LOCALITY-ATTENDING VISION TRANSFORMER

Anonymous authors
Paper under double-blind review

ABSTRACT

Vision transformers have demonstrated remarkable success in classification by leveraging global self-attention to capture long-range dependencies. However, this same mechanism can obscure fine-grained spatial details crucial for tasks such as segmentation. In this work, we seek to enhance the segmentation performance of vision transformers after being trained using the usual image-level classification objective. More specifically, we present a simple yet effective add-on for vision transformers that improve their performance on segmentation tasks while retaining their image-level recognition capabilities. In our approach, we modulate the self-attention with a learnable Gaussian kernel that biases the attention toward neighboring patches. We further refine the patch representations to learn better embeddings at patch positions. These modifications ensure meaningful representations at spatial positions and encourage tokens to focus on local surroundings, while still preserving the model's ability to incorporate global information. Experiments demonstrate the effectiveness of our modifications, evidenced by substantial segmentation gains on three benchmarks (*e.g.*, over 6% and 4% on ADE20K for ViT Tiny and Base), without changing the training regime or sacrificing classification performance. The code is available at `https://anonymous.4open.science/r/LocAtViTRepo/`.

1 INTRODUCTION

Vision transformers (ViT, Dosovitskiy et al., 2021) have emerged as powerful visual backbones by modeling images as sequences of patch tokens, processed with self-attention. Unlike convolutional neural networks (CNN, LeCun et al., 2015), which aggregate local information in a restricted receptive field, ViTs can capture long-range dependencies at any layer. This global attention mechanism has proven highly effective for image classification, enabling ViT models to surpass CNN performance when sufficient data is available (Touvron et al., 2021a). A key factor behind this success is the ability to integrate global context that leads to more uniform and holistic representations across layers, which enhances the recognition of high-level image semantics (Raghu et al., 2021).

The same global focus that makes ViTs excel in classification, however, poses challenges for dense prediction tasks such as semantic segmentation. These tasks require precise localization and fine-grained spatial detail, properties that convolutional inductive biases naturally encourage but vanilla ViTs lack (Hassani et al., 2023). As a result, the design of spatial attention and feature hierarchy has been found critical for adapting transformers to dense tasks (Wang et al., 2021; Liu et al., 2021). Still, a tension remains between capturing global context and preserving local detail. Global attention can dilute local cues, whereas purely local schemes may miss long-range dependencies needed for holistic understanding. Besides, the classification objective used by models often neglects the necessities of dense prediction, motivating a need for a "segmentation-in-mind" pretraining. Empirically, we show in Appendix G that, in a ViT trained for classification, patch tokens progressively lose distinct local structure and become increasingly aligned with the `[CLS]` token.

More recently, foundation models trained at large-scale (Radford et al., 2021; Oquab et al., 2023), which learn versatile visual representations, have seen broad adoption in a breadth of visual tasks. Despite the availability of more intricate designs, these models still mostly adopt vanilla ViT due to its simplicity and ease of integration. This widespread reliance underscores the practical value of enhancing ViT's capabilities rather than pursuing more complex new designs. A prominent example is CLIP (Radford et al., 2021), which couples a ViT-based image encoder with a text encoder to align image-text representations, enabling zero-shot classification and open-vocabulary recognition. Such

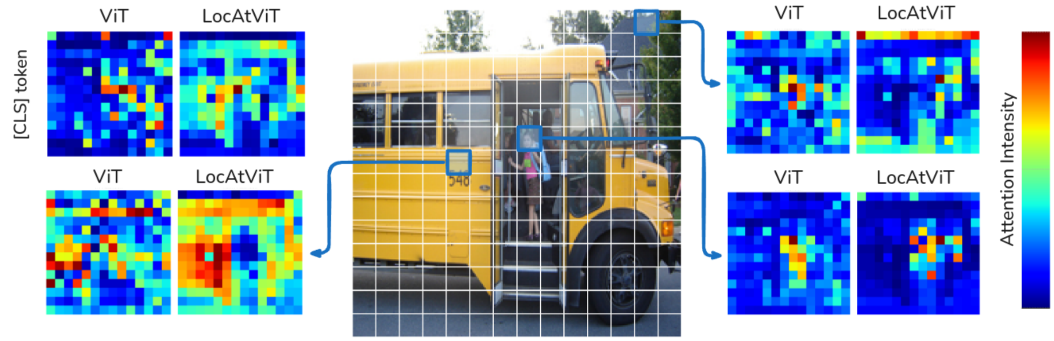


Figure 1: **Qualitative evaluation on the attention maps.** The final attention map of ViT and LocAtViT for the `[CLS]` token and three patches are illustrated for an image with label *school bus*.

representations can be repurposed for dense predictions, for instance, by comparing local features to text prompts, but this adaptation is non-trivial. Furthermore, recent studies try to harness CLIP's knowledge for segmentation without any task-specific training (Zhou et al., 2022; Wang et al., 2024; Hajimiri et al., 2025). However, as CLIP and similar models are not trained for quality local representations, their features often lack the spatial granularity needed for precise dense prediction.

Contributions. In this paper, we propose a modular *Locality-Attending* (LocAt) add-on, which incorporates two ideas: *(i)* We modulate the attention logits with a learnable Gaussian kernel centered on each query token's location, ensuring that patches closer to the query receive higher attention. This acts as an explicit inductive bias encouraging each token to attend to its local neighborhood while still allowing global interactions. We denote the resulting self-attention module as the *Gaussian-Augmented* (GAug) attention (Sec. 4.1). *(ii)* We enhance patch representations for segmentation by introducing minor changes prior to the classification head, preserving the meaningfulness of spatial tokens, that are most important for dense prediction. We term this procedure as *Patch Representation Refinement* (PRR) that addresses the gradient flow issue in ViTs for segmentation, which is overlooked in the literature (see Sec. 4.2). Hence, LocAt refers to the combination of GAug and PRR. Figure 2 demonstrates that it improves different baselines, yielding significant segmentation performance gains (arrows pointing upward), while preserving or improving classification accuracy (no arrow pointing to the left). The proposed add-on also enhances the quality of attention maps, as illustrated in Fig. 1. LocAt is a lightweight and objective-agnostic add-on, also compatible with self-supervised pretraining. Importantly, the minimal architectural changes required to integrate LocAt make it readily applicable to any ViT with marginal changes, facilitating its usage in foundation models. To the best of our knowledge, we are the first to offer this perspective on ViT pretraining: designing pretraining with downstream dense prediction in mind, while being faithful to vanilla ViT's training regime and architecture.

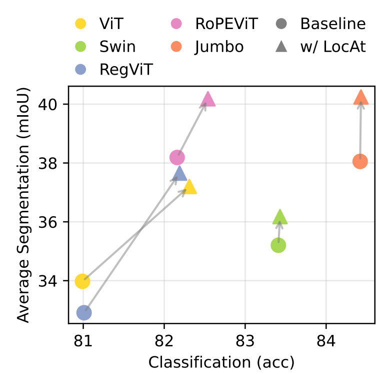


Figure 2: **LocAt considerably enhances different baselines** in segmentation, while preserving or even improving classification.

2 Related Work

Hierarchical ViT backbones for dense prediction. While the original ViT targets image classification and produces low-resolution features with weak locality priors (Dosovitskiy et al., 2021), dense prediction has motivated backbones that retain or recover spatial detail across stages. Some works use pyramid and token-merging designs to introduce multi-scale features and lightweight decoders for segmentation (Wang et al., 2021; Xie et al., 2021), while others build parallel branches for local and global processing (Chu et al., 2021). These works show that topology substantially helps dense tasks. However, they typically require non-trivial architectural changes (new stages or merging blocks) and may rely on local window attention that limits full-image interaction.

Convolution-based hybrids. Another line injects convolutional priors either inside attention or in the feed-forward network to encourage local bias while keeping global modeling. Works use convolutional projections (Wu et al., 2021a), add gated positional self-attention to softly bias toward convolutional behavior (d'Ascoli et al., 2021), couple local convolutional features with global representations (Peng et al., 2021), or add convolutions in the feed-forward network (Li et al., 2021). These hybrid models add extra modules that require tuning, and they can reduce plug-and-play compatibility with off-the-shelf ViTs, as they often introduce branches or replace core components. Besides, convolution offers a spatially-shared kernel which is independent of patch information.

Locality mechanisms inside attention. Orthogonal to backbone design, many papers modify the attention pattern itself to introduce locality. Many of the works use fixed or structured windows (Liu et al., 2021; Dong et al., 2021; Yang et al., 2021). Other ideas include utilizing sliding or dilated neighborhoods to expand receptive fields efficiently (Hassani et al., 2023; Hassani & Shi, 2023), sampling content-relevant keys (Xia et al., 2023), selecting regions using dynamic sparse routing (Zhu et al., 2023), or using explicit global-local mixers to balance context with locality (Ding et al., 2022; Tu et al., 2022; Chen et al., 2022; Hatamizadeh et al., 2023). Most of these approaches restrict or mask interactions (using windows or patterns) or add mixing subsystems that complicate design, impeding their widespread adoption.

Positional encodings that strengthen locality. Beyond absolute embeddings, relative positional encoding (RPE), and rotary positional encodings (RoPE) improve spatial awareness in ViTs (Shaw et al., 2018; Liu et al., 2021; Wu et al., 2021b; Su et al., 2021; Heo et al., 2024). These approaches are orthogonal to attention locality, and we briefly mentioned them to emphasize that they encode locality as well. Our work complements rather than replaces them, as we show in the experiments.

Improving token representation. Recent work on *register tokens* augments ViTs with dedicated auxiliary tokens that absorb non-informative computation and yield smoother feature maps helpful for dense prediction (Darcet et al., 2024). Unlike this approach, we do not require auxiliary tokens, and we also address the issue of gradient flow to spatial patch outputs, overlooked in the prior work. CaiT (Touvron et al., 2021b) introduces class-attention layers that specialize the last blocks to refining only the class token, while keeping patch tokens fixed in those layers, leading to suboptimal dense prediction performance. Token-labeling methods (Jiang et al., 2021) require a modified training regime and assign patch-level pseudo-labels from an external teacher. Finally, pooling heads such as global average pooling (GAP) and multihead attention pooling (MAP) (Zhai et al., 2022) aim to produce a stronger pooled representation for classification by aggregating patch tokens, while our work is explicitly designed for segmentation-in-mind training with an emphasis on improving the spatial token representations themselves rather than only the pooled vector.

