

PPS: Plug-and-Play Saccadic Vision for Fine-Grained Classification

Johann Schmidt¹
johann.schmidt@ovgu.de
Sebastian Stober¹
stober@ovgu.de
Joachim Denzler²
joachim.denzler@uni-jena.de
Paul Bodesheim²
paul.bodesheim@uni-jena.de

¹ Otto von Guericke University
Universitaetsplatz 2
39106 Magdeburg
Germany
² Friedrich Schiller University
Fuerstengraben 1
07737 Jena
Germany

Abstract

Fine-grained visual classification benefits from high-resolution inputs to resolve subtle, localized cues, but at the cost of large compute and memory budgets. Inspired by human saccadic vision, we propose a lightweight extension that turns any pretrained backbone with spatial feature maps into a saccadic classifier through glimpse-induced inference. The only architectural additions are a small priority head and a single cross-attention fusion block. The same weight-tied encoder encodes a downsampled peripheral glance, exposes the spatial feature map from which the priority head predicts a glimpse-priority distribution and re-encodes high-resolution glimpse crops sampled from that distribution in parallel. A sequential sampler with non-maximum suppression draws a runtime-configurable number of glimpses. A residual multi-head cross-attention block then fuses the glance against the glimpses. To improve the quality of the priority maps, we train the priority head by a per-glimpse information-gain target from a lightweight per-image memory bank. This requires no reinforcement learning. Complexity is linear in the number of glimpses and peak memory is decoupled from input resolution. Across nine benchmarks our framework improves consistently over its backbones.

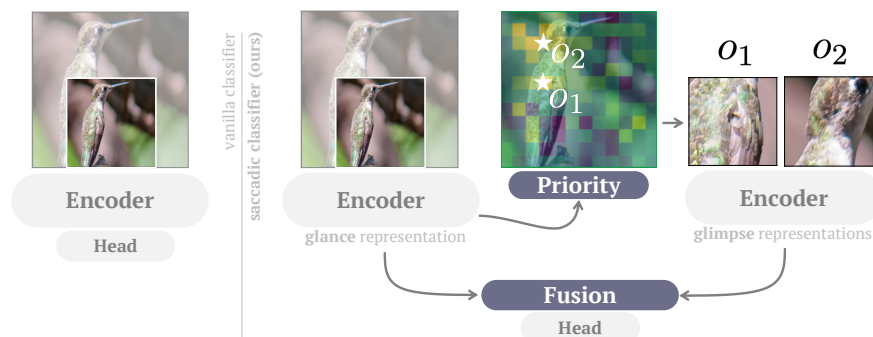


Figure 1: We reuse the backbone encoder to capture a glance, generate a priority map and encode sampled glimpses. Glance and glimpse representations are fused for prediction.

1 Introduction

Fine-grained visual classification (FGVC) has emerged as a critical area of research within computer vision, aiming to distinguish between visually very similar categories, such as different bird species [48, 51] or insect breeds [25, 28, 42]. Unlike traditional image classification, FGVC focuses on identifying subtle, local patterns and nuanced variations across categories, which is crucial for applications like biodiversity monitoring [28]. Despite its significance, FGVC remains challenging due to inherent complexities: (i) data is often sparse, (ii) categories exhibit high intra-class variability (pose, lighting, occlusion), (iii) inter-class differences are typically subtle and localized. While (i) and (ii) are tackled by large pre-trained backbones, (iii) requires models to process high-resolution signals to capture minute visual details. To keep the emerging costs low, model reuse is essential. Recent methods take inspiration from human visual perception, which is highly efficient and effective. Instead of uniformly scanning a scene, the human visual system employs rapid eye movements, known as *saccades* [57], interspersed with brief, high-acuity *glimpses* (fixations) [23]. This visual search process is not random but guided by various sources of pre-attentive information that coalesce to form a spatial *priority map* [23, 52]. This map dynamically directs selective attention to the most salient locations. Once attention is focused, the high-acuity central vision (fovea) extracts fine-grained details, while the rest of the scene is processed via peripheral vision at a lower resolution (the *glance*) [26]. This allows for capturing a high-resolution image using multiple low-resolution windows [1]. In computer vision, this strategy is realized by glimpse based models, which form a prediction from a few localized views instead of a glance alone [13, 14, 20, 36]. Because the prediction rests on the extracted views, these models stay interpretable and faithful [21, 50]. Yet three design choices recur across them and limit reuse. Most attach a dedicated localization network to the recognition backbone [14, 20, 56]. The glimpse count is fixed at architecture time [7, 14, 56]. And placing or aggregating views relies on reinforcement learning (RL) or sequential encoding [13, 20, 36].

Our Contributions We introduce the *Plug-and-Play Saccader* (PPS)¹, a saccadic extension that grants any vision classifier high resolution glimpses at a cost set by the glance resolution and a small glimpse budget rather than the input resolution. One weight tied encoder encodes a low resolution peripheral glance, reuses the resulting feature map as a spatial priority map, and encodes the sampled glimpse patches. This removes the separate localization network and the memory overhead of prior glimpse models [13, 14, 20]. Our contributions are:

- PPS wraps any backbone that emits spatial feature maps, convolutional or transformer, including those built on sparse token subsets. The shared encoder processes the glance and all glimpses alike, so input resolution and glimpse count become runtime settings rather than architectural commitments.
- Glimpses are read directly from the encoder’s own feature map and supervised by an auxiliary priority loss, leaving training fully end to end without RL.
- A single cross-attention block aggregates all glimpses and fuses them with the glance in one pass. Glimpses are thus processed in parallel rather than sequentially. Across nine FGVC benchmarks and three backbones, PPS improves consistently over the vanilla backbone.

¹Our source code is available at github.com/johSchm/Saccader.

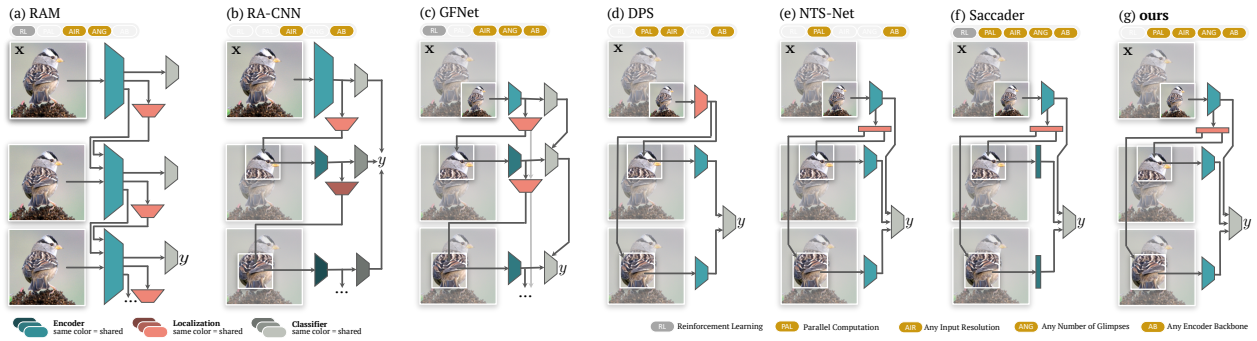


Figure 2: Visual side-by-side comparison of related glimpse-based approaches covering RAM [36], RA-CNN [14], GFNet [20], DPS [7], NTS-Net [56], Saccader [13] and our saccadic vision extension. The sharing of encoders, additional localization networks for glimpse localization, and the classifier are highlighted by different colors. Tags are used to highlight training based on reinforcement learning, whether glimpse encoding is parallelized (or sequential) and whether the approach is agnostic to the input resolution, the number of glimpses or the choice of backbone.

2 Related Work

Glimpse-based vision models extend a classifier with a mechanism that selects a small set of localized views and aggregates evidence from them, rather than processing the image uniformly. RAM [36] casts this as a reinforcement-learning problem in which a recurrent controller emits glimpse locations and is rewarded by the final classification accuracy. RA-CNN [14] replaces the controller with a stack of attention proposal networks that recursively zoom into a single discriminative region, NTS-Net [56] introduces a separate navigator whose proposals are ranked by a teacher classifier, and Saccader [13] returns to an explicitly saccadic policy trained with reinforcement learning. GFNet [20] and DPS [7] reduce compute by selecting glimpses from coarse feature maps, but still attach a dedicated localization or routing network on top of the recognition backbone. As summarized in Fig. 2, the recurring design commitments across this body of work are an additional localization network, a fixed number of glimpses, and either reinforcement learning or sequential glimpse encoding.

