A Spitting Image: Modular Superpixel Tokenization in Vision Transformers

Marius Aasan[1,2], Odd Kolbjørnsen[1,2,3], Anne Schistad Solberg[1,2], and Adín Ramirez Rivera[1,2]

[1] University of Oslo, Box 1072 Blindern, 0316 Oslo, Norway
{mariuaas,anne,oddkol,adinr}@uio.no
[2] SFI Visual Intelligence, Box 6050 Langnes, 9037 Tromsø, Norway
[3] Aker BP ASA, Box 65, 1324 Lysaker, Norway odd.kolbjornsen@akerbp.com

Abstract. Vision Transformer (ViT) architectures traditionally employ a grid-based approach to tokenization independent of the semantic content of an image. We propose a modular superpixel tokenization strategy which decouples tokenization and feature extraction; a shift from contemporary approaches where these are treated as an undifferentiated whole. Using on-line content-aware tokenization and scale- and shape-invariant positional embeddings, we perform experiments and ablations that contrast our approach with patch-based tokenization and randomized partitions as baselines. We show that our method significantly improves the faithfulness of attributions, gives pixel-level granularity on zero-shot unsupervised dense prediction tasks, while maintaining predictive performance in classification tasks. Our approach provides a modular tokenization framework commensurable with standard architectures, extending the space of ViTs to a larger class of semantically-rich models.

Keywords: ViT · Tokenization · Superpixels · XAI · Saliency

1 Introduction

Vision Transformers [14] (ViTs) have become the cynosure of vision tasks in the wake of convolutional architectures. In the original transformer for language [12, 42], *tokenization* serves as a crucial preprocessing step, with the aim of optimally partitioning data based on a predetermined entropic measure [20, 34]. As models were adapted to vision, tokenization was simplified to partitioning images into square patches. This approach proved effective [7, 25, 38, 39, 40, 41], and soon became canonical; an integral part of the architecture.

Despite apparent successes, we argue that patch-based tokenization has inherent limitations. Firstly, the scale of the tokens are rigidly linked to the model architecture by a fixed patch size, ignoring any redundancy in the original images. These limitations result in a significant increase in computation for larger

Code available at: https://github.com/dsb-ifi/SPiT
To appear in ECCV (MELEX) 2024 Workshop Proceedings.

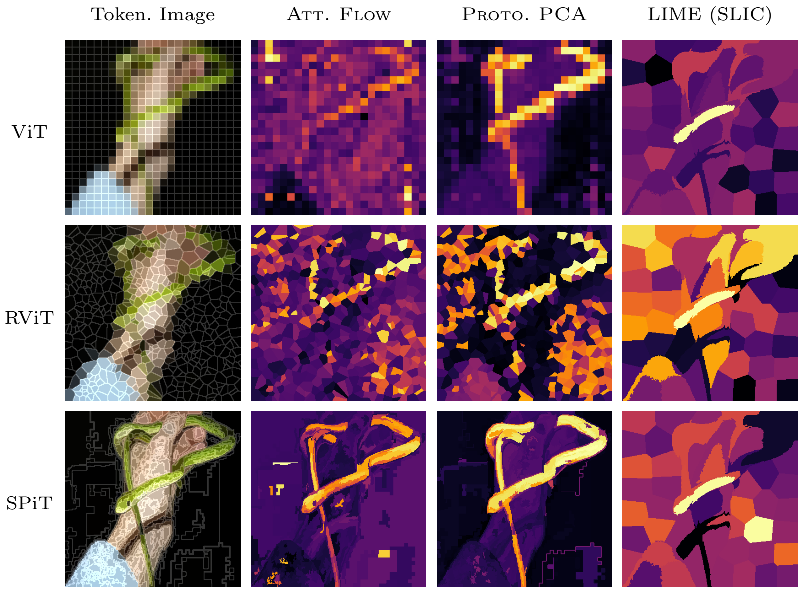

Fig. 1: Tokenized image and attributions for prediction “grass snake” with different tokenizers: square patches (ViT), Voronoi tesselation (RViT) and superpixels (SPiT). We show more results in Appendix D.

resolutions, as complexity and memory scales quadratically with the number of tokens. Moreover, regular partitioning assumes an inherent uniformity of the distribution of semantic content while effectively reducing spatial resolution.

Several works have since leveraged attention maps to visualize class token attributions for interpretability [8, 28], which has been exploited in dense prediction tasks [17]. However, attention maps with square partitions incur a loss of resolution in the patch representation, and subsequently do not inherently capture the resolution of the original images. For dense predictions with pixel level granularity, a separate decoder for upscaling is required [21, 47, 50].

1.1 Motivation

We take a step back from the original ViT architecture to re-evaluate the role of patch-based tokenization. By focusing on a somewhat overlooked component in the architecture, we look to establish image partitioning as the role of an *adaptive modular tokenizer*; an untapped potential in ViTs.

In contrast to square partitions, *superpixels* offer an opportunity to mitigate the shortcomings of patch-based tokenization by allowing for adaptability in scale and shape while leveraging inherent redundancies in visual data. Superpixels have been shown to align better with semantic structures within images [37], providing a rationale for their potential utility in vision transformer architectures. We compare the canonical square tokenization in standard ViTs with our proposed superpixel tokenized model (SPiT) as well as a control using random Voronoi tokenization (RViT), selected for being well defined mathematical objects for tessellating a plane. The three tokenization schemes are illustrated in Fig. 1, and their innate segmentation capabilities in Fig. 4.

1.2 Contributions

Our research induces three specific inquiries: **(a)** *Is a rigid adherence to square patches necessary?*, **(b)** *What effect does irregular partitioning have on tokenized representations?*, and **(c)** *Can tokenization schemes be designed as a modular component in vision models?* In this work we establish the following;

- **Generalized Framework:** Superpixel tokenization generalize ViTs in a modular scheme, providing a richer space of transformers for vision tasks where *the transformer backbone is independent of tokenization framework.*
- **Efficient Tokenization:** We propose an efficient on-line tokenization approach which provides *competitive training and inference times* as well as *strong performance in classification tasks.*
- **Refined Spatial Resolution:** Superpixel tokenization provides semantically aligned tokens with pixel-level granularity. We demonstrate that our method yields *significantly more faithful attributions compared to established explainability methods*, as well as *strong results in unsupervised segmentation.*
- **Visual Tokenization:** The main contribution of our work is *the introduction of a novel way of thinking about tokenization in ViTs*, an overlooked but central component of the modeling process—*cf*. discussion in Section 4.

Our primary objective is to evaluate tokenization schemes for ViTs, underscoring the intrinsic properties of alternative tokenization. In the interest of a fair comparative analysis, *we perform our study using vanilla ViT architectures and established training protocols* [36]. Hence, we design experiments to establish a fair comparison against well-known baselines *without architectural optimizations.* This controlled comparison is crucial for attributing observed disparities specifically to the tokenization strategy, and eliminates confounding factors from specialized architectures or training regimes.

Notation: We let $H \times W = \big\{(y,x) : 1 \leq y \leq h, 1 \leq x \leq w\big\}$ denote the coordinates of an image of spatial dimension (h,w), and let $\mathcal{I}$ be an index set for the mapping $i \mapsto (y,x)$. We consider a C-channel image as a signal $\xi\colon \mathcal{I} \to \mathbb{R}^C$. We use the vectorization operator $\operatorname{vec}\colon \mathbb{R}^{d_1 \times \dots \times d_n} \to \mathbb{R}^{d_1 \dots d_n}$, and denote function composition by $f(g(x)) = (f \circ g)(x)$.

2 Methodology

To evaluate and contrast different tokenization strategies, we require methods for partitioning images and extracting meaningful features from these partitions. While these tasks can be performed using a variety of deep architectures, such approaches add a layer of complexity to the final model, which would invalidate

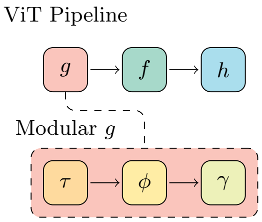

Fig. 2: Illustration of modular tokenization in ViT architecture.

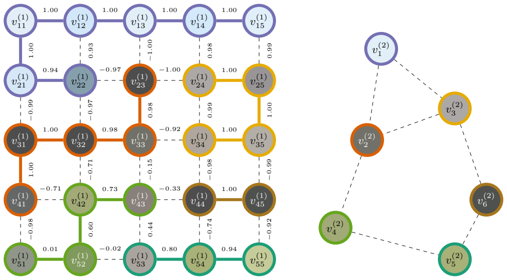

Fig. 3: Visualization of superpixel aggregation.

any direct comparison between tokenization strategies. Furthermore, this would also complicate any meaningful transfer learning between architectures. In line with this reasoning, we construct an effective heuristic superpixel tokenizer, and propose an uninvasive feature extraction method which aligns with the canonical ViT architecture, and facilitates direct comparison.

2.1 Framework

We generalize the canonical ViT architecture by allowing for a modular tokenizer and different methods of feature extraction. Note that a canonical ViT is generally presented as a three-component system with a tokenizer-embedder g, a backbone f consisting of a sequence of attention blocks, and a subsequent prediction head h. Contrarily, language transformers explicitly decouples g from the backbone f. Following this lead, we note that we can essentially rewrite a patch embedding module as a three component modular system, featuring a tokenizer τ, a feature extractor ϕ, and an embedder γ such that $g = \gamma \circ \phi \circ \tau$, emphasizing that these are inherent components in the original architecture obscured by a simplified tokenization strategy—*cf*. Fig 2. This provides a more complete assessment of the model as a five component feedforward system

$$\begin{aligned} \varPhi(\xi;\theta) &= (h \circ f \circ g)(\xi;\theta), && \text{(1a)} \\ &= (h \circ f \circ \gamma \circ \phi \circ \tau)(\xi;\theta), && \text{(1b)} \end{aligned}$$

where θ denotes the set of learnable parameters of the model. In a standard ViT model, the tokenizer τ acts by partitioning the image into fixed-size square partitions. This directly provides vectorized features since patches are of uniform dimensionality and ordering, hence $\phi = \text{vec}$ in standard ViT architectures. The embedding γ is typically a learnable linear layer, mapping features to the embedding dimension of the specific architecture. Alternatively, g can be taken as a convolution with kernel size and stride equal to the desired patch size ρ.

2.2 Partitioning and Tokenization

Tokenization in language tasks involves partitioning text into optimally informative tokens, analogous to how superpixels [37] partition spatial data into discrete connected regions. Hierarchical superpixels [48, 53] are highly parallelizable graph-based approaches suitable for on-line tokenization. We introduce a novel method that leverages fully parallel aggregation over batches of image graphs at each step t, in addition to regularization for size and compactness—*cf*. Appendix B. Our method yields a variable number of superpixels at each step, adapting dynamically to the complexity of an image.

Superpixel Graphs: Let $E^{(0)} \subset \mathcal{I} \times \mathcal{I}$ denote the four-way adjacency edges under $H \times W$. We consider a superpixel as a set $S \subset \mathcal{I}$, and we say that S is connected if for any two pixels $p, q \in S$, there exists a sequence of edges in $\big((i_j, i_{j+1}) \in E^{(0)}\big)_{j=1}^{k-1}$ such that $i_1 = p$ and $i_k = q$. A set of superpixels form a partition π of an image if for any two distinct superpixels $S, S' \in \pi$, their intersection $S \cap S' = \emptyset$, and the union of all superpixels is equal to the set of all pixel positions in the image, i.e., $\bigcup_{S \in \pi^{(t)}} S = \mathcal{I}$.

Let $\varPi(\mathcal{I}) \subset 2^{2^{\mathcal{I}}}$ denote the space of all partitions of an image, and consider a sequence of partitions $(\pi^{(t)})_{t=0}^{T}$. We say that a partition $\pi^{(t)}$ is a refinement of another partition $\pi^{(t+1)}$ if for all superpixels $S \in \pi^{(t)}$ there exists a superpixel $S' \in \pi^{(t+1)}$ such that $S \subseteq S'$, and we write $\pi^{(t)} \sqsubseteq \pi^{(t+1)}$. Our goal is to construct a T-level hierarchical partitioning of the pixel indices $\mathcal{H} = \big(\pi^{(t)} \in \varPi(\mathcal{I}) : \pi^{(t)} \sqsubseteq \pi^{(t+1)}\big)_{t=0}^{T}$ such that each superpixel is connected.

To construct $\mathcal{H}$, the idea is to successively join vertices by parallel edge contraction to update the partition $\pi^{(t)} \mapsto \pi^{(t+1)}$. We do this by considering each level of the hierarchy as a graph $G^{(t)}$ where each vertex $v \in V^{(t)}$ is the index of a superpixel in the partition $\pi^{(t)}$, and each edge $(u, v) \in E^{(t)}$ represent adjacent superpixels for levels $t = 0, \ldots, T$. The initial image can thus be represented as a grid graph $G^{(0)} = (V^{(0)}, E^{(0)})$ corresponding to the singleton partition $\pi^{(0)} = \big\{\{i\} : i \in \mathcal{I}\big\}$.

