Foveated Vision via Prediction Error–Augmented Reinforcement Learning

Brion Ye brionqye@stanford.edu
CS224R: Deep Reinforcement Learning, Spring 2026 Stanford University

Extended Abstract

Motivation Biological vision is foveated: the eye allocates high resolution only to a small central region and moves to point it at whatever is relevant. Reproducing this technique in a computer vision scenario has the potential to reduce classification cost while preserving accuracy where it counts. This project asks whether a reinforcement learning agent can learn a good patch-selection policy from sparse classification reward, and whether augmenting the policy with a forward-model prediction error signal (either as intrinsic reward or as an observation feature) improves over a vanilla baseline.

Method We formulate foveated classification as a finite-horizon MDP. A 224×224 ImageNet-100 image is divided into a 7×7 patch grid (49 patches of 32×32 pixels each). The agent starts with a globally downsampled blurry view, sequentially selects patches to reveal at full resolution within a fixed horizon H, and receives a sparse terminal reward from a frozen classifier applied to the resulting composite image. We evaluate four agents: **Agent 0** (random baseline), **Agent A** (vanilla PPO), **Agent B** (PPO with prediction error added as ICM-style intrinsic reward), and **Agent C** (PPO with per-patch prediction-error norms concatenated to the observation). Agents B and C rely on a forward dynamics model that predicts the next ViT embedding and its uncertainty for each candidate patch action.

Implementation The image feature extractor is a ViT-small/16 pretrained on ImageNet-100, producing a 384-d CLS token per step. A 2-layer MLP classifier head is fine-tuned on synthetically foveated images, then frozen during RL training. The dynamics model is a 3-layer Gaussian-output MLP trained offline on Agent A rollouts. All RL agents use PPO with 64 parallel environments for 2M environment steps. We sweep blur scale $s \in \{4, 8, 16\}$ and patch horizon $H \in \{8, 16, 24\}$ across 4 agents and 3 seeds, for 96 total runs on NVIDIA A10 GPUs via Modal.

Results RL agents substantially outperform random selection in every configuration, with the gap growing sharply under heavier blur. The most striking case is $s = 16$, $H = 8$: Agent A achieves 65.2% versus the random baseline's 16.8%—a 48.4-point improvement—where the selection policy accounts for nearly all useful information. Performance improves consistently with higher horizon H; blur scale s has diminishing impact as more patches are revealed. Neither the ICM intrinsic reward (Agent B) nor the per-patch uncertainty observation (Agent C) improves consistently over Agent A: all three trained agents achieve nearly identical accuracy across all nine configurations.

Discussion All trained agents converge to a nearly image-independent, center-first selection strategy; the heatmaps for Agents A, B, and C are essentially identical regardless of which prediction error signal is provided. There are two likely reasons for the null result. First, the dynamics model was trained on a narrow distribution of center-biased rollouts and likely produces nearly uniform uncertainty estimates across patches, providing no directional signal. Second, uncorrupted ImageNet subjects are typically centered, so a center-first policy is already near-optimal in expectation and PPO finds it quickly, leaving little room for an uncertainty-based strategy to improve.

Conclusion This project establishes that PPO can learn a meaningful foveated patch-selection policy on ImageNet-100, with the largest gains under heavy blur where patch ordering is the dominant factor in classifiability. The null result on Agents B and C is itself informative: prediction error does not help when center-biased selection already saturates the available accuracy. The natural next experiment is ImageNet-C, where distribution shift should make prediction errors larger and spatially uneven, giving the uncertainty signal a genuine opportunity to influence where the agent looks.

Foveated Vision via Prediction Error–Augmented Reinforcement Learning

Brion Ye brionqye@stanford.edu
CS224R: Deep Reinforcement Learning, Spring 2026
Stanford University, Electrical Engineering

Abstract

We study foveated image classification as a reinforcement learning problem: an agent sequentially reveals high-resolution patches from a globally blurry view, selecting the most informative regions within a fixed budget to maximize classification accuracy. We train and evaluate four agents across a 96-run sweep over blur scale and patch horizon on ImageNet-100. RL agents substantially outperform random patch selection under heavy blur, but augmenting the policy with a forward-model prediction error signal seems to provide no consistent benefit over vanilla PPO. All trained agents converge to a center-biased selection strategy, and we characterize why this happens and what it implies for future work.