Our framework removes these commitments: glimpse priorities are read directly from the encoder’s own feature map with a single weight-tied backbone, all glimpses are processed in parallel, the input resolution and the number of fixations are runtime-configurable, and training is fully end-to-end without reinforcement learning. Within FGVC more broadly, part-based feature localization can be grouped into two complementary regimes: *Single-Patch Part* (SPP) methods, which extract rectangular crops from the source image, and *Multi-Patch Part* (MPP) methods, which extract sparse token subsets from an encoded feature map. Our approach falls into the SPP family but is compatible with both convolutional and transformer backbones (including MPP encoders).

SPP methods. SPP methods have their roots in object detection [15, 16, 18, 32, 40, 40, 41], where two-stage detectors process individual region proposals at high accuracy but with significant computational overhead, while single-stage detectors combine proposal and prediction in a single forward pass for improved efficiency. In FGVC, similar techniques are used to extract object parts [21], but no ground-truth part annotations are available. Glimpses (part crops) are therefore learned implicitly, guided by the classification loss. To impose

some structure on the extracted parts, several frameworks introduce semantically clustered parts [3, 50]. Part-Stacked CNNs [21] adopt a single-stage framework to simultaneously generate region proposals and integrate their encodings for classification, while Dongliang et al. [3] learn multi-grained parts using top-down spatial attention.

To locate glimpses, many frameworks take inspiration from region proposal networks [41] and either regress locations and ratios without inductive biases [36] or read out activation peaks on spatial location maps [13, 14, 20]. Either a fixed number of glimpses are extracted [7, 14, 17, 56, 60] or recurrent mechanisms sample a variable number [13, 20, 36]. Variable-count sampling typically requires reinforcement learning [13, 20, 36], which renders training challenging. Many frameworks also rely on encoders tailored to a specific sub-task such as feature extraction for the peripheral view, for glimpses, or for locations, complicating the use of off-the-shelf backbones [21, 36]. Because these sub-tasks extract very similar features, the resulting encoders are highly redundant; this slows down learning and increases computational demand. This motivated per-encoder losses to ensure consistent and robust learning signals [59] and weight-tied encoders [7, 54, 56]. A critical aspect of these methods is the choice of context representation, which can rely on high-resolution images [28] or downsampled views [21]; computational demand grows quickly when the entire high-resolution input must be encoded [14, 21, 36]. Finally, glimpse features are fused with peripheral features through recurrent encoding [20, 36], concatenation [14, 21], multi-scale cross-attention between parts [54], or graph attention modelling explicit spatial relationships [46].

MPP methods. MPP methods encode the image as a whole and interpret the resulting feature maps as hierarchies of part maps, potentially sparsified by thresholds. While this view applies to ConvNets, it is most prominent in vision transformers [12, 47], where the global attention mechanism naturally yields sparse part maps. To remove background noise, improve efficiency, and further sparsify the representation, token sparsification (token dropout or pruning) is commonly used; this process corresponds to localization in SPP methods. TransFG [17] uses accumulated attention scores for token dropout. DynamicViT [39] performs dynamic token pruning that creates spatially coherent token selections, which inherently form part maps. A-ViT [58] introduces a halting probability that enables adaptive token selection, where high-probability tokens correspond to discriminative parts. ACC-ViT [61] mixes high-attention tokens from different classes, effectively isolating class-specific part regions and excluding background noise. Evo-ViT [55] uses a slow-fast token evolution mechanism that dynamically separates informative from redundant tokens to accelerate inference. FAL-ViT [22] applies coarse-to-fine token masking to create hierarchical part representations, progressing from global object structure to fine-grained details. PIM [5] uses a graph neural network to model intra- and inter-stage relationships between the top- k preserved tokens. HERBS [6] retains only top- k spatial features at different network stages, with higher temperatures in early layers to encourage exploration while providing guidance.

3 Methodology

Let $(\mathbf{x}, y) \sim \mathcal{D}$ represent a sampled tuple from a dataset $\mathcal{D}$, where $\mathbf{x} \in \mathbb{R}^{C \times H \times W}$ is an image and $y \in \mathbb{N}$ is its corresponding class label. $C \in \mathbb{N} \setminus \{0\}$ is the number of channels (we use $C = 3$ for RGB images) and $H, W \in \mathbb{N} \setminus \{0\}$ denote the spatial resolution of the pixel grid. To obtain a glance, $\mathbf{x}$ is downsampled to $\mathbf{x}_L \in \mathbb{R}^{C \times H_L \times W_L}$ with $H_L \ll H$ and $W_L \ll W$. Let

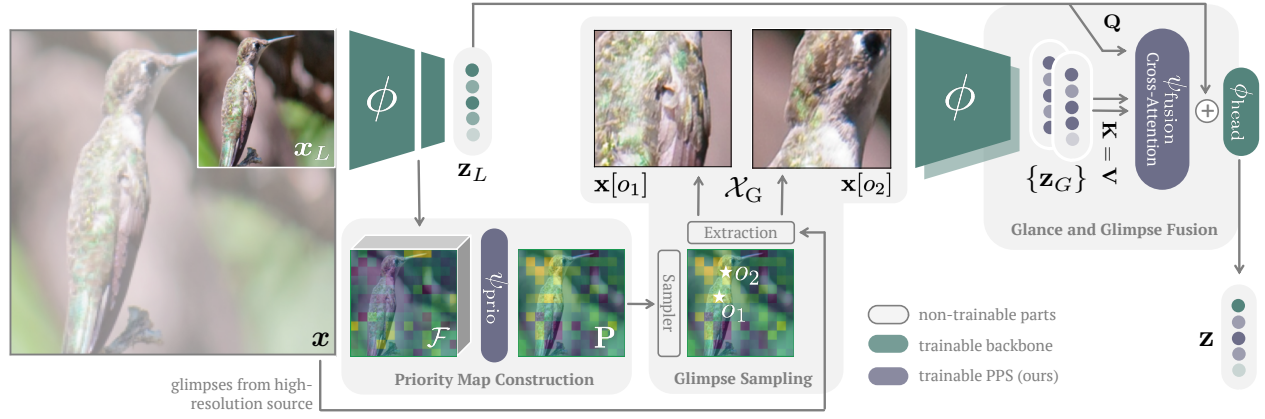


Figure 3: **Overview of PPS.** A glance $\mathbf{x}_L$ is obtained by downsampling the input $\mathbf{x}$. The backbone ϕ encodes it into spatial feature maps $\mathcal{F}$ and a glance embedding $\mathbf{z}_L$. A lightweight priority head ψ_{prio} turns $\mathcal{F}$ into a priority map $\mathbf{P}$, read as a spatial distribution over locations. From $\mathbf{P}$ we sample N glimpse locations $\mathcal{O}$ and crop the corresponding patches $\mathcal{X}_G$ from the high resolution source $\mathbf{x}$. The same (weight-tied) backbone ϕ encodes each patch into a glimpse embedding, giving $\{\mathbf{z}_G\}$. A cross-attention block ψ_{fusion} fuses the glimpse set with the glance $\mathbf{z}_L$, and the head ϕ_{head} maps the result to class logits $\mathbf{z}$.

$(\phi, \phi_{\text{head}})$ be a vision backbone with encoder $\phi : \mathbb{R}^{C \times H_L \times W_L} \rightarrow \mathbb{R}^D$ and head $\phi_{\text{head}} : \mathbb{R}^D \rightarrow \mathbb{R}^Y$, where $D \in \mathbb{N} \setminus \{0\}$ is embedding dimension and $Y \in \mathbb{N} \setminus \{0\}$ is the number of labels. Let $\mathcal{F} = \{\mathbf{F}^{(s)}\}_{s=1}^S$ denote the encoder’s spatial feature maps over S encoding stages on the glance view $\mathbf{x}_L$. Our proposed framework equips any vision backbone with the ability to leverage higher resolution inputs with only very little additional parameters. The process is illustrated in Fig. 3 and discussed in the following.

