ADAPTIVE GEOMETRIC PERCEPTION KERNELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Modern vision architectures typically construct visual representations using predefined spatial structures, such as fixed patch grids or convolutional receptive fields. We introduce **Adaptive Geometric Perception Kernels (AdaGPK)**, a vision architecture that learns a compact set of content-dependent anisotropic Gaussian kernels. Each kernel adapts its location, scale, and orientation to the input image and directly aggregates a feature map into a visual token, allowing them to be processed by a standard Transformer architecture. Across image-classification benchmarks, AdaGPK consistently outperforms fixed-patch Vision Transformer baselines and is competitive with or exceeds the evaluated convolutional baselines on ImageNet-scale classification. Compared with unconstrained adaptive pooling, Gaussian parameterization preserves classification performance while producing more spatially structured representations and substantially stronger cross-dataset zero-shot localization. Analysis of the learned representation shows that increasing the kernel budget produces progressively smaller, less overlapping spatial supports and more differentiated token features, while the learned geometry remains stable across augmentations and training seeds. Together, these results suggest that learned spatial decomposition provides an effective way to construct compact visual representations with explicit spatial support.

1 INTRODUCTION

Modern vision architectures typically construct representations using predefined spatial structures. Vision Transformers (ViTs) represent images as sequences of fixed-size patches, while convolutional neural networks (CNNs) aggregate information through receptive fields whose spatial organization is determined by the architecture. However, neither explicitly adapts the initial spatial decomposition to image content.

Natural images contain structures that vary substantially in position, scale, orientation, and spatial extent. A fixed patch grid therefore provides the same spatial partition for every input, regardless of image content. Representations must subsequently combine these predefined regions into features that capture the relevant spatial structure of the image.

Several lines of work have explored more adaptive spatial representations. Multi-scale architectures capture information across multiple resolutions Liu et al. (2021); Wang et al. (2021); Fan et al. (2021), object-centric methods learn latent representations associated with image regions Locatello et al. (2020); Burgess et al. (2019); Engelcke et al. (2019), and adaptive tokenization or sampling methods dynamically select or aggregate visual information Ryoo et al. (2021); Rao et al. (2021); Liang et al. (2022). Related approaches also incorporate spatial relationships through relative positional encodings Shaw et al. (2018); Bao et al. (2021) and deformable attention Zhu et al. (2020); Xia et al. (2022). However, visual tokens are often defined either by a fixed spatial partition or as abstract feature vectors whose spatial support is not explicitly parameterized.

We introduce **Adaptive Geometric Perception Kernels (AdaGPK)**, a vision architecture that replaces fixed patch tokenization with a compact set of learned, content-dependent Gaussian kernels. AdaGPK predicts anisotropic Gaussian kernels as a function of the input image. Each kernel acts as a continuous spatial weighting function that softly aggregates visual information into a token while preserving the spatial support from each associated kernel. Tokens are therefore determined and supported by the geometry of the spatial regions from which they are constructed, allowing them to be processed by a standard Transformer architecture.

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.

We evaluate AdaGPK across standard image classification benchmarks and compare it with convolutional and Transformer baselines. AdaGPK consistently outperforms the ViT baselines and is competitive with or exceeds the performance of convolutional baselines on ImageNet-scale classification. An architecture-matched ablation further shows that while adaptive pooling accounts for the majority of the performance gains, constrained Gaussian kernels allow AdaGPK to achieve substantially stronger zero-shot cross-dataset object localization. Analysis of the learned representation shows that increasing the number of kernels produces progressively smaller, less overlapping spatial supports while the learned geometry remains stable across augmentations and training seeds. Together, these results suggest that learned spatial decomposition provides an effective way to construct compact visual representations with explicit spatial support.

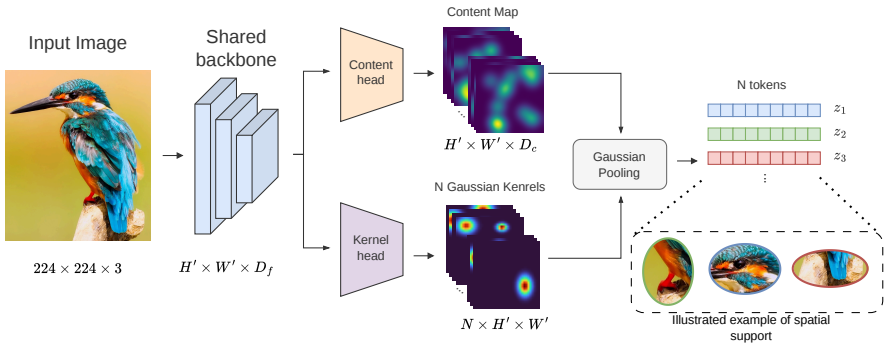

Figure 1: **Overview of AdaGPK.** An input image is mapped to a dense feature representation by a shared convolutional backbone. A kernel prediction head estimates N spatial heatmaps whose moments define anisotropic Gaussian kernels. In parallel, a content head produces the feature representation from which visual information is aggregated. Each Gaussian kernel acts as a continuous spatial weighting function over this content map, producing a visual token.

Our main contributions are:

- We introduce AdaGPK, a vision architecture that constructs tokens using content-dependent anisotropic Gaussian kernels with explicitly parameterized spatial support.
- We show that AdaGPK consistently outperforms fixed-patch ViT baselines and matches or exceeds the performance of convolutional baselines on ImageNet-scale classification.
- We show that AdaGPK achieves substantially stronger zero-shot cross-dataset object localization than unconstrained adaptive maps, while maintaining competitive classification performance.

2 Related Work

Convolutional neural networks CNNs LeCun et al. (1998) construct visual representations by applying local convolutional operations over regular spatial grids. Architectures such as AlexNet Krizhevsky et al. (2012), VGG Simonyan & Zisserman (2014), ResNet He et al. (2016), and ConvNeXt Liu et al. (2022) demonstrate the effectiveness of hierarchical feature extraction through local receptive fields. CNN receptive fields are largely determined by the architecture rather than the input. AdaGPK retains convolutional feature extraction but introduces learned Gaussian kernels that adaptively define the spatial support of each visual token.

Vision Transformers Vision Transformers (ViTs) Dosovitskiy et al. (2020) represent images as sequences of fixed-size patch tokens and model their interactions using self-attention. Subsequent architectures introduce hierarchical processing, local windows, and multi-scale representations Liu et al. (2021); Wang et al. (2021); Fan et al. (2021), substantially improving both performance and computational efficiency. Nevertheless, these approaches generally retain a predefined spatial decomposition at the tokenization stage, with token support determined by the patch or window struc-

ture. AdaGPK instead learns the spatial support of its tokens from image features, replacing fixed patch tokenization with content-dependent Gaussian receptive fields.

Adaptive Tokenization and Token Selection Several approaches modify the fixed-token representation used by standard Vision Transformers. TokenLearner Ryoo et al. (2021) constructs a compact set of tokens by predicting spatial weighting maps over dense feature representations and using them for weighted spatial aggregation. Other methods, including DynamicViT Rao et al. (2021) and EViT Liang et al. (2022), improve efficiency by dynamically selecting or reorganizing tokens after initial patch tokenization. AdaGPK differs in that each learned token is associated with an explicitly parameterized spatial support. The same kernel is used directly as the spatial weighting function for aggregating the token's visual content.

Object-Centric Representations Object-centric representation learning aims to decompose visual scenes into a set of latent components associated with distinct regions or entities. Methods such as MONet Burgess et al. (2019), GENESIS Engelcke et al. (2019), and Slot Attention Locatello et al. (2020) learn sets of latent variables that can capture different components of a scene. These methods are primarily designed to discover object-level or compositional representations, often through reconstruction or generative objectives. AdaGPK does not assume that individual kernels correspond to objects or semantic parts, it learns task-dependent spatial regions for discriminative representation learning.

Geometric and Spatially-Aware Attention Spatial information has also been incorporated directly into Transformer interactions. Relative positional representations Shaw et al. (2018), hierarchical positional mechanisms Liu et al. (2021), and deformable attention Zhu et al. (2020); Xia et al. (2022) introduce spatial relationships or learned sampling locations into the attention mechanism. Detection architectures such as DETR Carion et al. (2020) use learned queries to aggregate information from dense image features through cross-attention. These methods primarily modify how existing features or tokens interact. In contrast, AdaGPK introduces geometry at the token construction stage: the spatial region associated with each token is itself learned and explicitly parameterized before the tokens are passed to a standard Transformer.

3 METHOD

We introduce **Adaptive Geometric Perception Kernels (AdaGPK)**, a vision architecture that constructs visual tokens as a function of an input image. AdaGPK predicts a compact set of anisotropic Gaussian kernels whose spatial support adapts to the input. Each kernel acts as a continuous weighting function over a feature representation and determines where and over what extent visual information is aggregated. The pooled visual features form the token representations.