1 Introduction

The human visual system is foveated: only a small central region (the fovea) has high acuity, and the eye moves continuously to point this spotlight at whatever is relevant. This lets us make sense of a scene without needing high-resolution coverage of the entire visual field. The computational version of this *adaptive sensing* idea asks whether a machine can learn *where to look*: given a fixed number of high-resolution measurements, which image regions carry the most classification information?

In modern computer vision, this matters when deploying classifiers on hardware where acquiring a full high-resolution image is expensive. A policy that can identify the most informative patches within a fixed sensing budget rather than always capturing everything would be useful in embedded cameras and low-power remote sensing, among other settings. Reinforcement learning is a natural fit: the agent takes measurements sequentially and must learn to prioritize which regions to reveal given a fixed number of looks.

This project addresses three concrete questions within this setting. First, can a PPO-based agent learn a foveated patch-selection policy that meaningfully outperforms random selection? Second, across what parameter regime does intelligent patch selection confer the most benefit? Third, does exposing the agent to a forward-model prediction error signal improve over vanilla PPO?

The third question can be further refined depending on how exactly the prediction error signal is integrated. The hypothesis is that prediction error from a forward model is a useful signal for deciding where to look next: if the model is uncertain about what a region looks like, that region is likely to be informative. We test two integration methods: prediction error as an intrinsic motivation reward signal, and prediction error concatenated directly as an observation feature.

We build a complete foveated RL pipeline over ImageNet-100 (frozen ViT-small backbone, fine-tuned classifier head, Gaussian-output dynamics model) and conduct a 96-run sweep across agents, scales, horizons, and seeds. We report test accuracy across the full grid, derive an information density analysis as a function of (s, H), and characterize the patch-selection strategies learned by each agent.

2 Related Work

Attention-based foveated vision. Foveated, attention-based vision for classification was pioneered by the Recurrent Model of Visual Attention (RAM) Mnih et al. (2014), which trains a recurrent network via REINFORCE to select glimpse locations, achieving competitive performance while processing a small fraction of the image. Subsequent work has extended this paradigm: Ibrayev et al. (2024) couple a ventral recognition pathway with a dorsal RL controller in a two-stream architecture for robotic manipulation. Neither framework incorporates model-based uncertainty to guide attention; the policy conditions only on accumulated observations, not on forward-model predictions of candidate regions.

Intrinsic motivation and prediction error. The Intrinsic Curiosity Module (ICM) Pathak et al. (2017) is the canonical approach to using forward-model prediction error as exploration motivation: a learned dynamics model predicts the next state embedding and prediction error is added as intrinsic reward. ICM enables progress in sparse-reward settings, but the uncertainty signal is consumed entirely through the reward channel—the policy cannot condition its behavior on per-location model uncertainty. This hypothetical gap is precisely what this project aims to address by holding architecture and training procedure constant and varying only how the prediction error signal is used.

Prediction error as policy input. The closest precedent to the observation-concatenation approach is Predictive Processing PPO (P4O) Küçükolu et al. (2024), which integrates prediction-error residuals into a recurrent PPO backbone, improving sample efficiency on Atari. However, the prediction error serves as an internal gating mechanism within the network rather than an explicit observation feature concatenated to the policy's input, and P4O is neither goal-conditioned nor evaluated under distribution shift. Our work addresses this by making prediction error a first-class observation feature and testing whether it improves foveated patch selection.

3 Method

3.1 Problem Setup

We model foveated classification as a finite-horizon MDP. A 224×224 image from ImageNet-100 is divided into a 7×7 grid of 32×32 patches (49 total). At episode start the agent receives a globally downsampled view at scale s: each patch is average-pooled to $(32/s)^2$ pixels and upsampled back to 32×32. The composite image (defocused background with revealed patches inpainted at full resolution) is passed through a frozen ViT-small/16 backbone to produce a 384-d CLS embedding z_t. The agent also observes a 49-d binary mask m_t of revealed patches. The action $a_t \in \{0, \ldots, 48\}$ selects one unrevealed patch (masked softmax prevents re-selection). After H reveals, a frozen classifier head produces prediction $\hat{y}$ and the agent receives sparse terminal reward $r = \mathbf{1}[\hat{y} = y]$. We sweep $s \in \{4, 8, 16\}$ and $H \in \{8, 16, 24\}$, for nine distinct task configurations.