3.1 Constructing Priority Maps

SPP and MPP methods rely on a map that marks where to look, or which tokens deserve attention. A common approach trains a dedicated proposal network [14, 20], sometimes conditioned on the raw input [7]. These networks are large and must learn the mapping from scratch. Following [13, 56], we reuse the backbone’s own representations and learn only a shallow head on top. The intuition: features used to *recognize* an object already encode where to *look*. We retain only the final map $\mathbf{F}^{(S)} \in \mathbb{R}^{C^{(S)} \times H_P \times W_P} \in \mathcal{F}$, whose channels carry the most discriminative responses. Aggregating one head per stage $s \in [1, \dots, S]$ did not improve accuracy and computation costs. A shallow convolutional head $\psi_{\text{prio}} : \mathbb{R}^{C^{(S)} \times H_P \times W_P} \rightarrow \mathbb{R}^{H_P \times W_P}$ projects $\mathbf{F}^{(S)}$ to a scalar priority map $\mathbf{P} \in \mathbb{R}^{H_P \times W_P}$. This generalizes the fixed Gaussian smoothing of Huang et al. [21] to a learned channel reweighting. The head adds negligible parameters, since $\mathbf{F}^{(S)}$ is already computed by the underlying vision backbone. As $\mathbf{F}^{(S)}$ is only trained for classification purposes and $\mathbf{P}$ is conditioned on it, $\mathbf{P}$ shows no accumulation at borders (sign of spatial collapses). This removes the need for an additional edge mask as in Huang et al. [20]. We support our claims empirically in Sec. 4.

3.2 Sampling Glimpses from the Priority Map

Direct coordinate regression [24], as used in prior work [13, 14, 20], does not encode spatial structure and fails to generalize [37]. We instead exploit the spatial structure of $\mathbf{P}$ and draw



Require: prior map $\mathbf{P}$, number of glimpses N

Ensure: glimpse locations $\mathcal{O} \subset [0, 1]^2$ with $|\mathcal{O}| = N$

```

1: Init:  $\mathcal{O} \leftarrow \emptyset$ 
2:  $\Omega_{ij} \leftarrow ((j + \frac{1}{2})/W, (i + \frac{1}{2})/H)$  // normal. centered grid
3: for  $n = 1$  to  $N$  do
4:    $k \leftarrow \arg \max_{H_P W_P} (\text{vec}(\mathbf{P}))$ 
5:    $\mathbf{o} \leftarrow \Omega_{\lfloor k/W \rfloor, k \bmod W}$  // index position on 2D grid
6:    $K_{ij} \leftarrow \exp(-\|\Omega_{ij} - \mathbf{o}\|_2^2 / (2\sigma^2))$  // std. deviation  $\sigma$ 
7:    $\mathbf{P} \leftarrow \mathbf{P} - \lambda \mathbf{K}$  // additive  $\lambda$ -weighted penalty
8:    $\mathcal{O} \leftarrow \mathcal{O} \cup \{\mathbf{o}\}$  // add the estimated glimpse location
9: end for
10: return  $\mathcal{O}$ 

```

Figure 4: The priority progression over N glimpses using our NMS with Gaussian kernels on $\mathbf{P}$. Left an example progression for two different resolutions of $\mathbf{P}$ and on the right the pseudo-code of our sampling method.

N glimpse locations $\mathcal{O} = \{\mathbf{o} \mid \mathbf{o} \sim_N \mathbf{P}\}$ from it. Unlike prior work [14, 20, 56] that fixes the number of glimpses architecturally, our sampler treats N as a runtime hyperparameter. Sampling budgets thus change without retraining or any architectural modification. We denote the train- and test-time counts by $N_{\text{train}}, N_{\text{test}} \in \mathbb{N}$. Setting $N = 0$ recovers the vanilla backbone forward pass, so our PPS can be disabled whenever walltime outweighs accuracy.

Locations are expressed in normalized, centered coordinates $2\mathbf{o} - 1 \in [-1, 1]^2$, decoupling regression from the input resolution $H \times W$. Pixel coordinates follow as $[(\mathbf{o} + 1)/2] \odot [H, W]^\top$. This lets us estimate locations directly on the low-resolution feature map $\mathbf{F} \in \mathbb{R}^{C \times H_P \times W_P}$ with $H_P, W_P \ll H, W$, at its native resolution. No upsampling is required: the gradient signal stays concentrated on each cell of $\mathbf{F}$ rather than being diffused across pixels. Continuous, pixel-precise localization [7] is unnecessary for FGVC and only adds compute (see Sec. 4.2). A coarse prior map also stabilizes training, as the smaller discrete location set reduces variance in the sampler’s output. To avoid redundant samples in adjacent regions, we apply non-maximum suppression (NMS) during sampling. Rather than discrete IoU-based suppression on detections [56], we apply sequential Gaussian soft-suppression on $\mathbf{P}$. A smooth radial kernel masks already-selected neighborhoods, with no anchors or box parameters required. The only hyperparameters are the kernel variance (we use $\sigma = 0.04$) and the update weight λ (we use $\lambda = 1e^{-4}$). Locations are selected greedily by $\arg \max$ [7, 56]. Fig. 4 gives the complete procedure with an example of suppression progression. Given locations $\mathcal{O}$, we extract a fixed-size glimpse of resolution $H_L \times W_L$ at each by a differentiable affine crop [24]. Sampling uses bicubic interpolation with (zero) border padding, yielding crops of identical resolution independent of the source size $H \times W$. This decouples the glimpse resolution from the input resolution. The result are a set of glimpses $\mathcal{X}_G = \{\mathbf{x}^n\}_{n=1}^N$, which are encoded next.

3.3 Fusing Glance and Glimpse Representations

We reuse the same backbone encoder $\phi : \mathbb{R}^{C \times H_L \times W_L} \rightarrow \mathbb{R}^D$, where D denotes the embedding dimension, for the glance and glimpse encoding. This is applied to each glimpse in $\mathcal{X}_G$, producing a set of glimpse embeddings $\{\mathbf{z}_G^n\}_{n=1}^N$ with $\mathbf{z}_G^n \in \mathbb{R}^D$. Furthermore, we have the glance embedding $\mathbf{z}_L \in \mathbb{R}^D$. The goal is to update the glance features by the glimpse features by fusion of these embeddings vectors. The informativeness of the two pathways is strongly

instance-dependent. Some target objects fill most of the canvas, so that the localized glimpses are largely redundant with the glance context. Others are small and off-center, so that the discriminative evidence resides in a few glimpses while the remaining ones capture uninformative background. A fusion rule with a fixed [13] or globally predicted mixing weight cannot represent this. We therefore replace static weighting by a *residual cross-attention* block $\psi_{\text{fusion}} : (N+1) \cdot \mathbb{R}^D \rightarrow \mathbb{R}^D$ in which the glance representation queries the glimpses, and the mixing coefficients are produced per glimpse, per attention head, and conditioned on the glance content itself. Aggregation over the N glimpses and fusion with the glance are thereby performed jointly by a single operator. We treat the glance embedding as a single query token and the N glimpse embeddings as the keys and values of a multi-head attention layer. Let h denote one of H heads with $d = D/H$. Both streams are first layer-normalized with independent statistics and linearly projected

$$\mathbf{q}^{(h)} = \mathbf{W}_Q^{(h)} \text{LN}_q(\mathbf{z}_L), \quad \mathbf{k}_n^{(h)} = \mathbf{W}_K^{(h)} \text{LN}_{kv}(\mathbf{z}_G^n), \quad \mathbf{v}_n^{(h)} = \mathbf{W}_V^{(h)} \text{LN}_{kv}(\mathbf{z}_G^n), \quad (1)$$

with $\mathbf{W}_Q^{(h)}, \mathbf{W}_K^{(h)}, \mathbf{W}_V^{(h)} \in \mathbb{R}^{d \times D}$. The attention distribution over glimpses for head h is the scaled dot-product softmax

$$a_n^{(h)} = \frac{\exp(\langle \mathbf{q}^{(h)}, \mathbf{k}_n^{(h)} \rangle / \sqrt{d})}{\sum_{m=1}^N \exp(\langle \mathbf{q}^{(h)}, \mathbf{k}_m^{(h)} \rangle / \sqrt{d})}, \quad a_n^{(h)} \geq 0, \quad \sum_{n=1}^N a_n^{(h)} = 1. \quad (2)$$