3.1 ARCHITECTURE OVERVIEW

Given an input image, AdaGPK first computes a dense feature representation using a convolutional backbone. From this representation, two lightweight heads process the shared features in parallel. *Kernel prediction head* produces N spatial heatmaps whose moments define the parameters of N anisotropic Gaussian kernels; where each heatmap defines exactly one Gaussian kernel. *Content head* produces the feature representation from which visual information is aggregated. Each Gaussian kernel is then used as a continuous spatial weighting function over the content feature representation, producing one pooled token vector per kernel. The resulting N tokens are passed directly to a conventional Transformer encoder.

3.2 GEOMETRIC KERNELS

Given an input image $X \in \mathbb{R}^{H\times W\times C}$,, where H, W and C are the height, width, and number of channels of the input image. Shared backbone produces a dense feature representation $F \in \mathbb{R}^{H'\times W'\times D_f}$, where H', W' and D_f are the height, width, and number of channels of the feature representation. Kernel prediction head maps F to N spatial heatmaps $M_i \in \mathbb{R}^{H'\times W'}, i = 1, \ldots, N$. Each heatmap is normalized independently using a temperature-scaled spatial softmax,

$$P_i(p) = \frac{\exp(M_i(p)/\tau)}{\sum_q \exp(M_i(q)/\tau)}, \tag{1}$$

where p and q denote spatial locations on the $H' \times W'$ feature grid and τ controls the concentration of the distribution. We use normalized spatial coordinates $p = (x_p, y_p)$ and estimate the geometry of each kernel from the first- and second-order moments of P_i. The kernel center is given by

$$\mu_i = (\mu_{x,i}, \mu_{y,i}), \qquad \mu_{x,i} = \sum_p P_i(p) x_p, \qquad \mu_{y,i} = \sum_p P_i(p) y_p. \tag{2}$$

The second-order moments determine its spatial extent and covariance,

$$\begin{aligned} \sigma_{x,i}^2 = \sum_p P_i(p)(x_p - \mu_{x,i})^2, \qquad \sigma_{y,i}^2 = \sum_p P_i(p)(y_p - \mu_{y,i})^2, \\ \mathrm{cov}_{xy,i} = \sum_p P_i(p)(x_p - \mu_{x,i})(y_p - \mu_{y,i}). \end{aligned} \tag{3}$$

The correlation between the two spatial axes is $\rho_i = \frac{\mathrm{cov}_{xy,i}}{\sigma_{x,i}\sigma_{y,i}}$, which gives the covariance matrix

$$\Sigma_i = \begin{bmatrix} \sigma_{x,i}^2 & \rho_i \sigma_{x,i} \sigma_{y,i} \\ \rho_i \sigma_{x,i} \sigma_{y,i} & \sigma_{y,i}^2 \end{bmatrix}. \tag{4}$$

The resulting parameters define an anisotropic Gaussian kernel

$$G_i(p) = \exp\left(-\frac{1}{2} (p - \mu_i)^\top \Sigma_i^{-1} (p - \mu_i) \right). \tag{5}$$

Each reconstructed kernel is normalized over the spatial domain,

$$\widetilde{G}_i(p) = \frac{G_i(p)}{\sum_q G_i(q)}, \tag{6}$$

so that it acts as a continuous spatial weighting function for token construction. The kernel parameters are predicted independently for each input; its location, scale, and orientation can vary continuously with image content.

3.3 Token Construction

In parallel with kernel prediction, the content head transforms the shared representation into a content feature representation $C \in \mathbb{R}^{H' \times W' \times D_c}$, where D_c is the dimensionality of the content features and, consequently, of each visual token. For each kernel, visual information is aggregated by Gaussian-weighted spatial pooling,

$$t_i = \sum_p \widetilde{G}_i(p)\, C(p), \tag{7}$$

where $t_i \in \mathbb{R}^{D_c}$ is the visual token associated with kernel i. The spatial support used to construct t_i is therefore determined directly by the learned kernel geometry.

3.4 Downstream Transformer

The resulting token set $T = [t_1, \ldots, t_N]$ is passed to a standard Transformer encoder. For image classification, the output token representations are aggregated into a single image representation and mapped to class logits using a linear classification head.

4 EXPERIMENTS

We evaluate AdaGPK on standard image classification benchmarks. We study the method across different model capacities and token budgets, compare its parameter and computational efficiency with convolutional and patch-based baselines, and analyze how the learned kernels change as the number of kernels changes. We evaluate AdaGPK against the unconstrained ablation and assess how the learned representations transfer across datasets for zero-shot object localization.

4.1 EXPERIMENTAL SETUP

Datasets We evaluate on CIFAR-10, CIFAR-100 (Krizhevsky et al., 2009), ImageNet-100, and ImageNet-1k (Krizhevsky et al., 2012). CIFAR-10 and CIFAR-100 contain 60k 32×32 RGB images, with 50k training and 10k test images and 10 and 100 classes, respectively. ImageNet-1k contains approximately 1.28 million training images and 50k validation images across 1,000 classes. ImageNet-100 is a 100-class subset of ImageNet-1k with approximately 126k training images and 5k validation images.

CUB-200 (Wah et al., 2011) and Pascal VOC 2012 (Everingham et al., 2015) are used to test whether spatial grounding learned from ImageNet transfers to target datasets with different visual statistics and label spaces, without any localization supervision. We restrict Pascal VOC to single-object images to avoid ambiguity from multiple targets.

Model Variants We evaluate multiple AdaGPK model scales to characterize the relationship between representation capacity, parameter count, and performance. We use one shared-backbone block for 32×32 inputs and three blocks for 224×224 inputs, as detailed in Appendix A.1. The kernel prediction branch, content branch, and token construction follow the same design across model variants. The variants differ primarily in token dimensionality and downstream Transformer depth. We denote them as TINY, SMALL, DEEP, and WIDE, summarized in Appendix Table 2.

Baselines We compare AdaGPK against convolutional and patch-based architectures. For convolutional baselines, we evaluate ResNet-18, ResNet-34, and ResNet-50. For patch-based architectures, we evaluate ViT-Tiny and ViT-Small using standard fixed-size patch embeddings. These models provide parameter matched reference points across a broad range of parameter counts and computational costs. All models are trained and evaluated on the same dataset splits and use the same training and evaluation protocols.

Training All models are trained from scratch with cross-entropy loss. We use AdamW with a learning rate of 3×10^{-4}, weight decay of 5×10^{-4}, and cosine learning-rate decay over 300 epochs. We apply label smoothing of 0.1 and clip the gradient norm to 1.0. We additionally apply MixUp with $\alpha = 0.2$ or CutMix with $\alpha = 1.0$ during training. Model selection is based on validation top-1 accuracy, and the corresponding checkpoint is used for final evaluation. For CIFAR-10, training images are augmented using random 32×32 crops with four-pixel padding and random horizontal flipping. For ImageNet-100 and ImageNet-1k, training images are processed using random resized crops to 224×224, random horizontal flipping, and RandAugment with two operations and magnitude 9. Additionally we reserve 10% of the original training set for validation and use the official validation split for final evaluation. All experiments are repeated over three random seeds.

4.2 COMPARISON WITH STANDARD VISION ARCHITECTURES

We compare AdaGPK with ViT and ResNet baselines across CIFAR-10, CIFAR-100, ImageNet-100, and ImageNet-1k. Figure 2 summarizes the trade-off between top-1 accuracy, parameter count, and computational cost (breakdown in Appendix A.2).

Parameter efficiency. AdaGPK consistently outperforms the fixed-patch ViT baselines across all benchmarks. ResNets achieve the highest accuracy on CIFAR, whereas AdaGPK becomes increasingly competitive on ImageNet-scale classification. The larger AdaGPK variants reach or exceed the accuracy of the evaluated ResNet models while remaining in a favorable parameter regime. Overall, AdaGPK provides a particularly strong accuracy-parameter trade-off, suggesting that learned spatial token construction makes effective use of downstream model capacity.

