PATHHUNTER: HIERARCHICAL EVIDENCE ACQUISITION FOR WEAKLY SUPERVISED WSI CLASSIFICATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Histopathology remains the gold standard for definitive disease diagnosis, , yet identifying diagnostically relevant evidence within gigapixel whole-slide images (WSIs) under slide-level supervision remains challenging. Most weakly supervised pipelines treat WSI classification as passive aggregation over a predefined bag of patches, without explicitly learning how that evidence set should be constructed. We propose PathHunter, a hierarchical reinforcement learning (HRL) framework for evidence selection over precomputed multi-scale WSI representations. A high-level Manager selects candidate regions of interest (ROI)·, while a low-level Worker selects fine-grained patches within each region, producing a compact evidence bag for downstream MIL classification. PathHunter is trained in three stages: a standard MIL baseline is first learned from slide labels, the Worker is then initialized using MIL-derived pseudo-expert rankings, and finally the Manager and Worker are jointly optimized with PPO while the prediction MIL model adapts to policy-selected evidence. Step-wise rewards combine improvements in label-consistent confidence with instance-score-based shaping to stabilize weakly supervised credit assignment. Across Camelyon16, TCGA-NSCLC, and PANDA, PathHunter improves a broad range of backbone–MIL combinations, with the largest gains on sparse-evidence metastasis detection and smaller, more task-dependent gains on morphology-driven classification. Post-hoc annotation analysis further shows improved tumor enrichment and lesion coverage in selected evidence. Overall, PathHunter provides a structured and auditable mechanism for learning compact diagnostic evidence sets under weak supervision, while its effectiveness depends on the spatial distribution and complexity of task-relevant morphology. Anonymous code is `https://anonymous.4open.science/r/pathhunter-08F1/README.md`.

1 INTRODUCTION

Whole-slide images (WSIs) remain the clinical standard for pathology diagnosis, but their gigapixel resolution makes exhaustive high-resolution analysis impractical. In weakly supervised settings, annotations are usually available only at the slide level, so modern pipelines typically combine pathology foundation models such as CTransPath Wang et al. (2022) and UNI Chen et al. (2024) with multi-instance learning (MIL) for slide-level prediction Ilse et al. (2018); Li et al. (2021); Zhang et al. (2022). These pipelines can produce strong classifiers, but they still solve diagnosis mostly as a passive aggregation problem: given a sampled bag, the model assigns weights and predicts a label, without explicitly learning how diagnostic evidence should be acquired.

Clinical diagnosis is inherently sequential and multi-scale: pathologists first scan globally at low magnification, then zoom into suspicious regions to confirm malignancy Chen et al. (2025). Under slide-level supervision, reproducing this behavior is difficult because informative lesions are sparse and localized, the action space contains thousands of candidate patches, and slide labels provide only delayed and ambiguous credit for individual selections. The bottleneck is given a precomputed WSI representation and only slide-level supervision, can a model learn to construct a compact, spatially structured evidence bag that is more informative than static score-based selection?

Recent multimodal and agentic pathology systems move closer to evidence-centered reasoning, but they often require language supervision, multiple specialized modules, or expensive coordination.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

CPath-Omni Sun et al. (2025a) studies multimodal pathology representations, while CPathAgent Sun et al. (2025b), PathFound Hua et al. (2025), and PathFinder Ghezloo et al. (2025) pursue richer multi-scale diagnostic behavior through agent-style pipelines. These approaches push the field toward more realistic evidence gathering, but they do not eliminate the core learning problem we study here: under slide-level labels alone, can a model replace passive aggregation with active evidence construction and learn a stable coarse-to-fine acquisition policy?

We address these challenges with PathHunter, a hierarchical reinforcement learning (HRL) framework for evidence selection over precomputed multi-scale WSI representations. A high-level Manager selects candidate regions of interest (ROIs), while a low-level Worker selects fine-grained patches within each chosen ROI. This decomposition reduces a large flat selection problem into coordinated region-level and patch-level decisions. Importantly, MIL-derived instance scores are used to construct and describe the candidate space and to provide weak local priors; they do not deterministically specify the final evidence bag. The learned Manager and Worker instead construct a bounded set of evidence patches that is subsequently aggregated by a MIL prediction head. The resulting hierarchy is therefore coarse-to-fine in its selection structure and evidence trace, rather than in avoiding slide-wide high-resolution preprocessing.

PathHunter is trained through three stages that progressively increase the difficulty of evidence selection. Stage 1 trains a conventional slide-level MIL classifier. Stage 2 freezes this classifier and initializes the Worker using pseudo-expert rankings derived from MIL instance scores, providing a stable local selection prior. Stage 3 jointly optimizes the Manager and Worker with PPO while updating the prediction MIL classifier on policy-selected bags. To provide denser credit under slide-level supervision, the reward combines changes in label-consistent confidence with instance-score-based shaping. This creates an interaction between evidence selection and classification: the selector changes the evidence distribution seen by the classifier, while the evolving classifier provides task-aligned feedback to the selector.

To make this loop trainable under weak supervision, we define step-wise rewards from increments in label-consistent confidence and add instance-score-based reward shaping to stabilize credit assignment. The resulting policy is encouraged not merely to pick locally salient patches, but to assemble trajectories whose acquired evidence improves slide-level prediction. Empirically, this yields interpretable coarse-to-fine evidence traces and consistent improvements across Camelyon16, TCGA-NSCLC, and PANDA.

Our experiments also suggest a broader empirical pattern. The gains are largest on Camelyon16, where sparse metastatic evidence makes targeted acquisition especially valuable, and smaller but still mostly consistent on broader morphology tasks such as NSCLC subtyping and prostate grading. This motivates a useful distinction between tasks dominated by sparse decisive evidence and tasks that depend more on distributed morphology aggregation.

Our contributions are threefold:

- (1) We present a hierarchical RL framework that reformulates weakly supervised WSI diagnosis as sequential evidence acquisition using visual features only, hierarchical evidence selection over precomputed multi-scale representations.
- (2) We introduce a three-stage weakly supervised optimization scheme that combines MIL-based Worker initialization, hierarchical PPO, dense confidence-based rewards, and selected-bag classifier adaptation, while explicitly separating learned selection from the MIL-derived priors used for candidate construction and warm-up.
- (3) We provide controlled empirical analyses across Camelyon16, TCGA-NSCLC, PANDA, and additional pathology tasks, including matched static and policy ablations, selector-classifier adaptation controls, comparisons with sequential-observation baselines, and annotation-based evidence-quality evaluation. The results identify sparse or locally decisive evidence as the regime in which PathHunter is most effective, while revealing more task and backbone dependent behavior for distributed morphology.