Foundation models for dense prediction. Large pre-trained foundation models, such as CLIP (Radford et al., 2021), demonstrate impressive zero-shot generalization on image-level recognition by leveraging ViT backbones. The preference for the standard ViT backbone can be attributed to its strong global attention, predictable scaling behavior with data and model size, and a uniform architecture that avoids the need for complex stage-wise tuning as the model grows (Zhai et al., 2022; Alabdulmohsin et al., 2023). However, despite excelling on image-level benchmarks, such models remain less effective for dense prediction because their representations are predominantly global and task-agnostic (Shao et al., 2024). As a result, additional adaptation or decoding layers are usually required to repurpose them for segmentation or detection (Li et al., 2022; Xu et al., 2023; Luo et al., 2023). While these adaptations yield improvements, they do not fully address the core issue: foundation-model ViTs—trained with classification objectives—tend to emphasize global semantics over local detail (Liang et al., 2023).

A ViT backbone that natively preserves both local detail and global context could enable foundation models to excel at dense prediction without extra adaptation layers or specialized fine-tuning. In this work, we take a step in that direction by refining the ViT backbone itself. Our approach aims to potentially bridge the gap between the powerful image-level understanding and the requirements of pixel-level prediction tasks.

Mechanism family	Intact backbone	Locality type	Easily applied on ViT	Query adaptive
Conv-based hybrids	✗	Fixed spatial kernels, shared across patches.	✗	✗
Local window/block attention	✗	Hard locality within windows, limited cross-window links.	✗	Partial
Positional encodings	✓	Implicit spatial bias, no explicit distance decay.	✓	✗
Gaussian-Augmented (ours)	✓	Soft, continuous decay over patch distances.	✓	✓

Table 1: **Qualitative comparison of locality mechanisms in ViT architectures.** Further details are provided in Appendix E.

3 Preliminaries

Each ViT layer l takes a sequence of tokens $\mathbf{x}^{(l-1)} \in \mathbb{R}^{(1+hw)\times C}$ as input, containing a `[CLS]` token and hw spatial patch tokens. Each token is a C-dimensional vector, and h and w denote the number of patches in each column and row. $\mathbf{x}^{(0)}$ is the partitioned and flattened input after adding the positional embeddings. At each layer l, the following operations are applied, where LN, attn, and MLP denote layer normalization, self-attention, and feed-forward network, respectively:

$$\mathbf{x}' = \mathbf{x}^{(l-1)} + \mathrm{attn}\Big(\mathrm{LN}(\mathbf{x}^{(l-1)})\Big), \tag{1}$$

$$\mathbf{x}^{(l)} = \mathbf{x}' + \mathrm{MLP}\Big(\mathrm{LN}(\mathbf{x}')\Big). \tag{2}$$

Each self-attention module (attn) consists of two sets of weight matrices: $\mathbf{W}^{qkv} \in \mathbb{R}^{C\times d\times 3}$ to compute d-dimensional query, key, and value matrices (*i.e.*, $\mathbf{q}, \mathbf{k}, \mathbf{v} \in \mathbb{R}^{(1+hw)\times d}$) based on the input, and $\mathbf{W}^o \in \mathbb{R}^{d\times C}$ for the final projection. After obtaining $\mathbf{q}$, $\mathbf{k}$, and $\mathbf{v}$, we calculate:

$$\mathbf{A} = \mathrm{softmax}\left(\mathbf{q}\mathbf{k}^\top/\sqrt{d}\right)\mathbf{v}. \tag{3}$$

Matrix $\mathbf{A} \in \mathbb{R}^{(1+hw)\times d}$ is then transformed by $\mathbf{W}^o$ to form the output of the layer. The *attention logits* of a patch p are represented by the p^{th} row of $\mathbf{q}\mathbf{k}^\top/\sqrt{d}$. Note that for simplicity, we present the formulation of a single-head self-attention.

4 Method

We now present **LocAtViT**, which enhances ViT with two modular components, GAug attention (Sec. 4.1) and PRR (Sec. 4.2), and is trained with the same classification objective as ViT.

4.1 Gaussian-Augmented attention

We aim to introduce explicit local attention into layers of vision transformer (for all tokens but `[CLS]`) via adding a patch-specific Gaussian kernel to attention logits. We first discuss the altered self-attention formulation, followed by details on computation of the kernel, and then the final form of the attention. Table 1 compares our approach to multiple related work categories, and motivates our choice of a Gaussian kernel as a simple, query-adaptive locality bias that can be added on top of vanilla self-attention without modifying the ViT architecture or training objective.

Modified self-attention. At every layer's self-attention, we add a *supplement* matrix $\mathbf{S}$ to the attention logits, aiming to emphasize the attention of each patch to its surrounding. With this addition, the self-attention formulation of Eq. (3) is modified as follows, which is also depicted in Fig. 3a:

$$\mathbf{A} = \mathrm{softmax}\left(\frac{\mathbf{q}\mathbf{k}^\top}{\sqrt{d}} + \mathbf{S}\right)\mathbf{v}. \tag{4}$$

We construct $\mathbf{S}$ so that a patch p attends more to its immediate surroundings, with increment gradually decreasing with distance from p. A natural choice for such a distance-based locality prior is an unnormalized Gaussian centered at p. A Gaussian kernel provides a smooth, monotone decay of influence with distance, controlled by a variance parameter σ^2 (in the isotropic case). This gives an interpretable handle on the effective receptive field: small σ yields a sharp, highly local focus, whereas large σ approaches a nearly uniform weighting over patches (more information in Appendix F). We parameterize the variance of the Gaussian kernel for each patch by a 2D vector, stored in the p^{th} row of $\mathbf{\Sigma} \in \mathbb{R}_+^{hw\times 2}$, controlling the attention span along both axes for each patch. Since patches might have different needs in how far they should attend to their neighbors, we compute the variances based on the query matrix derived from the input, using a learnable weight matrix $\mathbf{W}^\sigma \in \mathbb{R}^{d\times 2}$ (with f being a scaled sigmoid function ensuring positive, bounded values):

$$\mathbf{\Sigma} = f(\mathbf{q}\mathbf{W}^\sigma), \tag{5}$$

Gaussian kernel. For a patch grid of size $h \times w$, we denote the set of coordinate vectors:

$$\mathbf{P} = [i \quad j]_{i\in\{1,2,\dots,h\},\, j\in\{1,2,\dots,w\}}\,, \tag{6}$$

in an $hw \times 2$ matrix. The $hw \times hw \times 2$ pairwise squared difference $\mathbf{D}$ is computed as:

$$\mathbf{D}_{ptm} = \Big(\mathbf{P}_{pm} - \mathbf{P}_{tm}\Big)^2, \quad \text{for } m \in \{1,2\}, \tag{7}$$

where p and t denote indices of source and target patch, and m indexes the coordinate dimensions. Given $\mathbf{\Sigma}$, the elements in the Gaussian kernel matrix $\mathbf{G} \in \mathbb{R}_+^{(hw+1)\times(hw+1)}$ are calculated as:

$$\mathbf{G}_{pt} = \exp\Big(-\frac{1}{2}\sum_{m=1}^{2}\frac{\mathbf{D}_{ptm}}{\mathbf{\Sigma}_{pm}}\Big), \tag{8}$$

which determines the addition to attention logits from patch p to t. We construct the Gaussian kernel only over spatial tokens. Since `[CLS]` has no spatial coordinates, entries involving `[CLS]` are zero and only the patch-patch attention logits are augmented. By pre-computing $\mathbf{D}$, *i.e.*, the numerator, we can efficiently compute $\mathbf{G}$ during training.

Supplement matrix. Based on Eq. (8), each entry in $\mathbf{G}$ lies in $[0, 1]$. Directly setting $\mathbf{S} = \mathbf{G}$ in Eq. (4) causes a scale mismatch between $\mathbf{S}$ and the attention logits. To mitigate this discrepancy, we assume a learnable weight matrix $\mathbf{W}^\alpha \in \mathbb{R}^{d\times 1}$ that computes the desired scaling for each patch, based on its $\mathbf{q}$ vector. Entries in $\boldsymbol{\alpha}$ scale rows of the Gaussian kernel, more specifically:

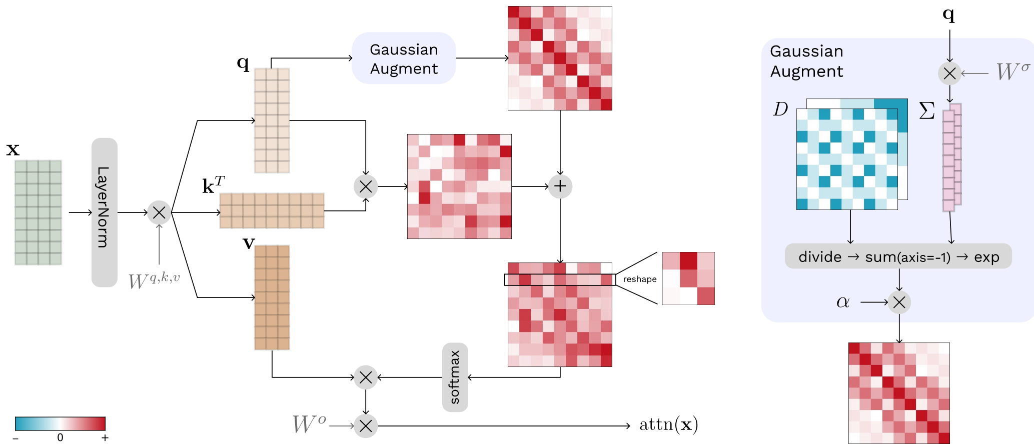
$$\boldsymbol{\alpha} = \text{softplus}(\mathbf{q}\mathbf{W}^\alpha) \in \mathbb{R}_+^{hw}, \tag{9}$$

$$\mathbf{S} = \text{diag}(\boldsymbol{\alpha})\, \mathbf{G}, \tag{10}$$

in which softplus ensures positive coefficients. Intuitively, $\boldsymbol{\alpha}$ acts as a per-query, row-wise balancing factor between the original attention logits and the Gaussian locality prior. For tokens where the network predicts small values of $\boldsymbol{\alpha}$, the contribution of $\mathbf{S}$ is negligible and the behavior approaches standard global self-attention (weak locality), whereas larger values of $\boldsymbol{\alpha}$ yield a stronger local bias. This makes our approach a soft, data-dependent locality mechanism rather than a hard constraint. We empirically analyze the effect of this scaling, as well as parameter-free alternatives, in Appendix D.3 and D.4. We refer to our modified self-attention as *Gaussian-Augmented* (GAug) attention. Figure 3b illustrates the generation process of the supplement matrix.