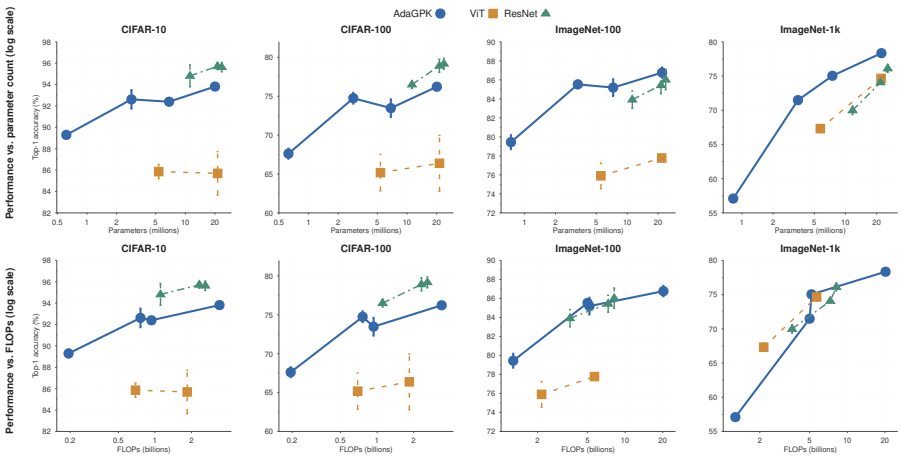

Figure 2: **Performance–efficiency trade-offs.** The top row compares top-1 accuracy against the number of trainable parameters, while the bottom row compares top-1 accuracy against single-image forward-pass FLOPs. Points show, for each model configuration and dataset, the mean top-1 accuracy over three runs, with error bars indicating two-sided 95% confidence intervals computed using Student's t-distribution. Connected points represent, from left to right, AdaGPK (32 kernels) TINY/SMALL/DEEP/WIDE, ViT-Tiny/ViT-Small, and ResNet-18/34/50.

Computational efficiency. The FLOP comparison shows that parameter matched AdaGPK requires more computation than the evaluated ViT baselines, although this is accompanied by substantially higher classification accuracy. Relative to ResNet, AdaGPK attains comparable or higher accuracy for several ImageNet configurations at similar computational scales. The primary efficiency advantage of AdaGPK lies in parameter efficiency and compact downstream token representations rather than uniformly minimizing FLOPs.

As an additional control, we compare against TokenLearner under the same training protocol. When trained from scratch, TokenLearner matches or underperforms the corresponding ViT baseline, while AdaGPK remains substantially more accurate, indicating that the observed gains are not a generic consequence of adaptive tokenization alone (Appendix A.3).

4.3 GAUSSIAN TOKENIZATION ABLATION

To isolate the effect of explicit Gaussian structure from adaptive spatial pooling itself, Table 1 compares AdaGPK with an architecture-matched unconstrained-map tokenizer. The ablation model retains the shared feature extractor, content head, map predictor, moment-derived geometry fusion, and downstream Transformer, but directly uses the predicted normalized heatmaps for spatial pooling instead of reconstructing anisotropic Gaussians. Both tokenizers use 32 content-dependent spatial weighting maps.

While the Gaussian constraint does not consistently improve classification accuracy over unconstrained adaptive maps, it imposes an explicit geometric structure on the learned spatial supports. We next test whether this structure provides benefits beyond classification by evaluating zero-shot object localization.

4.4 ZERO-SHOT OBJECT LOCALIZATION

We evaluate frozen ImageNet-100 checkpoints on CUB-200 and Pascal VOC 2012 without any localization supervision. For each model we select the largest evaluated configurations with approximately 20 million parameters. For each token, we measure the Jensen-Shannon divergence between the original predictive distribution and the distribution obtained after perturbing that token. For AdaGPK and the unconstrained ablation, these token-level importance scores weight the corresponding Gaussian kernels or learned spatial maps, respectively, and are summed to obtain a spatial

Table 1: Spatial-map parameterization ablation across model configurations. Both tokenizers use 32 content-dependent spatial weighting maps, which are either free-form or constrained to anisotropic Gaussians. Results are mean ± standard deviation over three runs.

Model	Tokenization	CIFAR-10	CIFAR-100	ImageNet-100
TINY	Unconstrained maps	88.75 ± 0.22	$\mathbf{68.14 \pm 0.21}$	$\mathbf{79.91 \pm 0.35}$
	AdaGPK	$\mathbf{89.30 \pm 0.12}$	67.62 ± 0.21	79.44 ± 0.24
SMALL	Unconstrained maps	92.53 ± 0.28	$\mathbf{75.17 \pm 0.39}$	$\mathbf{85.59 \pm 0.16}$
	AdaGPK	$\mathbf{92.62 \pm 0.29}$	74.75 ± 0.23	85.52 ± 0.13
DEEP	Unconstrained maps	$\mathbf{92.79 \pm 0.02}$	$\mathbf{74.32 \pm 0.45}$	$\mathbf{85.56 \pm 0.18}$
	AdaGPK	92.40 ± 0.05	73.47 ± 0.38	85.19 ± 0.29
WIDE	Unconstrained maps	$\mathbf{93.98 \pm 0.22}$	76.08 ± 0.51	$\mathbf{88.09 \pm 0.15}$
	AdaGPK	93.82 ± 0.07	$\mathbf{76.22 \pm 0.13}$	86.77 ± 0.18

importance map. For ViT, each patch importance score is assigned uniformly to its corresponding image patch. For ResNet, we analogously perturb one location of the final 7×7 feature map at a time by replacing its feature vector with the mean feature vector of that image, and assign the resulting Jensen-Shannon distribution change to that spatial location.

Figure 3 evaluates whether the resulting spatial importance is concentrated on annotated foreground regions. For each spatial budget $a \in \{0.1, 0.2, 0.3, 0.4\}$, we select the most relevant a fraction of locations and compute foreground precision f_p. To correct for object size, we report $\frac{f_p - b}{\min(1, b/a) - b}$, where b is the ground-truth foreground area fraction.

We additionally report the Pointing Game, which counts an image as correctly localized when the maximum-relevance location falls inside a ground-truth box.

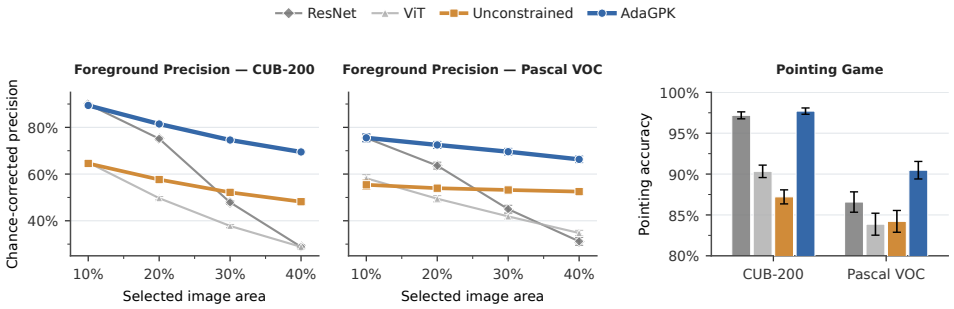

Figure 3: **Class-agnostic zero-shot object localization.** Foreground precision as a function of selected image area on CUB-200 and single-object Pascal VOC, together with Pointing Game accuracy. Error bars show 95% confidence intervals over images. Higher is better.

Despite similar classification performance, AdaGPK consistently yields higher foreground-selective relevance than the matched unconstrained tokenizer as well as ViT and Resnet baselines. This suggests that the Gaussian constraint does not primarily improve recognition accuracy, but instead imposes a more structured spatial organization on the learned representation. Qualitative examples in Appendix A.4 further illustrate this behavior, with AdaGPK producing smoother and more spatially coherent object-aligned importance maps than the unconstrained tokenizer. Across matched configurations, AdaGPK maintains stronger localization than the unconstrained tokenizer at comparable or lower classification accuracy (Appendix A.5), suggesting that the localization gain is not driven by recognition performance alone.

4.5 LEARNED GEOMETRIC KERNEL VISUALIZATIONS

To examine how AdaGPK uses its learned spatial representation, we visualize prediction-relevant kernels from the WIDE model with 32 kernels on representative ImageNet images (Figure 4). We

rank kernels using leave-one-token-out importance. For an input with predicted class $\hat{y}$, the importance of kernel i is defined as $\Delta z_i = z_{\hat{y}} - z_{\hat{y}}^{(-i)}$, where $z_{\hat{y}}$ is the predicted-class logit of the full model and $z_{\hat{y}}^{(-i)}$ is the corresponding logit when token i is removed. To avoid displaying multiple highly overlapping kernels, we retain the top $2K$ kernels by importance and greedily select K spatially diverse kernels from this set.

