et al., 2026; Yin et al., 2025), we employ a small convolution-based model as the patch encoder. Specifically, we modify the patch encoder to use a small CNN with GELU activation functions. We provide additional results on baselines with identical patch encoders in the

Appendix. The patch representation is obtained by global average pooling and a linear projection to hidden dimensions specified by the ViT backbone size (Dosovitskiy et al., 2020; Wightman, 2019) with $B = 4$ tokens available for selection and merging. The final representation is obtained by mean-pooling the Transformer output over the dynamic-length token sequence. All experiments use the AdamW optimizer with β_1=0.9, β_2=0.999 and cosine annealed learning rate decay. For classification tasks we use RandAugment (Cubuk et al., 2020) during training. For dense prediction tasks, we use the EVA-02 (Fang et al., 2024) backbone and modify only the patch encoder. The STE temperatures τ_{sel} and τ_{merge} are learnable and linearly warmed up. We provide additional details in the Appendix.

3.6 Baselines

We take the widely used Masked Autoencoder architecture (MAE) (He et al., 2022) as a representative baseline for traditional uniform patch-based Vision Transformer architectures. We also consider simple masking and resizing baselines built upon MAE (Li et al., 2023a; He et al., 2022). We also compare against adaptive patching methods APT (Choudhury et al., 2026) for all tasks, and DART (Yin et al., 2025) for image classification and dense prediction tasks.

3.7 Tasks and Datasets

Classification. We fine-tune APE from the MAE (He et al., 2022) checkpoint on ImageNet (Deng et al., 2009) using the ViT-B and ViT-L backbones (Dosovitskiy et al., 2020), applying RandAugment (Cubuk et al., 2020) during training as consistent with baselines. ImageNet comprises over 1.2M training images across 1000 classes. To assess fine-grained transfer, we further fine-tune from the same MAE checkpoint on iNaturalist 2021 (iNat21) (Van Horn & Mac Aodha, 2021) and on CUB-200-2011 (Wah et al., 2011). Both benchmarks require discriminating small visual differences between visually similar classes within a narrower domain than general natural imagery. We provide additional evaluations in the Appendix.

Dense prediction. For object detection on COCO (Lin et al., 2014) and semantic segmentation on ADE20K (Zhou et al., 2017; 2019), we adopt the default EVA-02 (Fang et al., 2024) dense-prediction recipe for APE and all baselines. COCO contains roughly 200K images annotated for detection across 80 object classes. ADE20K contains over 27K images with pixel-level labels for semantic segmentation.

Vision-language. To evaluate APE as a vision encoder for vision-language models, we substitute the vision encoder of the standard LLaVA-1.5-7B and LLaVA-1.5-13B (Liu et al., 2024a) backbones with an APE encoder while leaving the projector, language model, and instruction-tuning recipe unchanged. For vision-language fine-tuning we follow the protocol of APT (Choudhury et al., 2026).

4 Results

We evaluate APE on image classification, dense prediction, and vision-language modeling, and study its inference-time scaling behavior in Sec. 4.4.

4.1 Image Classification

Does learned, content-aware token allocation actually change how well a ViT classifies natural images? Tab. 1 reports ImageNet top-1 accuracy and inference GFLOPs for APE against MAE (He et al., 2022), APT (Choudhury et al., 2026), and DART (Yin et al., 2025). At every scale APE strictly beats dense MAE in both accuracy and computational cost. At matched resolution and patch size, APE also consistently beats the adaptive baselines in performance with only small additional overhead cost. Notably, APE's efficiency allows training at higher patch resolutions for stronger models, while still maintaining less computational cost when compared to dense baselines.

Fine-grained ecological classification. Ecological recognition requires content-aware tokenization in a domain that is visually different from the general ImageNet domain. Ecological images often contain high-frequency details (e.g. grass, branches, and waves) that contain low pixel compressibility but high semantic compressibility. In this domain, APE's *learned* token allocation mechanism strictly

Model	Res/Patch	Acc ↑	GFLOPs ↓
ViT-B^{MAE}	384/16	84.2	49.4
APT	384/16	84.2	**21.9**
DART	384/16	84.2	34.1
ours	384/16	**84.7**	30.5
APT	384/12	84.6	**29.7**
DART	384/12	84.5	38.3
ours	384/12	**85.2**	42.13

(a) ViT-B scale

Model	Res/Patch	Acc ↑	GFLOPs ↓
ViT-L^{MAE}	336/14	86.1	174.7
APT	336/14	86.1	**76.8**
DART	336/14	86.2	120.6
ours	336/14	**86.5**	128.78
APT	336/12	86.3	**97.7**
DART	336/12	86.5	131.9
ours	336/12	**87.3**	150.24