4.2 Patch Representation Refinement

Problem statement. In a classification task using ViT, only the `[CLS]` token's output of the model is used for computing the loss. While effective for classification, this approach has fundamental limitations for dense prediction from a gradient flow perspective. More concretely, the patch positions' outputs receive no *direct* supervision, *i.e.*, it is not important to the model what ViT's final outputs are at those positions. However, these output representations are crucial for further dense prediction. This is problematic because the fine-grained spatial information carried by individual patch tokens is not effectively learned at the final layer.



(a) Modified self-attention. (b) Supplement matrix $\mathbf{S}$.

Figure 3: **Illustration of the Gaussian-Augmented attention** for a 3×3 grid. For simplicity, the `[CLS]` token is not shown. *(a)* The Gaussian addition, *i.e.*, $\mathbf{S}$ in Eq. (4), is obtained based on $\mathbf{q}$ and is added to the attention logits. The p-th row in the attention logits matrix presents the attention of patch p to all patch tokens. The reshaped matrix illustrates that with the GAug add-on, both local and global attentions are integrated. *(b)* The supplement matrix $\mathbf{S}$ encourages attending to the locality and is computed using the pairwise squared difference tensor $\mathbf{D}$ from Eq. (7). For simplicity, we fix the Gaussian variances and scaling coefficients to one for all patches in this visualization.

Some subsequent methods, such as Swin (Liu et al., 2021), remove the `[CLS]` token and use global average pooling (GAP) before the classification head. However, this forces an undesirable behavior from a dense prediction standpoint, *i.e.*, a *uniform gradient flow* across all positions. For example, in an image of a bird with other objects in the background, GAP compels the model to match all patch representations—including background regions—with the classifier's prototype of bird. The uniform gradient flow means that all patch tokens receive equal importance, regardless of their relevance, potentially leading to representations particularly suboptimal for tasks like segmentation. Moreover, GAP has been shown to reduce localization in higher layers (Raghu et al., 2021).

Proposed solution. To encourage meaningful patch representations at the final layer's output, $\mathbf{x}^{(l)}$, we propose the following operation before the classification head:

$$\mathbf{x}^{+} = \mathrm{softmax}\left(\frac{\mathbf{x}^{(l)}\mathbf{x}^{(l)\top}}{\sqrt{d}}\right)\mathbf{x}^{(l)}, \tag{11}$$

which acts like a *parameter-free* self-attention. This operation, which introduces no new parameter, aggregates information from all patch positions in a non-uniform manner, thereby preserving their unique contributions and ensuring diverse gradient flow across patch locations. The resulting representation at the `[CLS]` token, $\mathbf{x}_0^{+}$, is then passed to the classification head. We refer to this strategy as *Patch Representation Refinement* (PRR). PRR can be seen as an alternative to GAP, suitable for segmentation-in-mind pretraining.

Our components share the common objective of making ViT representations more suitable for dense prediction, and they act at different stages. GAug operates inside the backbone, modifying self-attention to bias information exchange toward local neighborhoods so that patch tokens can better encode fine spatial details. PRR, in contrast, acts right before the classification head and changes how tokens are aggregated to explicitly route supervision and gradients to patch outputs. In practice, each module can be attached independently to a ViT backbone (see ablations in Sec. 5.4). However, they are coupled through the gradient path: with standard `[CLS]` classification, adding GAug in the last block has little effect, because its parameters receive no gradient from the loss, whereas PRR routes gradients to those GAug parameters so they can be effectively learned.

5 EXPERIMENTS

5.1 EXPERIMENTAL SETUP

Datasets. For the main experiments, where we assess both classification and segmentation performance, we first train models on ImageNet-1K (Deng et al., 2009; Russakovsky et al., 2015), which contains 1.28M training images from 1,000 classes. Then, we further utilize these models for training on segmentation datasets: ADE20K (Zhou et al., 2019), PASCAL Context (Mottaghi et al., 2014), and COCO Stuff (Caesar et al., 2018; Lin et al., 2014), which contain 150, 59, and 171 semantic categories, respectively. ADE20K and COCO Stuff images are resized to 512×512 and PASCAL Context images to 480×480. Furthermore, we also assess classification performance on smaller scale datasets: CIFAR-100 (Krizhevsky & Hinton, 2009) and mini-ImageNet (Vinyals et al., 2016), a subset of ImageNet-1K, consisting of 100 classes with 500 training and 100 validation examples each. In all classification experiments, images are resized to 224×224.

Implementation details. Our method is implemented using the PyTorch Image Models (`timm`) (Wightman, 2019) library. We train models on ImageNet-1K for 300 epochs, with initial learning rate (LR) 0.001, and on CIFAR-100 and mini-ImageNet for 600 epochs, with LR 0.0005. Global batch size is set to 1024, linear warm-up to 20 epochs, and we use AdamW (Kingma & Ba, 2014; Loshchilov & Hutter, 2019) optimizer with a weight decay of 0.05. As in Ding et al. (2022), a simple triangular learning rate scheduler (Smith & Topin, 2018) is applied, and the stochastic depth drop rates (Huang et al., 2016) for the Tiny, Small, and Base backbones are set to 0.1, 0.2, and 0.4, respectively. We follow Liu et al. (2021) for data augmentation and use RandAugment (Cubuk et al., 2020), Mixup (Zhang et al., 2018), Cutmix (Yun et al., 2019), and random erasing (Zhong et al., 2020). The sigmoid function f in Eq. (5) is scaled to have a maximum of $\max(h, w)$, and shifted to satisfy $f(0) = 1$.

For semantic segmentation, we utilize the MMSegmentation toolbox (OpenMMLab, 2020) and employ a simple 1-layer MLP on top of the frozen classification-trained models. This configuration ensures that segmentation performance mainly reflects the discriminative power of the classification-trained backbone in dense prediction. This setup aligns with our goal of isolating and assessing patch representation quality under a low-tuning regime (more information in Appendix J). Training on segmentation datasets is performed over 20K iterations with a batch size of 32.

5.2 MAIN RESULTS

Segmentation performance. The LocAt add-on can be applied on several ViT-based models, and Tab. 2 evaluates its effect, in terms of classification performance on ImageNet-1K, as well as segmentation performance on three benchmarks, when applied to five models: ViT (Dosovitskiy et al., 2021), Swin Transformer (Liu et al., 2021), ViTs with registers (denoted as RegViT, we use 4 registers, Darcet et al., 2024), Rotary Position Embedding for ViTs (denoted as RoPEViT, Heo et al., 2024), and the recent Jumbo (Fuller et al., 2025). Comparing each baseline with its enhanced counterpart (gray row below), proves LocAt's addition is useful in improving the segmentation performance of all. For instance, LocAtViT Tiny achieves a substantial improvement of **+6.17%**, **+4.86%**, and **+5.86%**, over ViT on ADE20K, PASCAL Context, and COCO Stuff, respectively. Importantly, LocAt-enhanced models' superior segmentation performance is achieved without compromising classification performance; in fact, they deliver comparable or even improved accuracy across different models (*e.g.*, LocAtViT outperforms ViT by **+1.55%** in Tiny backbone).

LocAt improves baselines that are architecturally close to ViT significantly, *e.g.*, RoPEViT, and interestingly, it brings improvements over Swin as well. We believe this is not trivial as the add-on was designed for ViT's architecture, in which there exists a `[CLS]` token and the attention width is not limited, while in Swin the windowed attention mechanism severely affects the extent to which LocAt can play a role. Furthermore, our add-on incurs a negligible increase in computational efficiency in terms of number of FLOPs over the corresponding counterparts (measured at 224×224 using Sovrasov, 2018-2024). Additional experiments are presented in Appendix B.

Classification performance. In addition to the ImageNet-1K classification results in Tab. 2, Tab. 3 investigates LocAt's classification effectiveness on small-scale datasets: mini-ImageNet (Vinyals

Table 2: **Segmentation performance** of models and their counterparts with our LocAt extension (in gray), along with their **classification performance** on ImageNet-1K, which the models are initially trained on. Results demonstrate that *(i)* LocAt substantially boosts segmentation performance (*our primary focus*), while preserving or even improving the classification performance, and *(ii)* this effect holds for a variety of methods, for different backbone sizes. Furthermore, *(iii)* the segmentation gains appear not only in weaker baselines, but also in strong, high-performing models, where classification improvements are harder to achieve.

	Method	Segmentation mIoU (%)			Top-1 (%)	#Params	FLOPs
		ADE	P-Context	C-Stuff	ImageNet	(M)	(G)
Tiny	ViT	17.30	33.71	20.29	72.39	6	1.26
	+ LocAt	23.47$_{+6.17}$	38.57$_{+4.86}$	26.15$_{+5.86}$	73.94$_{+1.55}$	6	1.27
	Swin	25.58	36.78	28.34	81.18	28	4.50
	+ LocAt	26.52$_{+0.94}$	37.65$_{+0.87}$	29.09$_{+0.75}$	81.43$_{+0.25}$	28	4.51
	RegViT	15.98	33.45	19.58	72.90	6	1.29
	+ LocAt	24.39$_{+8.41}$	39.90$_{+6.45}$	27.38$_{+7.80}$	74.08$_{+1.18}$	6	1.30
	RoPEViT	19.17	38.16	22.75	73.60	6	1.26
	+ LocAt	24.48$_{+5.31}$	40.79$_{+2.63}$	27.98$_{+5.23}$	74.34$_{+0.74}$	6	1.27
	Jumbo	20.33	36.36	22.13	78.71	9	1.40
	+ LocAt	21.62$_{+1.29}$	37.22$_{+0.86}$	23.87$_{+1.74}$	78.78$_{+0.07}$	9	1.42
Base	ViT	28.40	43.10	30.43	80.99	86	17.58
	+ LocAt	32.64$_{+4.24}$	45.35$_{+2.25}$	33.62$_{+3.19}$	82.31$_{+1.32}$	86	17.64
	Swin	31.90	40.11	33.60	83.41	88	15.46
	+ LocAt	32.89$_{+0.99}$	41.44$_{+1.33}$	34.20$_{+0.60}$	83.43$_{+0.02}$	88	15.47
	RegViT	27.93	41.81	28.99	81.01	86	17.95
	+ LocAt	32.71$_{+4.78}$	46.14$_{+4.33}$	34.12$_{+5.13}$	82.19$_{+1.18}$	86	18.02
	RoPEViT	31.38	48.83	34.35	82.16	86	17.58
	+ LocAt	34.94$_{+3.56}$	49.24$_{+0.41}$	36.37$_{+2.02}$	82.54$_{+0.38}$	86	17.64
	Jumbo	32.20	47.31	34.65	84.42	130	19.74
	+ LocAt	35.69$_{+3.49}$	49.20$_{+1.89}$	35.84$_{+1.19}$	84.43$_{+0.01}$	130	19.81

Table 3: **Classification top-1 accuracy** of ViT and LocAtViT for different backbone sizes, on mini-ImageNet and CIFAR-100, showcasing LocAt's effectiveness on small-scale datasets.