Weight function: To apply the edge contraction, we define an edge weight functional $w_\xi^{(t)} \colon E^{(t)} \to \mathbb{R}$. We retain self-loops in the graph to constrain regions by weighting loop edges by relative size. This acts as a regularizer by constraining the variance of region sizes. For non-loop edges, we use averaged features $\mu_\xi^{(t)}(v) = \sum_{i \in \pi_v^{(t)}} \xi(i)/|\pi_v^{(t)}|$ and apply a similarity function $\mathrm{sim} \colon E^{(t)} \to \mathbb{R}$. Loops are weighted using the empirical mean $\mu_{|\pi|}^{(t)}$ and standard deviation $\sigma_{|\pi|}^{(t)}$ of region sizes at level t. This gives us weights on the form

$$w_\xi(u, v) = \begin{cases} \mathrm{sim}\Big(\mu_\xi^{(t)}(u), \mu_\xi^{(t)}(v)\Big), & \text{for } u \neq v; \\ \Big(|\pi_u^{(t)}| - \mu_{|\pi|}^{(t)}\Big)/\sigma_{|\pi|}^{(t)}, & \text{otherwise.} \end{cases} \tag{2}$$

Compactness can optionally be regulated by computing the infinity norm density

$$\delta_\infty(u, v) = \frac{4(|\pi_u|^{(t)} + |\pi_v|^{(t)})}{\mathrm{per}_\infty(u, v)^2}, \tag{3}$$

where per_∞ is the perimeter of the bounding box that encapsulates superpixels u and v. This emphasizes how tightly two neighbouring superpixels u and v are packed in their bounding box, resulting in a regularized weight functional

$$w_\xi^{(t)}(u, v; \lambda) = \lambda\delta_\infty(u, v) + (1 - \lambda)w_\xi^{(t)}(u, v) \tag{4}$$

where $\lambda \in [0, 1]$ serves as a hyperparameter for compactness.

Update rule: We use a greedy parallel update rule for the edge contraction, such that each superpixel joins with a neighboring superpixel with the highest edge weights, including self-loops for all $G^{(t)}$ for $t \geq 1$. Let $\mathfrak{N}^{(t)}(v)$ denote the neighborhood of adjacent vertices of the superpixel with index v at level t. We construct an intermediate set of edges, given by

$$\hat{E}^{(t)} = \left(v, \operatorname*{arg\,max}_{u \in \mathfrak{N}^{(t)}(v)} w_\xi(u, v; \lambda) : v \in V^{(t)}\right). \tag{5}$$

Then the transitive closure $\hat{E}_+^{(t)}$, *i.e.* the connected components of $\hat{E}^{(t)}$, explicitly yields a mapping $V^{(t)} \mapsto V^{(t+1)}$ such that

$$\pi_v^{(t+1)} = \bigcup_{u \in \hat{\mathfrak{N}}_+^{(t)}(v)} \pi_u^{(t)}, \tag{6}$$

where $\hat{\mathfrak{N}}_+^{(t)}(v)$ denotes the connected component of vertex v in $\hat{E}_+^{(t)}$. This update rule for the partitions ensures that each partition at level $(t + 1)$ is a connected region, as it is formed by merging adjacent superpixels with the highest edge weights. We illustrate the aggregation step in Fig. 3.

Iterative refinement: We repeat the steps of computing aggregation maps, regularized edge weights, and edge contraction until the desired number of hierarchical levels T is reached. At each level, the partitions become more coarse, representing larger homogeneous regions in the image. The hierarchical structure provides a multiscale representation of the image, capturing both local and global structures. At level T we have obtained a sequence of partitions $(\pi^{(t)})_{t=0}^T$, where each partition at level t is a connected region with $\pi^{(t)} \sqsubseteq \pi^{(t+1)}$ for all t.

We conduct experiments to empirically verify the relationship between the number of tokens produced by varying the steps T and patch size ρ in canonical ViT tokenizers. Let $N_{\mathrm{SPiT}}, N_{\mathrm{ViT}}$ denote the number of tokens for the SPiT tokenizer and ViT tokenizer respectively. Remarkably, we are able to show with a high degree of confidence that the relationship is $\mathbb{E}(T \mid N_{\mathrm{SPiT}} = N_{\mathrm{ViT}}) = \log_2 \rho$, *regardless of image size*. Details can be found in Appendix B.

2.3 Feature Extraction with Irregular Patches

While we conjecture the choice of square patches in the ViT architecture to be motivated by simplicity, it is naturally also a result of the challenge posed by the alternative. Irregular patches are unaligned, exhibit different shapes and dimensionality, and are generally non-convex. These factors make the embedding of irregular patches to a common inner product space nontrivial. In addition to consistency and uniform dimensionality, we propose a minimal set of properties any such features would need to capture; *color, texture, shape, scale*, and *position.*

Positional Encoding: ViTs generally use a learnable positional embedding for each patch in the image grid. Noting that this corresponds to a histogram over positions over a downsampled image (*cf*. Prop. 1) we can extend learnable positional embeddings to handle more complex shapes, scales, and positions by using a kernelized approach. We propose applying a joint histogram over the coordinates of a superpixel S_n for each of the $n = 1, \dots, N$ partitions. First, we normalize the positions such that $(y', x') \in [-1, 1]^2$ for all $(y', x') \in S_n$. We decide on a fixed number of bins β, denoting the dimensionality of our features in each spatial direction using a Gaussian kernel K_σ such that

$$\hat{\xi}^{(\text{pos})}_{n,y,x} = \text{vec}\left(\sum_{(y_j, x_j) \in S_n} K_\sigma(y - y_j, x - x_j) \right), \tag{7}$$

typically with low bandwith $\sigma \in [0.01, 0.05]$. This, in effect, encodes the position of the patch within the image, as well as its shape and scale.

Color Features: To encode the light intensity information from the raw pixel data into our features, we interpolate the bounding boxes of each patch to a fixed resolution of $\beta \times \beta$ using a bilinear interpolation operator, while masking out the pixel information in other surrounding patches. These features essentially capture the raw pixel information of the original patches, but resampled and scaled to uniform dimensionality. We refer to the feature extractor ϕ as an *interpolating feature extractor*. Similar to positional and texture features, the RGB features are normalized to $[-1, 1]$ and vectorized such that $\hat{\xi}^{(\text{col})} \in \mathbb{R}^{3\beta^2}$.

Texture Features: Gradient operators provides a simple robust method of extracting texture information [10, 24]. We use the gradient operator proposed by Scharr [33] due to improved rotational symmetry and discretization errors. We normalize the operator such that $\nabla \xi \in [-1, 1]^{H \times W \times 2}$, where the last dimensions correspond to gradient directions $\nabla y, \nabla x$. Mirroring the procedure for the positional features, we then construct a joint histogram with a Gaussian kernel over the gradients within each superpixel S_n such that $\hat{\xi}^{(\text{grad})}_n \in \mathbb{R}^{\beta^2}$.

The feature modalities are concatenated as $\hat{\xi}_n = [\hat{\xi}^{(\text{col})}_n, \hat{\xi}^{(\text{pos})}_n, \hat{\xi}^{(\text{grad})}_n] \in \mathbb{R}^{5\beta^2}$. While our proposed gradient features are commensurable with the canonical ViT architecture, they represent an additional dimension of information. We therefore ablate the effect of including or omitting gradient features. For models where these features are omitted, *i.e.* $\hat{\xi}_n \setminus \hat{\xi}^{(\text{grad})}_n = [\hat{\xi}^{(\text{col})}_n, \hat{\xi}^{(\text{pos})}_n] \in \mathbb{R}^{4\beta^2}$, we say that the extractor ϕ is *gradient excluding.*

Table 1: Accuracy (Top 1) for Base (B) capacity models on classification.

Model			INReaL		IN1k		Caltech256		Cifar100	
Name	Grad.	Im./s.[‡]	Lin.	kNN	Lin.	kNN	Lin.	kNN	Lin.	kNN
ViT-B16	✗	793.04	0.853	0.849	0.802	0.737	0.879	0.879	0.892	0.897
ViT-B16	✓	721.12	0.854	0.844	**0.805**	0.748	**0.889**	0.885	**0.899**	**0.899**
RViT-B16[†]	✗	619.86	0.843	0.832	0.788	0.718	0.873	0.882	0.894	0.838
RViT-B16[†]	✓	585.64	0.841	0.836	0.789	0.725	0.864	0.861	0.888	0.762
SPiT-B16	✗	690.72	0.793	0.818	0.760	0.569	0.833	0.829	0.813	0.634
SPiT-B16	✓	640.59	**0.858**	**0.853**	0.804	**0.752**	0.888	**0.891**	0.884	0.845

[†] Uncertainty measures for RViT are detailed in Appendix Table G.1.

[‡] Median throughput over full training with 4× MI250X GPUs using `float32` precision.

2.4 Generalization of Canonical ViT

By design, our framework acts as a generalization of the canonical ViT tokenization, and is equivalent to applying an canonical patch embedder using a fixed patch size ρ with interpolated gradient excluding feature extraction.

Proposition 1 (Embedding Equivalence). *Let τ^* denote an canonical ViT tokenizer with a fixed patch size ρ, let ϕ denote a gradient excluding interpolated feature extractor, and let γ^*, γ denote embedding layers with equivalent linear projections $L^*_\theta = L_\theta$. Let $\hat{\xi}^{(\text{pos})} \in \mathbb{R}^{N \times \beta^2}$ denote a matrix of joint histogram positional embeddings under the partitioning induced by τ^*. Then for dimensions $H = W = \beta^2 = \rho^2$, the embeddings given by $\gamma \circ \phi \circ \tau^*$ are equivalent to the canonical ViT embeddings given by $\gamma^* \circ \phi^* \circ \tau^*$ up to proportionality.*

We provide necessary definitions and proofs for Prop. 1 in Appendix A, demonstrating that our proposed framework includes the canonical ViT architecture as a special case; an essential property for modularity.

3 Experiments and Results

We train ViTs with different tokenization strategies (ViT, RViT, SPiT) using base (B) and small (S) capacities on a general purpose classification task on ImageNet [11] (IN1k). We design our experiments with the goal of evaluating the quality of the resulting tokenized representations of the images. See details about the training setup in Appendix C.

3.1 Classification

We evaluate the models by fine-tuning on Cifar100 [22] and Caltech256 [16], in addition to validation using the INReaL labels [4], ablating the effect of gradient features. We also evaluate our models by replacing the linear classifier head with a k-nearest neighbours (kNN) classifier over the representation space of different models, focusing solely on the clustering quality of the class tokens in the embedded space [8, 28]. Table 1 gives an overview of the results. We include results for the Small (S) capacity models in Table C.1.

Our results show that ViTs with superpixel tokenization can be effectively trained for classification tasks. For models with gradient texture features, superpixel tokenization performs comparably to square partitioning, noting that superpixel tokenization with gradient excluding feature extraction underperforms. We conjecture that this is likely due to high irregularity in regions, and confirms our conjecture that gradient features can compensate for loss of information from interpolation. Our findings in Section 2.4 also supports this.

When comparing validation results, we note that SPiT performs better than the ViT over INReaL. This indicates that the model is more robust to label noise or localized-multiclass tasks, and likely generalizes better in real-world scenarios. This is further evident by the fact that SPiT performs better with kNN classification for higher resolution images in IN1k and CalTech256 than the ViT model. We note that square tokens perform better on Cifar100. *This is to be expected* as quantization artifacts from low resolution images persist under upscaling, favoring square patches.

Overall, our results indicates that SPiT with gradient features outperforms the vanilla ViT in classification tasks. However, when including our proposed gradient features in the standard ViT, *the results are not significant enough to claim a clear benefit on general purpose classification tasks*. We emphasize that *comparable performance is a positive result*, since our focus is on demonstrating the feasibility of modular superpixel tokenization as a new research direction for vision transformers. For more details, see Appendix G.

3.2 Evaluating Tokenized Representations

To evaluate the cohesive quality of the tokenized representations, we look to quantify the *faithfulness of attributions*, and the model's performance on *zero-shot unsupervised segmentation*. These were selected to give insight into the embedded context of the tokenized representation of the image.

Faithfulness of Attributions: One of the attractive properties of ViTs is the inherent interpretability provided by their attention mechanisms. Techniques such as attention rollout [8, 14], attention flow [1], and PCA projections [28] have been leveraged to visualize the reasoning behind the model's decisions. Unlike gradient-based attributions, which often lack clear causal links to model predictions [3], attention based attributions are intrinsically connected to the flow of information in the model, and provide direct insight into the decision-making process in an interpretable manner. They are, however, constrained by the granularity and semantic alignment of the original tokenization scheme. Classical methods such as LIME [30] provides a well-established counterfactual framework for post-hoc explainability with superpixel partitions using Quickshift [43] or SLIC [2] with local linear surrogate models.