(b) ViT-L scale

Table 1: ImageNet classification compared to MAE He et al. (2022) , APT Choudhury et al. (2026), and DART Yin et al. (2025) at ViT-B (left) and ViT-L (right) scales.

	iNat21-species	CUB-200-2011	avg. red. (time)	avg. red. (GLOPS)
ViT-B^{MAE}	76.3	86.7	0%	0%
APT	76.1	86.3	-20%	-44%
ours	**76.7**	**87.4**	**-24%**	**-47%**
ViT-L^{MAE}	84.2	89.8	0%	0%
APT	84.1	89.6	-22%	-45%
ours	**85.2**	**89.9**	**-24%**	**-49%**

Table 2: Classification performance and efficiency on iNaturalist-2021 and CUB-200-2011.

outperforms all baselines both in computational cost and performance on two benchmarks. Notably, APT fails to match the performance of the dense baseline across both datasets, indicating that static methods for token allocation are insufficient to prevent performance loss across all domains. We show additional geospatial domain classification results in the Appendix.

Model	Res	Speedup	mAP	AP50
EVA-02-B	1536	–	58.93	77.85
APT	1536	+14%	58.79	77.65
DART	1536	+8%	58.94	77.75
ours	1536	+10%	59.28	78.12
APT	1920	+4%	59.89	78.85
DART	1920	-2%	60.14	79.15
ours	1920	0%	**60.43**	**79.59**
EVA-02-L	1536	–	62.28	80.80
APT	1536	+30%	62.07	80.64
DART	1536	+20%	62.64	80.81
ours	1536	+25%	64.83	81.25
APT	1920	-5%	64.57	81.94
DART	1920	-15%	65.44	82.21
ours	1920	-12%	**66.14**	**82.65**

(a) Object detection results on COCO

Model	Res	Speedup	aAcc	mIoU
EVA-02-L	512	–	86.67	59.77
Resizing	512	+11%	86.09	58.81
APT	512	+11%	86.68	59.70
DART	512	+6%	87.03	59.86
ours	512	+8%	87.17	60.13
EVA-02-L	640	–	86.83	60.05
Resizing	640	+11%	86.06	58.83
APT	640	+11%	86.82	60.01
DART	640	+4%	86.80	59.92
ours	640	+6%	87.24	60.87
APT	800	+3%	87.12	60.51
DART	800	-4%	87.35	60.92
ours	800	-2%	**87.54**	**61.15**

(b) Semantic segmentation results on ADE20K

Table 3: Dense prediction results on COCO object detection (left) and ADE20K semantic segmentation (right). Within each block, all speedups are reported relative to the single dense EVA-02 reference run (greyed).

4.2 Dense Prediction

We report performance and computational cost on COCO detection and ADE20K segmentation in Tab. 3. We show results against APT (Choudhury et al., 2026), DART (Yin et al., 2025), and a resizing baseline. On equivalent resolutions, APE achieves better performance with less compute

compared to dense baselines, and consistently outperforms adaptive baselines with similar compute consumption. On higher resolutions, APE scales to even stronger performance over baselines with similar compute scaling cost.

4.3 APE AS A VISION-LANGUAGE ENCODER

Does APE's allocation pattern still help when its tokens feed a language model rather than a classification head? We swap the vision encoder of LLaVA-1.5-7B and LLaVA-1.5-13B (Liu et al., 2024a) for an APE encoder, and replicate the visual instruction tuning evaluation established by APT (Choudhury et al., 2026). Full details are in App. A.2. Tab. 4 reports results on nine standard VQA benchmarks (Goyal et al., 2017; Hudson & Manning, 2019; Lu et al., 2022; Singh et al., 2019; Li et al., 2023b; Fu et al., 2023; Liu et al., 2024b; Yu et al., 2023). Across all backbone $\times$ evaluation datasets pairs, APE consistently matches or exceeds the performance of the dense baseline on all but one time (VQA^T benchmark with 7B backbone), while achieving $> 17\%$ speedup. APE also outperforms the APT baseline on all evaluations, and unlike APT, our model does not fall below the performance of the dense baseline by more than 0.1 in any case. The learned nature of our token allocation mechanism allows APE to learn a powerful estimation of patch saliency from pretraining without relying on static priors that degrade in certain tasks.