3.2 Agent Design and Prediction Error Signals

All RL agents share the same 2-layer MLP policy network, feeding into a masked softmax policy head and a scalar value head. The four agents differ only in what they observe and what reward they receive:

- **Agent 0**: Selects patches uniformly at random. No training.
- **Agent A**: Observes (z_t, m_t), a 433-dimensional vector. Receives terminal classification reward only.
- **Agent B**: Same observation as Agent A. Receives terminal reward plus an ICM-style intrinsic bonus: the mean normalized prediction error across unrevealed patches, weighted by $\beta = 0.01$.
- **Agent C**: Observes $(z_t, m_t, \mathbf{e}_t)$, a 482-dimensional vector, where $\mathbf{e}_t \in \mathbb{R}^{49}$ is a per-patch prediction error vector. Receives terminal reward only.

Agents B and C rely on a shared forward dynamics model. Given the current CLS embedding z_t and a one-hot patch action a_t, the model outputs a Gaussian distribution $\mathcal{N}(\hat{\mu}_{t+1}, \hat{\sigma}^2_{t+1})$ over the next embedding. The per-patch prediction error norm for patch k is:

$$e_k = \frac{\|\hat{\mu}_k - z^{(k)}_{t+1}\|_2}{\hat{\sigma}_k}.$$

This normalization weights residuals by the model's own confidence. For Agent B, the intrinsic bonus at the terminal step is:

$$r_{\text{B}} = r + \beta \cdot \frac{1}{|U|} \sum_{k \in U} e_k,$$

where U is the set of unrevealed patches. For Agent C, the full vector $\mathbf{e}_t$ (with revealed-patch entries zeroed) is concatenated to the observation at each step.

3.3 Information Density Analysis

Before interpreting results, it is useful to ask: how much pixel-level information does the classifier actually see for a given (s, H) configuration? The composite image presented to the ViT contains H fully-revealed patches and $49 - H$ defocused patches. Regardless of upsampling, each defocused patch contributes only $(32/s)^2$ unique pixels of information. The total unique pixel information is:

$$P(s, H) = H \cdot 32^2 + (49 - H) \cdot \left(\frac{32}{s}\right)^2.$$

Normalized by the full-image pixel count $49 \times 1024 = 50{,}176$:

$$\rho(s, H) = \frac{H}{49} + \frac{49 - H}{49} \cdot \frac{1}{s^2}.$$

Table 1 reports $\rho(s, H)$ for all nine configurations.

Table 1: Information density $\rho(s, H)$: fraction of full-image unique pixel information available to the classifier. Horizon H is the dominant driver; scale s has diminishing impact at large H.

Scale s (defocused res.)	Horizon H		
	8	16	24
$s = 4$ (56^2)	21.6%	36.9%	52.2%
$s = 8$ (28^2)	17.6%	33.7%	49.8%
$s = 16$ (14^2)	16.6%	32.9%	49.2%

Two patterns stand out. **Horizon dominates:** each step from $H = 8 \to 16 \to 24$ adds 8,192 new high-resolution pixels, raising ρ by ≈15–16 points regardless of scale. **Scale has diminishing returns:** at $H = 24$ the spread across scales collapses to just 3 points (49.2%–52.2%), because the defocused background contributes only 0.2%–3.2% of total pixel information once 24 patches are revealed. The extremes are most illuminating: at $s = 16$, $H = 8$ the defocused background provides 164 unique pixels (0.3% of the image), so the 8 selected patches account for 98.0% of all available information; the selection policy is almost entirely what determines classifiability. At $s = 4$, the defocused view provides 2,624 pixels (5.2%) of much higher spatial fidelity, giving the backbone a richer global context even without any revealed patches.