To quantify the faithfulness of interpretations under different tokenization strategies, we compute the attention flow of the model in addition to PCA projected features and contrast this with attributions from LIME with indepen-

Table 2: Faithfulness of Attributions, w. CI (95%).

	ViT-B16 (IN1k)		RViT-B16 (IN1k)		SPiT-B16 (IN1k)	
	Comp ↑	Suff ↓	Comp ↑	Suff ↓	Comp ↑	Suff ↓
LIME/SLIC	**0.244 ± 0.004**	**0.543 ± 0.006**	**0.236 ± 0.004**	**0.591 ± 0.007**	0.244 ± 0.005	**0.520 ± 0.006**
Att.Flow	0.160 ± 0.004	0.664 ± 0.006	0.223 ± 0.005	0.685 ± 0.007	**0.259 ± 0.006**	0.558 ± 0.006
Prot.PCA	0.206 ± 0.005	0.710 ± 0.006	0.209 ± 0.005	0.691 ± 0.007	0.256 ± 0.005	0.592 ± 0.006

Color coding: baseline, weaker than baseline, stronger than baseline.

Table 3: Results for unsupervised salient segmentation with TokenCut. Models using additional postprocessing are included for completeness are colored in gray.

		ECSSD			DUTS			DUT-OMRON		
Model	Postproc.	max F_β	IoU	Acc.	max F_β	IoU	Acc.	max F_β	IoU	Acc.
DINO-B14[†]	✓	0.874	0.772	**0.934**	0.755	0.624	**0.914**	0.697	**0.618**	**0.897**
DINO-B14[†]	✗	0.803	0.712	0.918	0.672	0.576	0.903	0.600	0.533	0.880
SPiT-B16	✗	**0.903**	**0.773**	**0.934**	**0.771**	**0.639**	0.894	**0.711**	0.564	0.868

[†] As reported by Wang et al. [47].

dently computed SLIC superpixels, and measure faithfulness using *comprehensiveness* (Comp) and *sufficiency* (Suff) [13]. These metrics have been shown to be the two strongest quantitative measures of attributions for transformers [9]. See Appendix D for details.

The results in Table 2 suggests that predictions extracted from the attention flow and PCA using the SPiT model provide *better comprehensiveness scores* than interpretations from LIME, indicating that SPiT models produce attributions that more effectively exclude irrelevant regions of the image. A one-sided t-test confirms that the improvement in comprehensiveness between Att.Flow and LIME for the SPiT model is statistically significant.[4] Contrarily, our results show that interpretations extracted from the ViT and RViT models are *less faithful to the predictions than interpretations* procured with LIME. Furthermore, we note that the sufficiency score for SPiT models are closer to the baseline LIME interpretations than what we observe for the ViT, indicating that the interpretations from SPiT model captures the most essential features better than a canonical ViT. Figs. 1, D.1, D.2, D.4, and D.5 shows that the granularity of superpixel tokens provide interpretations that closely align with the semantic content of the image.

Unsupervised Segmentation: Superpixels have historically been applied in dense prediction tasks such as segmentation and object detection [23, 51] as a lower-dimensional prior for dense prediction tasks. To evaluate our tokens, we are particularly interested in tasks for which the outputs of the pre-trained model can be leveraged directly, without the addition of a downstream decoder. Wang et al. [47] propose an unsupervised methodology for extracting salient segmentation maps for any transformer model using normalized graph cut [35].

[4] One-sided t-test (Att.Flow > LIME): $(t = 6.54, p < 10^{-10}, \mathrm{df} = 49664)$.

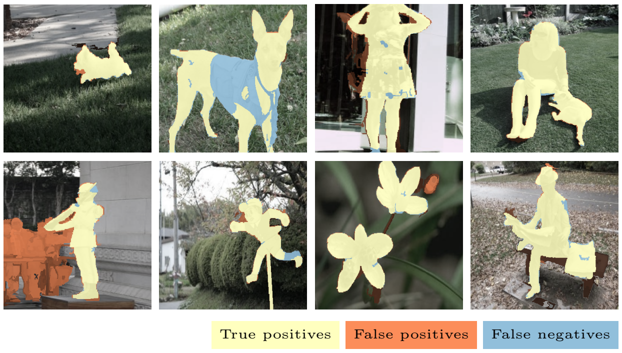

Fig. 4: Non-cherry picked samples ({0257..0264}.jpg) of unsupervised zero-shot segmentation results on ECSSD.

We conduct experiments extending this well-established method to showcase preliminary out-of-the-box capabilities on dense prediction tasks, with details of the experimental setup in Appendix E.

Table 3 shows results for the ECSSD [52], DUTS [44] and DUT-OMRON [54] datasets, and demonstrates that SPiT compares favorably to DINO [8] under the TokenCut framework, *notably without any form of postprocessing*. The results indicate that our tokenizer has strong semantic alignment with image content, and that our proposed framework is capable of dense predictions without learnable tokenization. We use the same metrics as the original TokenCut framework; for $\max F_\beta$ we set $\beta = 1/3$ and take the maximum F-score over 255 uniformly sampled thresholds. A series of non-cherry picked results are featured in Fig. 4.

3.3 Ablations

Tokenizer Generalization: In in Section 2.4 we showed that our framework generalizes the canonical ViT. This allows us to contrast different tokenization strategies across models by directly swapping tokenizers, emphasizing the modularity of our framework. We report the relative change in accuracy (Δ Acc.) of models when swapping tokenizers in Table 4.

Our results show that ViTs with square tokenization performs poorly when evaluated on irregular patches. We observe an increase in accuracy for RViT models when evaluated over square patches. Furthermore, we see that the SPiT models also generalize well to both to square and Voronoi tokens, but is highly dependent on the gradient features. With gradient features, we note a minor drop in accuracy when evaluating Voronoi tokens with SPiT, and superpixel tokens with RViT. This supports our conjecture that gradient features help encode texture information for irregular patches.

Table 4: Tokenizer Generalization.

		Δ Acc. ↑ (IN1K)		
Model	Grad.	ViT	RViT	SPiT
ViT-B16	✗	0.000	−0.551	−0.801
ViT-B16	✓	0.000	−0.494	−0.798
RViT-B16	✗	0.006	0.000	−0.593
RViT-B16	✓	0.003	0.000	−0.163
SPiT-B16	✗	−0.407	−0.464	0.000
SPiT-B16	✓	−0.200	−0.063	0.000

Table 5: Superpixel Evaluation.

	BSDS500		SBD		Time
	R^2↑	$\lvert\pi\rvert$ ↓	R^2↑	$\lvert\pi\rvert$ ↓	s/Im. ↓
ETPS†	0.924	651.0	0.955	648.1	0.3268
SEEDS†	0.901	670.6	0.944	644.9	0.4501
SLIC†	0.847	575.3	0.897	592.2	0.0729
Watershed†	0.803	608.1	0.871	641.1	0.0038
SPiT	0.914	595.0	0.948	570.2	0.0047

†As reported by Stutz et al. [37].

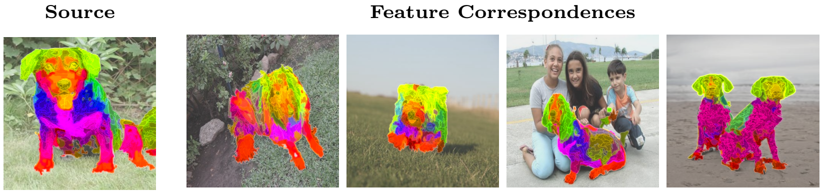

Fig. 5: Feature correspondences from a source image (left) to target images (right), mapped via normalized single head cross attention and colored using low rank PCA. We show more results in Appendix F.

Quality of Superpixels To evaluate the quality of superpixels, we compute the explained variation [27, 37] given by

$$R^2(\pi \mid \xi) = \frac{1}{\mathrm{Var}(\xi)} \sum_{S \in \pi} \Pr(S)\big(\mathbb{E}(\xi \cap S) - \mathbb{E}(\xi)\big)^2, \tag{8}$$

where $\Pr(S) = |S|/|\xi|$. The explained variation quantifies how well the superpixels capture the inherent structures in an image by measuring the amount of dispersion which can be attributed to the partitioning π. An ideal algorithm would produce a high R^2 with a minimal number of superpixels. We compare our approach with SotA superpixel methods [37] in Table 5, demonstrating that our superpixel algorithm performs comparably to top performing methods with substantially lower inference time, which is crucial for on-line tokenization.

Feature Correspondences Oquab et al. [28] visualize feature correspondences between images to examine the consistency of token representations across images for models trained with contrastive learning. Given the strong attribution scores of superpixel tokenization, we were interested to see how features correspond across images with similar, but not necessarily identical classes. We compute cross attention over normalized features between a source and target images, and visualize the correspondences using a low rank PCA with three channels. Figs. 5, F.1, and F.2 demonstrates that the features from SPiT provide strong feature correspondence properties without self-supervised pretraining, which is generally considered to provide more robust representations independent of downstream tasks.

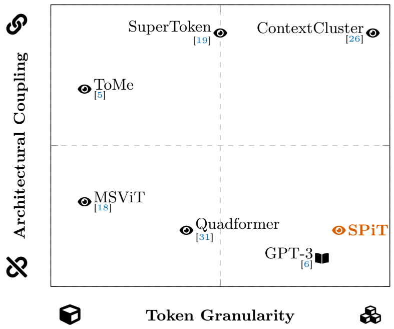

Fig. 6: Taxonomy of adaptive tokenization in transformers. Tokenization ranges from decoupled (⛓️‍💥) to coupled (🔗) to the transformer architecture, and from coarse (📦) to fine (🧊) token granularity. To contextualize vision models (👁) with LLMs (📖), GPT-3 [6] is included for reference.

4 Discussion and Related Work

Related Work Interest in adaptive tokenization is burgeoning in the field. We propose a taxonomy of adaptive tokenization with two main dimensions illustrated in Fig. 6. The first dimension illustrates the *coupling or integration* of tokenization into the transformer architecture. Several approaches [5, 19, 26] are inherently coupled to the architecture, while others adopt a decoupled approach [18, 31] which more closely aligns with our framework. The taxonomy is extended by a dimension of *token granularity*, measuring the proximity to modelling with pixel-level precision. Together, these dimensions facilitate an understanding of adaptive tokenization approaches for ViTs.

A significant body of current research is primarily designed to improve scaling and overall compute for attention [5, 32, 55] by leveraging token merging strategies in the transformer layers with square patches, and can as such be considered *low-granularity coupled approaches*. Distinctively, SuperToken [19] applies a coupled approach to extract a non-uniform token representation. The approach is fundamentally patch based, and does not aim for pixel-level granularity.

In contrast, multi-scale tokenization [18, 31] apply a *decoupled approach* where the tokenizer is independent of the transformer architecture. These are commensurable with *any transformer backbone*, and improve computational overhead. While square tokens operate on a *lower level of granularity*, there is significant potential for synergy between these approaches and our own, particularly given the hierarchical nature of SPiT. On the periphery, Ma et al. [26] propose a pixel-level clustering method with a *coupled high granularity approach*.

Limitations Our proposed framework is not optimizable with gradient based methods. Ideally, adaptable tokenization should be learnable in an end-to-end framework. However, such an approach needs to be carefully designed to not add undue computational overhead, and should ideally not be limited by a predefined number of tokens. Moreover, we see that irregular tokenization require additional gradient features to perform. While our framework provides competitive performance, it should be seen as an early step towards more flexible tokenization strategies, with several opportunities for further optimization. We provide visualization of edge cases for attributions in Fig. D.3.

Further Work Our work is distinguishable as a *decoupled high-granularity apprach* with multiple paths for further work. We see strong potential in exploring graph neural networks (GNNs) for tokenization, and hierarchical properties could be leveraged in self-supervised frameworks such as DINO [8], or pyramid models [45, 46] in a coupled approach. The modularity of our framework provides opportunites for research into the dynamic between ViTs and tokenization. Coupling SPiT with gating [18] or merging [5] could further improve scalability, and allow for a learnable framework. More work can be done in studying the effects of irregularity in feature extraction, as discussed in Section 3.3.

5 Conclusion

In this work, we posit tokenization as a modular component that generalize the canonical ViT backbone, and show that irregular tokenization with superpixels is commensurable with transformer architectures. Our experiments demonstrate that superpixel tokens have a significant impact on extracted attributions for predictions, and are amenable to unsupervised segmentation tasks without a separate decoder model. Moreover, we show that concatenated gradient features improve performance of base capacity ViTs, and that irregular tokenizers generalize between different tokenization strategies. Our experiments were performed with standard models and training to limit confounding factors in our results.