2 Related Work

Pathology foundation models and MIL. Recent pathology foundation models have substantially improved WSI classification by providing transferable patch representations that can be paired with slide-level MIL heads. Representative examples include CTransPath Wang et al. (2022) and UNI Chen et al. (2024), while common MIL formulations include attention-based MIL Ilse et al. (2018), DSMIL Li et al. (2021), and related feature-distillation variants Zhang et al. (2022). These pipelines are strong classifiers, but they usually aggregate a static bag of instances rather than explicitly learning how evidence should be acquired across scales.

Evidence-centered and agentic pathology systems. Recent pathology systems increasingly emphasize interpretable evidence gathering. ZoomMIL Thandiackal et al. (2022) uses differentiable zooming to focus computation on informative regions, while RLogist Zhao et al. (2023) formulates WSI observation as a reinforcement-learning problem for efficient region exploration. Sparse and sequential sampling approaches such as SASHA Gogisetty et al. (2026) similarly aim to classify slides from a restricted set of informative observations. CPath-Omni Sun et al. (2025a) studies unified multimodal pathology representations, while CPathAgent Sun et al. (2025b), PathFound Hua et al. (2025), and PathFinder Ghezloo et al. (2025) move toward agent-style or multi-stage diagnostic pipelines. Our goal is narrower and more controlled: we study whether slide-level labels alone are sufficient to train a coarse-to-fine visual acquisition policy, without relying on language supervision or multi-agent coordination.

Hierarchical decision making for evidence search. Hierarchical reinforcement learning is well suited to long-horizon problems with sparse rewards because it decomposes decision making across temporal or spatial scales Bacon et al. (2017); Vezhnevets et al. (2017); Nachum et al. (2018). PathHunter adapts this idea to WSIs by coupling high-level ROI selection with low-level patch retrieval, and by training both policies against slide-level diagnostic feedback. In contrast to generic HRL applications, the central challenge here is that the reward itself is weakly supervised and must be stabilized through interaction with an evolving MIL classifier.

3 Preliminaries

3.1 Multiple instance learning.

In weakly supervised WSI classification, each slide is represented as a bag of patch features $\mathcal{X} = \{x_i\}_{i=1}^{N}$ with $x_i \in \mathbb{R}^D$, and only the slide-level label y is observed. MIL models learn a bag-level predictor by aggregating instance features via pooling, attention, or transformer-based modules. In this sense, standard MIL is a passive evidence aggregation mechanism: it reweights a sampled bag rather than learning how the bag itself should be constructed. Let f_θ denote a MIL encoder and g_ϕ the aggregation head; a common formulation is

$$\hat{p}(y \mid \mathcal{X}) = \mathrm{softmax}(g_\phi(f_\theta(\mathcal{X}))). \tag{1}$$

The training objective is typically the cross-entropy loss between $\hat{p}(y \mid \mathcal{X})$ and the slide label y. While effective, MIL does not explicitly model the sequential acquisition of evidence and can be sensitive to noisy or redundant instances when sampling is limited.

3.2 Hierarchical reinforcement learning.

HRL addresses long-horizon decision-making by decomposing control into multiple levels operating at different temporal or spatial scales. A common formalism is the *Options Framework*, where an option is a temporally extended action specified by an initiation set, an internal policy, and a termination condition. This enables a high-level policy to select abstract actions while a low-level policy executes fine-grained actions, improving exploration and long-term credit assignment. We consider a two-level hierarchy. The high-level policy

$$\pi_H(a^H \mid s^H), \quad a^H \in \mathcal{A}^H,\ s^H \in \mathcal{S}^H, \tag{2}$$

selects an abstract action a^H given s^H. Conditioned on a^H, the low-level policy

$$\pi_L(a^L \mid s^L, a^H), \quad a^L \in \mathcal{A}^L,\ s^L \in \mathcal{S}^L, \tag{3}$$

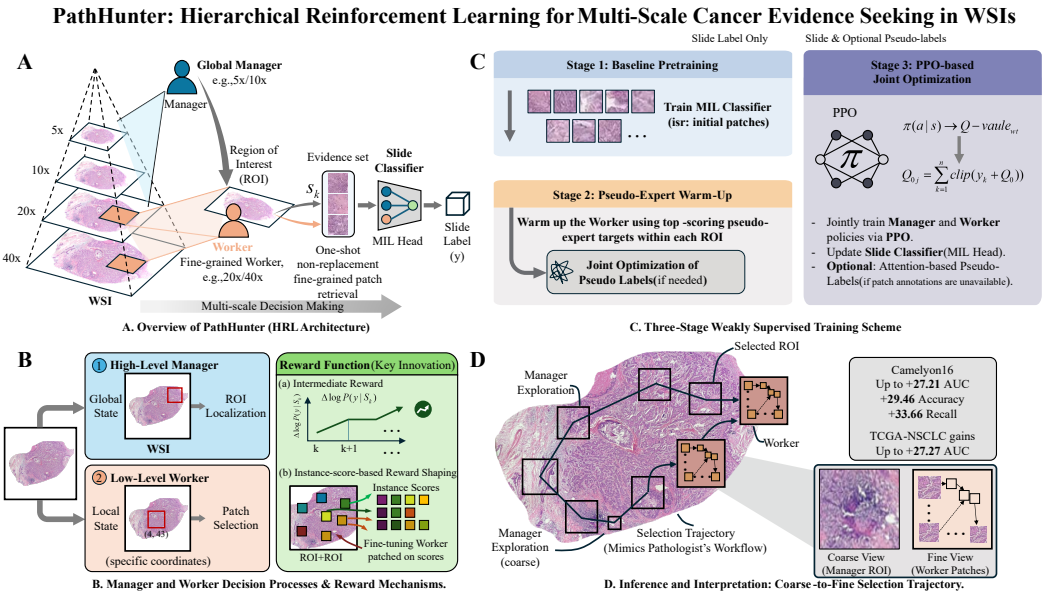


Figure 1: PathHunter framework overview. (A) The Manager selects an ROI on the WSI, and the Worker retrieves discriminative cancer patches within it; the selected trajectory is aggregated by a MIL classifier for slide-level prediction. (B) A coarse-to-fine decision process guided by an evolving heatmap with step-wise process rewards. (C) A self-evolving acquisition curriculum: train a base MIL model for static evidence aggregation, warm up the Worker from pseudo-expert Top-M evidence within each ROI (MIL frozen), then jointly optimize Manager+Worker with PPO while fine-tuning the base model on selected trajectories, closing the acquisition-classifier loop. (D) Inference yields coarse-to-fine search trajectories and interpretable patch evidence.