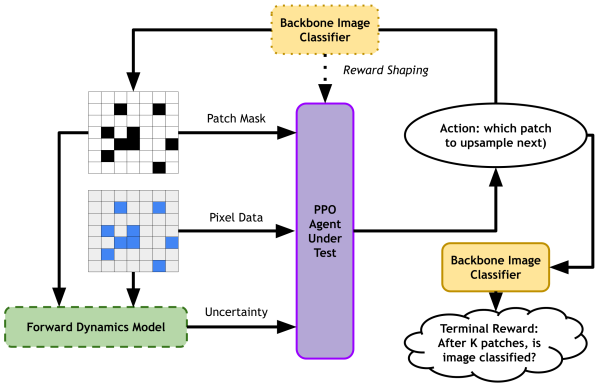


Figure 1: System overview. The agent observes the ViT-small CLS embedding of the composite image and a binary patch mask. Agents B and C additionally receive a signal from the forward dynamics model (dashed path): as intrinsic reward for Agent B, or as a per-patch uncertainty observation for Agent C. After H reveals the frozen classifier delivers a terminal reward.

4 Experimental Setup

Dataset. We train and evaluate on ImageNet-100, a 100-class subset of ImageNet-1k. Agents train on the full training split (∼130k images) sampled uniformly at random each episode, and evaluation is reported on the held-out validation set. We do not apply any data augmentation beyond the foveation process itself.

Backbone and classifier head. We use a ViT-small/16 pretrained on ImageNet-100 (88.2% top-1 accuracy on full-resolution ImageNet-100) as a frozen feature extractor. A pretrained ViT head trained only on full-resolution inputs underperforms on the partially-revealed foveated images that arise during RL training, depressing reward and slowing convergence. We therefore fine-tune a separate 2-layer MLP classifier head (384→256→100 dimensions) for 3 epochs on synthetically foveated views—images with a random number of revealed patches drawn uniformly from $[1, 49]$—then freeze it during RL training.

Dynamics model training. The forward dynamics model is a 3-layer MLP (hidden size 256) with a Gaussian output head, trained offline on transitions collected from a partially-trained Agent A. The training objective is negative log-likelihood under the predicted Gaussian. The model is trained to convergence, then frozen before any RL training begins. All agents B and C in the sweep share the same dynamics model checkpoint.

Policy training and sweep. All RL agents use PPO with clipping parameter $\varepsilon = 0.2$, entropy coefficient 0.01, value loss coefficient 0.5, and GAE-$\lambda = 0.95$. We use 64 parallel environments and train for 2M environment steps per run, evaluating over 256 episodes every 50k steps. Training runs on NVIDIA T4 GPUs via Modal. The full sweep is 3 scales × 3 horizons × 4 agents × 3 seeds = 108 runs; 96 completed successfully.

5 Results

5.1 Quantitative Evaluation

Table 2 reports mean validation accuracy across 3 seeds for all agents and configurations. We highlight three key findings.

Table 2: Test accuracy (%) for all agents, scales, and horizons (mean over 3 seeds). Bold indicates the top-performing RL agent per row. † denotes rows mixing 1- and 3-epoch head checkpoints due to partial training reruns. ViT full-resolution accuracy: 88.2%.

Config		Agent			
s	H	**0 (random)**	**A (PPO)**	**B (ICM)**	**C (Uncert.)**
4	8	74.5	**83.0**	82.4	80.6
	16	75.8	83.3†	83.6†	83.4†
	24	78.5	**84.0**†	82.9†	82.4†
8	8	60.5	70.5	71.3	**71.5**
	16	67.2	76.7	78.0	**78.8**
	24	73.0	81.9	**82.1**	81.4
16	8	16.8	**65.2**	61.7	64.4
	16	35.0	72.0	**73.9**	71.0
	24	55.2	76.4	75.9	**77.3**

RL vs. random gap. All three RL agents outperform random selection in every configuration. The gap scales strongly with s: at $s = 4$ the improvement over Agent 0 is 6–9 points; at $s = 8$, 7–12 points; at $s = 16$, 10–48 points. The extreme at $s = 16$, $H = 8$, where Agent A reaches 65.2% against the random baseline's 16.8%, is explained directly by the information density analysis: with only 16.6% of pixel information available and 98.0% of it concentrated in the 8 chosen patches, the selection policy almost entirely determines classifiability. At $s = 4$, $H = 8$, Agent 0 already achieves