Size	mini-ImageNet		CIFAR-100	
	ViT	LocAtViT	ViT	LocAtViT
Tiny	74.94	78.47$_{+3.53}$	73.84	80.43$_{+6.59}$
Small	78.98	84.30$_{+5.32}$	76.33	81.13$_{+4.80}$
Base	79.91	84.86$_{+4.95}$	76.90	82.20$_{+5.30}$

Table 4: **Self-supervised performance of LocAtViT used in DINO**, showcasing LocAt's effectiveness in the self-supervised regime.

Experiment		ViT-S/16	LocAtViT-S/16
Linear classification		65.52	67.65$_{+2.13}$
Nearest neighbor	10-NN	61.69	63.96$_{+2.27}$
	20-NN	61.53	63.74$_{+2.21}$
	100-NN	59.30	61.19$_{+1.89}$
	200-NN	57.90	59.78$_{+1.88}$

et al., 2016) and CIFAR-100 (Krizhevsky & Hinton, 2009). Although designed to enhance segmentation, these results demonstrate LocAt's classification effectiveness even when trained on small-scale datasets. LocAt improves ViT's performance by 3-6% on mini-ImageNet and 4-7% on CIFAR-100, while only introducing $2,340$ new parameters (0.003% increase for Base). Please note that segmentation results are not included for models trained on these datasets since, due to their scale and number of classes, representations are not expected to generalize well to segmentation benchmarks.

Foundation models. In the previous sections, we described our interest in improving ViT's segmentation capabilities without changing their training scheme. Our experiments support that our minor modifications lead to better dense prediction performance, while performing on par or superior to the vanilla models in classification. One reason for our interest in the mentioned problem is that ViTs have been widely used across computer vision foundation models and are the go-to choice for many of the recent methods (Radford et al., 2021; Kirillov et al., 2023; Caron et al., 2021; Oquab et al., 2023). One of the popular models that yields versatile image representations and transfers well to different computer vision tasks is DINO (Caron et al., 2021), which is trained in a self-supervised

Table 5: **Hummingbird dense NN retrieval** (mIoU %) on PASCAL VOC and ADE20K.

Method	Tiny				Base			
	PASCAL		ADE20K		PASCAL		ADE20K	
	Vanilla	+ LocAt	Vanilla	+ LocAt	Vanilla	+ LocAt	Vanilla	+ LocAt
ViT	39.2	**50.3**	12.0	**15.2**	55.8	**58.7**	19.5	**21.5**
Swin	45.2	**45.3**	16.1	**16.3**	57.6	**62.8**	23.3	**24.6**
RegViT	39.4	**52.3**	12.5	**15.9**	55.5	**60.3**	19.4	**22.8**
RoPEViT	50.7	**54.7**	16.0	**17.5**	61.0	**61.4**	22.4	**23.7**
Jumbo	40.0	**45.5**	13.3	**14.5**	58.5	**63.8**	21.6	**23.7**

manner and can serve as a general-purpose backbone. Two of the main evaluation protocols used by Caron et al. (2021) are learning a linear classifier on top of the frozen backbone and nearest neighbor classification (k-NN) on top of the features.

We train DINO ViT-S/16 and DINO LocAtViT-S/16 on ImageNet-1K for 50 epochs using the setting provided in the official repository, and evaluate them on the mentioned tasks. Table 4 demonstrates that replacing ViT with LocAtViT in DINO actually improves its performance on both linear and k-NN classification. We report the k-NN performance on $k \in \{10, 20, 100, 200\}$ as advised by Caron et al. (2021). These findings reveal our objective-agnostic modifications' effectiveness in the self-supervised regime and the potential of our method on backbones that learn general-purpose representations. While interesting, further investigation on larger foundation models is beyond our computational reach and lies outside the scope of this work.

Hummingbird evaluation. To further assess whether LocAt improves quality of image features, we evaluate our models using Hummingbird (Balažević et al., 2023), a protocol proposed for evaluating *in-context scene understanding* in a purely frozen-feature regime. We use the implementation by Pariza et al. (2024) and follow its dense nearest-neighbor (NN) retrieval setup. Table 5 shows that LocAt consistently improves NN retrieval performance relative to the corresponding vanilla backbones on PASCAL VOC (Everingham et al., 2010) and ADE20K, across architectures, suggesting that LocAt enhances spatial representations, even without any task-specific fine-tuning or decoder.

5.3 Qualitative analysis

An interesting implication of our proposed modifications is the refinement of ViT's patch outputs, which makes it more suitable for use cases on dense prediction tasks. Figure 1 offers a visual comparison of attention maps from a vanilla ViT and our LocAtViT, both trained for classification, for an image labeled as *school bus*. From the `[CLS]` token's attention, we observe that ViT's focus is broadly dispersed, whereas LocAtViT shows more concentrated and coherent activation on key features of the bus. Furthermore, we present the attention maps of three patch tokens to other patches. For instance, a patch on the bus side attends to nearly the entire bus in LocAtViT, whereas ViT's map is harder to interpret. A patch covering the child's face generates meaningful attention in both models, but ViT seems to highlight unrelated regions more. Interestingly, for a patch near the top-right corner, LocAtViT not only focuses on some tree patches, but also extends attention to the sky and road, all corresponding to the image background. Despite being trained solely for classification, LocAtViT exhibits an improved ability to detect some scene structures, suggesting that our proposed local interactions can enrich the model's contextual understanding without sacrificing global attention. Further qualitative examples are presented in Appendix C.

5.4 Ablation study

In this section, we provide an ablation study on the architectural choices we made. We also provide ablation study on the self-attention module's design in the Appendix D, and we compare PRR to pooling heads and class-attention in Appendix H and I.

Effect of GAug and PRR. Part ❶ of Tab. 6 ablates on GAug and PRR defined in Secs. 4.1 and 4.2. Results demonstrate that both GAug and PRR indeed enhance the performance of the model in both classification and segmentation, and their combination pushes the performance even further.

Table 6: **Ablations on model's architecture.** We report segmentation performance (mIoU %) over three benchmarks and classification accuracy (top-1 %) on ImageNet-1K. PE and GAP stand for positional embeddings and global average pooling.

	Method	Tiny				Base			
		ADE	P-Context	C-Stuff	ImageNet	ADE	P-Context	C-Stuff	ImageNet
❶	ViT	17.30	33.71	20.29	72.39	28.40	43.10	30.43	80.99
	ViT + GAug	18.98	34.97	21.51	73.16	30.26	44.36	32.21	82.00
	ViT + PRR	21.60	37.93	25.85	73.71	29.89	44.03	32.16	82.19
	LocAtViT	23.47	38.57	26.15	73.94	32.64	45.35	33.62	82.31
❷	ViT - PE	15.13	31.94	19.35	69.36	24.59	40.18	28.79	79.39
	LocAtViT - PE	22.69	38.15	26.05	73.10	29.73	44.69	32.17	82.17
❸	ViT	17.30	33.71	20.29	72.39	28.40	43.10	30.43	80.99
	ViT + GAP	19.65	34.94	22.86	72.50	27.99	41.97	29.88	81.84
	ViT + PRR	21.60	37.93	25.85	73.71	29.89	44.03	32.16	82.19

Effect of positional embeddings. Part ❷ of Tab. 6 evaluates the impact of the default absolute positional embeddings (PE) on our proposed LocAt add-on. For both backbone sizes, LocAtViT without PE not only outperforms ViT without PE, but also surpasses ViT with PE. This indicates that LocAt captures the spatial information embedded into PE and more, with much fewer learnable parameters. It is worth noting that our approach is not an alternative to positional encoding and we did not intend to propose a new PE method. Therefore, these results are included just to demonstrate empirically that LocAt indeed captures the spatial information that the default PE captures, which is the agent for capturing locality in vanilla ViT. We have shown in Tab. 2 that LocAt is applicable alongside other, newer positional encoding approaches, such as RoPE, as well.

Comparison between PRR and GAP. As discussed in Sec. 4.2, PRR addresses patch locations' gradient flow issues while overcoming GAP's limitations in segmentation. Part ❸ of Tab. 6 compares how vanilla ViT performs when equipped with PRR versus GAP. PRR shows superior segmentation performance and interestingly, it improves classification accuracy more than GAP. Moreover, although GAP helps ViT in classification, it hurts the segmentation performance in the Base backbone, which is in line with the discussions in Sec. 4.2 about GAP's problems in segmentation.

6 Conclusion

Summary. We present the *Locality-Attending Vision Transformer*, a modular framework that enhances vision transformers for dense prediction while preserving image-level capabilities and integrating seamlessly into existing ViTs. This introduces a segmentation-in-mind pretraining perspective: by adding *GAug* attention, our method biases self-attention toward local regions to capture fine-grained spatial details, while *PRR* ensures meaningful gradient flow to patch tokens, strengthening representations for dense prediction. Extensive experiments across multiple ViT baselines show that LocAt delivers superior segmentation performance without compromising classification accuracy. Our objective is not to surpass state-of-the-art architectures, but to improve classification-trained ViT backbones for segmentation with a method largely orthogonal to prior advancements, motivated by the trend of them being widely used, *e.g.*, by foundation models. Consistent with Heo et al. (2024), we therefore emphasize comparisons between baselines and their LocAt-enhanced counterparts. We hope that these lightweight modifications will be adopted in ViT-based foundation models.