Acknowledgments

Computations were performed on resources provided by Sigma2 (NRIS, Norway), Project NN8104K. We acknowledge Sigma2 for awarding access to the LUMI supercomputer, owned by the EuroHPC Joint Undertaking, hosted by CSC (Finland) and the LUMI consortium through Sigma2, Norway, Project no. 465000262. This work was funded in part by the Research Council of Norway, via the Visual Intelligence Centre for Research-based Innovation (grant no. 309439), and Consortium Partners.

Bibliography

[1] Abnar, S., Zuidema, W.H.: Quantifying attention flow in transformers. In: Jurafsky, D., Chai, J., Schluter, N., Tetreault, J.R. (eds.) Conf. Assoc. Comput. Ling. (ACL), pp. 4190–4197, Association for Computational Linguistics (2020), URL https://doi.org/10.18653/v1/2020.acl-main.385

[2] Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., Süsstrunk, S.: Slic superpixels compared to state-of-the-art superpixel methods. IEEE Trans. Pattern Anal. Mach. Intell. **34**(11), 2274–2282 (2012), https://doi.org/10.1109/TPAMI.2012.120

[3] Adebayo, J., Gilmer, J., Muelly, M., Goodfellow, I.J., Hardt, M., Kim, B.: Sanity checks for saliency maps. In: Bengio, S., Wallach, H.M., Larochelle, H., Grauman, K., Cesa-Bianchi, N., Garnett, R. (eds.) Adv. Neural Inf. Process. Sys. (NeurIPS), pp. 9525–9536 (2018), URL https://proceedings.neurips.cc/paper/2018/hash/294a8ed24b1ad22ec2e7efea049b8737-Abstract.html

[4] Beyer, L., Hénaff, O.J., Kolesnikov, A., Zhai, X., van den Oord, A.: Are we done with ImageNet? CoRR **abs/2006.07159** (2020), URL https://arxiv.org/abs/2006.07159

[5] Bolya, D., Fu, C.Y., Dai, X., Zhang, P., Feichtenhofer, C., Hoffman, J.: Token merging: Your vit but faster. In: Inter. Conf. Learn. Represent. (ICLR) (2023), URL https://openreview.net/forum?id=JroZRaRw7Eu

[6] Brown, T.B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D.M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., Amodei, D.: Language models are few-shot learners. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., Lin, H. (eds.) Adv. Neural Inf. Process. Sys. (NeurIPS) (2020), URL https://proceedings.neurips.cc/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html

[7] Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J. (eds.) European Conf. Comput. Vis. (ECCV), Lecture Notes in Computer Science, vol. 12346, pp. 213–229, Springer (2020), https://doi.org/10.1007/978-3-030-58452-8_13, URL https://doi.org/10.1007/978-3-030-58452-8_13

[8] Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., Joulin, A.: Emerging properties in self-supervised vision transformers. In: IEEE/CVF Inter. Conf. Comput. Vis. Pattern Recog. (CVPR), pp. 9630–9640, IEEE (2021), URL https://doi.org/10.1109/ICCV48922.2021.00951

[9] Chan, C.S., Kong, H., Guanqing, L.: A comparative study of faithfulness metrics for model interpretability methods. In: Conf. Assoc. Comput. Ling.

(ACL), pp. 5029–5038, Association for Computational Linguistics, Dublin, Ireland (2022), URL https://aclanthology.org/2022.acl-long.345

[10] Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: IEEE/CVF Inter. Conf. Comput. Vis. Pattern Recog. (CVPR), vol. 1, pp. 886–893 vol. 1 (2005), https://doi.org/10.1109/CVPR.2005.177

[11] Deng, J., Dong, W., Socher, R., Li, L., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: IEEE/CVF Inter. Conf. Comput. Vis. Pattern Recog. (CVPR), pp. 248–255, IEEE Computer Society (2009), URL https://doi.org/10.1109/CVPR.2009.5206848

[12] Devlin, J., Chang, M., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: Burstein, J., Doran, C., Solorio, T. (eds.) Conf. North Amer. Ch. Assoc. Comput. Ling. (NAACL), pp. 4171–4186, Association for Computational Linguistics (2019), URL https://doi.org/10.18653/v1/n19-1423

[13] DeYoung, J., Jain, S., Rajani, N.F., Lehman, E., Xiong, C., Socher, R., Wallace, B.C.: ERASER: A benchmark to evaluate rationalized NLP models. In: Conf. Assoc. Comput. Ling. (ACL), pp. 4443–4458, Association for Computational Linguistics, Online (Jul 2020), URL https://aclanthology.org/2020.acl-main.408

[14] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N.: An image is worth 16x16 words: Transformers for image recognition at scale. In: Inter. Conf. Learn. Represent. (ICLR) (2021), URL https://openreview.net/forum?id=YicbFdNTTy

[15] Fellbaum, C.: WordNet: An Electronic Lexical Database. Bradford Books (1998), URL https://mitpress.mit.edu/9780262561167/

[16] Griffin, G., Holub, A., Perona, P.: Caltech 256 (Apr 2022), https://doi.org/10.22002/D1.20087

[17] Hamilton, M., Zhang, Z., Hariharan, B., Snavely, N., Freeman, W.T.: Unsupervised semantic segmentation by distilling feature correspondences. In: Inter. Conf. Learn. Represent. (ICLR) (2022), URL https://openreview.net/forum?id=SaKO6z6Hl0c

[18] Havtorn, J.D., Royer, A., Blankevoort, T., Bejnordi, B.E.: Msvit: Dynamic mixed-scale tokenization for vision transformers. In: IEEE Inter. Conf. Comput. Vis. (ICCV), pp. 838–848 (October 2023), URL https://doi.org/10.1109/ICCVW60793.2023.00091

[19] Huang, H., Zhou, X., Cao, J., He, R., Tan, T.: Vision transformer with super token sampling (2022)

[20] Johnson, M., Schuster, M., Le, Q.V., Krikun, M., Wu, Y., Chen, Z., Thorat, N., Viégas, F.B., Wattenberg, M., Corrado, G., Hughes, M., Dean, J.: Google's multilingual neural machine translation system: Enabling zero-shot translation. Trans. Assoc. Comput. Linguistics **5**, 339–351 (2017), URL https://doi.org/10.1162/tacl_a_00065

[21] Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W., Dollár, P., Girshick, R.B.:

Segment anything (2023), URL https://doi.org/10.1109/ICCV51070.2023.00371

[22] Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images (2009)

[23] Ladický, L., Russell, C., Kohli, P., Torr, P.H.: Associative hierarchical crfs for object class image segmentation. In: IEEE Inter. Conf. Comput. Vis. (ICCV), pp. 739–746 (2009), https://doi.org/10.1109/ICCV.2009.5459248

[24] Leung, T., Malik, J.: Representing and recognizing the visual appearance of materials using three-dimensional textons. Inter. J. Comput. Vis. **43**(1), 29–44 (Jun 2001), ISSN 1573-1405, URL https://doi.org/10.1023/A:1011126920638

[25] Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. In: IEEE Inter. Conf. Comput. Vis. (ICCV), pp. 9992–10002, IEEE (2021), https://doi.org/10.1109/ICCV48922.2021.00986, URL https://doi.org/10.1109/ICCV48922.2021.00986

[26] Ma, X., Zhou, Y., Wang, H., Qin, C., Sun, B., Liu, C., Fu, Y.: Image as set of points. In: Inter. Conf. Learn. Represent. (ICLR) (2023), URL https://openreview.net/forum?id=awnvqZja69

[27] Moore, A.P., Prince, S.J.D., Warrell, J., Mohammed, U., Jones, G.: Superpixel lattices. In: IEEE/CVF Inter. Conf. Comput. Vis. Pattern Recog. (CVPR) (2008), https://doi.org/10.1109/CVPR.2008.4587471

[28] Oquab, M., Darcet, T., Moutakanni, T., Vo, H.V., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., Assran, M., Ballas, N., Galuba, W., Howes, R., Huang, P., Li, S., Misra, I., Rabbat, M., Sharma, V., Synnaeve, G., Xu, H., Jégou, H., Mairal, J., Labatut, P., Joulin, A., Bojanowski, P.: Dinov2: Learning robust visual features without supervision. Trans. Mach. Learn. Res. (2024), URL https://openreview.net/forum?id=a68SUt6zFt

[29] Perona, P., Malik, J.: Scale-space and edge detection using anisotropic diffusion. IEEE Trans. Pattern Anal. Mach. Intell. **12**(7), 629–639 (1990), https://doi.org/10.1109/34.56205

[30] Ribeiro, M.T., Singh, S., Guestrin, C.: "why should I trust you?" explaining the predictions of any classifier. In: ACM Conf. Knowl. Discov. Data Min. (ACM SIGKDD), pp. 1135–1144 (2016), URL https://doi.org/10.1145/2939672.2939778

[31] Ronen, T., Levy, O., Golbert, A.: Vision transformers with mixed-resolution tokenization. In: IEEE/CVF Inter. Conf. Comput. Vis. Pattern Recog. (CVPR), pp. 4612–4621 (2023)

[32] Ryoo, M.S., Piergiovanni, A.J., Arnab, A., Dehghani, M., Angelova, A.: TokenLearner: Adaptive space-time tokenization for videos. In: Ranzato, M., Beygelzimer, A., Dauphin, Y.N., Liang, P., Vaughan, J.W. (eds.) Adv. Neural Inf. Process. Sys. (NeurIPS), pp. 12786–12797 (2021), URL https://proceedings.neurips.cc/paper/2021/hash/6a30e32e56fce5cf381895dfe6ca7b6f-Abstract.html

[33] Scharr, H.: Optimal filters for extended optical flow. In: Jähne, B., Mester, R., Barth, E., Scharr, H. (eds.) Int. Wksp. Compl. Mot. (IWCM), pp. 14–29, Springer Berlin Heidelberg, Berlin, Heidelberg (2007), ISBN 978-3-540-69866-1

[34] Sennrich, R., Haddow, B., Birch, A.: Neural machine translation of rare words with subword units. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers, The Association for Computer Linguistics (2016), URL https://doi.org/10.18653/v1/p16-1162

[35] Shi, J., Malik, J.: Normalized cuts and image segmentation. IEEE Trans. Pattern Anal. Mach. Intell. **22**(8), 888–905 (2000)

[36] Steiner, A., Kolesnikov, A., Zhai, X., Wightman, R., Uszkoreit, J., Beyer, L.: How to train your vit? data, augmentation, and regularization in vision transformers. Trans. Mach. Learn. Res. (2022), URL https://openreview.net/forum?id=4nPswr1KcP

[37] Stutz, D., Hermans, A., Leibe, B.: Superpixels: An evaluation of the state-of-the-art. Comput. Vis. Image Underst. **166**, 1–27 (2018), ISSN 1077-3142, URL https://www.sciencedirect.com/science/article/pii/S1077314217300589

[38] Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., Jégou, H.: Training data-efficient image transformers & distillation through attention. In: Meila, M., Zhang, T. (eds.) Inter. Conf. Mach. Learn. (ICML), Proceedings of Machine Learning Research, vol. 139, pp. 10347–10357, PMLR (2021), URL http://proceedings.mlr.press/v139/touvron21a.html

[39] Touvron, H., Cord, M., Jégou, H.: Deit III: revenge of the vit. In: Avidan, S., Brostow, G.J., Cissé, M., Farinella, G.M., Hassner, T. (eds.) European Conf. Comput. Vis. (ECCV), Lecture Notes in Computer Science, vol. 13684, pp. 516–533, Springer (2022), https://doi.org/10.1007/978-3-031-20053-3_30, URL https://doi.org/10.1007/978-3-031-20053-3_30

[40] Touvron, H., Cord, M., Sablayrolles, A., Synnaeve, G., Jégou, H.: Going deeper with image transformers. In: IEEE Inter. Conf. Comput. Vis. (ICCV), pp. 32–42, IEEE (2021), URL https://doi.org/10.1109/ICCV48922.2021.00010

[41] Touvron, H., Cord, M., Sablayrolles, A., Synnaeve, G., Jégou, H.: Going deeper with image transformers. In: IEEE Inter. Conf. Comput. Vis. (ICCV), pp. 32–42, IEEE (2021), URL https://doi.org/10.1109/ICCV48922.2021.00010

[42] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. In: Guyon, I., von Luxburg, U., Bengio, S., Wallach, H.M., Fergus, R., Vishwanathan, S.V.N., Garnett, R. (eds.) Adv. Neural Inf. Process. Sys. (NeurIPS), pp. 5998–6008 (2017), URL https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html

[43] Vedaldi, A., Soatto, S.: Quick shift and kernel methods for mode seeking. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) European Conf. Comput.