outputs a primitive action a^L. The goal is to maximize the expected discounted return:

$$J(\theta) = \mathbb{E}_{\pi_H, \pi_L}\left[\sum_{t=0}^{\infty} \gamma^t r_t\right], \tag{4}$$

where $\gamma \in [0, 1)$ and θ collects parameters of both policies. Each policy π_θ (high-level or low-level) is optimized with PPO Schulman et al. (2017):

$$L^{\text{PPO}}(\theta) = \mathbb{E}_t\left[\min\left(r_t(\theta)\hat{A}_t,\ \text{clip}\big(r_t(\theta), 1-\epsilon, 1+\epsilon\big)\hat{A}_t\right)\right], \tag{5}$$

with

$$r_t(\theta) = \frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{\text{old}}}(a_t \mid s_t)}, \tag{6}$$

where $\hat{A}_t$ is the advantage estimate and ϵ controls the update size.

4 METHODS

We consider weakly supervised WSI classification in which each slide is represented by precomputed multi-scale patch features and only the slide-level label is available. Let a slide be a bag of instances $\mathcal{X} = \{x_i\}_{i=1}^N$ with $x_i \in \mathbb{R}^D$, and the slide-level label be $y \in \{1, \ldots, C\}$. In the fully weak setting, only y is observed. We consider a hyperparameterized selection process where the high-level policy runs for K rounds and the low-level policy selects m patches per round, yielding a total of $K \cdot m$ patches.

4.1 OVERVIEW: SELF-EVOLVE HIERARCHICAL EVIDENCE SELECTION

PathHunter performs hierarchical evidence selection over precomputed multi-scale WSI representations. It can also be viewed as a curriculum over action-space complexity: (i) train a slide-level MIL baseline to learn static evidence aggregation, (ii) freeze the baseline and warm up the low-level

patch selector (Worker) to learn local evidence refinement within an ROI, (iii) jointly optimize the hierarchical policy (Manager+Worker) with PPO while continuing to update the baseline classifier on policy-selected patches, thereby learning full-slide evidence acquisition. This closes the loop between acquisition and classification: better trajectories produce cleaner sub-bags for the baseline, and the improved baseline yields sharper rewards and stronger local priors for subsequent policy updates.

At inference time, the hierarchical policy selects a fixed-size set of informative patches controlled by (K, m), which are then classified by the baseline model. The goal is not to summarize tissue uniformly, but to acquire a compact evidence set that is sufficient for reliable slide-level prediction under the chosen patch budget. Compact selection is particularly effective when slide-level decisions depend on sparse or locally decisive evidence. For tasks requiring more distributed morphology, the same framework can acquire broader evidence by increasing the number of the total evidence budget, although the appropriate budget becomes more task dependent.

4.2 Hierarchical ROI-to-Patch Selection

We partition each slide into a set of candidate cells (coarse regions with low resolution) $\mathcal{C} = \{c_j\}_{j=1}^{M}$. Within each round $k \in \{1, \dots, K\}$, a high-level policy (Manager) chooses a cell index

$$a_k^H \sim \pi_H(a^H \mid s_k^H), \tag{7}$$

where s_k^H summarizes the current slide state (e.g., cell features, visited mask, and candidate mask). Conditioned on the chosen cell $c_{a_k^H}$, a low-level policy (Worker) chooses m patches (implemented as a sequence over discretized bins) from that cell:

$$a_k^L \sim \pi_L(a^L \mid s_k^L, a_k^H), \tag{8}$$

where a_k^L corresponds to a set of patch indices $\mathcal{I}_k$. The total number of selected patches is $K \cdot m$, controlled by the hyperparameters K (high-level rounds) and m (low-level patches per round). The environment updates its state after each low-level selection, producing an intermediate reward r_k, and terminates after K rounds.

4.3 Stage 1: Baseline Training with MIL

Stage 1 trains a baseline classifier f_θ with a bag-level MIL objective. Given a subsampled bag $\mathcal{X}' \subseteq \mathcal{X}$ with $|\mathcal{X}'| = N'$, a MIL model predicts slide logits

$$\hat{p}(y \mid \mathcal{X}') = \mathrm{softmax}(f_\theta(\{x_i\}_{i=1}^{N'})). \tag{9}$$

We optimize the cross-entropy loss

$$\mathcal{L}_{\mathrm{MIL}}(\theta) = \mathrm{CE}(y, \hat{p}(y \mid \mathcal{X}')). \tag{10}$$

We instantiate f_θ with standard MIL backbones such as attention-MIL/ABMIL, DSMIL, TransMIL, or DTFD-MIL. We use stratified 4-fold cross-validation when applicable. For datasets with multiple slides per patient, splits are performed at the patient level to avoid leakage. We select checkpoints by validation AUC (or accuracy) with early stopping.

4.4 Stage 2: Pseudo-Expert Warm-Up of the Worker

Stage 2 freezes the baseline f_θ and learns the Worker policy π_L to imitate pseudo-expert targets derived from baseline patch scores. This stage reduces the low-level action space before the full hierarchical policy is exposed to slide-level RL.

For each training step, a cell c is sampled. We compute baseline patch scores $\{s_i\}_{i \in c}$ from the frozen Stage 1 MIL model. The score is taken from the MIL model's predicted class logit, so the pseudo-target is learned from the Stage 1 model rather than manually fixed. We then form an expert set by selecting the top-m patches within the cell:

$$\mathcal{I}^\star(c) = \mathrm{TopM}(\{s_i\}_{i \in c}, m). \tag{11}$$

Patch indices are mapped to bin indices (each cell is discretized into M_{low} bins); we deduplicate bin indices while preserving order to obtain expert bin targets $\mathcal{B}^\star(c)$. The warm-up loss maximizes the

summed log-probability of the expert bins under the Worker, with masking so that after predicting each bin the corresponding bin is masked out for the next step:

$$\mathcal{L}_{\text{cur}}(\phi) = -\sum_{b \in \mathcal{B}^\star(c)} \log \pi_L(b \mid s^L, c; \phi), \tag{12}$$

where ϕ denotes the Worker parameters. This pseudo-expert warm-up stabilizes low-level selection before full reinforcement learning.