Model	Speedup	VQA^{v_2}	GQA	SQA^I	VQA^T	POPE	MME	MMB	MMB^C	MMV
LLaVA-1.5-7B	–	78.5	61.0	67.8	58.2	86.9	1510.1	64.6	58.1	30.7
LLaVA-1.5-7B-FT	–	78.4	61.1	67.6	58.1	87.0	1502.5	64.6	58.1	30.8
Random	+24%	76.9	60.9	67.2	54.1	86.1	1460.5	62.7	57.6	30.5
Resizing	+22%	77.5	61.1	66.8	56.5	86.6	1473.8	63.2	58.1	30.2
APT	+22%	77.8	61.1	67.5	56.6	86.3	1465.3	63.4	57.8	30.5
ours	+17%	78.6	61.6	67.9	58.1	87.1	1520.4	65.0	58.4	31.2
LLaVA-1.5-13B	–	80.0	63.2	72.7	61.2	87.1	1530.6	68.5	63.4	35.4
LLaVA-1.5-13B-FT	–	80.0	63.2	72.7	61.3	87.0	1524.7	68.5	63.5	35.6
Random	+26%	78.0	60.7	72.0	55.7	86.5	1484.0	64.7	60.8	32.0
Resizing	+23%	78.9	61.1	72.0	59.1	86.8	1496.9	65.8	62.5	33.9
APT	+23%	79.2	62.9	72.5	59.3	87.2	1488.2	66.0	63.3	33.6
ours	+18%	80.4	63.5	73.8	61.4	87.7	1545.8	69.1	64.0	35.4

Table 4: Vision-language results on nine VQA benchmarks with LLaVA-1.5-7B and LLaVA-1.5-13B Liu et al. (2024a), where APE replaces the standard vision encoder. Speedups are reported relative to the original LLaVA-1.5 backbone of the corresponding scale (greyed reference).

4.4 BUDGET-CONSTRAINED INFERENCE

Fig. 3 reports accuracy under varying token budgets on classification, detection, and segmentation against MAE (He et al., 2022), APT (Choudhury et al., 2026), DART (Yin et al., 2025), and a resizing baseline. Because t_{select} and t_{merge} are continuous and we design a positional and nesting-level embedding to be smoothly interpolated, we can cleanly and continuously vary the desired inference compute scale. By scaling both thresholds to more conservative values, APE selects fewer tokens and merges more to reduce the length of the input token sequence. Across all three tasks APE degrades least under tight budgets, while APT and DART show larger accuracy drops as the budget shrinks. Because patch importances are learned, APE can achieve strong performance by maintaining the small set of patches that contribute most to the output. APE also does not tie token allocation to whole rows or columns like DART, which prevent smooth scaling of the total sequence length.

4.5 VISUALIZATIONS

Fig. 4 shows the learned entropy field: high-entropy patches concentrate on object boundaries and discriminative textures, while homogeneous regions are absorbed into a single summary token.

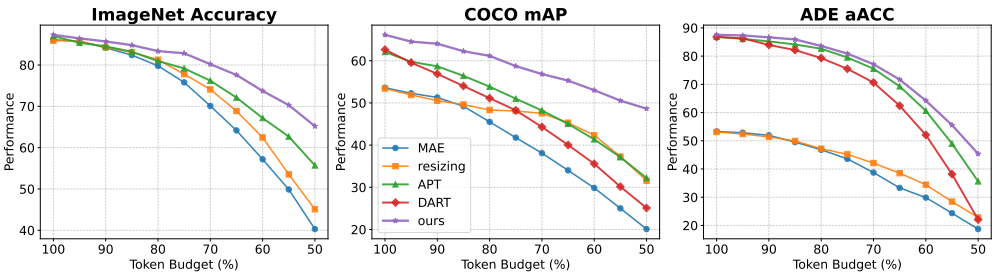


Figure 3: Performance on classification, detection, and segmentation under varying token budgets.

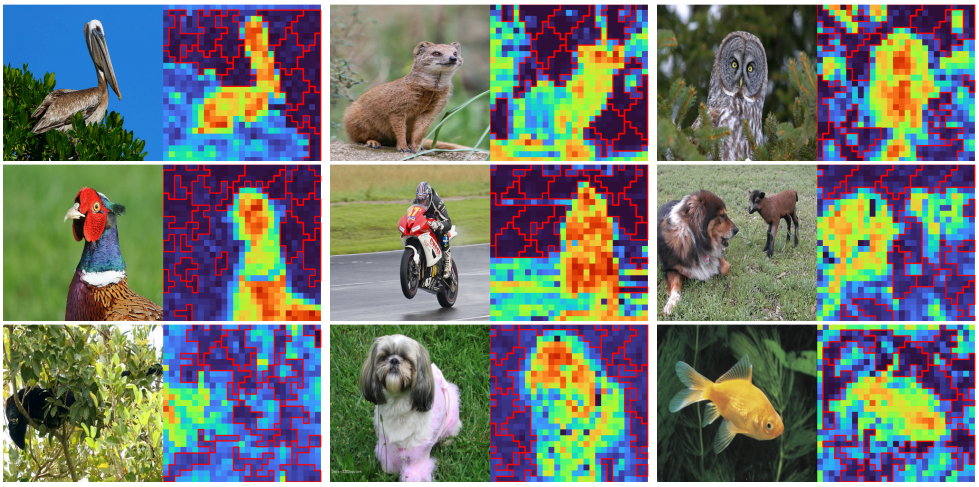


Figure 4: Visualizations of the learned entropy of patches and merged regions (red borders).