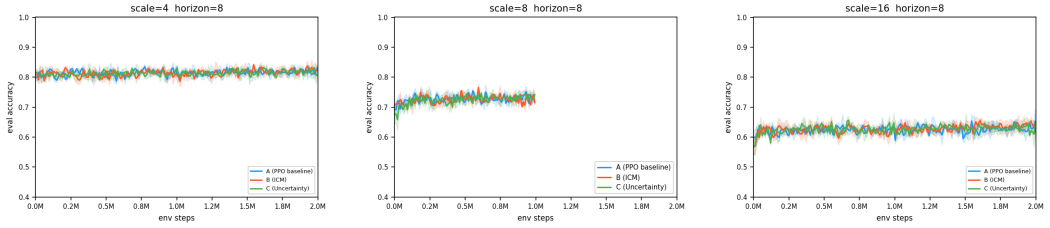
74.5%, consistent with a defocused background that provides a much richer 56^2 global context even before any patches are revealed.

Scale and horizon effects. Performance improves as s decreases (less blur) and as H increases (more reveals). The interesting tension is that the information density analysis predicts scale should matter much less than horizon—at $H = 24$, the spread in ρ across all three scales is only 3 points (49.2%–52.2%)—yet actual agent performance still spans 7.6 points (76.4%→84.0%). At $H = 8$ the discrepancy is even starker: ρ spans only 5 points across scales (16.6%–21.6%), while performance spans 17.8 points (65.2%→83.0%). In other words, the performance gap between $s = 4$ and $s = 16$ is consistently 3–4× larger than the pixel-count difference alone would suggest. This implies that the *quality* of the background context matters beyond its raw pixel count: at $s = 4$ the defocused view retains global structural information (edges, color blobs, rough object layout), while at $s = 16$ the 2×2-pixel-per-patch background is nearly blank, depriving the classifier of any spatial context for unrevealed regions. Horizon growth also benefits the random baseline at a similar rate as the trained agents: the gap between Agent 0 and the RL agents stays roughly constant as H increases within each scale, suggesting the primary benefit of RL is in ordering the first few reveals, not in selecting a globally better set.

Agent parity. There is no configuration where one of Agents A, B, or C consistently and substantially outperforms the others. At $s = 16$, $H = 8$, Agent B (61.7%) slightly underperforms Agent A (65.2%), suggesting that the ICM bonus may draw the policy toward uncertain patches that are not the most useful ones for classification. Differences across configurations are within 3–4 points and are difficult to interpret confidently with 3 seeds. What is clear is that neither signal provides a consistent and repeatable improvement over the vanilla PPO baseline.

5.2 Qualitative Analysis

Figure 2 shows evaluation accuracy during training for three representative configurations. In all three cases, all agents converge within approximately 200k–400k environment steps, after which performance plateaus for the remainder of the 2M step budget. The improvement from initialization to convergence is much larger at $s = 16$ (roughly 60%→65%) than at $s = 4$ (agents start around 80% and improve only a few points): at mild blur the random-initialization policy is already near-optimal, so there is less headroom. The curves for A, B, and C overlap throughout training at every configuration, confirming that the prediction error signal does not alter the optimization trajectory.



(a) $s = 4$, $H = 8$ **(b)** $s = 8$, $H = 8$ **(c)** $s = 16$, $H = 8$

Figure 2: Evaluation accuracy (mean ± std, 3 seeds) during training for Agents A (blue), B (red), C (green). Curves overlap tightly across all three difficulty levels.

Figure 3 shows a single episode at $s = 8$, $H = 16$ for Agent A (top) and Agent 0 (bottom). Agent A immediately focuses on the central subject, revealing the most discriminative regions early. Agent 0 spreads reveals more evenly. By $t = 16$ both agents have covered a similar fraction of the image, but Agent A's center-first ordering means the classifier received the useful information sooner.