4.5 Stage 3: Joint Optimization with PPO and Baseline Updates

Stage 3 jointly trains the Manager (π_H), the Worker (π_L), and continues to update the baseline classifier f_θ. This stage extends local Worker initialization to jointly learned hierarchical ROI-to-patch selection. For each slide, we collect an episode trajectory over K rounds:

$$\tau^H = \{(s_k^H, a_k^H, \log \pi_H, V_H, r_k)\}_{k=1}^K, \qquad \tau^L = \{(s_k^L, a_k^L, \log \pi_L, V_L, r_k)\}_{k=1}^K, \tag{13}$$

along with the final reward R_{final} and the selected patch features.

Let S_k denote the set of selected patches after round k, and the slide label y. The baseline f_θ defines $\log P_\theta(y \mid S)$ as the log-probability of the bag label given the selected bag S. The *per-step reward* at round k is

$$r_k = \log P_\theta(y \mid S_k) - \log P_\theta(y \mid S_{k-1}) - \lambda + \eta \cdot h(\mathcal{I}_k), \tag{14}$$

where λ is a per-round cost, η is the reward-shaping weight, $\mathcal{I}_k$ is the set of patch indices selected in round k, and $h(\mathcal{I}_k)$ is a shaping term based on the baseline instance scores $\{s_i\}$ of those patches. The *final reward* is the MIL log-probability of the full selected bag:

$$R_{\text{final}} = \log P_\theta(y \mid S_K). \tag{15}$$

We compute reward-to-go returns

$$G_k = R_{\text{final}} + \sum_{t=k}^{K} \gamma^{t-k} r_t, \tag{16}$$

with discount factor γ. Advantages are computed as $A_k = G_k - V(s_k)$ for the corresponding value function.

For both the Manager and the Worker, we use the clipped surrogate objective:

$$\mathcal{L}_{\text{PPO}} = -\mathbb{E}\left[\min\left(\rho_k A_k,\ \text{clip}(\rho_k,\ 1-\epsilon,\ 1+\epsilon)\, A_k\right)\right] + \lambda_v \mathcal{L}_V - \lambda_e \mathcal{H}(\pi), \tag{17}$$

where $\rho_k = \pi(a_k \mid s_k)/\pi_{\text{old}}(a_k \mid s_k)$, ϵ is the clip range, $\mathcal{L}_V$ is the value regression loss $\mathcal{L}_V = \mathbb{E}[(V(s_k) - G_k)^2]$, and $\mathcal{H}(\pi) = \mathbb{E}_{a \sim \pi(\cdot|s)}[-\log \pi(a \mid s)]$ is an entropy bonus to stabilize policy updates. For the Worker, which selects a sequence of bin actions of length up to m, losses are masked so that only valid steps contribute.

Baseline update on selected patches. Let $\mathcal{X}_{\text{sel}} = \bigcup_{k=1}^K \{x_i : i \in \mathcal{I}_k\}$ denote the set of patch features selected by the hierarchical policy. We fine-tune the baseline MIL classifier on $\mathcal{X}_{\text{sel}}$ using the bag-level cross-entropy:

$$\mathcal{L}_{\text{base}}(\theta) = \text{CE}(y,\ \text{softmax}(f_\theta(\mathcal{X}_{\text{sel}}))). \tag{18}$$

Updating the prediction MIL model on policy-selected bags adapts the classifier to the evidence distribution induced by the selector. The updated classifier in turn provides the confidence signal used by subsequent policy optimization. By updating the baseline on policy-selected patches, the classifier becomes specialized to the induced sub-bag, which keeps the terminal reward R_{final} informative. This forms a positive feedback loop (*selection* $\rightarrow$ *baseline update* $\rightarrow$ *sharper rewards*), jointly improving both the selector and the classifier.

5 EXPERIMENTS

We organize the evaluation around five research questions. RQ1 evaluates whether PathHunter improves diverse MIL pipelines and pathology tasks. RQ2 disentangles learned evidence selection from static MIL-score ranking and prediction-MIL adaptation. RQ3 compares PathHunter with dedicated sequential-observation methods. RQ4 evaluates whether the selected evidence is pathologically relevant and remains useful beyond the source evaluator. RQ5 examines the computational and evidence-budget trade-offs A.0.2. Test data are never used for checkpoint selection. Detailed datasets, splits, metrics, hyperparameters, and efficiency numbers are deferred to Appendix D, Appendix G, Appendix F.

5.1 MAIN RESULTS

5.1.1 RQ1:DOES PATHHUNTER IMPROVE DIVERSE MIL PIPELINES AND PATHOLOGY TASKS?

Table 1 evaluates three pathology foundation models (CTransPath, UNI2-H, and CONCH1.5) paired with four MIL heads (ABMIL, DSMIL, TransMIL, and DTFD-MIL) on Camelyon16 and TCGA-NSCLC. Appendix O reports additional five-seed uncertainty analysis.

Table 1: Performance comparison of baseline MIL models and PathHunter on Camelyon16 and TCGA-NSCLC. Positive changes are shown in green; negative changes are shown in gray. Results are tested averaged over 5 random seeds.