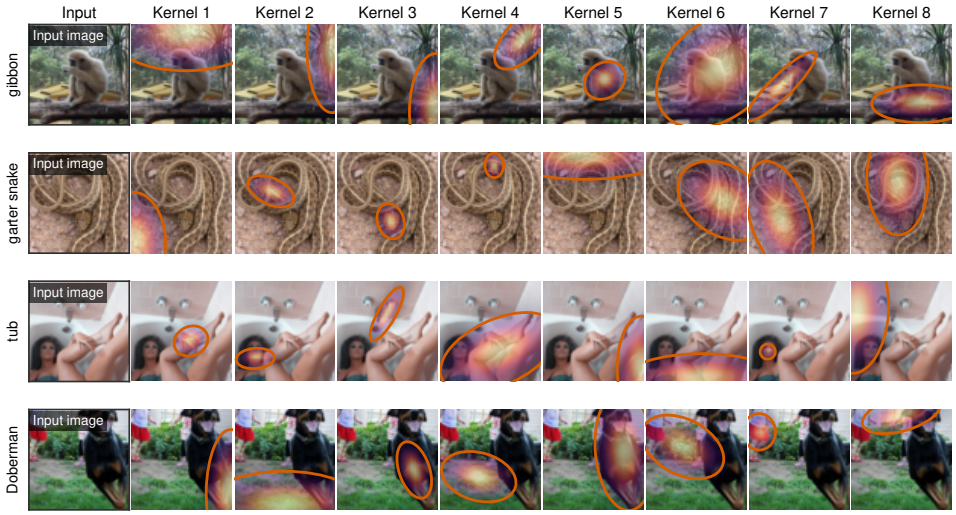

Figure 4: **Prediction-relevant learned Gaussian kernels.** Each row shows an input image followed by eight selected kernels. The contours indicate the spatial support of the corresponding Gaussians. Kernels are selected using leave-one-token-out importance followed by a diversity-aware spatial selection step.

The learned kernels adapt substantially across images in position, scale, and orientation. Within each image, prediction-relevant kernels cover localized parts of the dominant object and background. Smaller kernels tend to focus on finer structures, while larger kernels provide broader spatial support. These examples illustrate that AdaGPK does not impose a fixed decomposition of the image.

Additional analysis shows that prediction relevance is not determined solely by kernel size or spatial overlap, suggesting that the importance of a token depends on both its spatial support and the visual information it aggregates (Appendix A.11).

4.6 Token Scaling and Kernel Analysis

We next examine how the learned representation changes as the number of kernels changes on ImageNet-100. For each model variant, we train configurations with $N \in \{8, 16, 32, 64\}$ kernels over three random seeds.

Figure 5 reports three measures of spatial and representational diversity together with top-1 accuracy. Median kernel area is defined as the area of the one-standard-deviation Gaussian ellipse normalized by image area. Kernel overlap is the mean off-diagonal cosine similarity between flattened Gaussian kernel maps, and token similarity is the mean off-diagonal cosine similarity between the resulting token representations.

Increasing the kernel budget consistently changes the granularity of the learned representation. Across model variants, median kernel area decreases as N increases, while pairwise kernel overlap and token similarity also generally decline. Larger kernel budgets produce finer spatial decompositions and more differentiated token representations.

Performance generally improves from very small kernel budgets but shows diminishing returns beyond a moderate number of kernels. The Wide model improves from 8 to 16 kernels and changes only modestly thereafter, while the Small and Deep variants remain comparatively stable at larger

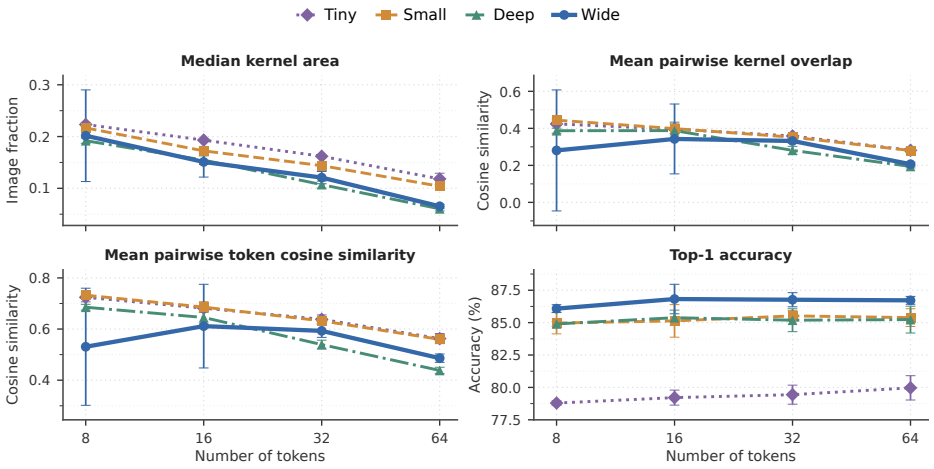

Figure 5: **Effect of kernel budget on learned spatial representations.** Results are reported on ImageNet-100 for the TINY, SMALL, DEEP, and WIDE variants with 8, 16, 32, or 64 kernels. The panels show median normalized kernel area, mean pairwise kernel overlap, mean pairwise token cosine similarity, and top-1 accuracy. Points indicate means over three runs, with two-sided 95% confidence intervals computed using Student's t-distribution.

kernel counts. The TINY model benefits more consistently from additional kernels, suggesting that kernel budget is more limiting when model capacity is small.

Taken together, these results show that increasing the number of kernels continues to refine the spatial representation even after classification performance begins to saturate. Additional geometric analysis shows that larger kernel budgets yield smaller and moderately more anisotropic kernels, with a preference for axis-aligned orientations (Appendix A.9).

5 CONCLUSION

We introduced Adaptive Geometric Perception Kernels (AdaGPK), a vision architecture that constructs visual tokens using learned, content-dependent anisotropic Gaussian kernels. Each kernel explicitly parameterizes the location, scale, and orientation of the spatial region used to aggregate visual information, producing compact tokens that can be processed by a standard Transformer.

Across image-classification benchmarks, AdaGPK consistently outperforms fixed-patch ViT baselines and is competitive with convolutional baselines on ImageNet-scale classification. Architecture-matched ablations show that adaptive spatial pooling accounts for much of this predictive performance, while Gaussian parameterization provides an explicit geometric structure without sacrificing competitive classification accuracy.

Importantly, this structure is useful beyond classification. Without localization supervision, AdaGPK produces substantially stronger zero-shot object localization than unconstrained adaptive pooling and baseline ResNet and ViT models. Further analysis shows that increasing the token budget yields progressively smaller, less overlapping spatial supports and more differentiated token representations, while the learned geometry remains stable across augmentations and training seeds.

Together, these results show that explicitly parameterizing content-dependent spatial support provides effective representations with meaningful and transferable spatial structure. This opens opportunities for applying geometric tokens to dense prediction, spatial reasoning, and adaptive visual computation.

REFERENCES

Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. Beit: Bert pre-training of image transformers. *arXiv preprint arXiv:2106.08254*, 2021.

Christopher P Burgess, Loic Matthey, Nicholas Watters, Rishabh Kabra, Irina Higgins, Matt Botvinick, and Alexander Lerchner. Monet: Unsupervised scene decomposition and representation. *arXiv preprint arXiv:1901.11390*, 2019.

Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pp. 213–229. Springer, 2020.

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.

Martin Engelcke, Adam R Kosiorek, Oiwi Parker Jones, and Ingmar Posner. Genesis: Generative scene inference and sampling with object-centric latent representations. *arXiv preprint arXiv:1907.13052*, 2019.

Mark Everingham, SM Ali Eslami, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes challenge: A retrospective. *International journal of computer vision*, 111(1):98–136, 2015.

Haoqi Fan, Bo Xiong, Karttikeya Mangalam, Yanghao Li, Zhicheng Yan, Jitendra Malik, and Christoph Feichtenhofer. Multiscale vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 6824–6835, 2021.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Youwei Liang, Chongjian Ge, Zhan Tong, Yibing Song, Jue Wang, and Pengtao Xie. Not all patches are what you need: Expediting vision transformers via token reorganizations. *arXiv preprint arXiv:2202.07800*, 2022.

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.

Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11976–11986, 2022.

Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-centric learning with slot attention. *Advances in neural information processing systems*, 33:11525–11538, 2020.

Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. Dynamicvit: Efficient vision transformers with dynamic token sparsification. *Advances in neural information processing systems*, 34:13937–13949, 2021.

Michael Ryoo, AJ Piergiovanni, Anurag Arnab, Mostafa Dehghani, and Anelia Angelova. Tokenlearner: Adaptive space-time tokenization for videos. *Advances in neural information processing systems*, 34:12786–12797, 2021.

Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. Self-attention with relative position representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pp. 464–468, 2018.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

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

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

Zhuofan Xia, Xuran Pan, Shiji Song, Li Erran Li, and Gao Huang. Vision transformer with deformable attention. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4794–4803, 2022.

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

A APPENDIX

A.1 ADAGPK IMPLEMENTATION DETAILS

This section provides the architectural details of the AdaGPK tokenizer used in our experiments.

Shared feature extractor Given an input image, AdaGPK first constructs a dense feature representation using a sequence of convolutional downsampling and residual blocks. Each downsampling block applies a stride-2 convolution, followed by group normalization and GELU activation, a depthwise convolution, and a pointwise projection, together with a residual skip projection. The shared backbone is followed by two lightweight residual convolutional blocks that further refine the spatial feature representation.