Vis. (ECCV), pp. 705–718, Springer Berlin Heidelberg, Berlin, Heidelberg (2008), ISBN 978-3-540-88693-8

[44] Wang, L., Lu, H., Wang, Y., Feng, M., Wang, D., Yin, B., Ruan, X.: Learning to detect salient objects with image-level supervision. In: IEEE/CVF Inter. Conf. Comput. Vis. Pattern Recog. (CVPR) (2017)

[45] Wang, W., Xie, E., Li, X., Fan, D., Song, K., Liang, D., Lu, T., Luo, P., Shao, L.: Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In: IEEE Inter. Conf. Comput. Vis. (ICCV), pp. 548–558, IEEE (2021), https://doi.org/10.1109/ICCV48922.2021.00061, URL https://doi.org/10.1109/ICCV48922.2021.00061

[46] Wang, W., Xie, E., Li, X., Fan, D., Song, K., Liang, D., Lu, T., Luo, P., Shao, L.: PVT v2: Improved baselines with pyramid vision transformer. Comput. Vis. Media **8**(3), 415–424 (2022), https://doi.org/10.1007/s41095-022-0274-8, URL https://doi.org/10.1007/s41095-022-0274-8

[47] Wang, Y., Shen, X., Hu, S.X., Yuan, Y., Crowley, J.L., Vaufreydaz, D.: Self-supervised transformers for unsupervised object discovery using normalized cut. In: IEEE/CVF Inter. Conf. Comput. Vis. Pattern Recog. (CVPR) (2022)

[48] Wei, X., Yang, Q., Gong, Y., Ahuja, N., Yang, M.: Superpixel hierarchy. IEEE Trans. Image Process. **27**(10), 4838–4849 (2018), URL https://doi.org/10.1109/TIP.2018.2836300

[49] Xiaohan, Y., Yla-Jaaski, J., Huttunen, O., Vehkomaki, T., Sipila, O., Katila, T.: Image segmentation combining region growing and edge detection. In: IEEE Inter. Conf. Pattern Recog. (ICPR), pp. 481–484 (1992), https://doi.org/10.1109/ICPR.1992.202029

[50] Xie, E., Wang, W., Yu, Z., Anandkumar, A., Alvarez, J.M., Luo, P.: SegFormer: Simple and efficient design for semantic segmentation with transformers. In: Ranzato, M., Beygelzimer, A., Dauphin, Y.N., Liang, P., Vaughan, J.W. (eds.) Adv. Neural Inf. Process. Sys. (NeurIPS), pp. 12077–12090 (2021), URL https://proceedings.neurips.cc/paper/2021/hash/64f1f27bf1b4ec22924fd0acb550c235-Abstract.html

[51] Yan, J., Yu, Y., Zhu, X., Lei, Z., Li, S.Z.: Object detection by labeling superpixels. In: IEEE/CVF Inter. Conf. Comput. Vis. Pattern Recog. (CVPR), pp. 5107–5116 (2015), https://doi.org/10.1109/CVPR.2015.7299146

[52] Yan, Q., Xu, L., Shi, J., Jia, J.: Hierarchical saliency detection. In: IEEE/CVF Inter. Conf. Comput. Vis. Pattern Recog. (CVPR), pp. 1155–1162 (2013), https://doi.org/10.1109/CVPR.2013.153

[53] Yan, T., Huang, X., Zhao, Q.: Hierarchical superpixel segmentation by parallel crtrees labeling. IEEE Trans. Image Process. **31**, 4719–4732 (2022), https://doi.org/10.1109/TIP.2022.3187563

[54] Yang, C., Zhang, L., Lu, H., Ruan, X., Yang, M.H.: Saliency detection via graph-based manifold ranking. In: IEEE/CVF Inter. Conf. Comput. Vis. Pattern Recog. (CVPR), pp. 3166–3173, IEEE (2013)

[55] Yuan, L., Chen, Y., Wang, T., Yu, W., Shi, Y., Jiang, Z., Tay, F.E.H., Feng, J., Yan, S.: Tokens-to-Token ViT: Training vision transformers from

scratch on imagenet. In: IEEE Inter. Conf. Comput. Vis. (ICCV), pp. 538–547, IEEE (2021), https://doi.org/10.1109/ICCV48922.2021.00060, URL https://doi.org/10.1109/ICCV48922.2021.00060

[56] Yun, S., Han, D., Chun, S., Oh, S.J., Yoo, Y., Choe, J.: Cutmix: Regularization strategy to train strong classifiers with localizable features. In: IEEE Inter. Conf. Comput. Vis. (ICCV), pp. 6022–6031, IEEE (2019), URL https://doi.org/10.1109/ICCV.2019.00612

A Equivalence of Frameworks

Definition 1 (ViT Tokenization). *Let $\xi\colon H\times W \to \mathbb{R}^C$ be an image signal with tensor representation $\boldsymbol{\xi} \in \mathbb{R}^{H\times W\times C}$. The canonical ViT tokenization operator $\tau^*\colon \mathbb{R}^{H\times W\times C} \to \mathbb{R}^{N\times\rho\times\rho\times C}$ partitions the image into $N = \lceil\frac{H}{\rho}\rceil \cdot \lceil\frac{W}{\rho}\rceil$ non-overlapping C-channel square zero-padded patches. For the case where we have $H \bmod \rho = W \bmod \rho = 0$, we get $N = \frac{H}{\rho}\cdot\frac{W}{\rho}$, and no padding is necessary.*

Definition 2 (ViT Features). *Let ρ denote the patch dimension of a canonical ViT tokenizer τ^*, and let $M = \rho^2 C$. The canonical ViT feature extractor $\phi^*\colon \mathbb{R}^{N\times\rho\times\rho\times C} \to \mathbb{R}^{N\times M}$ is given by $\phi^* = \mathrm{vec}_M$, where vec_M denotes the vectorization operator applied to each of the N patches via $\rho\times\rho\times C \mapsto M$.*

Definition 3 (ViT Embedder). *Let ϕ^* be a canonical ViT feature extractor, and let $Q \in \mathbb{R}^{N\times D}$ denote a positional encoding. The canonical ViT embedder $\gamma^*\colon \mathbb{R}^{N\times M} \to \mathbb{R}^{N\times D}$ is given by*

$$\gamma^*(z) = L_\theta z + Q$$

where $L_\theta\colon \mathbb{R}^{N\times M} \to \mathbb{R}^{N\times D}$ is a learnable linear transformation, and Q is either a learnable set of parameters or a function of the positions of the N blocks in the partitioning induced by the canonical tokenizer τ^*.

Lemma 1 (Feature Equivalence). *Let τ^* denote a canonical ViT tokenizer with a fixed patch size ρ, and let ϕ denote a gradient excluding interpolating feature extractor with $\beta = \rho$. Then the operations $\phi \circ \tau^*$ are equivalent to the canonical ViT operations $\phi^* \circ \tau^*$.*

Proof. The proof is highly trivial but illustrative. Note that for each of the N square patches generated by τ, the extractor ϕ performs an interpolation to rescale the patch to a fixed resolution of $\beta\times\beta$. However, for $\beta = \rho$ the patches already match the target dimensions exactly. It follows that the interpolation operation reduces to identity. The vectorization operator is equivalent for both mappings, hence $\phi = \mathrm{vec}_N = \phi^*$.

Proposition 1 (Embedding Equivalence). *Let τ^* denote an canonical ViT tokenizer with a fixed patch size ρ, let ϕ denote a gradient excluding interpolated feature extractor, and let γ^*, γ denote embedding layers with equivalent linear projections $L_\theta^* = L_\theta$. Let $\hat{\xi}^{(\mathrm{pos})} \in \mathbb{R}^{N\times\beta^2}$ denote a matrix of joint histogram positional embeddings under the partitioning induced by τ^*. Then for dimensions $H = W = \beta^2 = \rho^2$, the embeddings given by $\gamma \circ \phi \circ \tau^*$ are equivalent to the canonical ViT embeddings given by $\gamma^* \circ \phi^* \circ \tau^*$ up to proportionality.*

Proof. We first note that we can assume $\hat{\xi}^{(\mathrm{pos})}$ is a matrix with single entry components, since under $\beta = \rho$ and $N = \beta^2$, each vectorized histogram feature is a scaled unit vector $c_n \boldsymbol{e}_n$ with $n = 1, \dots, N$. Moreover, since the partitioning inferred by τ^* exhaustively covers the spatial dimensions $H\times W$, the histograms essentially span the standard basis, such that $\hat{\xi}^{(\mathrm{pos})}$ is diagonal. Furthermore, since each patch is of the same size we have equal contribution towards each entry, such that $c_n = c_m$ for all $m \neq n$. Therefore, without loss of generality, we can ignore the scalars and simply consider $\hat{\xi}^{(\text{pos})} = I$ as an identity matrix. From Lemma 1 we have that $z = (\phi^* \circ \tau^*)(\xi) = (\phi \circ \tau^*)(\xi)$. Then, since

$$\gamma^*(z) = L_\theta z + Q = [L_\theta, Q] \begin{bmatrix} z \\ I \end{bmatrix} = \gamma(z) \tag{9}$$

we have that $\gamma = \gamma^*$ up to proportionality for some constant $c = c_n$.

Remark 1. While we only demonstrate the equality up to proportionality, this can generally be ignored since we can effectively choose our linear projection under γ to be L_θ / c. We note that while the equality holds for empirical histograms, equality does not strictly hold for $\hat{\xi}^{(\text{pos})}$ computed using KDE with a Gaussian kernel, however we point out that the contribution from the tails of a kernel K_σ with a small bandwidth is effectively negligible.

B Preprocessing and Superpixel Features

Compared to standard preprocessing, we use a modified normalization scheme for the features for improving the superpixel extraction. We apply a combined contrast adjustment and normalization function using a reparametrized version of the Kumaraswamy CDF. which is computationally efficient and allows more fine-grained control of the distribution of intensities than empirical normalization, which improves the superpixel partitioning.

The normalization uses a set of means μ shape parameters λ for normalizing the image and adjusting the contrast. The normalization is given by

$$\left(1 - \left(1 - x^{\lambda}\right)^{b}\right), \tag{10}$$

where b is defined by

$$b = -\frac{\ln(2)}{\ln\left(1 - \mu^{\lambda}\right)}, \tag{11}$$

and we set means $\mu_r = 0.485, \mu_g = 0.456, \mu_b = 0.406$ and $\lambda_r = 0.539, \lambda_g = 0.507, \lambda_b = 0.404$, respectively.

The features used for the superpixel extraction are further processed using anisotropic diffusion, which smoothes homogeneous regions while avoiding blurring of edges. This technique was advocated for superpixel segmentation by Xiaohan et al. [49]. We use the algorithm proposed by Perona and Malik [29] over 4 iterations, with $\kappa = 0.1$ and $\gamma = 0.5$. Note that these features are only applied for constructing the superpixels in the tokenizer. We emphasize that we do not apply anisotropic diffusion for the features in the predictive model.

Table B.1: Expected no. superpixels with SPiT over IN1K (train, CI 95%).

Im.Size	$\mathbb{E}(\|\pi^{(1)}\|)$	$\mathbb{E}(\|\pi^{(2)}\|)$	$\mathbb{E}(\|\pi^{(3)}\|)$	$\mathbb{E}(\|\pi^{(4)}\|)$
224	11 940.278 ± 2.848	3155.512 ± 0.808	794.650 ± 0.209	197.411 ± 0.052
256	15 496.020 ± 3.786	4097.510 ± 1.074	1031.727 ± 0.277	256.051 ± 0.071
384	34 084.297 ± 9.188	9047.289 ± 2.586	2287.822 ± 0.669	567.690 ± 0.172

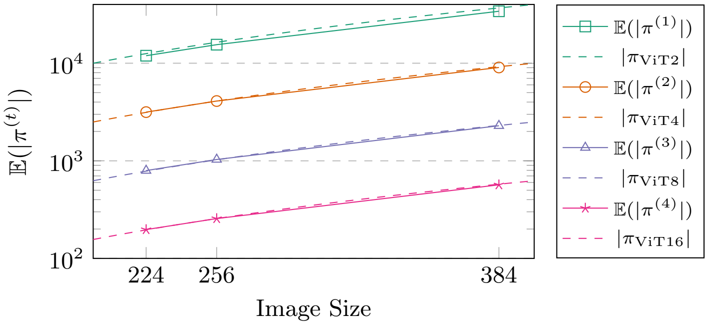

Fig. B.1: Expected no. superpixels with SPiT compared with no. ViT patches.

Number of Superpixels: In Section 2.2, we mention that SPiT gives comparable numbers of partitions to a ViT with different patch sizes. Table B.1 shows empirical results for superpixel sizes using the SPiT tokenizer over the training images of ImageNet1k, and Fig. B.1 compares the results to number of patches with canonical ViT tokenization, demonstrating the validity of our claims.