Backbone	Method	Camelyon16				TCGA-NSCLC			
		Prec.	Rec.	Acc.	AUC	Prec.	Rec.	Acc.	AUC
				CTransPath					
	ABMIL	87.09	83.21	86.05	85.94	91.82	91.53	91.41	98.24
CTransPath	PathHunter + ABMIL	89.36 +2.27	85.71 +2.50	90.70 +4.65	94.87 +8.93	87.86 -3.96	98.40 +6.87	92.58 +1.17	98.34 +0.10
	DSMIL	85.40	66.33	74.42	72.78	93.08	93.04	92.97	98.14
CTransPath	PathHunter + DSMIL	100 +14.59	100 +33.66	100 +25.57	100 +27.21	94.62 +1.54	98.40 +5.36	96.48 +3.51	99.75 +1.61
	TransMIL	87.15	84.63	86.82	91.12	92.75	92.33	92.19	97.19
CTransPath	PathHunter + TransMIL	100 +12.84	97.96 +13.33	99.22 +12.40	100 +8.87	91.73 -1.02	97.60 +5.27	94.53 +2.34	98.71 +1.52
	DTFD-MIL (MaxS)	86.06	77.32	82.17	87.19	93.80	93.80	93.75	98.51
CTransPath	PathHunter + DTFD-MIL	99.20 +13.14	100 +22.67	99.69 +17.52	99.97 +12.78	95.42 +1.62	100 +6.19	97.66 +3.91	99.56 +1.05
				UNI2-H					
	ABMIL	91.67	83.67	87.60	98.65	95.32	95.35	95.31	98.96
UNI2-H	PathHunter + ABMIL	100 +8.32	91.84 +6.13	95.66 +8.52	99.86 +1.21	93.18 -2.14	98.40 +3.05	95.70 +0.39	99.20 +0.24
	DSMIL	90.79	89.34	90.70	95.87	94.54	94.56	94.53	99.04
UNI2-H	PathHunter + DSMIL	100 +9.20	100 +10.66	100 +9.30	100 +4.13	96.09 +1.55	98.40 +3.84	97.27 +2.74	99.60 +0.56
	TransMIL	98.19	96.94	97.67	97.37	92.28	91.93	91.80	98.73
UNI2-H	PathHunter + TransMIL	99.18 +0.99	100 +3.06	99.69 +2.02	100 +2.63	99.68 +7.40	99.52 +7.59	99.61 +7.81	100 +1.27
	DTFD-MIL (MaxS)	95.87	94.54	95.35	96.18	95.00	94.98	94.92	99.17
UNI2-H	PathHunter + DTFD-MIL	100 +4.13	100 +5.46	100 +4.65	100 +3.82	96.12 +1.12	99.20 +4.22	97.66 +2.74	99.90 +0.73
				CONCH1.5					
	ABMIL	93.01	86.73	89.92	92.73	96.12	96.07	96.09	98.52
CONCH1.5	PathHunter + ABMIL	100 +6.99	81.63 -5.10	93.02 +3.10	95.03 +2.30	99.16 +3.04	94.40 -1.67	96.88 +0.79	99.04 +0.52
	DSMIL	89.60	78.57	83.72	81.35	95.70	95.71	95.70	99.07
CONCH1.5	PathHunter + DSMIL	100 +10.40	100 +21.43	100 +16.28	100 +18.65	99.21 +3.51	100 +4.29	99.61 +3.91	99.95 +0.88
	TransMIL	74.45	73.81	68.99	93.19	94.58	94.58	94.53	98.91
CONCH1.5	PathHunter + TransMIL	100 +25.55	100 +26.19	100 +31.01	100 +6.81	97.50 +2.92	93.60 -0.98	95.70 +1.17	99.04 +0.13
	DTFD-MIL (MaxS)	92.11	84.69	88.37	96.30	95.31	95.31	95.31	99.06
CONCH1.5	PathHunter + DTFD-MIL	98.00 +5.89	100 +15.31	99.22 +10.85	99.64 +3.34	100 +4.69	99.20 +3.89	99.61 +4.30	100 +0.94

The largest gains occur on Camelyon16, where metastatic evidence is often sparse and spatially localized. Representative improvements include $+27.21$ AUC points for DSMIL+CTransPath and $+31.01$ accuracy points for TransMIL+CONCH1.5. Because several configurations approach ceiling performance, we additionally quantify uncertainty for the canonical H-Optimus-1+AttentionMIL setting. AUC increases from 99.23 to 100.00 ($+0.77$ points; 95% CI $[0, 1.94]$), whereas macro-F1 increases from 94.06 to 99.18 ($+5.12$ points; 95% CI $[0.87, 9.65]$). We therefore view the fixed-threshold classification improvement as clearer evidence than the near-ceiling AUC change. On TCGA-NSCLC, improvements are generally smaller and more metric dependent. Unlike sparse metastasis detection, subtype classification requires evidence distributed over broader tissue morphology. This suggests that the relevant distinction is not simply binary versus multi-class prediction, but the amount and spatial diversity of evidence required for a reliable slide-level decision. Camelyon16 also represents a favorable search regime because the Stage 1 class-1 (tumor) score is used during Worker warm-up and to construct and characterize the candidate space at inference time. These scores guide the search space but do not deterministically define the final evidence bag, which is selected by the learned Manager and Worker.

Table 2: Strict re-evaluation on the official Patho-Bench PANDA `isup_grade` split using **ABMIL** as the slide-level aggregator for both baseline and PathHunter. Weighted kappa (WK), accuracy (Acc), and macro F1 are reported in %. Positive changes are shown in green and negative changes in gray. Average results across two configurations to show robustness.

Backbone	Domain	WK: Baseline → PathHunter	Acc: Baseline → PathHunter	F1: Baseline → PathHunter
ABMIL-based PANDA re-evaluation				
CONCH1.5	Radboud	92.21%→93.79%(+1.58%)	74.95%→78.88%(+3.94%)	74.18%→77.90%(+3.72%)
CONCH1.5	Karolinska	93.79%→93.82%(+0.04%)	80.28%→81.69%(+1.41%)	67.17%→71.90%(+4.73%)
UNI2	Radboud	96.80%→97.32%(+0.52%)	91.36%→91.03%(-0.33%)	91.06%→90.49%(-0.57%)
UNI2	Karolinska	93.00%→94.73%(+1.73%)	81.89%→82.60%(+0.70%)	70.93%→72.23%(+1.30%)
H-Optimus1	Radboud	96.64%→97.03%(+0.39%)	90.92%→91.68%(+0.77%)	90.67%→91.47%(+0.80%)
H-Optimus1	Karolinska	95.11%→95.51%(+0.41%)	85.71%→82.70%(-3.02%)	75.88%→69.93%(-5.95%)

PANDA provides an important complementary setting. It is a six-class, cross-institution grading benchmark whose pseudo-expert targets are derived from the Stage 1 model prediction rather than from a manually specified tumor class. As shown in Table 2, weighted kappa improves across all six evaluated backbone–domain pairs, with four of six pairs also improving both accuracy and macro-F1. Additional PathoBench experiments in the appendix further show positive results on grading and immune-classification tasks, while also revealing task- and backbone-dependent behavior on more spatially distributed problems.

Takeaway. RQ1. PathHunter improves a broad range of feature-extractor–MIL combinations and its benefit is not restricted to binary metastasis detection. The largest gains occur when decisive evidence is sparse, whereas tasks requiring broader morphology show greater sensitivity to the amount and spatial coverage of the selected evidence.

5.1.2 RQ2: Where do the gains come from?

A central question is whether the Stage 3 improvement arises from learned hierarchical selection, from simply replaying the Stage 1 MIL ranking, or from adapting the prediction MIL model to the selected bags. We therefore construct a strictly matched comparison on Camelyon16 using H-Optimus-1 and AttentionMIL. All conditions use the same Stage 1 scorer, candidate construction, data split, maximum budget $(K, m, B) = (16, 16, 256)$, optimization protocol, and validation rule; only the selector or prediction-MIL adaptation condition is changed.