Kernel prediction head From the shared feature representation, a kernel prediction head produces N spatial heatmaps using group normalization followed by a 1×1 convolution with N output channels. The heatmaps are normalized independently over spatial locations using a temperature-scaled softmax. Their first- and second-order moments are then used to estimate the center, variance, and covariance of each anisotropic Gaussian kernel.

Gaussian reconstruction Each kernel is parameterized by its center (μ_x, μ_y), standard deviations (σ_x, σ_y), and correlation coefficient ρ. A minimum scale is added to the estimated variances for numerical stability, and ρ is constrained to the interval $[-0.95, 0.95]$. The resulting anisotropic Gaussian kernels are reconstructed analytically and normalized to sum to one over the spatial domain.

Content head In parallel to the kernel prediction head, a content head transforms the shared feature representation into the content feature representation used for spatial aggregation. The head consists of a 1×1 channel projection followed by group normalization, GELU activation, and an efficient residual block with depthwise spatial convolution. Each Gaussian kernel is then used as a soft spatial weighting function over this content feature representation, producing one pooled content vector per kernel.

Table 2: AdaGPK model configurations.

Variant	Token dim.	Transformer depth	Params (32×32)	Params (224×224)
TINY	96	2	0.62M	0.74M
SMALL	192	3	2.85M	3.30M
DEEP	192	12	6.85M	7.30M
WIDE	384	8	20.10M	21.89M

A.2 Full Classification Results

Table 3 reports the complete top-1 classification results for AdaGPK and the evaluated ViT and ResNet baselines. All values are reported as mean ± standard deviation over three independent runs. The best mean performance for each benchmark is highlighted in bold.

Table 3: Comparison with standard vision architectures across image classification benchmarks. Results are reported as mean ± standard deviation over three runs.

Model	CIFAR-10	CIFAR-100	ImageNet-100	ImageNet-1k
AdaGPK-Tiny	89.30 ± 0.12	67.62 ± 0.21	79.44 ± 0.24	57.09 ± 0.09
AdaGPK-Small	92.62 ± 0.29	74.75 ± 0.23	85.52 ± 0.13	71.48 ± 0.13
AdaGPK-Deep	92.40 ± 0.05	73.47 ± 0.38	85.19 ± 0.29	75.03 ± 0.11
AdaGPK-Wide	93.82 ± 0.07	76.22 ± 0.13	$\mathbf{86.77 \pm 0.18}$	$\mathbf{78.34 \pm 0.07}$
ViT-Tiny	85.86 ± 0.22	65.18 ± 0.77	75.90 ± 0.44	67.33 ± 0.19
ViT-Small	85.70 ± 0.67	66.37 ± 1.18	77.77 ± 0.10	74.66 ± 0.18
ResNet-18	94.82 ± 0.33	76.50 ± 0.16	83.91 ± 0.30	69.99 ± 0.20
ResNet-34	$\mathbf{95.72 \pm 0.10}$	78.92 ± 0.28	85.43 ± 0.29	74.08 ± 0.05
ResNet-50	95.63 ± 0.15	$\mathbf{79.18 \pm 0.23}$	86.02 ± 0.35	76.12 ± 0.19

A.3 Adaptive Tokenization

To test whether AdaGPK's classification gains can be explained by adaptive tokenization alone, we additionally compare against TokenLearner under the same training protocol used for the corresponding ViT baseline.

Table 4 reports top-1 accuracy and parameter count for the matched ViT, TokenLearner, and AdaGPK configurations. Both TokenLearner and AdaGPK use 32 adaptive tokens.

Table 4: Classification accuracy and efficiency of AdaGPK, TokenLearner, and ViT variants. Top-1 accuracy is mean $\pm$ standard deviation over three seeds. Trainable parameter count reports the largest evaluated instance across the three datasets. ViT patch sizes match TokenLearner.

Model	CIFAR-10 (%)	CIFAR-100 (%)	ImageNet-100 (%)	Parameters (M)
AdaGPK-Tiny	89.30 ± 0.12	67.62 ± 0.21	79.44 ± 0.24	0.74
AdaGPK-Small	92.62 ± 0.29	74.75 ± 0.23	85.52 ± 0.13	3.30
AdaGPK-Deep	92.40 ± 0.05	73.47 ± 0.38	85.19 ± 0.29	7.30
AdaGPK-Wide	93.82 ± 0.07	76.22 ± 0.13	86.77 ± 0.18	21.89
ViT-Tiny	85.86 ± 0.22	65.18 ± 0.77	75.90 ± 0.44	5.54
ViT-Small	85.70 ± 0.67	66.37 ± 1.18	77.77 ± 0.10	21.70
TokenLearner-Tiny	83.25 ± 0.49	66.01 ± 0.28	75.60 ± 0.21	5.56
TokenLearner-Small	84.82 ± 0.66	65.52 ± 0.71	68.46 ± 1.20	21.73

When trained from scratch under the matched setup, TokenLearner matches or underperforms the corresponding ViT baseline, whereas AdaGPK remains substantially more accurate. This suggests that the gains of AdaGPK are not a generic consequence of replacing fixed patches with adaptive tokenization, but arise from the specific spatial token construction used by the model.

A.4 Qualitative Zero-Shot Localization Results

Figures 6 and 7 show representative class-agnostic localization maps on CUB-200 and Pascal VOC 2012, respectively. All four models use the same input crop and the same Jensen-Shannon distribution-change importance described in Section 4.4. Each relevance map is normalized by its maximum for visualization.

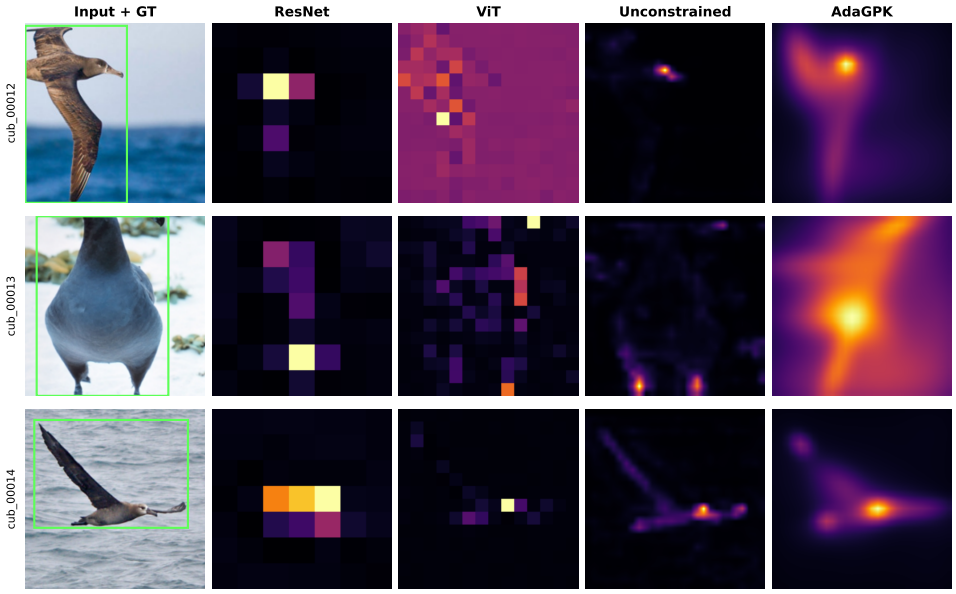

Figure 6: **Qualitative zero-shot localization on CUB-200.** The first column shows the evaluation crop and ground-truth bounding box. The remaining columns show the class-agnostic relevance maps produced by ResNet, ViT, the unconstrained adaptive tokenizer, and AdaGPK.