4.6 Ablations

Tab. 5 ablates the key design choices of APE on ImageNet at the ViT-B/$384/12$ setting. Sweeping nesting width B shows optimality at $B{=}4$: $B{\in}\{2,3\}$ results in decreased performance while $B{=}5$ adds negligible performance. Removing the nesting loss decouples Shannon entropy from information density, degrading performance without conserving compute. Removing the merging step and removing our 3D RoPE formulation both cost performance with minimal computational savings.

Table 5: Ablations on ImageNet classification. Speedups are the relative FLOPs reduction vs. dense ViT-B$^{\text{MAE}}$ baseline at $384/16$ (49.4 GFLOPs).

Variant	Acc ↑	Speedup ↑	GFLOPs ↓
APE ($B{=}4$)	85.2	+14.7%	42.13
$B{=}2$	83.9	**+37.0%**	**31.12**
$B{=}3$	84.7	+25.0%	37.05
$B{=}5$	**85.3**	+2.0%	48.41
− Merging	84.8	+15.1%	41.92
− Nesting Loss	84.1	+14.6%	42.21
− 3D RoPE	84.6	+14.8%	42.05

5 Conclusion

We introduced APE, a Vision Transformer that allocates tokens to where information lives. Via nested representation learning, token selection, and token merging, APE decides *how many* tokens each image region deserves directly from a task-dependent learned representation rather than from pixel statistics. The result is a single recipe that consistently improves the accuracy/compute trade-off across image classification, dense prediction, and vision-language modeling. A single APE checkpoint also accurately scales to continuous inference compute budgets without retraining. These results demonstrate that latent-space, information-theoretic patch importance is a general inductive bias for Vision Transformers to boost both performance and computational efficiency.

AI USE STATEMENT

In this work, we used generative AI tools for implementing methods and designing or providing feedback on research methodology or experiments (providing feedback). We have not used generative AI tools for propose or refine hypotheses, assist with translation, clean and reformat dataset, support qualitative and thematic data analysis, interpret results. and: generate synthetic data sets, help develop theoretical models or conceptual frameworks, formulate mathematical claims, provide critical ingredients for proving mathematical claims, assist in the writing of proofs are not applicable to this work. Additionally, we used generative AI tools for create or edit software code, draft parts of a research paper, brainstorming, sourcing/searching for information. We have reviewed all AI-assisted work. AI generated code and prose was verified by authors for correctness. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

Source code and trained models will be released upon acceptance.

REFERENCES

Guillaume Astruc, Nicolas Gonthier, Clement Mallet, and Loic Landrieu. Anysat: One earth observation model for many resolutions, scales, and modalities. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 19530–19540, 2025.

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

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.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *2021 IEEE/CVF international conference on computer vision (ICCV)*, pp. 9630–9640. IEEE, 2021.

Rohan Choudhury, JungEun Kim, Jinhyung Park, Eunho Yang, László A. Jeni, and Kris Kitani. Faster vision transformers with adaptive patches. In *International Conference on Learning Representations (ICLR)*, 2026. To appear.

Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pp. 702–703, 2020.

Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. Vision transformers need registers. *arXiv preprint arXiv:2309.16588*, 2023.

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

Yuxin Fang, Quan Sun, Xinggang Wang, Tiejun Huang, Xinlong Wang, and Yue Cao. Eva-02: A visual representation for neon genesis. *Image and Vision Computing*, 149:105171, 2024.

Mohsen Fayyaz, Soroush Abbasi Koohpayegani, Farnoush Rezaei Jafari, Sunando Sengupta, Hamid Reza Vaezi Joze, Eric Sommerlade, Hamed Pirsiavash, and Jürgen Gall. Adaptive token sampling for efficient vision transformers. In *European conference on computer vision*, pp. 396–414. Springer, 2022.

Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, et al. Mme: A comprehensive evaluation benchmark for multimodal large language models. *arXiv preprint arXiv:2306.13394*, 2023.

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6904–6913, 2017.

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

Wenbo Hu, Zi-Yi Dou, Liunian Li, Amita Kamath, Nanyun Peng, and Kai-Wei Chang. Matryoshka query transformer for large vision-language models. *Advances in Neural Information Processing Systems*, 37:50168–50188, 2024.

Drew A Hudson and Christopher D Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 6700–6709, 2019.