Figure 4 shows mean patch visit frequency aggregated over ≈1,000 validation images at $s = 8$, $H = 16$. The most striking feature is how concentrated all trained agents are: the central 3×3 region is visited in nearly every episode, while corner patches are selected only rarely. Agent 0, by contrast, shows the expected near-uniform distribution ($H/49 \approx 0.33$ per patch). This center bias is not driven by image content—inspecting individual episodes shows that Agents A, B, and C select largely the same patches regardless of the image class. The second striking feature is that the heatmaps for Agents A, B, and C are essentially identical. Whether the agent receives no prediction error signal,

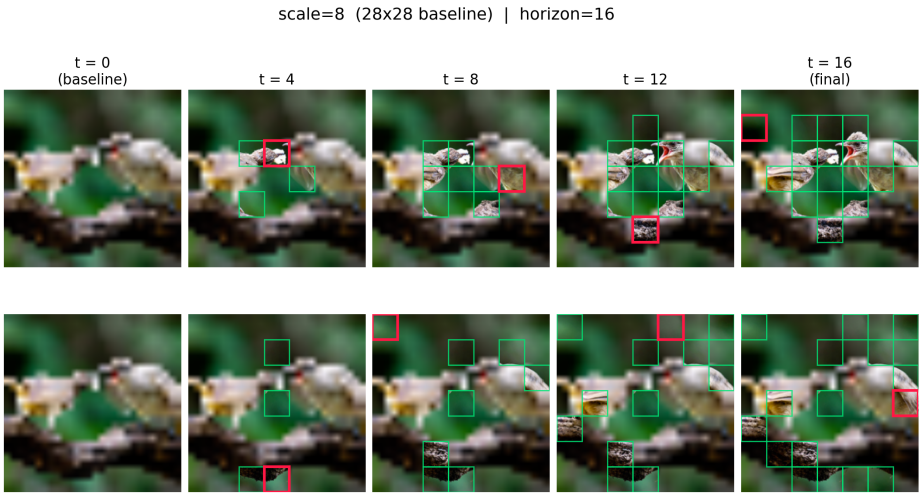


Figure 3: Patch selection at $s = 8$, $H = 16$. **Top:** Agent A (PPO). **Bottom:** Agent 0 (random). Columns show the composite image at $t \in \{0, 4, 8, 12, 16\}$; green borders = previously revealed; red border = current selection.

an intrinsic bonus, or a full 49-dimensional uncertainty observation, it ends up looking at the same places. The prediction error channel does not change the spatial distribution of patch selection at all.

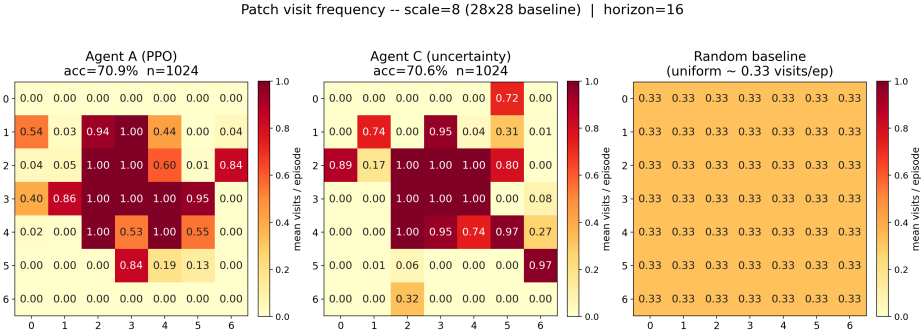


Figure 4: Mean patch visit frequency per episode over ∼1,000 validation images ($s = 8$, $H = 16$). Agent A (left) and Agent C (center) exhibit strong center bias; Agent 0 (right) visits patches uniformly. Agents B and C show heatmaps indistinguishable from Agent A.

6 Discussion

The two main behavioral observations from the results—that all trained agents converge to the same center-biased strategy, and that prediction error provides no measurable benefit—call for explanation.

6.1 Why the Prediction Error Signal Has No Effect

There are two likely reasons. First, the dynamics model may not be producing useful per-patch uncertainty estimates. It was trained offline on rollouts collected from a center-biased Agent A, so it has rarely seen corner patches and probably assigns nearly uniform uncertainty everywhere. If $\mathbf{e}_t$ is roughly constant across patches, it adds no useful information to Agent C's observation and gives Agent B's reward bonus no directional signal either. Second, uncorrupted ImageNet may not require image-conditional patch selection to achieve near-optimal performance. Most ImageNet subjects are centered, so always revealing center patches first is already a near-optimal strategy in expectation. PPO finds this solution quickly, and training longer doesn't push the agents away from it. We can't say whether the null result is fundamental or a symptom of poor dynamics model

calibration; diagnosing the model's spatial uncertainty distribution is a necessary first step before drawing stronger conclusions.