Limitations. We evaluated our method on multiple classification and segmentation datasets. However, these datasets all only contain natural images, and we have left evaluation on other domains such as medical imaging or remote sensing as future work. Furthermore, while we have demonstrated the effectiveness of LocAtViT used in a small foundation model, evaluation on large foundation models, such as CLIP, has been out of our computational reach.

References

Ibrahim M Alabdulmohsin, Xiaohua Zhai, Alexander Kolesnikov, and Lucas Beyer. Getting vit in shape: Scaling laws for compute-optimal model design. *Advances in Neural Information Processing Systems*, 36:16406–16425, 2023.

Ivana Balažević, Daniel Steiner, Nikhil Parthasarathy, Relja Arandjelović, and Olivier J Hénaff. Towards in-context scene understanding. *arXiv preprint arXiv:2306.01667*, 2023.

Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. Coco-stuff: Thing and stuff classes in context. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2021.

Chun-Fu (Richard) Chen, Rameswar Panda, and Quanfu Fan. RegionViT: Regional-to-Local Attention for Vision Transformers. In *International Conference on Learning Representations*, 2022.

Xiangxiang Chu, Zhi Tian, Yuqing Wang, Bo Zhang, Haibing Ren, Xiaolin Wei, Huaxia Xia, and Chunhua Shen. Twins: Revisiting the design of spatial attention in vision transformers. *Advances in neural information processing systems*, 34:9355–9366, 2021.

Xiangxiang Chu, Zhi Tian, Bo Zhang, Xinlong Wang, and Chunhua Shen. Conditional positional encodings for vision transformers. In *International Conference on Learning Representations*, 2023.

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

Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. Vision transformers need registers. In *International Conference on Learning Representations*, 2024.

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

Mingyu Ding, Bin Xiao, Noel Codella, Ping Luo, Jingdong Wang, and Lu Yuan. Davit: Dual attention vision transformers. In *European conference on computer vision*, pp. 74–92. Springer, 2022.

Xiaoyi Dong, Jianmin Bao, Dongdong Chen, Weiming Zhang, Nenghai Yu, Lu Yuan, Dong Chen, and Baining Guo. Cswin transformer: A general vision transformer backbone with cross-shaped windows, 2021.

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

Stéphane d'Ascoli, Hugo Touvron, Matthew L Leavitt, Ari S Morcos, Giulio Biroli, and Levent Sagun. Convit: Improving vision transformers with soft convolutional inductive biases. In *International conference on machine learning*, pp. 2286–2296. PMLR, 2021.

Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2): 303–338, 2010.

Anthony Fuller, Yousef Yassin, Daniel G Kyrollos, Evan Shelhamer, and James R Green. Simpler fast vision transformers with a jumbo cls token. 2025.

Sina Hajimiri, Ismail Ben Ayed, and Jose Dolz. Pay attention to your neighbours: Training-free open-vocabulary semantic segmentation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2025.

Ali Hassani and Humphrey Shi. Dilated neighborhood attention transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

Ali Hassani, Steven Walton, Jiachen Li, Shen Li, and Humphrey Shi. Neighborhood attention transformer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 6185–6194, 2023.

Ali Hatamizadeh, Hongxu Yin, Greg Heinrich, Jan Kautz, and Pavlo Molchanov. Global context vision transformers. In *Proceedings of the 40th International Conference on Machine Learning*, pp. 12633–12646, 2023.

Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pp. 2961–2969, 2017.

Byeongho Heo, Song Park, Dongyoon Han, and Sangdoo Yun. Rotary position embedding for vision transformer. In *European Conference on Computer Vision*, pp. 289–305. Springer, 2024.

Gao Huang, Yu Sun, Zhuang Liu, Daniel Sedra, and Kilian Q Weinberger. Deep networks with stochastic depth. In *European conference on computer vision*, pp. 646–661. Springer, 2016.

Zi-Hang Jiang, Qibin Hou, Li Yuan, Daquan Zhou, Yujun Shi, Xiaojie Jin, Anran Wang, and Jiashi Feng. All tokens matter: Token labeling for training better vision transformers. *Advances in neural information processing systems*, 34:18590–18602, 2021.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything. *arXiv:2304.02643*, 2023.

Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. 2009.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

Boyi Li, Kilian Q Weinberger, Serge Belongie, Vladlen Koltun, and René Ranftl. Language-driven semantic segmentation. *arXiv preprint arXiv:2201.03546*, 2022.

Yawei Li, Kai Zhang, Jiezhang Cao, Radu Timofte, and Luc Van Gool. Localvit: Bringing locality to vision transformers. *arXiv preprint arXiv:2104.05707*, 2021.

Feng Liang, Bichen Wu, Xiaoliang Dai, Kunpeng Li, Yinan Zhao, Hang Zhang, Peizhao Zhang, Peter Vajda, and Diana Marculescu. Open-vocabulary semantic segmentation with mask-adapted clip. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7061–7070, 2023.

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

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.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

Huaishao Luo, Junwei Bao, Youzheng Wu, Xiaodong He, and Tianrui Li. SegCLIP: Patch aggregation with learnable centers for open-vocabulary semantic segmentation. *ICML*, 2023.

Roozbeh Mottaghi, Xianjie Chen, Xiaobai Liu, Nam-Gyu Cho, Seong-Whan Lee, Sanja Fidler, Raquel Urtasun, and Alan Yuille. The role of context for object detection and semantic segmentation in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2014.

OpenMMLab. MMSegmentation: Openmmlab semantic segmentation toolbox and benchmark. https://github.com/open-mmlab/mmsegmentation, 2020.

Maxime Oquab, Timothée Darcet, Theo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Russell Howes, Po-Yao Huang, Hu Xu, Vasu Sharma, Shang-Wen Li, Wojciech Galuba, Mike Rabbat, Mido Assran, Nicolas Ballas, Gabriel Synnaeve, Ishan Misra, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision, 2023.

Valentinos Pariza, Mohammadreza Salehi, and Yuki Asano. Hummingbird evaluation for vision encoders, 4 2024. URL https://github.com/vpariza/open-hummingbird-eval.

Zhiliang Peng, Wei Huang, Shanzhi Gu, Lingxi Xie, Yaowei Wang, Jianbin Jiao, and Qixiang Ye. Conformer: Local features coupling global representations for visual recognition. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 367–376, 2021.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pp. 8748–8763, 2021.

Maithra Raghu, Thomas Unterthiner, Simon Kornblith, Chiyuan Zhang, and Alexey Dosovitskiy. Do vision transformers see like convolutional neural networks? *Advances in neural information processing systems*, 34:12116–12128, 2021.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. doi: 10.1007/s11263-015-0816-y.

Tong Shao, Zhuotao Tian, Hang Zhao, and Jingyong Su. Explore the potential of clip for training-free open vocabulary semantic segmentation. In *European Conference on Computer Vision*. Springer, 2024.

Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. Self-attention with relative position representations. In Marilyn Walker, Heng Ji, and Amanda Stent (eds.), *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, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-2074.

Leslie N. Smith and Nicholay Topin. Super-convergence: Very fast training of residual networks using large learning rates, 2018.

Vladislav Sovrasov. ptflops: a flops counting tool for neural networks in pytorch framework. https://github.com/sovrasov/flops-counter.pytorch, 2018-2024.

Jianlin Su, Yu Lu, Shengfeng Pan, Bo Wen, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding, 2021.

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

Hugo Touvron, Matthieu Cord, Alexandre Sablayrolles, Gabriel Synnaeve, and Hervé Jégou. Going deeper with image transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 32–42, 2021b.

Zhengzhong Tu, Hossein Talebi, Han Zhang, Feng Yang, Peyman Milanfar, Alan Bovik, and Yinxiao Li. Maxvit: Multi-axis vision transformer. In *European conference on computer vision*, pp. 459–479. Springer, 2022.

Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. *Advances in neural information processing systems*, 29, 2016.

Feng Wang, Jieru Mei, and Alan Yuille. Sclip: Rethinking self-attention for dense vision-language inference. *European Conference on Computer Vision*, 2024.

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.

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

Haiping Wu, Bin Xiao, Noel Codella, Mengchen Liu, Xiyang Dai, Lu Yuan, and Lei Zhang. Cvt: Introducing convolutions to vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 22–31, 2021a.

Kan Wu, Houwen Peng, Minghao Chen, Jianlong Fu, and Hongyang Chao. Rethinking and improving relative position encoding for vision transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 10033–10041, 2021b.

Zhuofan Xia, Xuran Pan, Shiji Song, Li Erran Li, and Gao Huang. Dat++: Spatially dynamic vision transformer with deformable attention. *arXiv preprint arXiv:2309.01430*, 2023.

Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 418–434, 2018.

Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. *Advances in neural information processing systems*, 34:12077–12090, 2021.

Mengde Xu, Zheng Zhang, Fangyun Wei, Han Hu, and Xiang Bai. San: Side adapter network for open-vocabulary semantic segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

Jianwei Yang, Chunyuan Li, Pengchuan Zhang, Xiyang Dai, Bin Xiao, Lu Yuan, and Jianfeng Gao. Focal self-attention for local-global interactions in vision transformers, 2021.

Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 6023–6032, 2019.

Xiaohua Zhai, Alexander Kolesnikov, Neil Houlsby, and Lucas Beyer. Scaling vision transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 12104–12113, 2022.

Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*, 2018.

Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 13001–13008, 2020.

Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ade20k dataset. *International Journal of Computer Vision*, 2019.

Chong Zhou, Chen Change Loy, and Bo Dai. Extract free dense labels from clip. In *European Conference on Computer Vision*, 2022.

Lei Zhu, Xinjiang Wang, Zhanghan Ke, Wayne Zhang, and Rynson WH Lau. Biformer: Vision transformer with bi-level routing attention. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10323–10333, 2023.

LOCALITY-ATTENDING VISION TRANSFORMER APPENDIX

A TECHNICAL DETAILS

A.1 CODE

Our code is anonymously and publicly available at `https://anonymous.4open.science/r/LocAtViTRepo/`. The `README.md` file provides guidelines on how to set up the environment, train the models, and perform different evaluations. For ViT (Dosovitskiy et al., 2021), Swin Transformer (Liu et al., 2021), RegViT (Darcet et al., 2024), and RoPEViT (Heo et al., 2024), we used the implementation provided by Wightman (2019), and for Jumbo (Fuller et al., 2025) we used their official repository. All of these models are reproduced. Jumbo is a new work the repository is incomplete, hence, we used the available code and implemented some of the components based on the paper.