Aditya Kusupati, Gantavya Bhatt, Aniket Rege, Matthew Wallingford, Aditya Sinha, Vivek Ramanujan, William Howard-Snyder, Kaifeng Chen, Sham Kakade, Prateek Jain, et al. Matryoshka representation learning. *Advances in Neural Information Processing Systems*, 35:30233–30249, 2022.

Alexandre Lacoste, Nils Lehmann, Pau Rodriguez, Evan Sherwin, Hannah Kerner, Björn Lütjens, Jeremy Irvin, David Dao, Hamed Alemohammad, Alexandre Drouin, et al. Geo-bench: Toward foundation models for earth monitoring. *Advances in Neural Information Processing Systems*, 36: 51080–51093, 2023.

Yanghao Li, Haoqi Fan, Ronghang Hu, Christoph Feichtenhofer, and Kaiming He. Scaling language-image pre-training via masking. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 23390–23400, 2023a.

Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Xin Zhao, and Ji-Rong Wen. Evaluating object hallucination in large vision-language models. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pp. 292–305, 2023b.

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

Chengxin Liu, Hao Lu, Zhiguo Cao, and Tongliang Liu. Point-query quadtree for crowd counting, localization, and more. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 1676–1685, 2023.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 26296–26306, 2024a.

Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, et al. Mmbench: Is your multi-modal model an all-around player? In *European conference on computer vision*, pp. 216–233. Springer, 2024b.

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

Pan Lu, Swaroop Mishra, Tanglin Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering. *Advances in neural information processing systems*, 35:2507–2521, 2022.

Nick Nikzad, Yi Liao, Yongsheng Gao, and Jun Zhou. Sata: spatial autocorrelation token analysis for enhancing the robustness of vision transformers. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 9730–9739, 2025.

Jan Olszewski, Dawid Damian Rymarczyk, Piotr Wójcik, Mateusz Pach, and Bartosz Zielinski. Tore: Token recycling in vision transformers for efficient active visual exploration. In *Proceedings of the Winter Conference on Applications of Computer Vision*, pp. 8595–8605, 2025.

Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alex Sorkine-Hornung, and Luc Van Gool. The 2017 davis challenge on video object segmentation. *arXiv preprint arXiv:1704.00675*, 2017.

Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. Dynamicvit: Efficient vision transformers with dynamic token sparsification. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Cedric Renggli, André Susano Pinto, Neil Houlsby, Basil Mustafa, Joan Puigcerver, and Carlos Riquelme. Learning to merge tokens in vision transformers, 2022. URL https://arxiv.org/abs/2202.12015.

Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. Towards vqa models that can read. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 8317–8326, 2019.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

Grant Van Horn and Oisin Mac Aodha. iNat Challenge 2021 - FGVC8. Kaggle, 2021. URL https://www.kaggle.com/competitions/inaturalist-2021. Accessed: Jan 2026.

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.

Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011.

Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

Ross Wightman. Pytorch image models. https://github.com/rwightman/pytorch-image-models, 2019.

Luyuan Xie, Tianyu Luan, Wenyuan Cai, Guochen Yan, Zhaoyu Chen, Nan Xi, Yuejian Fang, Qingni Shen, Zhonghai Wu, and Junsong Yuan. dflmoe: Decentralized federated learning via mixture of experts for medical data analysis. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 10203–10213, 2025.

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*, pp. 10809–10818, 2022.

Shicheng Yin, Kaixuan Yin, Yang Liu, Weixing Chen, and Liang Lin. Dart: Differentiable dynamic adaptive region tokenizer for vision transformer and mamba. *arXiv e-prints*, pp. arXiv–2506, 2025.

Seungdong Yoa, Seungjun Lee, Hye-Seung Cho, Bumsoo Kim, and Woohyung Lim. Imagepiece: Content-aware re-tokenization for efficient image recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 9544–9552, 2025.

Weihao Yu, Zhengyuan Yang, Linjie Li, Jianfeng Wang, Kevin Lin, Zicheng Liu, Xinchao Wang, and Lijuan Wang. Mm-vet: Evaluating large multimodal models for integrated capabilities. *arXiv preprint arXiv:2308.02490*, 2023.

Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip HS Torr, and Vladlen Koltun. Point transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 16259–16268, 2021.

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*, 2017.

Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ade20k dataset. *International Journal of Computer Vision*, 127(3):302–321, 2019.

A Appendix

A.1 Limitations

Both the selection and merging steps rely on straight-through estimator gradients, which may have unpredictable effects on the optimization process; we attempt to stabilize training via the described temperature-warmup schedule section 3.5. Dynamic sequence lengths produced by APE may be difficult to optimize for batched GPU execution on all hardware; we do not explore these low-level optimizations in this work. We also do not explore metrics additional to Shannon entropy for proxying patch informativeness; we leave this to future work.