Each head returns the induced convex combination of glimpse values. The head outputs are concatenated and linearly mixed, and a further output projection yields the glimpse summary

$$\mathbf{o}^{(h)} = \sum_{n=1}^N a_n^{(h)} \mathbf{v}_n^{(h)}, \quad \hat{\mathbf{z}}_G = \mathbf{W}_g [\mathbf{W}_O \text{concat}(\mathbf{o}^{(1)}, \dots, \mathbf{o}^{(H)})], \quad (3)$$

where $\mathbf{W}_O \in \mathbb{R}^{D \times D}$ is the projection internal to the attention layer and $\mathbf{W}_g \in \mathbb{R}^{D \times D}$ is an additional learned output projection (followed, during training, by dropout on $\hat{\mathbf{z}}_G$). The fused representation is formed by a residual connection that adds the glimpse summary onto the *normalized* glance, which is then encoded by the linear classification head, such that

$$\mathbf{z} = \phi_{\text{head}}(\text{LN}_{\text{fusion}}(\hat{\mathbf{z}}_L + \hat{\mathbf{z}}_G)) = \phi_{\text{head}}(\text{LN}_{\text{fusion}}(\text{LN}_q(\mathbf{z}_L) + \hat{\mathbf{z}}_G)), \quad (4)$$

where $\phi_{\text{head}} : \mathbb{R}^D \rightarrow \mathbb{R}^Y$ is the head of the backbone projecting to the class logits. The residual is essential as our PPS first has to learn where to look, so glimpses are not very useful early in training. When no glimpses are taken ($N = 0$) the fusion block is bypassed and the model reduces to a pure-glance classifier (resembling the vanilla backbone pass).

3.4 Learning to Focus

Our model is trained end-to-end. The downstream task supplies a per-example loss $\mathcal{L}_{\text{task}}(\mathbf{z}, y)$ on the fused logits $\mathbf{z}$ of Eq. (4). For classification this is the categorical cross-entropy against the label y . However, the `argmax` used to select glimpse locations in Sec. 3.2 is non-differentiable², so no gradient of $\mathcal{L}_{\text{task}}$ reaches the priority map $\mathbf{P}$. Prior work [13, 20, 36] closes this gap with reinforcement learning, whose policy-gradient estimators are high-variance, slow to converge, and reliant on hand-tuned baselines. We instead detach the

²In Sec. 4.2 we test with a Gumbel softmax, rendering PPS fully differentiable but also reducing performance.

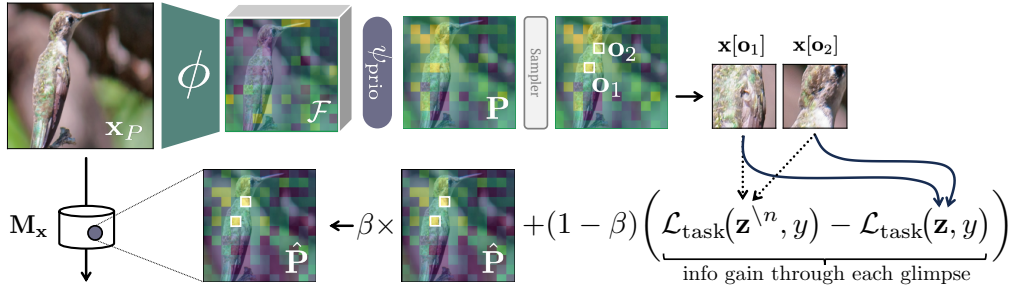


Figure 5: The training process of PPS with focus on our auxiliary priority loss.

sampler from the backward graph and supervise $\mathbf{P}$ with an auxiliary priority loss $\mathcal{L}_{\text{prio}}$. The full objective is

$$\mathcal{L} = \mathcal{L}_{\text{task}}(\mathbf{z}, y) + \lambda_{\text{prio}} \mathcal{L}_{\text{prio}}, \quad (5)$$

with $\lambda_{\text{prio}} > 0$ a balancing scalar, which we set to $\lambda_{\text{prio}} = 1$. The procedure is illustrated in Fig. 5 and discussed in the following.

Glimpse Utility. We quantify the contribution of each glimpse by a leave-one-out (LOO) ablation in logit space. This is computed under exponential-moving-average (EMA) teacher parameters for the classification head and the fusion ($\bar{\theta}_{\text{head}}, \bar{\theta}_{\text{fusion}}$). Let $\mathbf{z}^{\setminus n}$ denote the fused logits from Eq. (4) with glimpse n masked out of the cross-attention keys and values. The utility of glimpse n is the resulting loss change,

$$\Delta \mathcal{L}_n = \mathcal{L}_{\text{task}}(\mathbf{z}^{\setminus n}, y) - \mathcal{L}_{\text{task}}(\mathbf{z}, y). \quad (6)$$

A positive $\Delta \mathcal{L}_n$ marks a glimpse that carried discriminative evidence. A near-zero value marks a redundant or background crop. $\Delta \mathcal{L}_n$ is the pointwise information gain that glimpse n contributes to the teacher’s log-posterior over y , so the priority loss aligns $\mathbf{P}$ with expected per-glimpse informativeness. All $N+1$ passes share the encoder embeddings and run as a single batched call on the lightweight fusion and head subnets, so the overhead is negligible. The teacher is refreshed after each step by $\bar{\theta}_{\text{head}} \leftarrow m\bar{\theta}_{\text{head}} + (1-m)\theta_{\text{head}}$ ³ (and likewise for $\bar{\theta}_{\text{fusion}}$), which insulates the targets from optimizer noise while tracking the evolving feature space. Furthermore, backbones without registers, such as Swin [33], can emit stray high-norm activations [8]. Anchoring $\mathbf{P}$ to measured glimpse utility rather than raw feature magnitude removes the issue without backbone-specific fixes.

Priority Memory Bank. A single step observes only N of the $H_P W_P$ cells, so per-step utilities are too sparse to supervise $\mathbf{P}$ directly. We accumulate them in a persistent bank $\mathbf{M} \in \mathbb{R}^{|\mathcal{D}_{\text{train}}| \times H_P \times W_P}$. With $\hat{\mathbf{o}} = \lfloor [H_P, W_P] \odot \mathbf{o} \rfloor$ denoting the cell index, each visit updates the bank by

$$\mathbf{M}_{\mathbf{x}}[\hat{\mathbf{o}}_n] \leftarrow \beta \mathbf{M}_{\mathbf{x}}[\hat{\mathbf{o}}_n] + (1-\beta) \Delta \mathcal{L}_n, \quad (7)$$

with $\beta \in [0, 1)$, where $\beta = 0$ is used on the first visit. $\mathbf{M}$ is initialized with zeros and updated over training time. Repeated visits across epochs yield a temporally smoothed utility field over the full grid.

³Here, $m \in (0, 1)$ is a EMA momentum term, which we set to $m = 0.99$ for all our experiments.

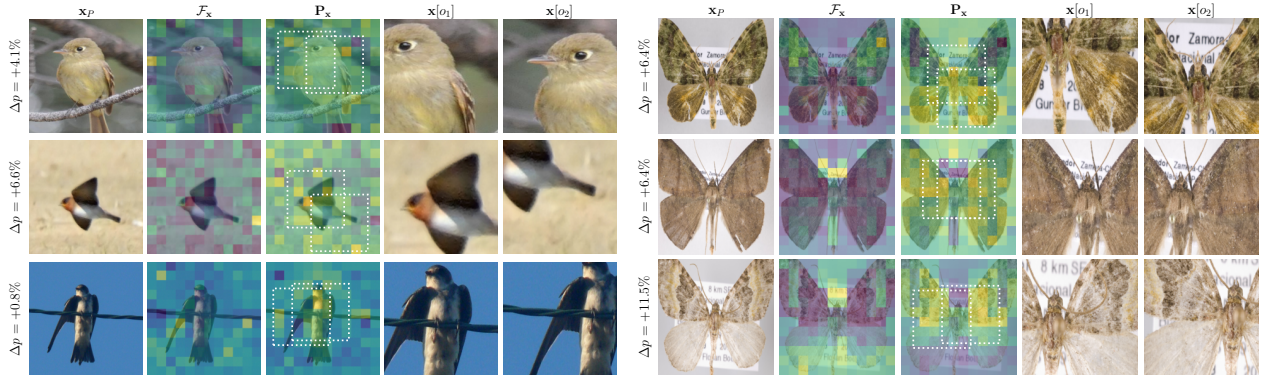


Figure 6: Test samples from NABirds and ECU with results obtained by our PPS-MPSA-SwinB. From left to right the columns show the input image $\mathbf{x}_L$, MPSA’s multi-granularity feature map (generally $\mathcal{F}$), our priority map $\mathbf{P}$, and two glimpses at $\mathbf{o}_1$ and $\mathbf{o}_2$. Annotated on the left is the prediction delta induced by the glimpses compared to the glance.