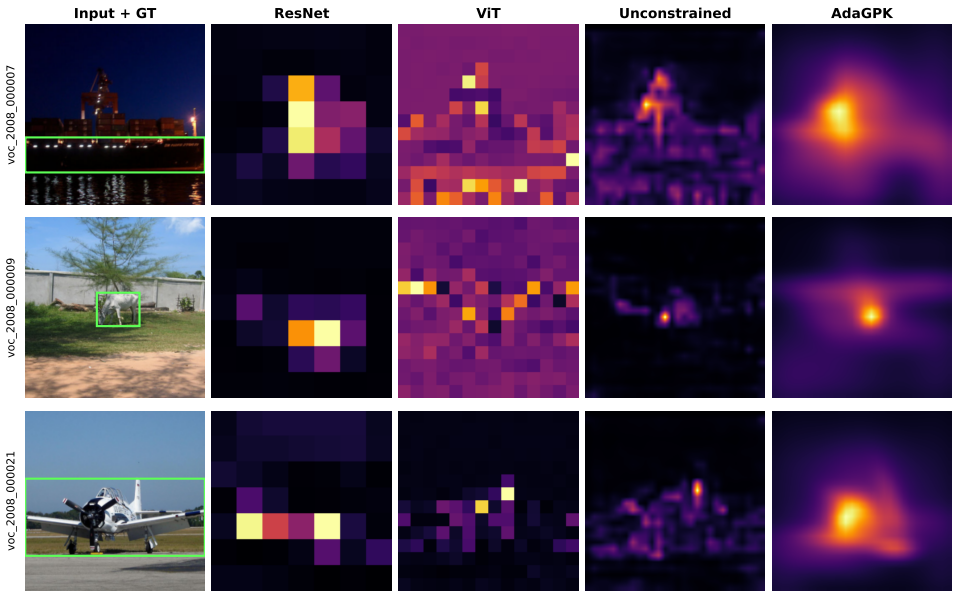

Figure 7: **Qualitative zero-shot localization on Pascal VOC 2012.** The first column shows the evaluation crop and ground-truth bounding box. The remaining columns show the class-agnostic relevance maps produced by ResNet, ViT, the unconstrained adaptive tokenizer, and AdaGPK.

A.5 IMAGENET-100 CLASSIFICATION AND LOCALIZATION

Figure 8 compares classification performance with localization quality across the TINY, SMALL, DEEP, and WIDE AdaGPK and unconstrained-map configurations. Each configuration uses 32 adaptive tokens and is evaluated over three independently trained seeds. This comparison shows that localization quality is not explained by classification accuracy alone: the unconstrained models can match or exceed AdaGPK classification accuracy while producing less foreground-concentrated relevance.

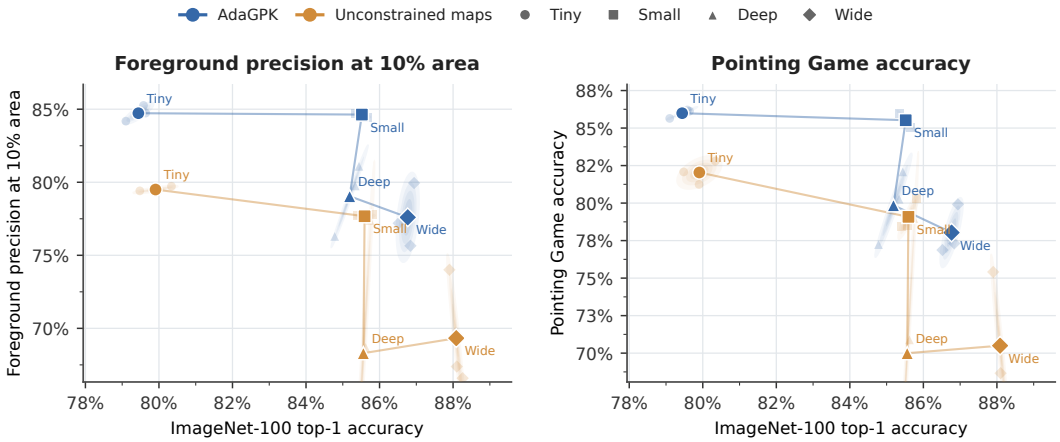

Figure 8: **ImageNet-100 classification performance versus localization quality.** Top-1 classification accuracy is compared with foreground precision at a 10% selected-area budget (left) and Pointing Game accuracy (right). Faint markers denote individual seeds, prominent markers denote three-seed configuration means, and layered bivariate Gaussian contours show joint uncertainty in classification and localization performance. Marker shapes identify the model configurations, and connected points trace increasing model capacity within each tokenizer family.

A.6 Robustness to Augmentations

To examine the effect of input perturbations on model predictions, we evaluate prediction robustness under small deterministic translations, rotations, and zoom perturbations. Translations move the image by 2% of its width or height in each cardinal direction, rotations use angles of $\pm 5^\circ$, and zoom perturbations rescale the image by $\pm 5\%$. Figure 9 compares the resulting Top-1 accuracy drop, agreement with the clean prediction, and Jensen-Shannon divergence from the clean predictive distribution on ImageNet-1k.

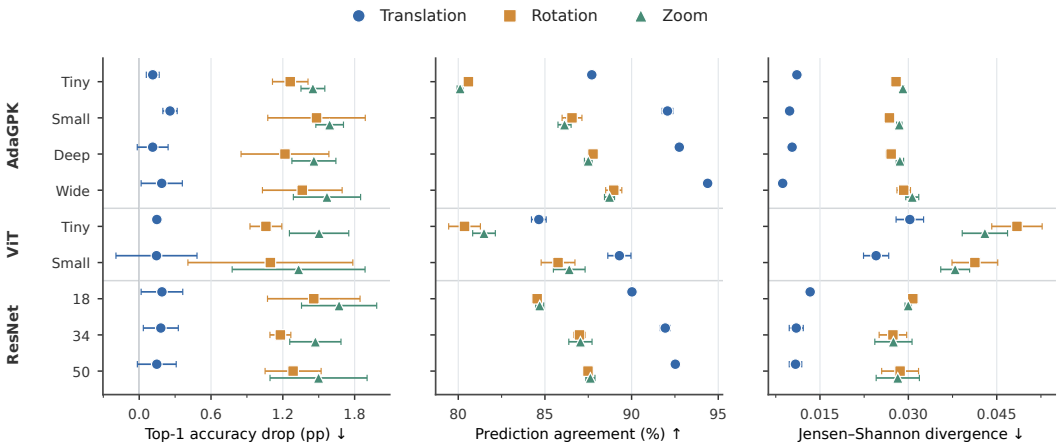

Figure 9: **Prediction robustness on ImageNet-1k.** Models are evaluated under small translations, rotations, and zoom perturbations. The panels report Top-1 accuracy drop, clean-prediction agreement, and Jensen-Shannon divergence; arrows indicate the preferred direction. Points show means over three runs, with error bars indicating two-sided 95% confidence intervals computed using Student's t-distribution.

AdaGPK exhibits strong prediction stability across the tested geometric perturbations. The larger AdaGPK variants achieve the highest prediction agreement across translation, rotation, and zoom, indicating that their predicted classes change less frequently under these transformations. AdaGPK also yields the lowest Jensen-Shannon divergence under translation and rotation, while remaining competitive with ResNet under zoom. Top-1 accuracy degradation is mixed; AdaGPK performs best under translation, whereas the evaluated ViT baselines show the smallest drops under rotation and zoom.

Together, these results indicate that AdaGPK maintains particularly stable predictions under modest geometric changes, especially when measured by prediction agreement and distributional shift.

Table 5 reports the complete ImageNet-1k robustness results underlying Figure 9. Results are grouped by perturbation type and include clean Top-1 accuracy, the Top-1 accuracy drop, agreement with the clean prediction, and Jensen-Shannon divergence from the clean predictive distribution.

Table 5: Prediction robustness on ImageNet-1k under small translations (2% of image size), rotations ($\pm 5^\circ$), and zoom perturbations ($\pm 5\%$). Results are mean $\pm$ standard deviation over three runs. Lower is better for Top-1 drop and Jensen–Shannon (JS) divergence; higher is better for agreement. The best robustness value in each column is bolded.

Translation

Model	Clean Top-1 (%)	Drop (pp)	Agreement (%)	JS divergence
AdaGPK-Tiny	57.10 ± 0.11	$\mathbf{0.11 \pm 0.02}$	87.70 ± 0.02	0.0111 ± 0.0001
AdaGPK-Small	71.48 ± 0.13	0.26 ± 0.02	92.07 ± 0.10	0.0098 ± 0.0002
AdaGPK-Deep	75.02 ± 0.12	$\mathbf{0.11 \pm 0.04}$	92.75 ± 0.06	0.0103 ± 0.0001
AdaGPK-Wide	78.35 ± 0.07	0.19 ± 0.06	$\mathbf{94.39 \pm 0.03}$	$\mathbf{0.0087 \pm 0.0001}$
ViT-Tiny	67.33 ± 0.19	0.15 ± 0.01	84.65 ± 0.14	0.0303 ± 0.0008
ViT-Small	74.65 ± 0.18	0.14 ± 0.11	89.30 ± 0.22	0.0245 ± 0.0007
ResNet-18	69.97 ± 0.18	0.19 ± 0.06	90.02 ± 0.05	0.0133 ± 0.0002
ResNet-34	74.07 ± 0.05	0.18 ± 0.05	91.94 ± 0.09	0.0110 ± 0.0004
ResNet-50	76.11 ± 0.18	0.15 ± 0.05	92.52 ± 0.06	0.0108 ± 0.0003