Table 3: Matched selector ablations on Camelyon16 using H-Optimus-1 and AttentionMIL. All conditions use the same Stage 1 scorer, candidate construction, data split, and $K/m/T = 16/16/20$.

Selector	Accuracy	Macro-F1	AUC
Static selection			
Stage-1 static top-K/top-m	93.80	93.47	98.09
Stage-3 static top-K/top-m	97.67	97.56	**100.00**
Joint-trained hierarchy: evaluation-only ablations			
Manager only (Worker disabled at eval.)	**99.23**	**99.18**	**100.00**
Worker only (Manager disabled at eval.)	97.67	97.56	**100.00**
Independently trained single-level policies			
Train Manager-only	98.45	98.34	99.95
Train Worker-only	98.45	98.35	99.31
Full PathHunter	**99.23**	**99.18**	**100.00**

The selector comparison in Table 3 shows that directly replaying the Stage 1 top-K/top-m ranking yields 93.80 accuracy, 93.47 macro-F1, and 98.09 AUC, whereas Full PathHunter reaches 99.23 accuracy, 99.18 macro-F1, and 100.00 AUC. A stronger static baseline based on the Stage 3 scores substantially narrows this gap, reaching 97.67 accuracy and 97.56 macro-F1, while already attaining 100.00 AUC. Thus, in this setting, the remaining benefit of the learned policy is most clearly reflected in the operating-point metrics rather than in ranking AUC. The evaluation-only ablations show that the Manager drives most of the gain in this setting. Disabling the Worker leaves performance unchanged at 99.23 accuracy, 99.18 macro-F1, and 100.00 AUC, whereas disabling the Manager reduces performance to 97.67 accuracy and 97.56 macro-F1, matching the Stage 3 static selector. Independently trained single-level policies show the same trend: Manager-only reaches

$98.45/98.34/99.95$ accuracy/macro-F1/AUC, while Worker-only reaches $98.45/98.35/99.31$. Removing Stage 3 PPO further reduces performance to 96.65 macro-F1 and 98.11 AUC, indicating that the final policy cannot be explained by the warm-up ranking alone. We also disentangle selector quality from prediction-MIL adaptation. Under the same frozen Stage 1 classifier, PathHunter retains a $+1.63$-point AUC advantage over static selection; after adapting separate MIL classifiers to the selected bags, the gap increases to $+2.93$ points. Thus, MIL adaptation amplifies an existing selector advantage rather than creating it. Additional stage-wise and Manager/Worker ablations are reported in Appendix L.

Takeaway. RQ2. The final gain cannot be explained solely by static Stage 1 ranking or prediction-MIL adaptation. Learned evidence selection contributes independently, with the Manager providing most of the gain in the canonical setting, while selected-bag classifier adaptation further amplifies this advantage.

5.1.3 RQ3: How does PathHunter compare with dedicated sequential-observation methods?

Standard MIL baselines do not directly test whether hierarchical policy learning improves over prior learned observation strategies. We therefore compare PathHunter with ZoomMIL and SASHA-0.2 on the same official 129-slide Camelyon16 test cohort. To reduce representation differences, their official implementations are adapted to the same H-Optimus-1 feature family used by PathHunter. These results should therefore be interpreted as controlled shared-representation comparisons rather than unchanged reproductions of the original published protocols.

Table 4: Comparison with dedicated sequential-observation methods on the official Camelyon16 test cohort over three random seeds. ZoomMIL and SASHA-0.2 are adapted to a shared H-Optimus-1 feature family.

Method	AUC	Acc.	Macro-F1
ZoomMIL (adapted)	88.20 ± 7.28	84.76 ± 5.82	82.40 ± 7.31
SASHA-0.2 (adapted)	94.19 ± 1.23	91.99 ± 0.45	91.41 ± 0.57
PathHunter	$\mathbf{99.52 \pm 0.78}$	$\mathbf{98.19 \pm 1.95}$	$\mathbf{98.06 \pm 2.12}$

As shown in Table 4, PathHunter achieves 99.52 ± 0.78 AUC, 98.19 ± 1.95 accuracy, 98.06 ± 2.12 macro-F1. PathHunter improves over adapted ZoomMIL by $+11.32$ AUC points and $+15.66$ macro-F1 points, and over adapted SASHA-0.2 by $+5.33$ AUC points and $+6.65$ macro-F1 points.

RLogist could not be reproduced unchanged because of release-level implementation and environment issues; we therefore report its published results only as descriptive context in the appendix rather than treating them as a matched comparison.

Takeaway. RQ3. Under a shared-feature, matched-test protocol, PathHunter outperforms adapted ZoomMIL and SASHA-0.2 with positive mean differences for both AUC and macro-F1. Because these are protocol-labelled adaptations, we interpret the comparison as controlled empirical positioning rather than an unchanged reproduction of the original methods.

6 Conclusion

We present PathHunter, a hierarchical RL framework for coarse-to-fine evidence acquisition in MIL. PathHunter consistently improves performance across backbone–MIL pairs and datasets, with the largest gains on sparse-evidence binary detection (up to +27.21 AUC on Camelyon16) and smaller but stable gains on multi-class tasks such as TCGA-NSCLC and PANDA. Stage-wise analysis shows that Stage 2 stabilizes local selection, while Stage 3 joint PPO with MIL adaptation drives the main slide-level improvements; a held-out evaluator study on PANDA further supports selector transferability. Limitations include fixed patch budgets, dataset-dependent warm-up priors, and reward-shaping sensitivity. Future work will focus on adaptive policies and more robust, label-efficient objectives, with potential extensions to other large sparse-evidence domains such as remote sensing and long-form video.

Reproducibility Statement

We provide an anonymized implementation of PathHunter together with the training, evaluation, and ablation scripts used in our experiments. The manuscript specifies the datasets, data splits, backbone–MIL combinations, stage-wise training procedure, patch budgets, optimization settings, and evaluation metrics required to reproduce the reported results. For stochastic experiments, we report results over multiple random seeds where applicable and keep the data split and evaluation protocol fixed across matched comparisons. We also include ablations that separately control the Manager, Worker, Stage 3 PPO, and prediction-MIL adaptation to make the contribution of each component traceable. All experiments use publicly available pathology datasets and standard pretrained feature encoders. Additional implementation details, hyperparameters, and dataset-specific preprocessing procedures are provided in the appendix and accompanying code.

Statement on the Use of Generative AI

An AI assistant assisted drafting, organization, and consistency checks against experiment records. The claims, experimental validity, and final submission remain the authors' responsibility.