A.2 Additional Implementation Details

Selection STE. The hard indicator $m_i = \mathbb{1}[H(\mathbf{e}_i^{(b)}) \geq t_{\text{select}}]$ is used in the forward pass, leaving each selected token's value unchanged. In the backward pass, autograd routes $\partial\mathcal{L}/\partial t_{\text{select}}$ through the smooth surrogate s_i, giving a non-zero gradient proportional to the contribution of marginally-selected tokens. Because the forward value of g_i equals m_i exactly, the token activations themselves are unaffected; only the threshold receives a gradient signal.

Merging STE. An analogous construction is used for the merging step. For each patch j absorbed by the BFS, the per-absorption gate is

$$g_j = 1 + (s_j - \text{sg}[s_j]), \qquad s_j = \sigma\big((\text{sim}(\mathbf{e}_j^{(0)}, \mathbf{e}_{\pi(j)}^{(0)}) - \sigma(t_{\text{merge}}))/\tau_{\text{merge}}\big), \tag{11}$$

where $\pi(j)$ is the parent patch that absorbed j during BFS and $\text{sg}[\cdot]$ denotes the stop-gradient operator. Each g_j has forward value 1, so the summary token $\mathbf{z}_k$ (Eq. 7) reduces to a plain mean in the forward pass. In the backward pass, gradients flow through s_j to t_{merge}, with the largest signal arising from absorptions whose similarity lies just above the threshold—precisely the marginal merges whose removal would alter $\mathbf{z}_k$.

STE surrogate temperatures. The STE scales τ_{sel} and τ_{merge} are learnable scalars scheduled per forward-pass counter t via $\log\tau = \log\tau^0 + \text{progress}(t) \cdot \Delta\log\tau$, $\text{progress}(t) = \text{clip}\big((t - T_0)/T_w,\, 0,\, 1\big)$. Because $\partial\log\tau/\partial(\Delta\log\tau) = \text{progress}$, gradients on $\Delta\log\tau$ are zero for the first T_0 forwards, ramp linearly over the next T_w forwards, and reach full magnitude afterward (equivalent to a linear learning-rate warmup on the temperature parameter). We use τ^0_{sel}=0.5, τ^0_{merge}=0.1, T_0=T_w=2,000, and clamp τ to $[\tau_{\min}, \tau_{\max}]$. The thresholds t_{select} and t_{merge} are separate learnable parameters optimized jointly with the backbone.

Replicating APT's VQA evaluation. We start with the fully fine-tuned model and all default configurations of the second-stage fine-tuning publicly released by the LLaVA-1.5 team. The deviation from the official fine-tuning procedure of LLaVA-1.5 is that we additionally unfreeze the new components of the APT and APE methods. For APT, this includes the newly introduced ZeroConv and MLP modules. For APE, this includes the newly introduced patch encoder. For all methods, the language model backbone and MLP projector are fine-tuned, as is consistent with LLaVA-1.5's second-stage fine-tuning. As in the APT paper, we only fine-tuned for 5% of the total iterations used in LLaVA-1.5 fine-tuning.

Hardware. Main experiments were performed on a single NVIDIA H100 GPU with 80GB VRAM. The supporting CPU environment consisted of an Intel Sapphire Rapids 8468 processor utilizing 24 threads.

A.3 Domain-Specific Classification on GeoBench

GeoBench (Lacoste et al., 2023) comprises six classification tasks over satellite and overhead imagery, a domain whose pixel statistics differ markedly from natural images. We use it to probe whether APE's latent-saliency metric continues to allocate compute effectively outside the natural-image distribution and at small backbone sizes. Following each baseline's pretraining protocol, we fine-tune ViT-T and ViT-S backbones on each GeoBench task.

	bigearthnet	brick-kiln	eurosat	forestnet	pv4ger	so2sat	Δtime
vit-t-timm	0.632	0.982	0.958	0.521	0.988	0.565	0%
vit-t-APT	0.624	0.980	0.950	0.501	0.982	0.563	−2%
vit-t-APE	**0.658**	**0.985**	**0.971**	**0.574**	**0.988**	**0.568**	−8%
vit-s-timm	0.662	0.984	0.971	**0.524**	**0.983**	0.567	0%
vit-s-APT	0.658	0.975	0.971	0.517	0.980	0.562	−16%
vit-s-APE	**0.670**	**0.988**	**0.972**	0.522	0.982	**0.583**	−24%

Table 6: Classification results on the six GeoBench (Lacoste et al., 2023) classification tasks, stratified by dataset.