Rotation

Model	Clean Top-1 (%)	Drop (pp)	Agreement (%)	JS divergence
AdaGPK-Tiny	57.10 ± 0.11	1.26 ± 0.05	80.59 ± 0.07	0.0279 ± 0.0002
AdaGPK-Small	71.48 ± 0.13	1.48 ± 0.13	86.56 ± 0.19	$\mathbf{0.0268 \pm 0.0001}$
AdaGPK-Deep	75.02 ± 0.12	1.22 ± 0.12	87.78 ± 0.09	0.0271 ± 0.0003
AdaGPK-Wide	78.35 ± 0.07	1.36 ± 0.11	$\mathbf{88.97 \pm 0.15}$	0.0292 ± 0.0004
ViT-Tiny	67.33 ± 0.19	$\mathbf{1.06 \pm 0.04}$	80.36 ± 0.30	0.0484 ± 0.0014
ViT-Small	74.65 ± 0.18	1.10 ± 0.23	85.76 ± 0.32	0.0413 ± 0.0013
ResNet-18	69.97 ± 0.18	1.46 ± 0.13	84.55 ± 0.07	0.0308 ± 0.0002
ResNet-34	74.07 ± 0.05	1.18 ± 0.03	86.99 ± 0.11	0.0274 ± 0.0008
ResNet-50	76.11 ± 0.18	1.29 ± 0.08	87.48 ± 0.03	0.0286 ± 0.0010

Zoom

Model	Clean Top-1 (%)	Drop (pp)	Agreement (%)	JS divergence
AdaGPK-Tiny	57.10 ± 0.11	1.45 ± 0.03	80.11 ± 0.06	0.0291 ± 0.0001
AdaGPK-Small	71.48 ± 0.13	1.59 ± 0.04	86.13 ± 0.12	0.0285 ± 0.0001
AdaGPK-Deep	75.02 ± 0.12	1.46 ± 0.06	87.50 ± 0.07	0.0286 ± 0.0002
AdaGPK-Wide	78.35 ± 0.07	1.57 ± 0.09	$\mathbf{88.73 \pm 0.10}$	0.0307 ± 0.0004
ViT-Tiny	67.33 ± 0.19	1.50 ± 0.08	81.48 ± 0.22	0.0430 ± 0.0013
ViT-Small	74.65 ± 0.18	$\mathbf{1.33 \pm 0.18}$	86.40 ± 0.30	0.0379 ± 0.0008
ResNet-18	69.97 ± 0.18	1.67 ± 0.10	84.70 ± 0.08	0.0300 ± 0.0002
ResNet-34	74.07 ± 0.05	1.47 ± 0.07	87.05 ± 0.22	$\mathbf{0.0275 \pm 0.0010}$
ResNet-50	76.11 ± 0.18	1.50 ± 0.13	87.62 ± 0.09	0.0282 ± 0.0012

A.7 Tokenizer Robustness to Augmentations

Figure 10 examines how the learned tokenizer geometry and representations respond to small input translations on ImageNet-100.

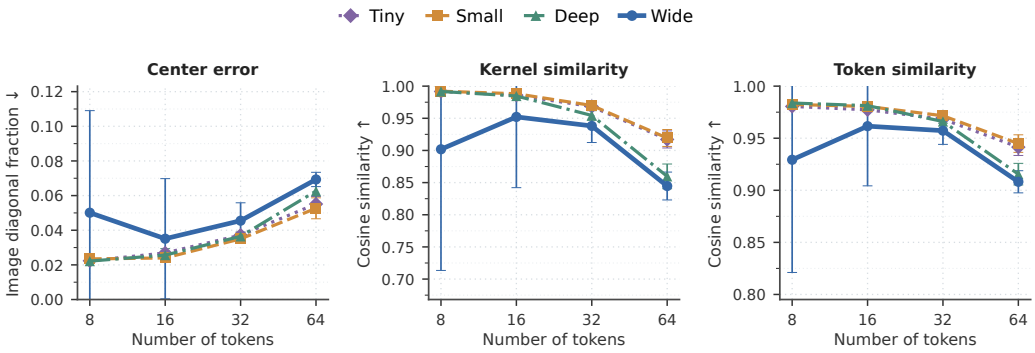

Figure 10: **Tokenizer robustness to small translations on ImageNet-100.** Center error measures translation equivariance as a fraction of the image diagonal after undoing the known input shift. Kernel and token similarity report cosine similarity between the aligned kernel maps and corresponding token vectors, respectively. Points show means over three runs, with error bars indicating two-sided 95% confidence intervals computed using Student's t-distribution.

Across model variants, the learned kernel geometry and resulting token representations remain stable under augmentation, particularly at moderate token budgets. Stability decreases at larger token counts, consistent with the emergence of finer and more localized spatial supports.

A.8 Tokenizer Stability Across Seeds

Figure 11 measures how consistently independently trained models learn their tokenizer geometry on ImageNet-100.

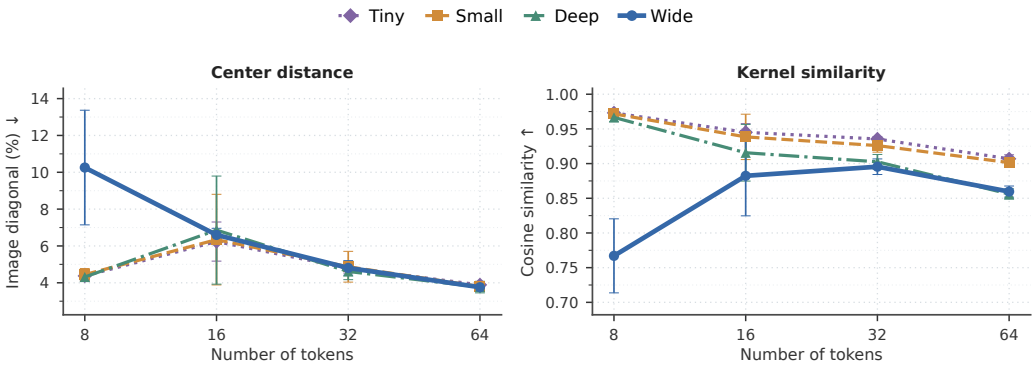

Figure 11: **Tokenizer stability across training seeds on ImageNet-100.** Kernel identities are unordered across independently trained models, center distance uses symmetric nearest-center matching and is reported as a percentage of the image diagonal. Kernel similarity is the symmetric best-match cosine similarity between Gaussian kernel maps. Points average pairwise comparisons among three training seeds, with error bars indicating two-sided 95% confidence intervals computed using Student's t-distribution.

Kernel maps are highly similar across training seeds for all model variants and token budgets. Center locations also remain close, although variability is greater for the Wide model at the smallest token budget.

A.9 Learned Kernel Geometry

Figure 12 characterizes the scale, shape, and orientation of the Gaussian kernels learned on ImageNet-100.

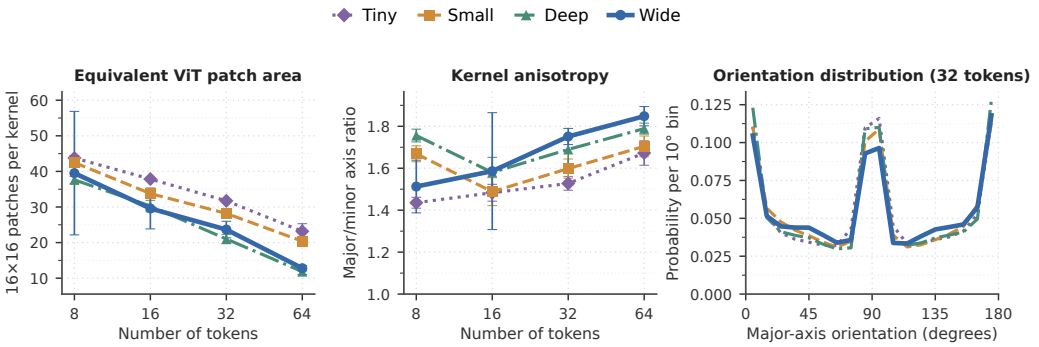

Figure 12: **Learned kernel geometry on ImageNet-100.** Equivalent ViT patch area expresses the median one-standard-deviation Gaussian ellipse area in units of 16×16 image patches. Kernel anisotropy is the median ratio between the major and minor covariance axes. The orientation panel reports the major-axis distribution for models with 32 tokens. Points show means over three runs, with error bars indicating two-sided 95% confidence intervals computed using Student's t-distribution.

As the token budget increases, individual kernels cover progressively smaller patch-equivalent areas while becoming modestly more anisotropic. The orientation distribution favors horizontal and vertical axes, while retaining substantial mass at intermediate orientations.

A.10 TOKENIZER SPATIAL ORGANIZATION

Figure 13 characterizes how the learned Gaussian kernels are arranged across the image on ImageNet-100.