Ethics Statement

PathHunter is developed for research on computational pathology and is not intended for clinical diagnosis or direct patient care. Our experiments use publicly available, de-identified pathology benchmarks and operate on pre-extracted image representations; no new patient data are collected in this work. As with other learning-based pathology systems, performance may vary across institutions, scanners, staining protocols, and patient populations, and strong benchmark performance should therefore not be interpreted as evidence of clinical validity or deployment readiness. In particular, the learned selection policy may inherit biases from the underlying data, feature encoders, and MIL models, potentially affecting which regions are prioritized. Any future clinical use would require independent external validation, subgroup and distribution-shift analyses, and appropriate oversight by domain experts. More broadly, PathHunter should be viewed as a method for studying efficient evidence acquisition rather than as an automated medical decision-making system.

References

Pierre-Luc Bacon, Jean Harb, and Doina Precup. The option-critic architecture. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31, 2017.

Chengkuan Chen, Luca L Weishaupt, Drew FK Williamson, Richard J Chen, Tong Ding, Bowen Chen, Anurag Vaidya, Long Phi Le, Guillaume Jaume, Ming Y Lu, et al. Evidence-based diagnostic reasoning with multi-agent copilot for human pathology. *arXiv preprint arXiv:2506.20964*, 2025.

Richard J Chen, Tong Ding, Ming Y Lu, Drew FK Williamson, Guillaume Jaume, Bowen Chen, Andrew Zhang, Daniel Shao, Andrew H Song, Muhammad Shaban, et al. Towards a general-purpose foundation model for computational pathology. *Nature Medicine*, 2024.

Fatemeh Ghezloo, Mehmet Saygin Seyfioglu, Rustin Soraki, Wisdom O Ikezogwo, Beibin Li, Tejoram Vivekanandan, Joann G Elmore, Ranjay Krishna, and Linda Shapiro. Pathfinder: A multi-modal multi-agent system for medical diagnostic decision-making applied to histopathology. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 23431–23441, 2025.

Tarun Gogisetty, Naman Malpani, Gugan Chandrashekhar Mallika Thoppe, and Sridharan Devarajan. Sequential attention-based sampling for histopathological analysis. *Advances in Neural Information Processing Systems*, 38:155532–155561, 2026.

Shengyi Hua, Jianfeng Wu, Tianle Shen, Kangzhe Hu, Zhongzhen Huang, Shujuan Ni, Zhihong Zhang, Yuan Li, Zhe Wang, and Xiaofan Zhang. Pathfound: An agentic multimodal model activating evidence-seeking pathological diagnosis. *arXiv preprint arXiv:2512.23545*, 2025.

Wisdom Ikezogwo, Saygin Seyfioglu, Fatemeh Ghezloo, Dylan Geva, Fatwir Sheikh Mohammed, Pavan Kumar Anand, Ranjay Krishna, and Linda Shapiro. Quilt-1m: One million image-text pairs for histopathology. *Advances in neural information processing systems*, 36:37995–38017, 2023.

Maximilian Ilse, Jakub Tomczak, and Max Welling. Attention-based deep multiple instance learning. In *International conference on machine learning*, pp. 2127–2136. PMLR, 2018.

Koray Kavukcuoglu et al. Gemini 2.5: Our most intelligent ai model. *Google DeepMind*, 25(3):2025, 2025.

Bin Li, Yin Li, and Kevin W Eliceiri. Dual-stream multiple instance learning network for whole slide image classification with self-supervised contrastive learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 14318–14328, 2021.

Ofir Nachum, Shixiang Shane Gu, Honglak Lee, and Sergey Levine. Data-efficient hierarchical reinforcement learning. *Advances in neural information processing systems*, 31, 2018.

AI Open. Introducing gpt-4.1 in the api, 2025.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Yuxuan Sun, Yunlong Zhang, Yixuan Si, Chenglu Zhu, Zhongyi Shui, Kai Zhang, Jingxiong Li, Xingheng Lyu, Tao Lin, and Lin Yang. Pathgen-1.6 m: 1.6 million pathology image-text pairs generation through multi-agent collaboration. *arXiv preprint arXiv:2407.00203*, 2024.

Yuxuan Sun, Yixuan Si, Chenglu Zhu, Xuan Gong, Kai Zhang, Pingyi Chen, Ye Zhang, Zhongyi Shui, Tao Lin, and Lin Yang. Cpath-omni: A unified multimodal foundation model for patch and whole slide image analysis in computational pathology. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10360–10371, 2025a.

Yuxuan Sun, Yixuan Si, Chenglu Zhu, Kai Zhang, Zhongyi Shui, Bowen Ding, Tao Lin, and Lin Yang. Cpathagent: An agent-based foundation model for interpretable high-resolution pathology image analysis mimicking pathologists' diagnostic logic. *arXiv preprint arXiv:2505.20510*, 2025b.

Kevin Thandiackal, Boqi Chen, Pushpak Pati, Guillaume Jaume, Drew FK Williamson, Maria Gabrani, and Orcun Goksel. Differentiable zooming for multiple instance learning on whole-slide images. In *European Conference on Computer Vision*, pp. 699–715. Springer, 2022.

Alexander Sasha Vezhnevets, Simon Osindero, Tom Schaul, Nicolas Heess, Max Jaderberg, David Silver, and Koray Kavukcuoglu. Feudal networks for hierarchical reinforcement learning. In *International conference on machine learning*, pp. 3540–3549. PMLR, 2017.

Xiyue Wang, Sen Yang, Jun Zhang, Minghui Wang, Jing Zhang, Wei Yang, Junzhou Huang, and Xiao Han. Transformer-based unsupervised contrastive learning for histopathological image classification. *Medical image analysis*, 81:102559, 2022.

Hongrun Zhang, Yanda Meng, Yitian Zhao, Yihong Qiao, Xiaoyun Yang, Sarah E Coupland, and Yalin Zheng. Dtfd-mil: Double-tier feature distillation multiple instance learning for histopathology whole slide image classification. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 18802–18812, 2022.

Boxuan Zhao, Jun Zhang, Deheng Ye, Jian Cao, Xiao Han, Qiang Fu, and Wei Yang. Rlogist: fast observation strategy on whole-slide images with deep reinforcement learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pp. 3570–3578, 2023.

A Interpretability

Figure 2 shows that PathHunter accumulates evidence along spatially coherent coarse-to-fine trajectories rather than isolated patch picks. At the manager level, selected ROIs concentrate around annotated tumor regions, indicating successful low-magnification localization. The worker then refines those ROIs into high-magnification patches with diagnostic morphology, providing a clinically interpretable trace of how slide-level evidence is gathered. This visibility is useful for auditing failure cases and distinguishes PathHunter from black-box bag aggregators.