Table 1: Average top-1 test accuracy on FGVC moth benchmarks.

Method	Backbone	Pretr.	Res.	ECU	EU	WE	NA	CA
Swin-B [33]	-	Im22k	224	75.41	97.06	94.03	97.57	98.98
MPSA [50]	SwinB	Im22k	224	77.04	97.64	94.17	97.68	99.07
PPS-MPSA (ours)	SwinB	Im22k	224	80.89	98.18	94.44	97.76	99.08
SPPs-Cond. [29]	Inc.V3	Im1k	299	-	91.50	-	-	-
SPPs-Cond. [29]	Inc.V3	iNat	299	-	93.13	-	-	-
Swin-B [33]	-	Im22k	384	77.19	97.10	94.84	97.95	99.02
MPSA [50]	SwinB	Im22k	384	80.30	98.36	94.78	97.94	99.09
PPS-MPSA (ours)	SwinB	Im22k	384	83.41	98.55	94.96	98.09	99.08

Priority Loss. We turn the priority logits $\mathbf{P}$ and the bank $\mathbf{M}$ into distributions over the $H_P W_P$ cells and align them by Kullback–Leibler divergence,

$$\mathcal{L}_{\text{prio}} = \mathbb{E}_{\mathbf{x} \in \mathcal{D}_{\text{train}}} \left[D_{\text{KL}} \left(\text{softmax}(\text{vec} \mathbf{M}) \parallel \text{softmax}(\text{vec} \mathbf{P}) \right) \right], \quad (8)$$

with the target $\mathbf{M}$ detached (stopped gradient). The softmax discards the instance-dependent scale and offset of $\mathbf{M}$ and supervises only the relative ranking of locations. A peaked target pulls $\mathbf{P}$ toward a confident, selective distribution. $\mathbf{P}$ learns to concentrate probability mass on the locations the LOO measurement identifies as informative. If $\mathcal{L}_{\text{prio}}$ would reach the encoder it would pronounce shortcut learning as the network reshaped its features to make the saliency easier to fit rather than learning where to look. We therefore stop the gradient at $\mathbf{F}$ that conditions $\mathbf{P}$.⁴ In this setup, $\mathcal{L}_{\text{task}}$ trains the encoder, the fusion, and the head, while $\mathcal{L}_{\text{prio}}$ trains the priority encoder alone. Hence, the two terms in Eq. (5) never compete for the same parameters.

4 Experiments

⁴A joint backbone objective would force the encoder to recognize *what* the object is and shape *where* the model looks at once, which we found to destabilize training.

Backbones and Pretraining PPS is backbone-agnostic and applies to any vision encoder that exposes a spatial feature map, including both convolutional and transformer architectures. All reported experiments use a ConvNextv2-Base (short CNB) [53], a Swin Base [33], a MPSA encoder [50] (32 heads, parts ratio 2) with a Swin Base backbone. Our models are pretrained on ImageNet [9] (iNaturalist [49] pretraining usually increase performance even more). We finetuned the ViT, SwinB, MPSA, and FET using their code and hyperparameters on several datasets. Even under these settings, we are not able to exactly replicate all claimed performances and hence report our achieved values here.

Hyperparameters We follow the training recipe from [50] with adaptations detailed. The peripheral input is downsampled to $H_F = W_F = 7$ grid cells for MPSA and $H_F = W_F = 14$ for SwinB. The resolution of glimpses is specified but generally 224×224 or 384×384 (specified in each experiment) using a crop scale of 25%. The fusion block uses 32-head residual cross-attention. The priority memory bank uses decay $\beta = 0.99$. Optimization uses AdamW [34] except for NABirds, CUB200, and Food101

where we used plain SGD with cosine learning-rate schedule. We use 5 warmup epochs, gradient clipping at norm 10, label smoothing 0.1 and 0.001 weight decay. Learning rates are dataset specific and vary between $6e^{-2}$ and $8e^{-5}$, and early stopping. If not otherwise stated we use 512px source images, which we downsample (bilinear interpolation) to the input size of the vision backbone. We extract 2 glimpses during training and 4 during testing of the same input size from the high resolution source image. This choice follows from findings in [7, 54], which we reduced during training to decrease walltime costs. We use standard augmentations comprising random cropping and light Gaussian blurring ($p = 0.1$). Because the random crop offset is symmetric and zero-mean, accumulating utilities in crop-frame cells smooths each image’s bank entry by at most one cell on MPSA’s 7×7 grid (sub-cell in expectation) without biasing its location, and the deterministic center crop at test time matches this expected frame. Training runs for 50–100 epochs and mini-batches depending on dataset size on A40 and V100 GPUs. We use a frozen seed = 42. For full reproducibility of our findings we make our code, configurations, and checkpoints publicly available.

Datasets We evaluate PPS on moth, bird, dog, and food benchmarks, spanning a wide

Table 2: Average top-1 test accuracy on CUB.

	Method	Backbone	Pretr.	Acc@1
224 px	RA-CNN [14]	VGG19	Im1k	85.3
	TASN [62]	ResNet50	Im1k	87.0
	NTSN [56]	ResNet50	Im1k	87.3
	MPSA [50]	SwinB	Im22k	91.4
	PPS-MPSA (ours)	SwinB	Im22k	91.5
384 px	CNB [53]	-	Im22k	90.2
	PPS-CNB (ours)	-	Im22k	90.5
	MetaFormer2 [11]	-	Im22k	91.8
	MPSA [50]	SwinB	Im22k	92.6
	PPS-MPSA (ours)	SwinB	Im22k	92.7
448 px	AFTrans [60]	ViT-B16	Im1k	91.5
	TransFG [17]	ViT-B16	Im1k	91.7
	FAL-ViT [22]	ViT-B16	Im1k	91.7
	ACC-ViT [61]	ViT-B16	Im1k	91.8
	FET [4]	SwinB	Im22k	92.9

Table 3: Avg. top-1 test accuracy on NABirds.

	Method	Backbone	Pretr.	Acc@1
224 px	MetaFormer2 [11]	-	Im1k	89.7
	MPSA [50]	SwinB	Im22k	90.2
	PPS-MPSA (ours)	SwinB	Im22k	90.4
384 px	MetaFormer2 [11]	-	Im22k	92.2
	MPSA [50]	SwinB	Im22k	92.4
	PPS-MPSA (ours)	SwinB	Im22k	92.5
448 px	CrossX [35]	ViT-B16	Im1k	88.9
	TransFG [17]	ViT-B16	Im1k	90.8
	FAL-ViT [22]	ViT-B16	Im1k	91.1
	ACC-ViT [61]	ViT-B16	Im1k	91.4

range of object scales and background conditions. This includes five moth FGVC benchmarks: EU [28] (1650 images, 200 species), ECU [42] (1445 images, 675 species) with significant center bias, and AMI-GBIF [25] subsets⁵ for North America (NA) (203k images, 2408 species), Central America (CA) (18k images, 551 species) and Western Europe (33k images, 2533 species) with a significantly higher variability and noise rate. EU and ECU feature relatively clean backgrounds, ideal for evaluating fixation mechanisms without background interference. Additionally, we evaluated our model on four established FGVC benchmarks with increasing background complexity: CUB200-2011 [51] (12k images, 200 bird species), NABirds [48] (49k images, 555 species), Stanford-Dogs [27] (21k images, 120 breeds) and Food101 [2] (101k images, 101 categories). These datasets progress from moderate natural backgrounds to highly complex real-world imaging conditions.

4.1 Fine-Grained Visual Classification

We found consistent improvement over the vanilla backbones with only very few exceptions. The margin is largest where small targets sit in cluttered scenes and where high intra-class variability rewards instance-adaptive attention, and smallest where the object already fills a clean, tightly cropped frame. On the moth benchmarks PPS-MPSA leads all baselines at 224px and at 384px with one exception (Tab. 1). On trails on the near-saturated AMI-CA with little remaining headroom and results remain withing run-to-run noise.