Importantly, these results also reveal much about effective inference times. In Table 5, we show that the overhead for constructing the superpixels is very low. However, the number of tokens depends on the image. Images with large homogeneous regions will be processed faster, while images with many independent regions will necessary incur a cost. Nevertheless, the results in Table B.1 show that we will, on average, have comparable inference times to a canonical ViT due to the beneficial properties of our proposed superpixel tokenization.

Final Thresholding : Adaptable tokenization frameworks does not necessarily entail an overall drop in inference throughput. Contrarily, it could potentially be leveraged to substantially improve inference speed by designing learnable methods to lower the number of tokens without decreasing performance, *e.g.* ToMe by Bolya et al. [5].

We apply an additional final merging step where we compute the euclidean distance between adjacent superpixels and merge all superpixels below a given threshold for our SPiT-B16 model with gradient features. Noting that a threshold of 0.0 retains the original model design, the results in Table B.2 indicate that models with superpixel tokenization can be optimized to improve inference throughput. We also note that taking the maximum performing tokens over all

Table B.2: Accuracy under final thresholding.

Threshold	No.Tok.	Im./s.	Avg.Acc.
0.00	556.5	718.7	0.804
0.05	513.2	749.2	0.804
0.10	441.6	844.4	0.802
0.15	365.0	950.5	0.797
0.20	293.7	1038.9	0.786

thresholds achieves an accuracy of 0.817, significantly improving the predictive performance.

C Training Details

As mentioned in Section 1.2, we use standardized ViT architectures and generally follow the recommendations provided by Steiner et al. [36]. We provide training logs, pre-trained models, and code for training models from scratch in our GitHub project repository (in the camera ready manuscript).

Classification: Training is performed over 300 epochs using the AdamW optimizer with a cosine annealing learning rate scheduler with 5 epochs of cosine annealed warmup from learning rate $\eta_{\text{start}} = 1\times10^{-5}$. The schedule maxima and minima are given by $\eta_{\max} = 3\times10^{-3}$, and $\eta_{\min} = 1\times10^{-6}$. We use a weight decay of $\lambda_{\text{dec}} = 2\times10^{-2}$ and set the smoothing term $\epsilon = 1\times10^{-7}$. In addition, we used stochastic depth dropout with a base probability of $p = 0.2$ and layer scaling. Models were pre-trained with spatial resolution 256×256.

For augmentations, we randomly select between using the RandAug framework at medium strength or using Aug3 framework by Touvron et al. [39] including CutMix [56] with parameter $\alpha = 1.0$. We use RandomResizeCrop using the standard scale $(0.08, 1.0)$ with randomly sampled interpolation modes. Since the number of partitions from the superpixel tokenizer are adapted on an image-to-image basis, we effectively constrain the maximum number of tokens during training using token dropout to balance number of tokens.

We found that a naive on-line computation of Voronoi tessellations was unnecessarily computationally expensive, hence we precompute sets of random Voronoi tessellations with 196, 256, and 576 partitions, corresponding to images of 224×224, 256×256, and 384×384 resolutions given patch size $\rho = 16$.

All training was performed on AMD MI250X GPUs. One important distinction is that we do not use quantization with `bfloat16` for training our models, instead opting for the higher 32-bit precision of `float32` since this improves consistency between vendor frameworks. Inference was carried out on a mixture of NVIDIA A100, RTX 2080Ti, Quadro P6000, and AMD MI250X to validate consistency across vendor frameworks.

Table C.1: Accuracy (Top 1) for Small (S) capacity models on classification. Note that the Small capacity models have been trained without final finetuning.

Model		INReaL		IN1k		Caltech256		Cifar100	
Name	Grad.	Lin.	kNN	Lin.	kNN	Lin.	kNN	Lin.	kNN
ViT-S16	✗	0.778	0.808	0.765	0.692	0.818	0.827	0.827	0.833
ViT-S16	✓	0.782	0.811	0.754	0.682	0.824	0.832	0.830	0.836
RViT-S16	✗	**0.829**	**0.814**	**0.767**	0.740	0.852	0.858	0.856	0.858
RViT-S16	✓	0.818	0.812	0.759	**0.741**	**0.856**	**0.861**	**0.856**	**0.859**
SPiT-S16	✗	0.746	0.796	0.689	0.628	0.767	0.771	0.761	0.769
SPiT-S16	✓	0.819	0.812	0.750	0.736	0.849	0.851	0.832	0.839

†Uncertainty measures for RViT tokenizer are detailed in Appendix Table G.1.

Fine Tuning: All base models were fine-tuned over 30 epochs with increased degrees of regularization. We increase the level of RandAug to "strong" using 2 operations with magnitude 20. Additionally, we increase the stochastic depth dropout to $p = 0.4$. Fine tuning was performed with spatial resolution 384×384, and we reduce the maximum learning rate to $\eta_{\max} = 1 \times 10^{-4}$. For the alternative classification datasets Cifar100 and Caltech256, fine tuning was performed by replacing the classification head and fine tuning for 10 epochs using AdamW with learning rate $\eta = 1 \times 10^{-4}$ and the same weight decay. No augmentation was used in this process, and images were re-scaled to 256×256 for training and evaluation.

D Interpretability and Attention Maps

For LIME explanations, we train a linear surrogate model L_{Φ} for predicting the output probabilities for the prediction of each model Φ. To encourage independence between tokenizers and LIME explanations, as well as promote direct comparability, we use SLIC with a target of $|\pi| \approx 64$ superpixels. We use Monte Carlo sampling of binary features for indicating the presence or omission of each superpixel with stochastic $p \in \mathrm{Uniform}(0.1, 0.3)$, and keep these consistent across model evaluations. We observed that certain images in the IN1k at times produced less than 5 superpixels using SLIC, hence these images were dropped from the evaluation.

The attention flow [1] of a transformer differs from the standard attention roll-out by accounting for the contributions of the residual connections in computations. The attention flow of an ℓ-layer transformer is given by

$$A_{\mathrm{Flow}} = \prod_{i=1}^{\ell} \big((1-\lambda)I + \lambda A_i\big). \tag{12}$$

where we set $\lambda = 0.9$ to account for stochastic depth and layer scaling factors while accentuating the contribution of the attention operators. We use max-aggregation over the heads to extract a unified representation. Following Dosovitskiy et al. [14] and Caron et al. [8], we extract the attention for the class token as an interpretation of the model's prediction.

For the PCA projection, we take inspiration from the visualizations technique used in the work of Oquab et al. [28]. In this work, the features of multiple images with comparable attributes are concatenated, and projected onto a set of the top principal components of the image. We compute a set of 5 prototype centroids $\nu \in \mathbb{R}^{1000 \times d \times 5}$ for each class token of each model over ImageNet using KMeans, while enforcing relative subclass orthogonality by introducing a regularization term

$$J(\nu) = \frac{\lambda_\nu}{1000} \sum_{c=1}^{1000} \| I - \nu_c^\intercal \nu_c \|_2^2, \tag{13}$$

selecting $\lambda_\nu = 0.1$. Given a prediction c, we concatenate the prototypes to the token embeddings to form a matrix $M = [\Phi(\xi;\theta)^\intercal, \nu_c^\intercal]^\intercal$. Letting $U\Sigma V^\intercal = M - \mu(M)$ be a low-rank SVD of the centered features, we then project the original features to the principal components by $\Phi(\xi;\theta)V$, and use max-aggregation to extract the attribution as an interpretation of the model's prediction. We experimented with different ranks, but found that simply using the first principal component aligned well with attention maps and LIME coefficients. This somewhat mirrors the procedure by Oquab et al. [28], where a thresholded projection on the first principal component is applied as a mask. In the interest of reproducibility, we provide links for downloading normalized attention maps for all attributions in our GitHub repository.

To quantify the faithfulness of the attributions for each model, we used comprehensiveness and sufficiency as proposed by DeYoung et al. [13]. Given a sequence of quantiles $Q \in [0,1]$ from an attribution, these metrics are given by

$$\textsc{Comp}_{Q|x,\Phi} = \frac{1}{|Q|} \sum_{q \in Q} \big(\Phi(x;\theta) - \Phi(x \setminus x_{>q};\theta) \big), \tag{14}$$

$$\textsc{Suff}_{Q|x,\Phi} = \frac{1}{|Q|} \sum_{q \in Q} \big(\Phi(x;\theta) - \Phi(x \setminus x_{\leq q};\theta) \big). \tag{15}$$

The benefit of these metrics is that they are symmetrical, and invariant to the scaling of the attributions due to using quantiles to produce the masks. Following the procedure outlined by DeYoung et al. [13] we set the quantiles to $Q = (0.01, 0.05, 0.2, 0.5)$. Figs. D.1 and D.2 show additional attributions, while Figs. D.4 and D.5 illustrate the occlusions with the selected quantiles. While we show that SPiT produces strong attributions, the proposed method is by no means free of failure cases. We find it informative to also include visualizations of the limiting edge cases for attributions in Fig. D.3.

E Unsupervised Salient Segmentation Details

The TokenCut [47] framework proposes to use a normalized cut [35] over the key features without class tokens in the last self-attention layer of the network. A

Table E.1: Extended unsupervised salient segmentation results. Models including extensive decoders and postprocessing are colored in gray.

			ECSSD			DUTS			DUT-OMRON		
Model	Method	Postproc.	max F_β	IoU	Acc.	max F_β	IoU	Acc.	max F_β	IoU	Acc.
DINO-B14[†]	TokenCut	BL	0.874	0.772	0.934	0.755	0.624	0.914	0.697	0.618	0.897
DINO-S8	SelfMask	MF	0.894	0.779	0.943	0.789	0.648	0.938	0.733	0.609	0.923
DINO-S8	SelfMask	MF+BL	0.911	0.803	0.951	0.819	0.694	0.949	0.774	0.677	0.939
DINO-S8	MOVE	Seg+MF	0.921	0.835	0.956	0.829	0.728	0.954	0.756	0.666	0.933
DINO-S8	MOVE	Seg+MF+BL	0.917	0.800	0.952	0.827	0.687	0.952	0.766	0.665	0.937
DINO-B14[†]	TokenCut	✗	0.803	0.712	0.918	0.672	0.576	0.903	0.600	0.533	0.880
SPiT-B16	TokenCut	✗	0.903	0.773	0.934	0.771	0.639	0.894	0.711	0.564	0.868

[†] As reported by Wang et al. [47].

soft adjacency A_{TC} is computed using cosine similarities, which are thresholded using a small threshold $\tau_{\text{TC}} = 1/3$ to estimate adjacency over the complete graph over token features. The normalized cut is performed by extracting the Fiedler vector; the second smallest eigenvector of the graph Laplacian, and gives a bipartition of the graph into foreground and background elements. The original paper [47] uses DINO [8] as a pre-trained base model.

We found that extracting the key tokens from the last self-attention operator in the network is less effective than simply using the final features for the SPiT framework. In TokenCut, the saliency map is refined using postprocessing with a bilateral solver, however, in the SPiT framework this step is clearly redundant. Instead, we simply standardize the Fiedler vector using its mean and standard deviation, and map the result on the segmentations from the SPiT tokenizer. For certain images, the foreground and background elements could be swapped under the standard unsupervised normalized cut method. From our experiments on interpretability, we found that simply taking the class token for the full image, and comparing it using cosine similarity to class tokens (produced given the saliency mask) will accurately provide a robust estimate of which element is the foreground and the background.

F Feature Correspondences

The work by Caron et al. [8] and Oquab et al. [28] established certain emergent properties in self-supervised models, where the tokenized features of ViT trained with self-supervised methods provide inherent interpretability and inter-image feature correspondence. Given our results on feature attributions from Section 3.2, we perform experiments to visualize feature correspondences to see if similar emergent properties can be observed from supervised training with superpixel tokenization.

Method: A sequence of support images $(\xi_n)_{n=1}^N$ are selected, with labels such that $y_n \neq y_m$ for all $1 \leq m, n \leq N$, as a set of features to search from. Furthermore, these images are selected such that the WordNet [15] hypernym of the labels are the same, *e.g.*, 'dog' in Fig. F.1. We compute the normalized superpixel token features $\big(z_n : z_n = \Phi(\xi_n)/\|\Phi(\xi_n)\|\big)_{n=1}^{N}$ where Φ is our SPiT-B16 model with gradient including feature extraction. We omit the class tokens, and compute correspondences via cross-attention for each pair $A^{\times}_{mn} = \sigma(z_m z_n^{\intercal})$ where σ is the softmax with a temperature of 0.01. For visualizing the high-dimensional features, we compute a three component PCA to produce pseudocolors $c_n = \mathrm{PC}_3(z_n) \in \mathbb{R}^{3\times k}$ for $z_n \in \mathbb{R}^{k\times d}$, where $\mathrm{PC}_3(\cdot)$ extracts the first 3 principal components. These are normalized to $[0, 1]$, and mapped to RGB channels directly in the respective order.[5] The idea is to visualize the correspondences using PCA pseudocolors from the source image mapped to a target image. The principal components are thresholded using normalized cut, *cf*. Section E, and the feature correspondences are computed via

$$c_{m\to n} = (A^{\times}_{mn})^{\intercal} c_m, \tag{16}$$

such that $c_{m\to n}$ are the projected feature correspondences from the cross attention $A^{\times}_{mn}$ over the full feature space using pseudocolors from the source image m into the target image n. In other words, for each superpixel in the target image, we mix the pseudocolors of the corresponding superpixels in the source image and visualize them as the transferred pseudocolors. Given that these correspondences are directed due to the softmax operator, we compute the correspondences for every support image to illustrate the effect of using different source image mappings.