Tab. 6 shows that APE with ViT-T outperforms the timm baseline on five of six tasks at an 8% time reduction, with the largest gain on *forestnet* (0.574 vs. 0.521, $+5.3$ pp). With ViT-S, APE wins four of six tasks at a 24% time reduction, while the pixel-space APT baseline trails the timm initialization on *every* task. Together with the fine-grained results in the main paper, this reinforces that grounding patch importance in learned features rather than raw pixel statistics is essential whenever the input distribution departs from the data on which a fixed pixel-space heuristic was calibrated.

A.4 Results with equivalent patch encoders

Table 7 gives additional results using the APE patch encoder CNN architecture for all models. We report on top-1 ImageNet classification using the ViT-B backbone and training GFLOPs. APE remains the strongest model and achieves significant computational reductions compared to the dense baseline.

Model	Res/Patch	Acc ↑	GFLOPs ↓
ViT-B$^{\text{MAE}}$	384/16	84.3	51.4
APT	384/16	84.3	22.5
DART	384/16	84.0	34.1
ours	384/16	84.7	30.5

Table 7: Results on ImageNet classification with equivalent patch encoder architectures across models. APE remains the strongest model.

A.5 Other Proxies for Patch Informativeness

We present additional metrics pre-training APE using norm-based selection instead of Shannon entropy in Table 8. In general, the exact metric is for token selection is easily modified in the APE pipeline. We choose Shannon entropy as the embedding scoring function because it has theoretical justification and empirically demonstrates strong performance.

	Res/Patch	Acc
ours-B	384/16	**84.7**
ours-B + norm scoring	384/16	84.5
ours-L	336/14	**86.5**
ours-L + norm scoring	336/14	85.2

Table 8: Shannon entropy vs. norm-based scoring.

A.6 Comparisons with In-Backbone Methods

In the main evaluations, our choice of MAE, APT, and DART is deliberate, because these baselines all tackle the task of maximizing the value of the input token sequence as the APE framework does.

APE bridges a core methodological gap relative to baselines by estimating token value from *learned latent representation* rather than from pixels.

Other methods, such as ImagePiece (Yoa et al., 2025) and DynamicViT (Rao et al., 2021), are related in motivation but operate beyond the patch encoding and within the transformer backbone. They address a different question using signal that does not exist at the patch encoder. We therefore see them as complementary rather than competing. In Table 9, we have also added a direct empirical comparison to ImagePiece and DynamicViT. Despite only modifying the patch encoder, APE achieves higher accuracy than these in-backbone methods while still delivering compute savings.

	Res/Patch	Acc	speedup
ImagePiece-B	384/16	84.5	+32%
DynamicViT-B-0.7	384/16	83.8	**+34%**
ours-B	384/16	**84.7**	+22%
ImagePiece-L	336/14	86.4	+29%
DynamicViT-L-0.7	336/14	85.5	**+36%**
ours-L	336/14	**86.5**	+20%

Table 9: Comparison of accuracy and speedup with ImagePiece and DynamicViT.

To demonstrate APE's composability with in-backbone methods, we show results with ToME (Bolya et al., 2023) in Table 10. Adding ToMe to the APE backbone increases throughput at small costs to accuracy. These results demonstrate that APE and ToMe are both composable and complementary.

	Res/Patch	Acc	speedup
MAE-B + ToMe	384/16	83.4	+40%
ours-B	384/16	**84.7**	+22%
ours-B + ToMe	384/16	84.1	**+56%**
MAE-L + ToMe	336/14	85.4	+38%
ours-L	336/14	**86.5**	+20%
ours-L + ToMe	336/14	85.8	**+47%**

Table 10: Combining APE with ToMe.

A.7 Varying Training Objective

We demonstrate APE is transferrable to other SSL paradigms. In Table 11, we replicate the DINO (Caron et al., 2021) SSL protocol on ImageNet pretraining data and modify the ViT's patch encoder under the APE paradigm. We report two different evaluation protocols from the original DINO paper: ImageNet classification by linear probe and KNN, and object segmentation from DAVIS2017 (Pont-Tuset et al., 2017). Models are pretrained on ImageNet without labels. We report accuracy on ImageNet and report averaged mean region similarity and mean contour accuracy $(\mathcal{J}\&\mathcal{F})_m$ on DAVIS2017. We see that the APE architecture generalizes both accuracy and speedup to the DINO SSL task.

	backbone	IN1K-Linear	IN1K-KNN	DAVIS2017	speedup
DINO	S/8	79.7	78.3	69.9	0%
DINO+APE	S/8	**80.2**	**79.2**	**70.5**	**+12%**
DINO	B/8	80.1	77.4	71.4	0%
DINO+APE	B/8	**80.4**	**78.5**	**72.3**	**+14%**

Table 11: Applying APE to DINO pretraining. Results on ImageNet-1K linear probing, k-NN classification, and DAVIS-2017 video object segmentation.