Two observations support our hypothesis. ECU backgrounds contain handwritten notes that invite overfitting, and the glimpses supply an additional self-augmentation path that reduces it in our experiments. Accuracy also rises on the noisier AMI subsets, indicating ro-

bustness to background clutter. Glimpses on ECU caused in 12.4% a positive class flip and in 7.3% a negative one. Examples are shown in Fig. 6, where Δp denotes the shift in the predicted class probability induced by the glimpses $\Delta p = p(y | \mathbf{z}) - p(y | \hat{\mathbf{z}}_L)$. A positive value indicates the glimpses reinforced the model’s decision. Repeating the EU experiment with PPS-SwinB over three seeds left the accuracy unchanged to the first decimal (std below 0.1), indicating the reported margins are stable rather than seed artifacts. CUB-200 presents tightly cropped birds with moderate clutter and little noise, so a downsampled glance already exposes the discriminative features and the gain from glimpses is small. We sample glimpses from 512px images, beyond which no further detail is available, and narrowly lead the best methods at 224 and 384px (Tab. 2). NABirds places small bird regions in complex, noisy

Table 4: Avg. top-1 test acc. on Stanford-Dogs. With [†] being PPS trained with $N_{\text{train}} = 4$.

	Method	Backbone	Pretr.	Acc@1
224 px	RA-CNN [14]	VGG-16	Im1k	87.3
	MPSA [50]	SwinB	Im22k	94.1
	PPS-MPSA (ours)	SwinB	Im22k	94.2
384 px	MPSA [50]	SwinB	Im22k	94.7
	PPS-MPSA (ours)	SwinB	Im22k	94.9
	PPS-MPSA [†] (ours)	SwinB	Im22k	95.1
448 px	TransFG [17]	ViT-B16	Im1k	92.9
	FAL-ViT [22]	ViT-B16	Im1k	91.1
	ACC-ViT [61]	ViT-B16	Im1k	92.3

Table 5: Avg. top-1 test acc. on Food-101.

Method	Backbone	Pretr.	Res.	Acc@1
FAL-ViT [22]	ViT-B16	Im1k	448	91.8
MPSA [50]	SwinB	Im22k	224	93.7
PPS-MPSA (ours)	SwinB	Im22k	224	94.0

⁵Due to link corruptions in the original AMI collection, we only included classes with ≥ 10 samples per class.

scenes with large scale and pose variation. Results are shown in Tab. 3, where the margin increased to 0.2 accuracy points for MPSA using 224px inputs. PPS suppresses background interference and recover fine detail when the target occupies a small fraction of the image. On Stanford Dogs and Food-101 the advantage instead stems from high intra-class variability. PPS-MPSA improves over vanilla MPSA by 0.3% on Food-101 and 0.4% with four glimpses on Dogs. The diverse sources, poses, lighting, and presentations in these sets reward an attention mechanism that adapts per instance and reads local geometry and texture. The ViT based MPP methods trail at 448px despite their larger inputs.

4.2 Ablation Study

Priority Map. We sample on $\mathbf{P}$ at its native feature resolution $H_P \times W_P$ rather than upsampling it to the image grid. Tab. 6 contrasts the two. Native sampling exceeds the upsampled variant on both backbones at strictly lower cost. Upsampling diffuses the gradient across pixels and adds compute with no accuracy return. The coarse grid keeps the signal concentrated on each cell of $\mathbf{F}^{(S)}$ and shrinks the discrete location set,

lowering sampler variance. Thus, pixel-precise localization is unnecessary for FGVC.

Sampling. Tab. 7 varies the sampler and its input map. Two findings stand out. First, the sampling source matters more than the sampler. On a plain SwinB, drawing locations from the learned priority map $\mathbf{P}$ beats drawing them from the raw feature map $\mathbf{F}$ for every sampler. MPSA shows the reverse: $\mathbf{F}$ already suffices (91.5) and $\mathbf{P}$ adds little (91.4). The reason is that MPSA’s CAM-condensed map [45, 50] is itself a strong saliency signal, so sampling on it needs no extra head.

The priority head supplies the same signal to any backbone that lacks one. Second, the edge mask of Huang et al. [20] is redundant under $\mathbf{P}$. On $\mathbf{F}$ the mask helps (90.6 vs. 90.0), as raw Swin activations accumulate at borders. On $\mathbf{P}$ it hurts (90.3 vs. 90.7): the learned head already avoids border accumulation, so masking only discards valid cells. This confirms the claim of Sec. 3.1. The sampler itself is secondary. Random, TopK, and NMS lie within 0.3 points, so the differences sit at run-to-run noise on saturated CUB. Random is a strong baseline, consistent with random-crop augmentation [30]. We also varied the number of test-time glimpses with results in Tab. 8. While the performance on EU is consistent (2 glimpses suffice for the well-aligned dataset), the performance increased with more glimpses on CUB. Similarly, we only found for Stanford Dogs an increase in performance when raising N_{train} see Tab. 4.

Table 6: Avg. top-1 test accuracy on CUB with a priority map and loss ablation.

	Method	SwinB	MPSA
$\mathbf{P}$	upsampled	90.4	91.4
	native (ours)	90.7	91.5
$\mathcal{L}$	REINFORCE [13]	90.5	91.4
	only $\mathcal{L}_{\text{task}}$ (Gumbel)	90.6	91.4
	Eq. (8) (ours)	90.7	91.5

Table 7: Avg. top-1 test accuracy on CUB using a SwinB with and without MPSA. The sampler’s input feature map and the sampler itself are varied.

Sampling	$\mathcal{F}$	$\mathbf{o} \sim \mathbf{F}$		$\mathbf{o} \sim \mathbf{P}$	
		SwinB	MPSA	SwinB	MPSA
Random	last	90.3	91.4	90.8	91.3
TopK	last	90.2	91.4	90.5	91.3
NMS	full	90.2	91.3	90.4	91.4
NMS	mask	90.6	91.4	90.3	91.3
NMS	last	90.0	91.5	90.7	91.4

Table 8: Avg. top-1 test acc. for MPSA with varying glimpses.

N_{test}	EU	CUB
2	98.2	91.4
4	98.2	91.5
8	98.2	91.5

Fusion and Augmentation. We isolate sampling against augmentation and ablate our fusion. Averaging is the standard aggregator for sampled patches. Cordonnier et al. [7] report that mean pooling of patch embeddings beats concatenation on CUB, attributing the gap to reduced classifier overfitting. We adopt this averaging as the fusion baseline. Tab. 9 compares our sampler against training-time data augmentation (DA) and test-time augmentation (TTA) [30], both of which crop randomly and average the resulting signals. Our sampler reaches the highest accuracy, which we argue stems from the contextual weighting in Eq. (4) (informed rather than random crops).

Loss. Tab. 6 compares three ways to train the priority map. REINFORCE [13] treats sampling as a policy and optimizes it by policy gradient. It is weakest (90.5), consistent with the high variance and baseline sensitivity of such estimators. Replacing the argmax with a Gumbel softmax makes the sampler differentiable, so $\mathcal{L}_{\text{task}}$ reaches $\mathbf{P}$ directly and no auxiliary term is needed. This removes the policy gradient but couples $\mathbf{P}$ to the task loss, inviting the encoder to re-

shape features for easy saliency rather than informative glimpses; it reaches 90.6. Our decoupled objective (Eq. (8)) supervises $\mathbf{P}$ with measured leave-one-out utility under a stop-gradient at $\mathbf{F}$, so the priority head and the backbone never compete. It is best on both backbones (90.7, 91.5). MPSA is near insensitive to this choice (91.4–91.5): its CAM features already localize, so how $\mathbf{P}$ is trained barely registers, and the loss matters mainly for a plain backbone. Margins are small on saturated CUB, but the ranking is consistent and our loss avoids both the variance of RL and the shortcut risk of end-to-end differentiation.

Table 9: Avg. top-1 test accuracies on EU between our PPS, (training-time) data augmentation (DA), test-time augmentation (TTA), and a combination of both. DA and TTA use random sampling of target object patches and signal averaging.