In Fig. F.1, we see that these feature correspondences pick up on the nuances of the different breeds of dogs, and are able to map similar parts between images, even with multiple instances of dogs in the same image. In Fig. F.2, we extend the experiment to a broader class of mammals with similar, albeit slightly less clear correspondence. In particular, the second row of Fig. F.2 illustrates a case where the feature correspondences exhibit less structure.

Notably, our model has not been trained with contrastive self-supervised approaches, and the features are derived from a model trained only supervisedly on IN1k. Moreover, the class tokens are removed before computing the cross attention and PCA, which confirms that the tokenized features themselves are informative for discriminative tasks.

G Extended Discussion on Classification

Certain interesting observations can be made from our results in Table 1. Firstly, random Voronoi tessellations perform better than data-driven superpixels for gradient excluding features, and despite its inherent stochasticity, tokenization with random Voronoi tessellations proves to be a relatively effective strategy, and demonstrate surprisingly consistent results over prediction tasks as reported in Table G.1. To account for the stochasticity in validation, we compute accuracy

[5] We use `torch.pca_lowrank`, which has nondeterministic behaviour. Slight deviations in pseudocolors could therefore occur when reproducing the visualizations.

Table G.1: Results w. CI (95%) for models with RViT tokenizers (5 runs).

ViT Model				IN1k	INReaL	Cifar100	Caltech256
Name	Tok.	Feat.	Grad.	Lin.	Lin.	Lin.	Lin.
RViT-S16	RV	Intp.	✗	0.7669 ± 0.0002	0.8285 ± 0.0003	0.8557 ± 0.0028	0.8521 ± 0.0007
RViT-S16	RV	Intp.	✓	0.7593 ± 0.0003	0.8183 ± 0.0002	0.8563 ± 0.0032	0.8558 ± 0.0006
RViT-B16	RV	Intp.	✗	0.7878 ± 0.0002	0.8436 ± 0.0002	0.8941 ± 0.0043	0.8731 ± 0.0007
RViT-B16	RV	Intp.	✓	0.7892 ± 0.0002	0.8414 ± 0.0001	0.8875 ± 0.0030	0.8644 ± 0.0006

scores over five runs and report 95% confidence intervals in Table G.1. We find that the segmentations based on the Voronoi tessellations produces remarkably consistent results over the validation set.

Additionally we note that gradient including tokenizers perform comparatively worse for small (S) models. This is particularly noteworthy, since the gradient features are essentially an added set of features to the model. We speculate that this could be an artifact of over-fitting on information-dense features, at the expense of the utility of the canonical pixel features.

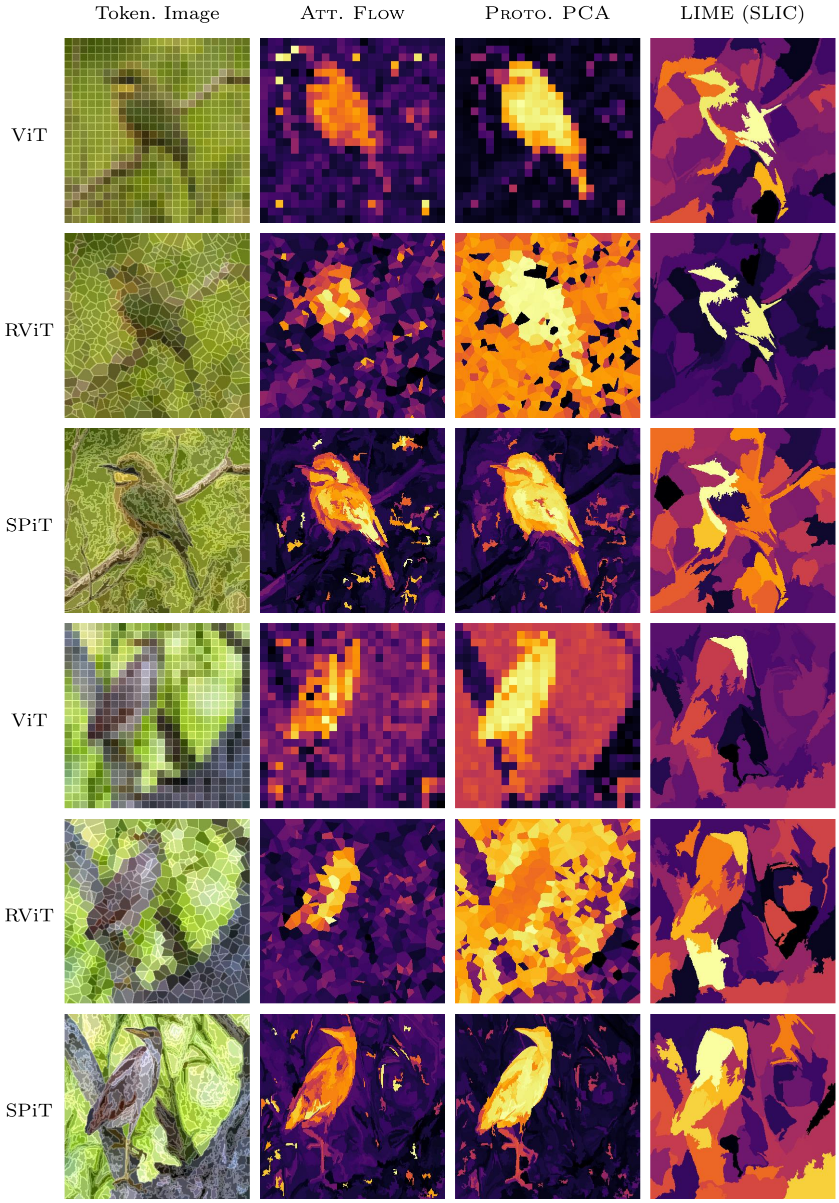

Fig. D.1: Visualization of feature attributions for prediction "*bee eater*" and "*bittern*" with different tokenization strategies: square partitions (ViT), random Voronoi tessellation (RViT) and superpixels (SPiT).

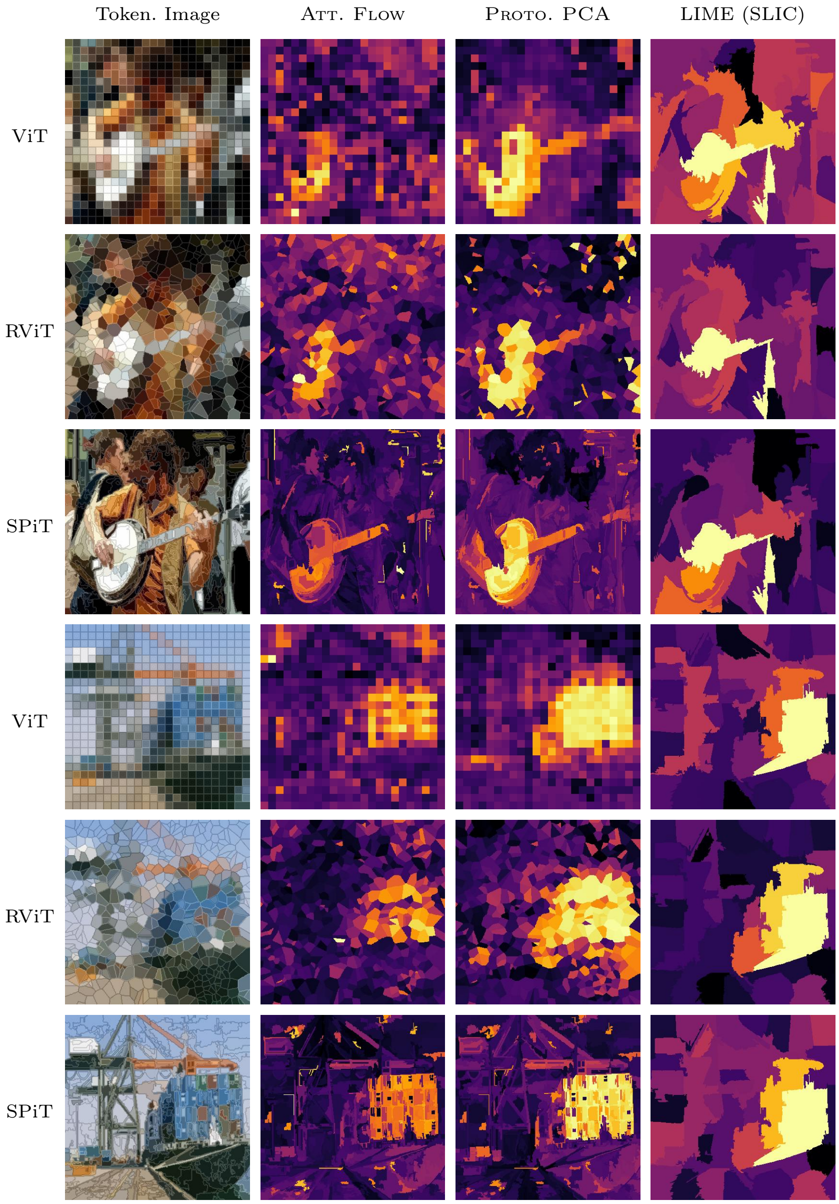

Fig. D.2: Visualization of feature attributions for prediction "*banjo*" and "*container ship*" with different tokenization strategies: square partitions (ViT), random Voronoi tesselation (RViT) and superpixels (SPiT).

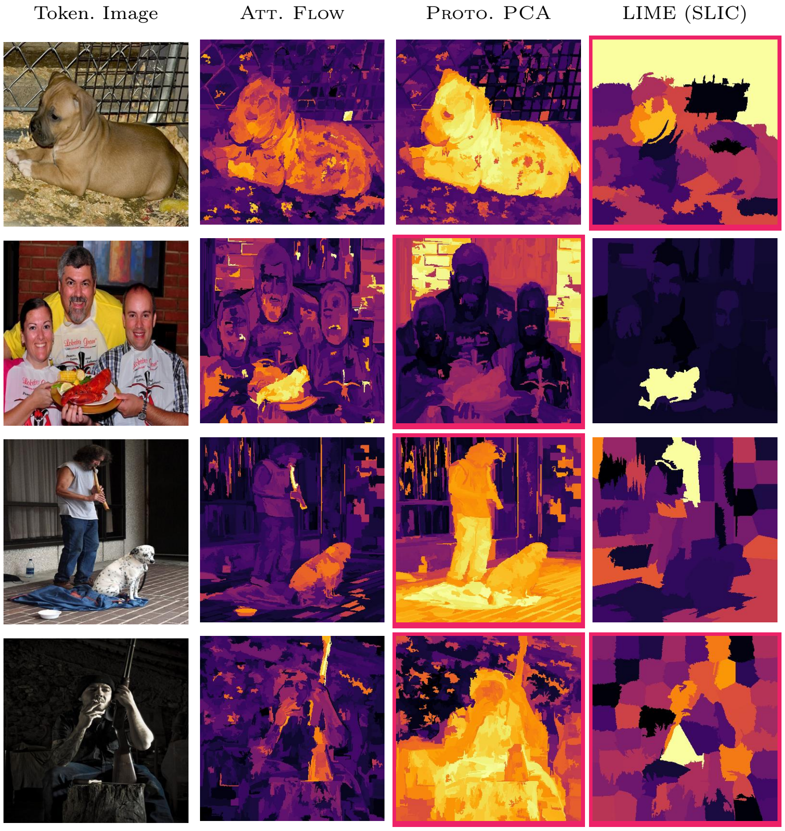

Fig. D.3: Examples of edge cases for attribution maps with SPiT. Row 1 demonstrates a case where LIME fails to provide coherent attributions for the prediction "*Staffordshire terrier*", and row 2–3 shows cases where the PCA prototype attributions fail for predictions "*lobster*" and "*flute*", respectively. Row 4 shows a case where attributions for both LIME and PCA prototypes are inadequate for the predicted label "*rifle*".