A.8 Varying Nesting Width on Ecological Classification

We find empirically that $B = 4$ is generally optimal across tested domains and tasks. Due to computational limitations, we cannot exhaustively ablate the optimal value of across domains, but we provide a small additional ablation on ecological classification in Table 2.

	iNat21-species	CUB-200-2011	avg. red. (time)	avg. red. (GFLOPs)
ours (B=3)	84.8	89.6	**-29%**	**-58%**
ours (B=4)	85.2	**89.9**	-24%	-49%
ours (B=5)	**85.3**	89.7	-17%	-35%

Table 12: Effect of B on classification performance and efficiency (ViT-B).

A.9 Additional Visualizations

Fig. 5 provides additional qualitative examples of the entropy field learned by APE. Selected tokens (high entropy) concentrate on object boundaries, fine-grained textures, and semantically discriminative regions, while unselected patches in homogeneous areas are collapsed into merged summary tokens (red borders). The spatial coherence of merged regions confirms that the BFS clustering respects semantic boundaries rather than grouping arbitrary background patches.

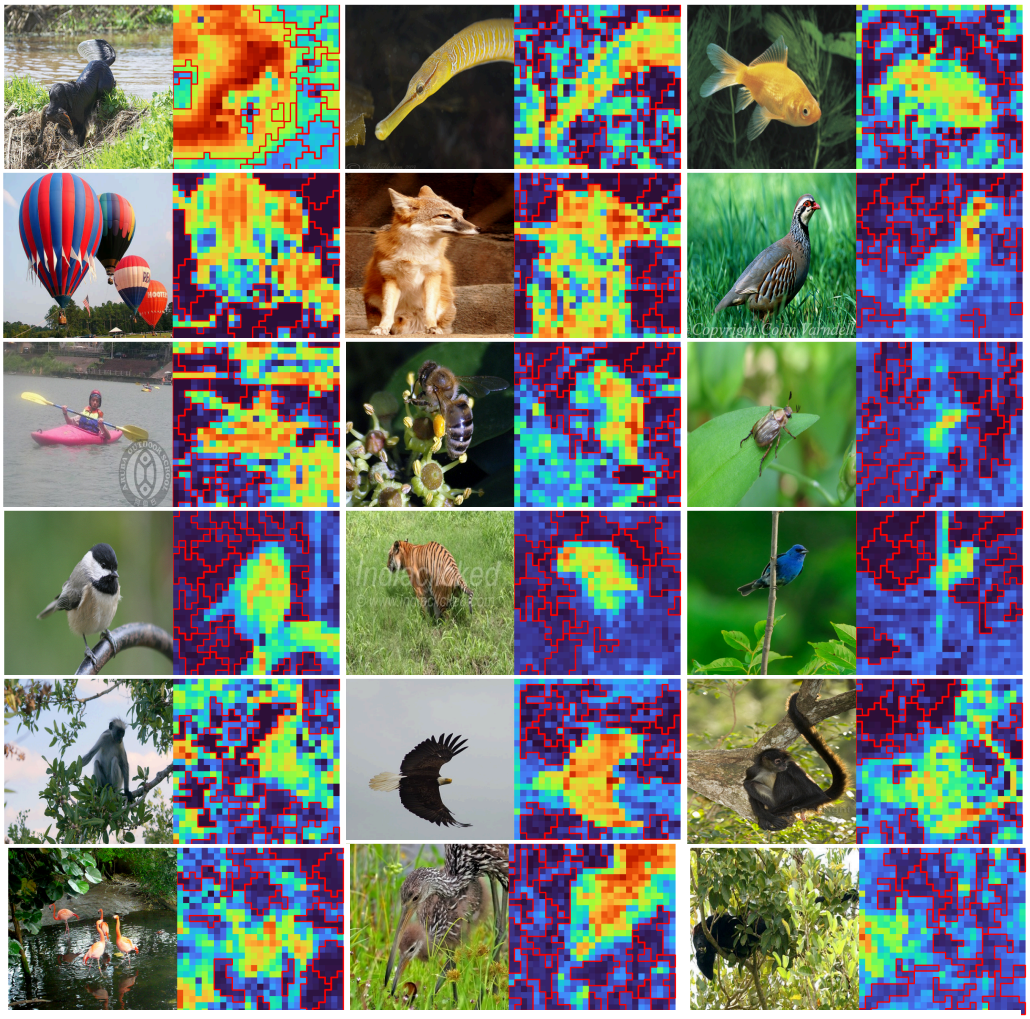


Figure 5: Additional qualitative visualizations of the APE entropy field. Brighter patches indicate higher entropy and are passed as individual tokens to the backbone; red-bordered regions are merged into a single summary token.