Method	Sampler	Fusion	N_{train}	N_{test}	MPSA
Vanilla	-	-	0	0	96.0
DA	Rnd	Avg	4	0	97.1
TTA	Rnd	Avg	0	4	96.0
DA+TTA	Rnd	Avg	4	4	96.4
PPS (ours)	NMS	Attn	4	4	98.0

4.3 Complexity

A forward pass comprises the glance encoding $\phi(\mathbf{x}_L)$, N glimpse encodings $\phi(\mathbf{x}_G^n)$, the location sampling from $\mathbf{P}$, and the cross-attention fusion. From these components, the encoding by the backbone dominates. Sampling and fusion are linear in N but act on D dimensional embeddings rather than the full feature tensor, so their cost is negligible against a backbone pass. The compute is therefore $(1+N)$ backbone passes, linear in the glimpse budget N , and $N=0$ recovers the plain backbone exactly. The $N+1$ encodings carry no inter-glimpse dependency and issue as a single batched call, which we exploit for throughput. The serialized glimpse policies of [20, 36] forbid this batching, since each glimpse conditions on the previous one. Relative to a single high

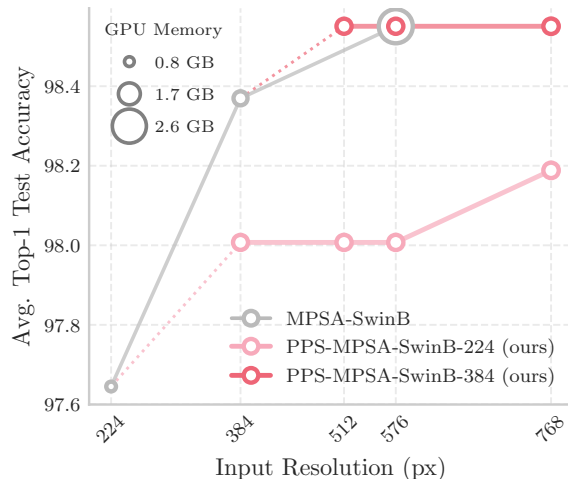


Figure 7: Performance and peak GPU memory over input resolutions on EU.

resolution pass we trade fewer pixels per pass for more passes, which are flexible to choose compared to [7, 14, 20]. On an A40 with a SwinB backbone the baseline reads a 384 px and 512 px input in 0.19 s and 0.53 s, while PPS ($N = 2$) processes a 512 px source image in 0.46 s, at 48.8, 116.1, and 146.4 GFLOPs, respectively. PPS is therefore not the cheapest option at every resolution. The benefit is memory rather than walltime. Peak memory is set by the glance resolution and the budget N , so it stays flat while a dense pass grows super-linearly. Methods that encode the full high resolution input for context [14, 21, 36] pay the dense pass our glance resolution avoids. Every encoder pass runs at $H_L \times W_L$, whereas a single high resolution pass scales with the full pixel grid $H \times W$. This decoupling is sharpest in memory. No full resolution activation map is ever materialized, so the peak memory of PPS stays essentially flat as the canvas grows, while the baseline rises super-linearly with resolution (Fig. 7). There is an intersection point (here at 576 px) of both curves. Below it a single low resolution pass is lighter than our $N+1$ passes, above it PPS is the cheaper model and the gap widens monotonically. At high resolution the saving is the central practical benefit, since it is exactly the regime where a dense backbone pass becomes memory bound or infeasible. The effect is strongest for backbones whose activation footprint grows with the input grid, convolutional networks in particular, which are moreover often widened to exploit higher resolution. The added parameters are fixed across resolutions but scale with the backbone embedding dimension D . The fusion projections are $D \times D$, so the count grows with wider backbones. For SwinB the priority head and fusion block add 18.4M parameters. A dedicated localization network [7, 14, 56] or recurrent controller [36] instead adds overhead that scales with the backbone, not a fixed shallow head. PPS thus decouples the memory cost of recognition from input resolution rather than trading it for a larger model.

5 Conclusion

We introduce the Plug and Play Saccader (PPS), a priority-guided sampling framework that equips any vision backbone with the ability to exploit high resolution inputs at the cost of the glance resolution and a small glimpse budget. A shallow head turns the backbone’s own spatial features into a priority map, from which we sample glimpse locations by sequential Gaussian non maximum suppression. The glimpses are cropped, encoded by the weight-tied backbone, and fused with the glance through a residual cross attention block that weights each glimpse per head and per instance. A decoupled auxiliary loss supervises the priority map with leave one out glimpse utility, avoiding both the variance of reinforcement learning and the shortcut risk of end-to-end differentiation. The design removes three constraints of prior attention guided methods: it needs no separate localization network, it treats the glimpse count as a runtime hyperparameter rather than an architectural choice, and it decouples memory cost from input resolution so peak memory stays essentially flat as the canvas grows. Across nine FGVC benchmarks, PPS improves consistently over its backbones. The added cost is a shallow priority head and a single fusion block fixed across resolutions.

Limitations and Future Work. Several avenues merit further investigation. The localization and extraction of glimpses can be read through the lens of affine transformations, which opens a perspective in which canonicalization [31, 43, 44] integrates naturally into saccadic vision pipelines to improve robustness. The additional backbone passes dominate the compute, so low rank approximations of these passes are a promising route to further reduce cost. Finally, the priority map offers a natural basis for sequential refinement through information theoretic view selection [10], where early glimpses inform later ones.

Acknowledgements

Johann Schmidt was funded by the ZIM (KK5540302MS4).

References

- [1] Dana H. Ballard. Animate vision. *Artificial Intelligence*, 1991. doi: 10.1016/0004-3702(91)90080-4.
- [2] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101 – mining discriminative components with random forests. In *European Conference on Computer Vision (ECCV)*, 2014.
- [3] Dongliang Chang, Yixiao Zheng, Zhanyu Ma, Ruoyi Du, and Kongming Liang. Fine-grained visual classification via simultaneously learning of multi-regional multi-grained features. In *ArXiv*, 2021.
- [4] Huazhen Chen, Haimiao Zhang, Chang Liu, Jianpeng An, Zhongke Gao, and Jun Qiu. Fet-fgvc: Feature-enhanced transformer for fine-grained visual classification. *Pattern Recognition*, 2024.
- [5] Po-Yung Chou, Chu-Hsing Lin, and Wen chung Kao. A novel plug-in module for fine-grained visual classification. *ArXiv*, 2022.
- [6] Po-Yung Chou, Yu-Yung Kao, and Cheng-Hung Lin. Fine-grained visual classification with high-temperature refinement and background suppression. *ArXiv*, 2023.
- [7] Jean-Baptiste Cordonnier, Aravindh Mahendran, Alexey Dosovitskiy, Dirk Weissenborn, Jakob Uszkoreit, and Thomas Unterthiner. Differentiable patch selection for image recognition. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.
- [8] Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. Vision transformers need registers. In *International Conference on Learning Representations (ICLR)*, 2024.
- [9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.
- [10] Joachim Denzler and C. Brown. Information-theoretic sensor data selection for active object recognition and state estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2002.
- [11] Qishuai Diao, Yi Jiang, Bin Wen, Jia Sun, and Zehuan Yuan. Metaformer: A unified meta framework for fine-grained recognition. *ArXiv*, 2022.
- [12] 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. *International Conference on Learning Representations (ICLR)*, 2021.