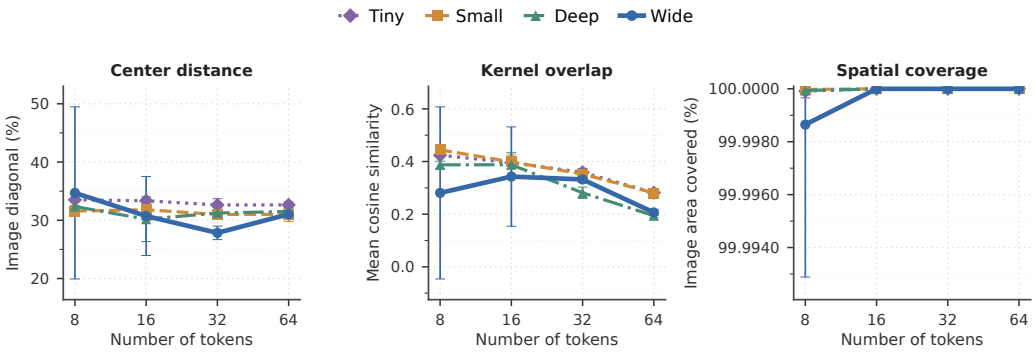

Figure 13: **Spatial organization of learned tokenizer kernels on ImageNet-100.** Center distance is the median pairwise distance between kernel centers, normalized by the image diagonal. Kernel overlap is the mean pairwise cosine similarity between flattened Gaussian kernel maps. Spatial coverage is the fraction of tokenizer-grid locations contained in the union of the kernels' two-standard-deviation ellipses. Points show means over three runs, with error bars indicating two-sided 95% confidence intervals computed using Student's t-distribution.

Kernel centers remain separated by approximately one-third of the image diagonal across token budgets, while mean kernel overlap decreases as more localized kernels are introduced. Together, the kernels maintain effectively complete spatial coverage for every model variant and token budget.

A.11 TOKENIZER PREDICTION RELEVANCE

Figure 14 examines whether kernel scale or spatial redundancy is associated with a token's contribution to the model prediction on ImageNet-100.

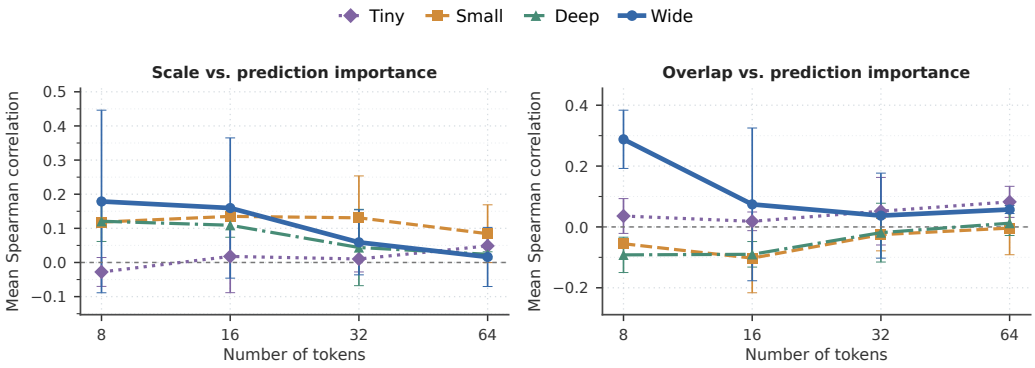

Figure 14: **Learned kernel geometry and prediction relevance on ImageNet-100.** Prediction importance is the decrease in the originally predicted-class logit when a token is removed. The left panel reports the within-image Spearman correlation between kernel area and importance. The right panel correlates importance with each kernel's mean cosine overlap with the other kernel maps. Points show means over three runs, with error bars indicating two-sided 95% confidence intervals computed using Student's t-distribution.

The correlations are generally weak and tend toward zero as the token budget increases. This suggests that prediction relevance is not determined solely by kernel size or spatial redundancy.

A.12 Exploratory Analysis of Geometry-Aware Transformer

The following is an exploratory overview of geometry-aware Transformers and positional encoding. This architecture extension is not used in the main experiments but may provide valuable insights for future work.

Figure 15 illustrates how the downstream Transformer can utilize the geometry retained by AdaGPK through geometric positional encoding and geometry-aware self-attention.

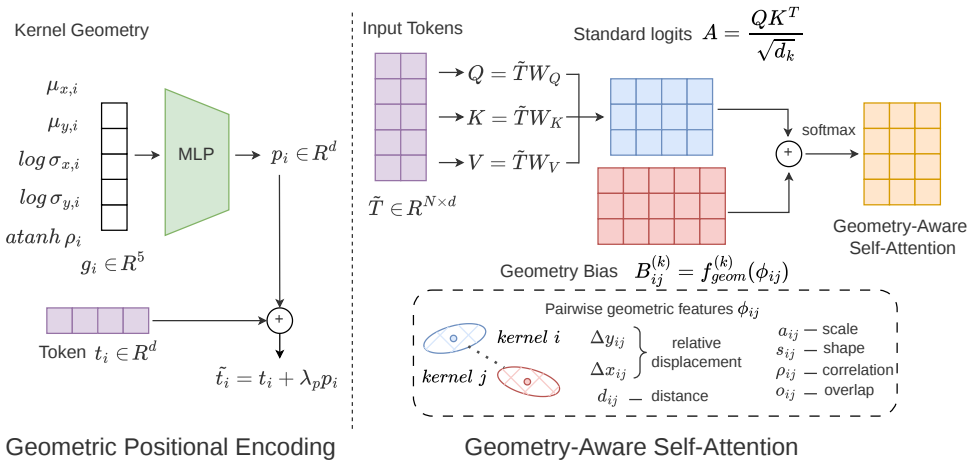

Figure 15: **Geometry-aware Transformer.** Kernel geometry provides both a per-token positional embedding (left) and a pairwise, head-specific bias to the self-attention logits (right).

Geometric positional encoding. Each kernel is represented by the five-dimensional descriptor $g_i = [\mu_{x,i}, \mu_{y,i}, \log \sigma_{x,i}, \log \sigma_{y,i}, \operatorname{atanh}(\rho_i)]$. After normalization, a two-layer MLP maps g_i to a D-dimensional positional embedding p_i. This embedding is combined with the normalized appearance token as

$$\widetilde{t}_i = \mathrm{LN}(t_i) + \lambda_p p_i, \tag{8}$$

where λ_p is a learned scalar initialized to a small value. Unlike a fixed grid-based positional embedding, p_i changes with the input because the kernel center, scale, and correlation are predicted independently for each image.

Geometry-aware self-attention. For every ordered pair of kernels (i, j), we construct a seven-dimensional feature vector ϕ_{ij} containing scale-normalized horizontal and vertical displacement, covariance-normalized distance, relative area, relative anisotropy, correlation difference, and the overlap of the two reconstructed Gaussian maps. A two-layer MLP maps ϕ_{ij} to one bias value per attention head. For head h, the attention logits become

$$A_{ij}^{(h)} = \frac{\langle q_i^{(h)}, k_j^{(h)} \rangle}{\sqrt{d_h}} + \beta_{ij}^{(h)}, \tag{9}$$

where $\beta_{ij}^{(h)}$ is the geometry bias, modulated by a learned per-head scale and a content-dependent gate. The attention weights are obtained by applying the softmax to $A^{(h)}$. This allows different heads to favor different spatial relationships while preserving conventional content-based attention.

A.13 Geometry-Aware Transformer Ablation

Table 6 compares geometric positional encoding, geometry-aware self-attention, and their combination across the three evaluated benchmarks.

Table 6: Ablation of the geometry-aware Transformer components for the DEEP model with 32 learned Gaussian kernels. Results are Top-1 accuracy (%) reported as mean ± standard deviation over three runs.

Transformer components	CIFAR-10	CIFAR-100	ImageNet-100
AdaGPK	92.40 ± 0.05	73.47 ± 0.38	85.19 ± 0.29
Positional encoding	92.53 ± 0.19	74.07 ± 0.18	85.32 ± 0.28
Geometry-aware attention	$\mathbf{92.54 \pm 0.13}$	74.00 ± 0.44	85.21 ± 0.20
Attention + positional encoding	92.43 ± 0.31	$\mathbf{74.27 \pm 0.13}$	$\mathbf{85.42 \pm 0.16}$

Positional encoding and geometry-aware self-attention showcase marginal gains in predictive performance across the evaluated classification benchmarks. However, they might prove beneficial for future work exploring more complex geometric relationships.

A.14 USE OF LLMS

Large language models were used to assist with language editing, proofreading, extending scripts and evaluating written code. All scientific ideas, analyses, experiments, results, and conclusions were developed and verified by the authors.