A.2 COMPUTE RESOURCES

Our experiments were conducted using NVIDIA RTX A6000 48GB, V100 32GB, and A100 40GB GPUs. The Tiny, Small, and Base backbones of LocAtViT require 15GB, 29GB, and 29GB of GPU memory with a local batch size of 512, 512, and 256, respectively.

A.3 LLM USAGE

We used LLMs to aid or polish writing. Adhering to ICLR's author guideline, we include additional information here. We used LLMs to generate codes for plotting figures, tables, and other code or LaTeX related issues. We also used LLMs to improve the writing, polish, or shorten the paragraphs, while double checking the output.

B LOCATVIT COMPARISON WITH RELATED WORK

In Tab. 2, we included five baseline methods and implemented LocAt for each. Table 7 compares LocAtViT to multiple related works from Sec. 2: CvT-21 (Wu et al., 2021a), Conformer (Peng et al., 2021), ConViT (d'Ascoli et al., 2021), Twins (Chu et al., 2023; 2021), DaViT (Ding et al., 2022), and GCViT (Hatamizadeh et al., 2023). We utilized the publicly available code and checkpoints, and evaluated the models on our segmentation pipeline, as described in Sec. 5. Although LocAtViT does not achieve the best classification performance, LocAt helps ViT outperform methods like Twins across all three segmentation benchmarks.

Table 7: **Segmentation and classification performance** of the Base backbone of related works and the proposed LocAtViT.

Method	Segmentation mIoU (%)			Top-1 (%)
	ADE	P-Context	C-Stuff	ImageNet
CvT-21	21.40	40.91	29.29	82.50
Conformer	22.11	40.03	26.37	83.83
ConViT	23.08	44.82	25.20	82.30
Twins	30.47	44.55	32.27	82.71
DaViT	30.68	44.87	32.38	**84.64**
GCViT	30.91	44.71	32.77	84.47
LocAtViT	**32.64**	**45.35**	**33.62**	82.31

C ADDITIONAL QUALITATIVE EXPERIMENTS

Figure 4 provides three additional images from the mini-ImageNet dataset, alongside the attention maps of the [CLS] token and several patches for ViT and LocAtViT.

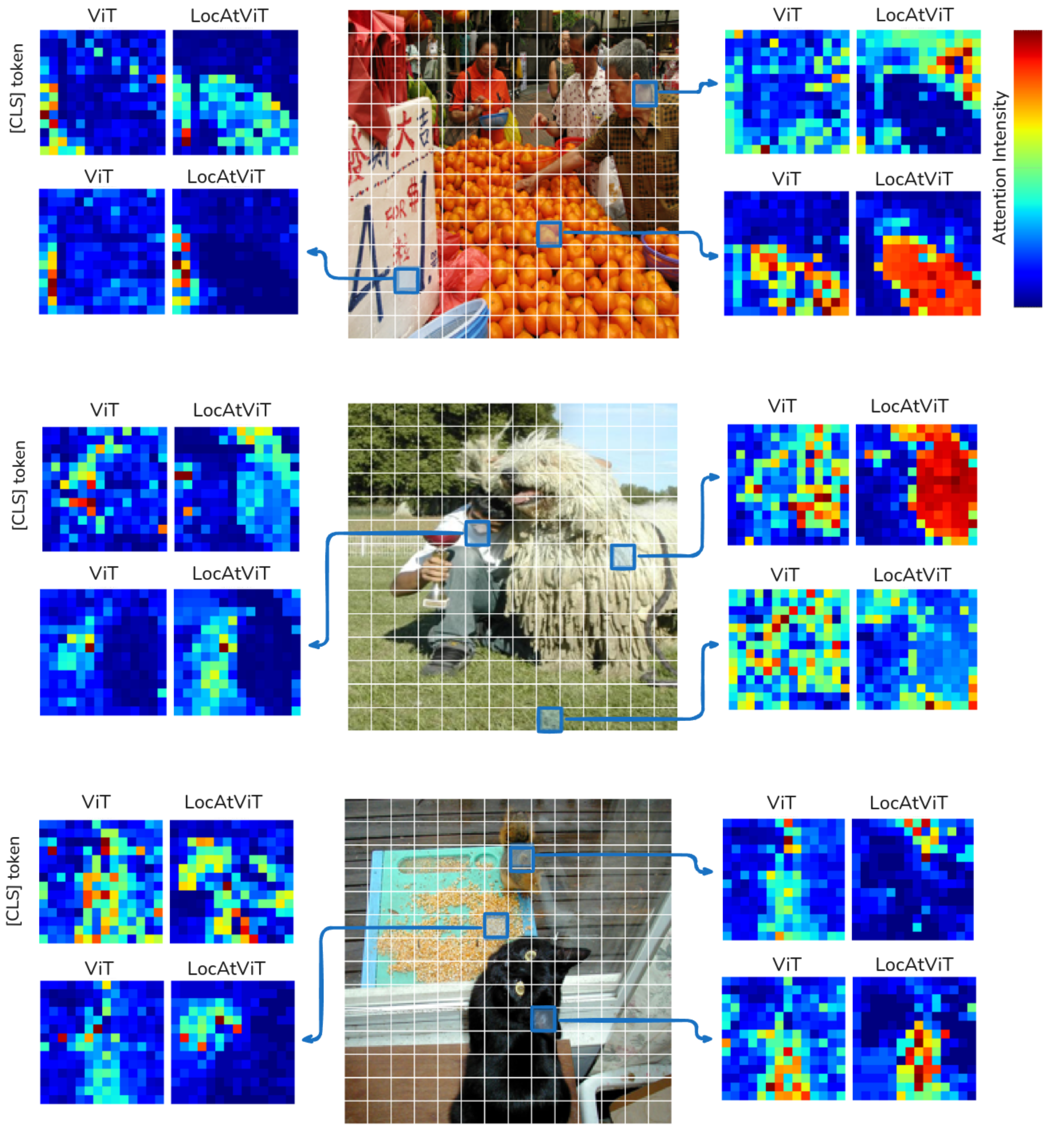


Figure 4: **Qualitative evaluation on the attention maps.** The final attention map of ViT and LocAtViT for the [CLS] token and three different patches are illustrated for three different images from mini-ImageNet with labels: *orange*, *Komondor*, and *corn*.

D ABLATION STUDY ON SELF-ATTENTION

In this section, we perform ablations on the design choices inside the GAug self-attention module.

D.1 GAUSSIAN BASED ON INPUT

In the original ViT, a query vector intuitively determines the information a patch should be looking for. Since the Gaussian variance controls how far a patch attends to its surroundings, we compute $\mathbf{\Sigma}$ based on the query matrix in Eq. (5). Table 8 compares this approach to computing $\mathbf{\Sigma}$ based on $\mathbf{x}$,

Table 8: **Ablations on GAug attention components**. Δ*#Params* shows the difference in the number of the parameters of each model compared to LocAtViT (first row). Experiments are conducted on mini-ImageNet.

		Tiny	Base	Δ#Params
LocAtViT (Sec. 4)		78.47	84.86	-
Gaussian from $\mathbf{x}$		79.10	85.18	$+18,504$, $+329,868$
Isotropic Gaussian		78.71	84.66	-780
Fixed width	$\sigma = 1$	75.20	82.81	$-2,340$
	$\sigma = 5$	76.41	82.65	$-2,340$
	$\sigma = 10$	75.53	82.42	$-2,340$
No scaling		76.26	83.07	-780
Auto $\boldsymbol{\alpha}$		78.48	84.54	-780

the self-attention input. While the latter improves performance, it significantly increases the number of parameters.

D.2 Variance matrix

To comply with a more general setting, we assigned separate variances for each image axis. An alternative is to use a single variance per patch, forming an isotropic Gaussian kernel. This simplifies Eq. (8) to:

$$\mathbf{G}_{pt} = \exp\Big(-\frac{\sum_{m=1}^{2} \mathbf{D}_{ptm}}{2\sigma_p^2}\Big). \tag{12}$$

The result of this modification is referred to as *Isotropic Gaussian* in Tab. 8. This table also compares this approach with another experiment where the Gaussian kernel width is fixed different constant values, instead of being patch-specific and query-based. These results indicate that an isotropic Gaussian kernel performs comparably, but a fixed kernel width substantially diminishes performance, demonstrating the importance of our dynamic input-dependent kernel width.

D.3 No supplement matrix scaling

In Sec. 4.1, we introduced a learnable scaling vector $\boldsymbol{\alpha}$ to match the scale of the supplement matrix $\mathbf{S}$ to that of the attention logits. To isolate its effect, Tab. 8 reports a variant (*No* α) in which the supplement matrix in Eq. (10) is not scaled, *i.e.*, we set $\mathbf{S} = \mathbf{G}$ (equivalently, $\boldsymbol{\alpha} = \mathbf{1}$) and directly add the raw Gaussian kernel to the logits. This no-scaling configuration corresponds to a harder use of the locality term and consistently reduces accuracy, confirming that unscaled addition of G is suboptimal and that the learnable scaling is important for balancing global attention with the Gaussian prior.

D.4 Automatic scaling of the supplement matrix

As mentioned, we motivated the need for scaling the supplement matrix before adding it to the attention logits in Sec. 4.1. We now propose a parameter-free, input-dependent scheme, *Auto* $\boldsymbol{\alpha}$, that automatically matches the scale of $\mathbf{S}$ to that of the original attention logits. Concretely, let $N = 1 + hw$, $\mathbf{q}, \mathbf{k} \in \mathbb{R}^{N \times d}$, and define the row-wise ℓ_2-norm vectors:

$$\mathbf{r} = \big[\|\mathbf{q}_1\|_2, \dots, \|\mathbf{q}_n\|_2\big]^\top, \tag{13}$$

$$\mathbf{u} = \big[\|\mathbf{k}_1\|_2, \dots, \|\mathbf{k}_n\|_2\big]^\top. \tag{14}$$

Then the standard attention logits satisfy:

$$\frac{\mathbf{q}\mathbf{k}^\top}{\sqrt{d}} = \Big(\frac{\mathbf{r}\mathbf{u}^\top}{\sqrt{d}}\Big) \circ \cos\big(\mathbf{q}, \mathbf{k}\big), \tag{15}$$

where $\circ$ denotes the Hadamard product, and $\cos(\mathbf{q}, \mathbf{k}) \in \mathbb{R}^{N \times N}$ has entries $\cos(\mathbf{q}_i, \mathbf{k}_j)$. Hence, if we set

$$\boldsymbol{\alpha} = \frac{\mathbf{r}\mathbf{u}^\top}{\sqrt{d}} \in \mathbb{R}^{N \times N}, \tag{16}$$

then the modified logits in Eq. (4) can be rewritten as

$$\frac{\mathbf{q}\,\mathbf{k}^\top}{\sqrt{d}} + \mathbf{S} = \boldsymbol{\alpha} \circ \big(\cos(\mathbf{q}, \mathbf{k}) + \mathbf{G}\big), \tag{17}$$

where both terms inside the parentheses are bounded (in $[-1, 1]$ and $[0, 1]$, respectively), ensuring that $\mathbf{S}$ scales comparably to the original logits.