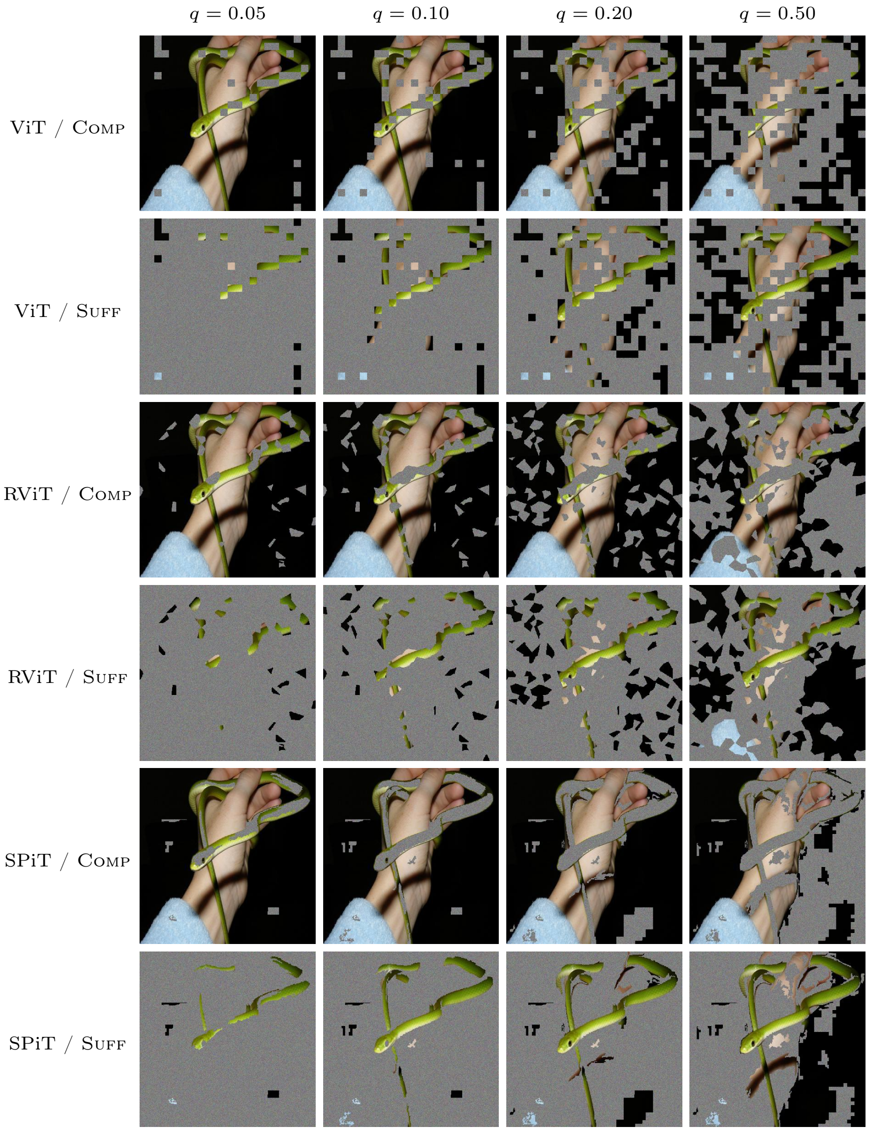

Fig. D.4: Visualization of attention flow occlusions at different quantiles q for prediction "*grass snake*". Note how the scaling of attention maps under superpixel tokenization improves occlusion for the predicted class.

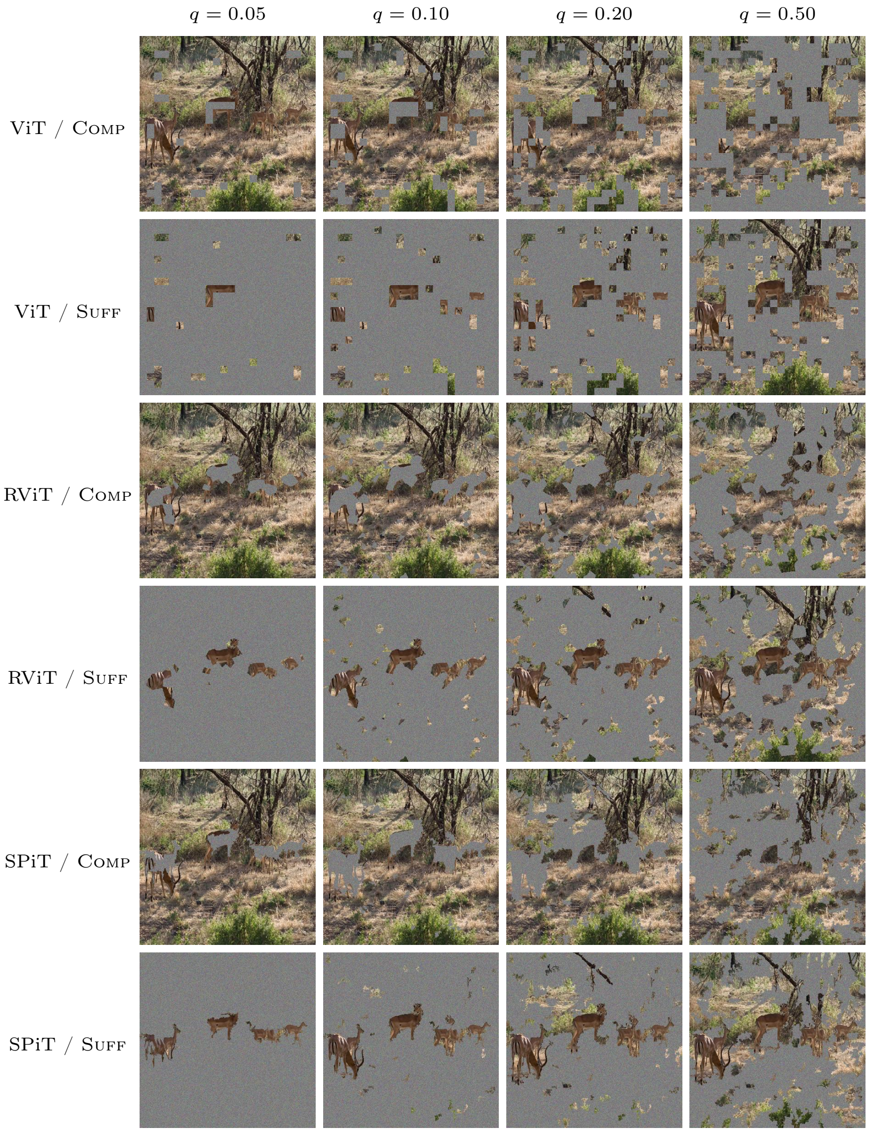

Fig. D.5: Visualization of attention flow occlusions at different quantiles q for prediction "*impala*". Note how the scaling of attention maps under superpixel tokenization improves occlusion for the predicted class.

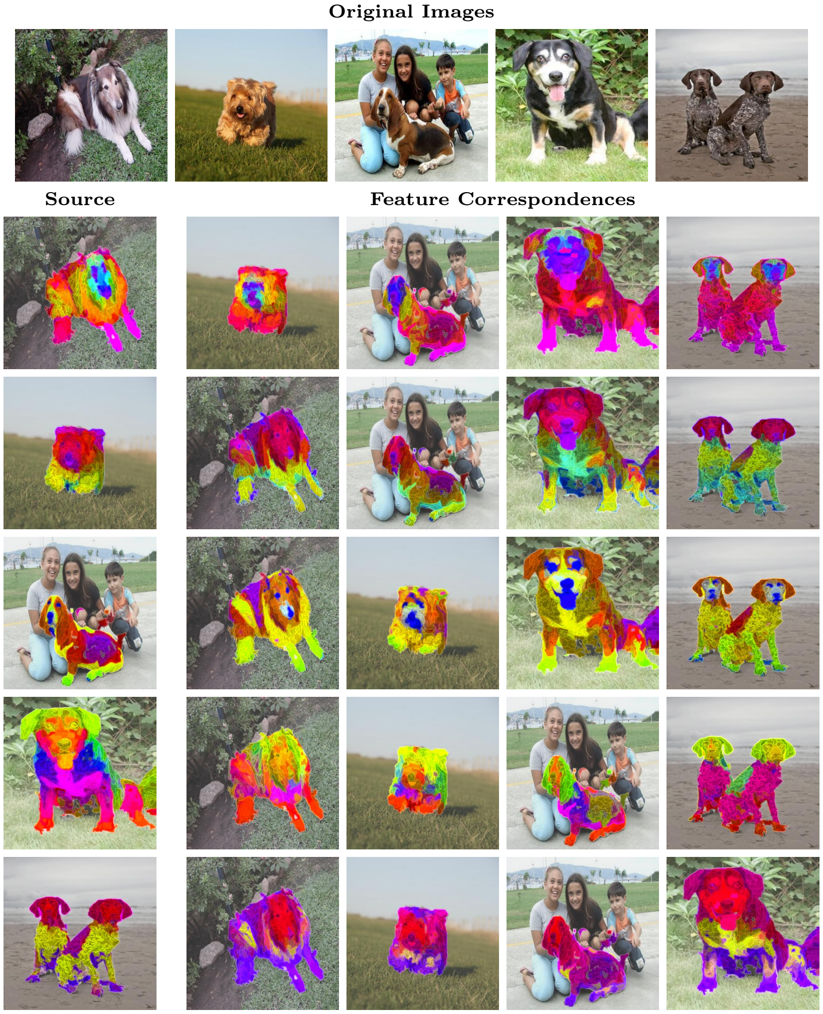

Fig. F.1: Visualization of feature correspondences from source features from superpixel tokens (left) to target images (right). Features are mapped via single head normalized cross attention between tokenized images, using pseudocolors from low rank PCA with three components. Images contain different classes (breeds) under the common hypernym "*domestic dog, canis familiaris*".

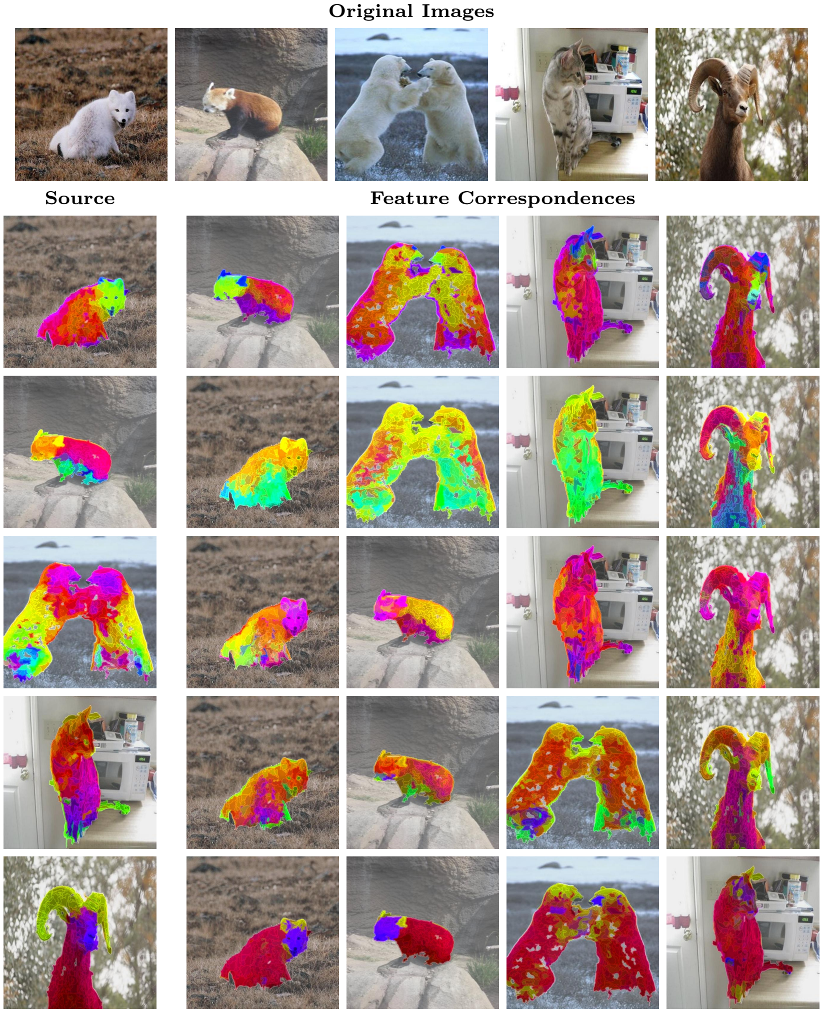

Fig. F.2: Visualization of feature correspondences from source features from superpixel tokens (left) to target images (right). Images contain different classes (species) under the common hypernym "*mammal*". The second row (red panda) illustrates a case where the visualized feature mappings exhibit less structure than in the other examples.