- [13] Gamaleldin F. Elsayed, Simon Kornblith, and Quoc V. Le. Saccader: improving accuracy of hard attention models for vision. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2019.
- [14] Jianlong Fu, Heliang Zheng, and Tao Mei. Look closer to see better: Recurrent attention convolutional neural network for fine-grained image recognition. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.
- [15] Ross Girshick. Fast r-cnn. *International Conference on Computer Vision (ICCV)*, 2015.
- [16] Ross B. Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.
- [17] Ju He, Jieneng Chen, Shuai Liu, Adam Kortylewski, Cheng Yang, Yutong Bai, Changhu Wang, and Alan Loddon Yuille. Transfg: A transformer architecture for fine-grained recognition. In *Association for the Advancement of Artificial Intelligence (AAAI)*, 2021.
- [18] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. Mask r-cnn. In *International Conference on Computer Vision (ICCV)*, 2017.
- [19] Xiaodi Hou and Liqing Zhang. Saliency detection: A spectral residual approach. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007.
- [20] Gao Huang, Yulin Wang, Kangchen Lv, Haojun Jiang, Wenhui Huang, Pengfei Qi, and Shiji Song. Glance and Focus Networks for Dynamic Visual Recognition. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 2023.
- [21] Shaoli Huang, Zhe Xu, Dacheng Tao, and Ya Zhang. Part-stacked cnn for fine-grained visual categorization. *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.
- [22] Yueting Huang, Zhenzhe Hechen, Mingliang Zhou, Zhengguo Li, and Sam Kwong. An attention-locating algorithm for eliminating background effects in fine-grained visual classification. *IEEE Transactions on Circuits and Systems for Video Technology*, 2025.
- [23] Laurent Itti and Christof Koch. Computational modelling of visual attention. *Nature Reviews Neuroscience*, 2001.
- [24] Max Jaderberg, Karen Simonyan, Andrew Zisserman, and Koray Kavukcuoglu. Spatial transformer networks. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2015.
- [25] Aditya Jain, Fagner Cunha, Michael James Bunsen, Juan Sebastián Cañas, Léonard Pasi, Nathan Pinoy, Flemming Helsing, JoAnne Russo, Marc Botham, Michael Sabourin, Jonathan Fréchette, Alexandre Anctil, Yacksecari Lopez, Eduardo Navarro, Filonila Perez Pimentel, Ana Cecilia Zamora, José Alejandro Ramirez Silva, Jonathan Gagnon, Tom August, Kim Bjerger, Alba Gomez Segura, Marc Bélisle, Yves Basset, Kent P. McFarland, David Roy, Toke Thomas Høye, Maxim Larrivée, and David Rolnick. Insect identification in the wild: The ami dataset. In *European Conference on Computer Vision (ECCV)*, 2024.

- [26] V. Javier Traver and Alexandre Bernardino. A review of log-polar imaging for visual perception in robotics. *Robotics and Autonomous Systems*, 2010.
- [27] Aditya Khosla, Nityananda Jayadevaprakash, Bangpeng Yao, and Li Fei-Fei. Novel dataset for fine-grained image categorization. In *CVPR Workshop on Fine-Grained Visual Categorization*, 2011.
- [28] Dimitri Korsch, Paul Bodesheim, and Joachim Denzler. Deep learning pipeline for automated visual moth monitoring: Insect localization and species classification. *INFORMATIK*, 2021.
- [29] Dimitri Korsch, Paul Bodesheim, Gunnar Brehm, and Joachim Denzler. Automated visual monitoring of nocturnal insects with light-based camera traps. In *CVPR Workshop on Fine-grained Visual Classification*, 2022.
- [30] Dmitry Laptev, Nikolay Savinov, Joachim M. Buhmann, and Marc Pollefeys. Tipooling: Transformation-invariant pooling for feature learning in convolutional neural networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
- [31] Dominik Lindner, Johann Schmidt, Tom Siegl, Martin Becker, and Sebastian Stober. Zero-shot test-time canonicalization using out-of-distribution scoring. *European Conference on Machine Learning Principles and Practice of Knowledge Discovery in Databases*, 2026.
- [32] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott E. Reed, Cheng-Yang Fu, and Alexander C. Berg. Ssd: Single shot multibox detector. In *European Conference on Computer Vision (ECCV)*, 2016.
- [33] 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 *International Conference on Computer Vision (ICCV)*, 2021.
- [34] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*, 2017.
- [35] Wei Luo, Xitong Yang, Xianjie Mo, Yuheng Lu, Larry S. Davis, Jun Li, Jian Yang, and Ser-Nam Lim. Cross-x learning for fine-grained visual categorization. *International Conference on Computer Vision (ICCV)*, 2019.
- [36] Volodymyr Mnih, Nicolas Heess, Alex Graves, and Koray Kavukcuoglu. Recurrent models of visual attention. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2014.
- [37] Aiden Nibali, Zhen He, Stuart Morgan, and Luke A. Prendergast. Numerical coordinate regression with convolutional neural networks. *ArXiv*, 2018.
- [38] Ninh Pham and Rasmus Pagh. Fast and scalable polynomial kernels via explicit feature maps. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2013.

- [39] Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. Dynamicvit: Efficient vision transformers with dynamic token sparsification. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.
- [40] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
- [41] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2015.
- [42] Erik Rodner, Marcel Simon, Gunnar Brehm, Stephanie Pietsch, Johann Wolfgang Wägele, and Joachim Denzler. Fine-grained recognition datasets for biodiversity analysis. *ArXiv*, 2015.
- [43] Johann Schmidt and Sebastian Stober. Tilt your head: Activating the hidden spatial-invariance of classifiers. *International Conference on Machine Learning (ICML)*, 2024.
- [44] Johann Schmidt and Sebastian Stober. Robust canonicalization through bootstrapped data re-alignment. *EurIPS: UPLB Workshop*, 2025.
- [45] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. *International Journal on Computer Vision*, 2020.
- [46] Arindam Sikdar, Yonghuai Liu, Siddhardha Kedarisetty, Yitian Zhao, Amr Ahmed, and Ardhendu Behera. Interweaving insights: High-order feature interaction for fine-grained visual recognition. *International Journal on Computer Vision*, 2024.
- [47] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Herve Jegou. Training data-efficient image transformers: distillation through attention. In *International Conference on Machine Learning (ICML)*, 2021.
- [48] Grant Van Horn, Steve Branson, Ryan Farrell, Scott Haber, Jessie Barry, Panos Ipeirotis, Pietro Perona, and Serge Belongie. Building a bird recognition app and large scale dataset with citizen scientists: The fine print in fine-grained dataset collection. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.
- [49] Grant Van Horn, Oisin Mac Aodha, Yang Song, Yin Cui, Chen Sun, Alex Shepard, Hartwig Adam, Pietro Perona, and Serge Belongie. The inaturalist species classification and detection dataset. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.
- [50] Jiahui Wang, Qin Xu, Bo Jiang, Bin Luo, and Jinhui Tang. Multi-granularity part sampling attention for fine-grained visual classification. *IEEE Transactions on Image Processing*, 2024.
- [51] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona. Caltech-ucsd birds 200. Technical report, 2010.
- [52] Jeremy Wolfe. Guided search 6.0: An updated model of visual search. *Psychonomic bulletin and review*, 2021.

- [53] Sanghyun Woo, Shoubhik Debnath, Ronghang Hu, Xinlei Chen, Zhuang Liu, In So Kweon, and Saining Xie. Convnext v2: Co-designing and scaling convnets with masked autoencoders. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.
- [54] Qin Xu, Sitong Li, Jiahui Wang, Bo Jiang, and Jinhui Tang. Context-semantic quality awareness network for fine-grained visual categorization. *ArXiv*, 2024.
- [55] Yifan Xu, Zhijie Zhang, Mengdan Zhang, Kekai Sheng, Ke Li, Weiming Dong, Liqing Zhang, Changsheng Xu, and Xing Sun. Evo-vit: Slow-fast token evolution for dynamic vision transformer. In *Association for the Advancement of Artificial Intelligence (AAAI)*, 2021.
- [56] Ze Yang, Tiange Luo, Dong Wang, Zhiqiang Hu, Jun Gao, and Liwei Wang. Learning to navigate for fine-grained classification. In *European Conference on Computer Vision (ECCV)*, 2018.
- [57] A. L. Yarbus. Eye movements and vision. *i-Perception*, 1967.
- [58] Hongxu Yin, Arash Vahdat, Jose Alvarez, Arun Mallya, Jan Kautz, and Pavlo Molchanov. A-ViT: Adaptive tokens for efficient vision transformer. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.
- [59] Fan Zhang, Meng Li, Guisheng Zhai, and Yizhao Liu. Multi-branch and multi-scale attention learning for fine-grained visual categorization. In *International Conference on MultiMedia Modeling (MMM)*, 2021.
- [60] Yuan Zhang, Jian Cao, Ling Zhang, Xiangcheng Liu, Zhiyi Wang, Feng Ling, and Weiqian Chen. A free lunch from vit: adaptive attention multi-scale fusion transformer for fine-grained visual recognition. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022.
- [61] Zi-Chao Zhang, Zhen-Duo Chen, Yongxin Wang, Xin Luo, and Xin-Shun Xu. A vision transformer for fine-grained classification by reducing noise and enhancing discriminative information. *Pattern Recognition*, 2024.
- [62] Heliang Zheng, Jianlong Fu, Zheng-Jun Zha, and Jiebo Luo. Looking for the devil in the details: Learning trilinear attention sampling network for fine-grained image recognition. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.