However, using $\boldsymbol{\alpha} \circ \mathbf{G}$ would independently scale each entry of $\mathbf{G}$, destroying the Gaussian kernel structure (each row of $\mathbf{G}$ is a kernel centered at one patch). To preserve each kernel's shape, we average $\boldsymbol{\alpha}$ across columns:

$$\bar{\alpha}_i \;=\; \frac{1}{N} \sum_{j=1}^{N} \boldsymbol{\alpha}_{ij}, \quad \bar{\boldsymbol{\alpha}} = [\bar{\alpha}_1, \ldots, \bar{\alpha}_n]^\top \in \mathbb{R}^N, \tag{18}$$

and then form:

$$\mathbf{S} \;=\; \operatorname{diag}(\bar{\boldsymbol{\alpha}})\, \mathbf{G}, \tag{19}$$

similar to Eq. (10). This row-wise scaling applies a single factor to each Gaussian kernel, preserving its shape while matching its magnitude to the attention logits.

Auto $\boldsymbol{\alpha}$ performs close to learnable $\boldsymbol{\alpha}$ in the original LocAtViT, with slightly fewer parameters. We nevertheless keep the learnable $\boldsymbol{\alpha}$ in our main model for simplicity of formulation and to give the network maximal flexibility in attenuating or amplifying locality where beneficial.

E Qualitative comparison of locality mechanisms in ViT architectures.

In Table 1 we provided a summarized qualitative comparison of locality mechanisms in ViT architectures, which highlights the benefits of the proposed method. In this section, we provide further details on the following properties considered in the table: **easily applicable on ViT architecture**, and **query-adaptive locality**.

Easily applicable on ViT architectures. *Convolution-based hybrids* are not easily applicable, since they require convolutional stems or intermediate convolutional stages that alter the backbone design. *Local window or block attention* also needs architectural changes, such as window partitioning and shifted windows, which makes them less straightforward to integrate into a standard ViT. *Positional encodings*, by contrast, are trivially applicable, as they can be added to the attention mechanism without modifying the backbone. Our *Gaussian-Augmented attention* is similarly easy to apply, since it simply adds a Gaussian bias term to the attention logits and does not require structural changes.

Query-adaptive locality. *Convolution-based hybrids* do not provide query-adaptive locality, as convolutional kernels are fixed after training and shared across spatial positions. *Local window or block attention* offers partial adaptivity, where attention weights are content-based, but restricted to a fixed window, so queries cannot flexibly extend beyond that boundary. *Positional encodings* are not query-adaptive, since they impose a static positional bias that does not depend on the query. In contrast, our *Gaussian-Augmented attention* is fully query-adaptive: Gaussian parameters are predicted from each query, allowing the locality radius and decay to vary dynamically depending on the query content.

F Ablation study on alternative distance-based kernels

In the main paper we model locality with a Gaussian kernel added to the attention logits (Sec. 4.1). The choice of a Gaussian is motivated by the desire for a smooth, distance-based attenuation function with a scale parameter that controls the effective receptive field, and that can be predicted from

Table 9: **Effect of different distance-based attenuation kernels.** Segmentation performance (mIoU %) over three benchmarks and classification accuracy (top-1 %) on ImageNet-1K are reported.

Kernel	Tiny				Base			
	ADE	P-Context	C-Stuff	ImageNet	ADE	P-Context	C-Stuff	ImageNet
No (ViT)	17.30	33.71	20.29	72.39	28.40	43.10	30.43	80.99
Gaussian	23.47	38.57	26.15	73.94	32.64	45.35	33.62	82.31
Inv-dist	22.18	38.16	25.25	74.00	28.42	43.48	30.82	81.94
Laplace	21.67	37.80	25.56	74.01	29.74	44.10	31.95	82.24

each query token. Nevertheless, other monotone distance-based kernels are also reasonable, and we compare with two other kernels in what follows.

Let $r_{pt} = \|P_p - P_t\|_2$ denote the Euclidean distance between patches p and t in the spatial grid. We construct two alternative kernels by predicting scale parameters γ and λ from the queries:

$$L_{pt} = \exp\left(-\gamma_p \, r_{pt}\right), \tag{20}$$

denoting the Laplace kernel, and the inverse-distance kernel denoted as:

$$I_{pt} = \frac{1}{1 + r_{pt}/\lambda_p}. \tag{21}$$

In both cases, the resulting kernel matrix replaces G in Eq. (10), and the rest of the GAug formulation (including the scaling with $\boldsymbol{\alpha}$) is kept unchanged.

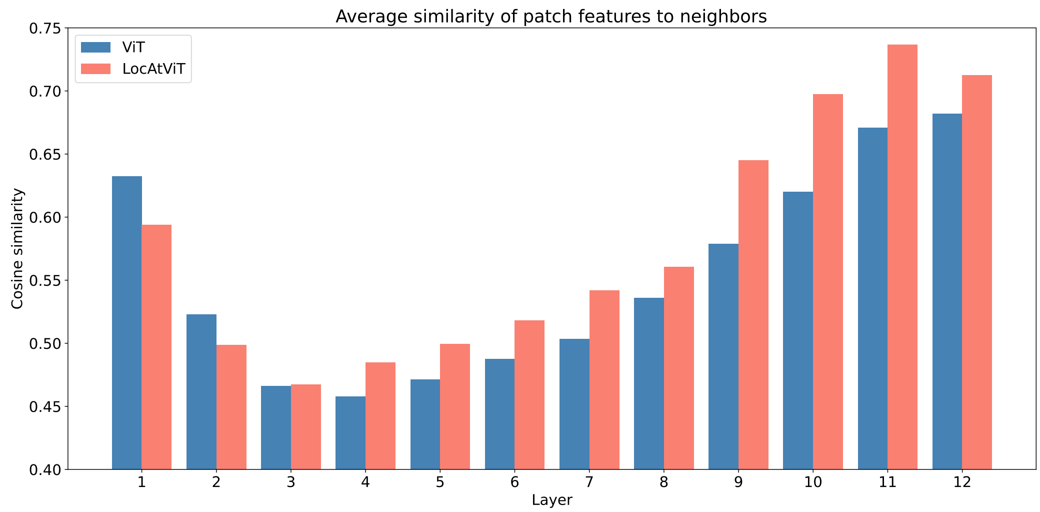
Table 9 compares performance of different choices of the kernel. All three locality-augmented variants improve over the baseline ViT, confirming that introducing a smooth distance-based prior is beneficial. Among them, the Gaussian kernel delivers the strongest segmentation gains on all three benchmarks, while remaining competitive in ImageNet-1K accuracy compared to the Laplace and inverse-distance kernels. This supports the choice of a Gaussian kernel in the main LocAtViT model as a simple yet effective way to inject adaptive locality into attention.

G Local feature analysis across layers

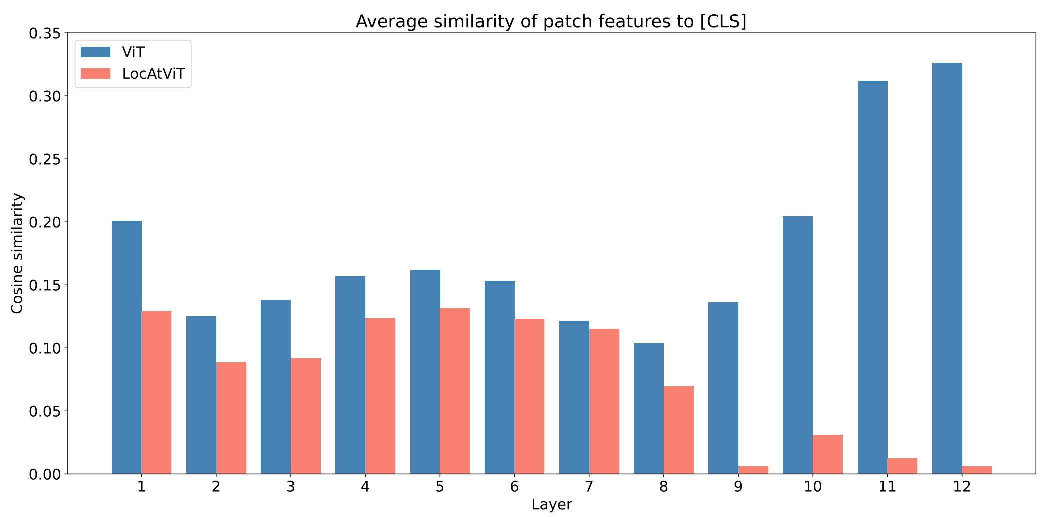
In the main paper, we argue that the global attention mechanism of vanilla ViT tends to obscure fine-grained local information that is important for dense prediction. Here, we provide a quantitative analysis of how local patch features evolve across layers in a standard ViT and in our LocAtViT. We focus on Base models of ViT and LocAtViT trained on ImageNet-1K and evaluate features on the ImageNet-1K validation set.

Locality score. For each layer l and each spatial patch token, we compute a locality score defined as the cosine similarity between that patch and its 8 immediate neighbors in the surrounding 3×3 window. We then average this score over all spatial locations and all validation images. Intuitively, a higher locality score indicates that nearby patches share more similar representations, which is desirable as long as representations do not collapse globally. Figure 5a reports this locality score per layer. After the third layer, LocAtViT consistently achieves a higher locality score than vanilla ViT, indicating that its patch features remain more coherent with their spatial neighbors as depth increases.