6.2 The Center Bias as a Backbone Artifact

The center-bias finding is consistent with what we know about ViT-small's attention behavior. ViT models trained on ImageNet tend to place disproportionate weight on central tokens in their self-attention layers, partly because ImageNet images are composed with subjects centered by convention. This means the backbone already rewards center-first selection regardless of what the image actually contains, which reinforces the center bias through the reward signal. The agents are optimizing against a signal that is itself center-biased: the center-biased strategy scores well, PPO reinforces it, and the agents never explore other orderings long enough to discover whether image-conditional strategies would do better. Breaking this cycle likely requires a backbone with more uniform spatial attention, or a training distribution where subjects are not systematically centered.

7 Conclusion

PPO agents learn a meaningful foveated patch-selection policy on ImageNet-100, outperforming random selection across all configurations, with gains as large as 48 points under heavy blur. The patch horizon H is the dominant factor in both information density and classification accuracy; blur scale has diminishing impact as more patches are revealed. Adding prediction error as intrinsic reward or as an observation feature does not improve over the vanilla PPO baseline; all three agents converge to the same center-biased strategy. The dynamics model calibration is an open question, and whether prediction error is genuinely uninformative or simply underutilized is not yet settled. The most compelling next experiment is ImageNet-C: the clean ImageNet sweep establishes a clear baseline, and distribution shift is where prediction errors should become spatially meaningful enough to influence patch selection. A per-patch sensing cost formulation, which would turn the problem into one of adaptive budget allocation rather than fixed-horizon ordering, is a longer-term direction the current results directly support.

8 Team Contributions

Brion Ye designed and implemented the full pipeline: the Gymnasium foveated vision environment, the ViT-based observation stack, the classifier head fine-tuning procedure, the dynamics model training, and all three PPO agent variants. He ran the full 96-run parameter sweep on Modal cloud infrastructure, produced all figures and analysis, and wrote the project deliverables.

Changes from Proposal The original proposal planned to evaluate agents on both clean ImageNet and ImageNet-C (15 corruption types), with distribution shift robustness as the primary test. The final project focused the parameter sweep on clean ImageNet-100, characterizing the baseline regime thoroughly before extending to corruptions; ImageNet-C evaluation is the most natural next step. The implementation also diverged from the initial spec in several ways: a custom PPO implementation replaced the originally planned stable-baselines3, ImageNet-100 replaced ImageNet-1k as the classification target, and the dynamics model architecture was finalized as a Gaussian-output MLP rather than a deterministic transformer.

Acknowledgements

The author thanks Mouhssine Rifaki (Arbabian Lab, Stanford EE) for advising this project. Mouhssine helped to pose an initial core research question—whether prediction error is more useful as an observation feature than as intrinsic reward—and provided some initial positive experimental runs that inspired the final 4-agent, 3-seed design. He also gave guidance on the dynamics model architecture and the intrinsic reward weighting. This project is part of the broader adaptive sensing research effort in the Arbabian Lab (Prof. Amin Arbabian).

References

Timur Ibrayev et al. 2024. Two-Stream Foveation-based Active Vision for Efficient Robot Manipulation. *arXiv preprint arXiv:2403.15977* (2024).

Burcu Küçükolu, Walraaf Borkent, Bodo Rueckauer, Nasir Ahmad, Umut Güçlü, and Marcel van Gerven. 2024. Efficient Deep Reinforcement Learning with Predictive Processing Proximal Policy Optimization. arXiv:2211.06236 [cs.LG] `https://arxiv.org/abs/2211.06236`

Volodymyr Mnih, Nicolas Heess, Alex Graves, and Koray Kavukcuoglu. 2014. Recurrent Models of Visual Attention. In *Advances in Neural Information Processing Systems (NeurIPS)*. 2204–2212.

Deepak Pathak, Pulkit Agrawal, Alexei A. Efros, and Trevor Darrell. 2017. Curiosity-driven Exploration by Self-supervised Prediction. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*. PMLR, 2778–2787.