Patch-`[CLS]` similarity. High neighbor similarity alone does not guarantee that meaningful local structure is preserved: if all patch tokens collapse to the same global representation, their mutual similarity (including to neighbors) will also be high. To distinguish this degenerate case from genuine locality, we additionally measure, for each layer l, the cosine similarity between every patch token and the `[CLS]` token, again averaged over all patches and validation images. Figure 5b shows that in vanilla ViT this patch-`[CLS]` similarity steadily increases with depth and peaks in the final layers, revealing a progressive pull of patch features toward a shared global representation dominated by the `[CLS]` token. In contrast, LocAtViT maintains substantially lower patch-`[CLS]` similarity across layers, while still achieving a higher locality score.



(a) Average cosine similarity to neighbors.



(b) Average cosine similarity to `[CLS]`.

Figure 5: **Degradation of local features in vanilla ViT.** Features in ViT collapse to the global information in the last layers while in LocAtViT, patch features encode local information.

Discussion. Taken together, these two measurements show that, in vanilla ViT, patch tokens gradually lose distinct local information and become dominated by global `[CLS]`-like content as depth grows. LocAtViT, on the other hand, preserves strong locality in patch features without collapsing them onto the `[CLS]` token. This behavior aligns with our design goal: to enhance the preservation of local structure while retaining the benefits of global attention, thereby producing representations that are better suited for dense prediction.

H COMPARISON WITH CLASSIFIER-SIDE REFINEMENT METHODS

We compared PRR to GAP in Sec. 5.4. To further isolate the classifier-side refinement, we compare PRR against several standard pooling heads: max pooling, AvgMax (average + max pooling), and multihead attention pooling (MAP). Table 10 reports the performance of these pooling mechanisms with comparable capacity with ViT Tiny backbone size (6M). PRR achieves the best performance across segmentation and classification among the pooling heads, indicating the effectiveness of explicitly refining patch representations.

Table 10: **Comparison of PRR with standard pooling heads.** on a Tiny backbone. We report segmentation performance (mIoU %) over three benchmarks and classification accuracy (top-1 %) on ImageNet-1K.

Pooling	ADE	P-Context	C-Stuff	ImageNet
GAP	19.7	34.9	22.9	72.5
Max	19.2	34.7	23.3	71.9
AvgMax	20.1	35.6	24.2	72.3
MAP	20.2	36.3	23.1	73.0
PRR	21.6	37.9	25.9	73.7

I Comparison to CaiT and class-attention

CaiT (Touvron et al., 2021b) introduces dedicated class-attention layers in the last blocks that process only the class token, while patch tokens remain fixed, and are primarily designed to stabilize optimization and improve classification in very deep transformers. Table 11 compares ViT+PRR Tiny and Base to CaiT backbone of similar size in both classification and segmentation. For a fair comparison, CaiT models are trained with the same data, augmentations, and optimization settings, and we evaluate all models in our segmentation pipeline as described in Sec. 5. In this setting, PRR consistently outperforms CaiT on both ImageNet-1K classification and all three segmentation benchmarks. We attribute this gap to a structural difference in how the final layers are used. In CaiT, the last class-attention blocks exclusively update the class token while keeping patch tokens fixed, so a fraction of the backbone capacity is devoted solely to refining a shallow class embedding. In our ViT+PRR design, all blocks maintain full self-attention among patch and class tokens, and PRR then applies one additional parameter-free self-attention over all tokens. This symmetric use of capacity allows the final layers to refine patch features and the class token jointly, which appears better aligned with the demands of dense prediction while remaining superior in classification.

Table 11: Comparison between PRR and CaiT at similar parameter budgets (*i.e.*, 6M and 86M parameters for Tiny and Base). We report mIoU (%) on three segmentation benchmarks and top-1 accuracy (%) on ImageNet-1K.

Method	Tiny				Base			
	ADE	P-Context	C-Stuff	ImageNet	ADE	P-Context	C-Stuff	ImageNet
ViT+PRR	21.6	37.9	25.9	73.7	29.9	44.0	32.2	82.2
CaiT	16.9	30.2	18.7	69.6	27.8	41.9	30.1	79.1

J Full fine-tuning on ADE20K

The segmentation results in Tab. 2 use a simple MLP decoder on top of a frozen backbone in order to keep the head lightweight and make performance primarily reflect the backbone representations. We adopt this pipeline deliberately to isolate the effect of the backbone representations: a strong decoder and long full fine-tuning can partially mask differences between pretraining strategies. To check that the gains of LocAt also hold under a standard segmentation protocol, we additionally attach a UperNet (Xiao et al., 2018) decoder to ViT and LocAtViT and fine-tune *all* parameters on ADE20K for 50K iterations. As shown in Tab. 12, LocAt improves mIoU over ViT in both Tiny and Base backbone sizes, confirming that the locality bias introduced by GAug and PRR yields more effective representations even when the entire network is trained end-to-end for segmentation.

Table 12: **End-to-end ADE20K fine-tuning with a standard UperNet decoder.** We report mIoU (%) after fine-tuning all parameters for 50K iterations.

	Tiny	Base
ViT	35.7	41.2
ViT + LocAt	**36.9**	**45.2**

K COCO DETECTION AND INSTANCE SEGMENTATION

To further assess the generality of LocAt beyond semantic segmentation, we conduct experiments on COCO 2017 object detection and instance segmentation using a Mask R-CNN (He et al., 2017) head. We use Swin Tiny backbone and evaluate two training regimes: *(i)* full end-to-end fine-tuning (FT), and *(ii)* frozen-backbone training where only the detection head is updated. Both settings follow a standard 1× schedule. Table 13 reports bounding-box AP and mask AP. LocAt improves performance in both FT and frozen settings, confirming that the locality-preserving representations produced by GAug and PRR benefit spatial localization tasks as well.

Table 13: **COCO 2017 object detection and instance segmentation** using a Mask R-CNN head. We report bounding-box AP (AP^b) and mask AP (AP^m) under both full fine-tuning (FT) and frozen-backbone settings.

		AP^b	AP^b_{50}	AP^b_{75}	AP^m	AP^m_{50}	AP^m_{75}
FT	Swin	42.3	65.0	46.0	38.9	61.9	41.7
	Swin + LocAt	42.8	65.4	46.7	39.3	62.5	42.0
Frozen	Swin	28.9	52.9	28.1	29.3	50.7	30.3
	Swin + LocAt	29.7	54.2	28.7	30.0	51.6	30.9

L STABILITY OF LEARNED STANDARD DEVIATIONS

The per-patch Gaussian variances are predicted from the queries through a bounded nonlinearity in Eq. (5), ensuring numerical stability; however, in principle these values could collapse to the lower or upper end of the admissible range. Figure 6 analyzes the mean and percentile ranges of the learned standard deviations across layers for a LocAtViT Base model trained on ImageNet-1K. We find that the predicted variances remain well inside the allowed interval and do not cluster near the bounds. Instead, they form non-trivial depth-dependent patterns: early layers tend to use narrower kernels, whereas deeper layers gradually broaden their spatial extent. These observations indicate that GAug learns meaningful locality scales rather than degenerately switching the Gaussian bias "off" (very small variance) or "fully on" (maximal variance) everywhere.

M HIGHER RESOLUTION OVERHEAD

We quantify the overhead introduced by LocAt when moving from the common 224×224 training resolution to a higher 512×512 setting, and report in Tab. 14, wall-clock time and peak GPU memory usage for one epoch of training on mini-ImageNet using a single A100 GPU with batch size 16.

N FAILURE MODES AND LIMITATIONS OF THE GAUSSIAN BIAS

Our design goal for the Gaussian augmentation is to gently bias attention toward local structure, rather than to hard-enforce locality. Empirically, across the backbones and tasks reported in the main paper, we observe performance gain when adding GAug and PRR. However, the magnitude of the gains depends on the underlying attention topology. The largest improvements appear on backbones with unrestricted patch-patch attention (*e.g.*, ViT, RegViT, RoPEViT, Jumbo), whereas

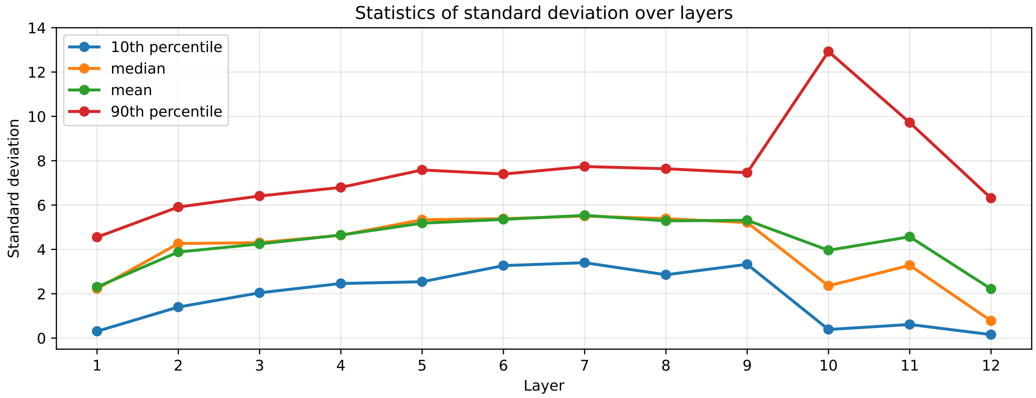


Figure 6: Standard deviation of GAug's Gaussian over layers

Table 14: Training wall-clock (minutes per epoch) and peak memory usage (GB) on mini-ImageNet for Tiny and Base backbones under different input resolutions. Batch size is 16 and all experiments use a single A100 GPU.

Size	Image side	Wall-clock time (m)	Memory (GB)
Tiny	224	2.0	1.2
	512	6.1	6.7
Base	224	3.1	4.1
	512	21.1	25.0

the gains on a windowed-attention backbone such as Swin are noticeably smaller. This suggests that GAug is most effective when attention is globally connected and locality is not already hard-coded by the architecture.

To further probe this limitation, we also applied our approach on top of GCViT (Hatamizadeh et al., 2023), a stronger windowed-attention model with attention confined to small grids. In this setting we did not obtain improvements in the performance. We attribute this negative result to the fact that when attention is restricted to narrow windows, the additional Gaussian bias has little room to meaningfully reshape the locality pattern. In contrast, even for powerful unrestricted-attention models such as Jumbo, there remains enough flexibility for GAug and PRR to provide noticeable benefits.