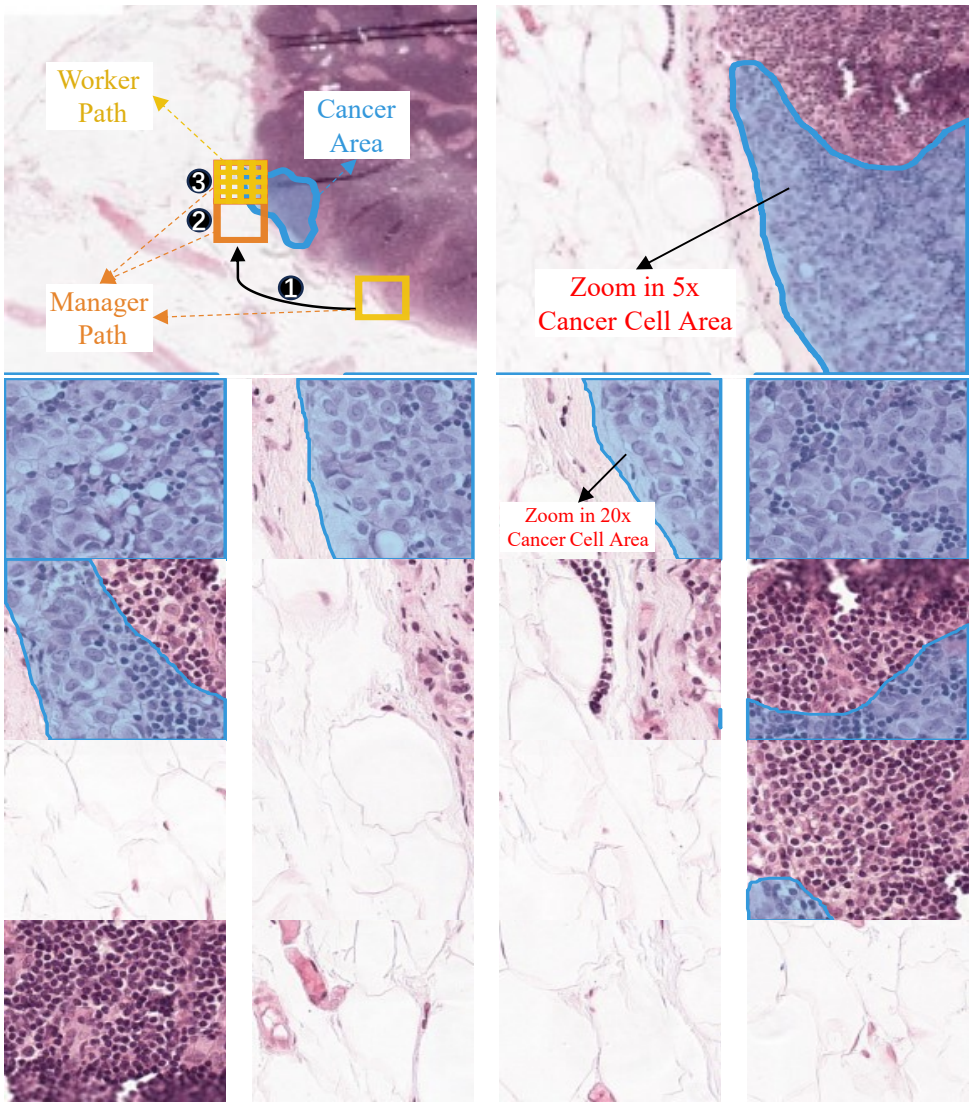


Figure 2: PathHunter search trajectories. Blue regions denote annotated cancer tissue, while yellow/orange boxes indicate the ROIs and patches selected by PathHunter. The first row shows whole-slide trajectories and manager views; the lower panels show the corresponding high-magnification Worker evidence.

A.0.1 RQ4: Does PathHunter select pathologically relevant and transferable evidence?

Table 5: Post-hoc evidence-quality analysis on Camelyon16. Lesion annotations for positive slides are used only after evidence selection and never for training, candidate construction, reward computation, or checkpoint selection. Favorable changes are shown in green. Values are in %.

Evidence Metric	**Static**	**PathHunter**	**Δ**
Positive slides: tumor enrichment and coverage			
Patch precision ↑	36.20	**42.78**	+6.58
Hit@budget ↑	88.89	**96.30**	+7.41
Tumor-patch recall ↑	52.55	**64.29**	+11.74
Lesion coverage ↑	43.34	**48.81**	+5.47
Negative slides: suppression of spurious positive evidence			
False-positive rate ↓	93.75	**1.25**	-92.50
Mean positive probability ↓	72.68	**7.02**	-65.66
High-positive patch fraction ↓	53.96	**4.12**	-49.84

Classification accuracy alone does not establish that the selected patches correspond to meaningful pathological evidence. We therefore perform an annotation-only analysis on 27 positive Camelyon16 validation slides. XML lesion annotations are used only after patch selection and do not enter training, candidate construction, reward computation, or checkpoint selection. Static selection and PathHunter use the same H-Optimus-1 features, AttentionMIL scorer, and realized per-slide evidence budget.

PathHunter improves all four positive-slide evidence metrics: patch precision increases from 36.20 to 42.78, Hit@budget from 88.89 to 96.30, tumor-patch recall from 52.55 to 64.29, and lesion coverage from 43.34 to 48.81. The effect is complementary on negative slides: the false-positive rate decreases from 93.75 to 1.25, mean positive probability from 72.68 to 7.02, and the fraction of high-positive patches from 53.96 to 4.12. Together, these results indicate that PathHunter both enriches tumor-relevant evidence on positive slides and suppresses spurious positive evidence on negative slides.

Table 6: Held-out evaluator validation on PANDA Karolinska `fold_0`. The PathHunter policy is trained with UNI2+ABMIL on Karolinska (`Karo f0`) and evaluated using an independent frozen Stage 1 UNI2+ABMIL model trained on Radboud (`Rad f0`). Positive changes are shown in green. Values are reported in %.

Evidence	WK	Acc.	AUC	Macro-F1
Full-slide evidence	25.84	44.47	75.37	35.52
PathHunter-selected evidence	**50.69** +24.85	**50.50** +6.04	**79.13** +3.76	**39.29** +3.77

We additionally evaluate whether the selected evidence remains useful outside the source evaluator. A PANDA policy trained on Karolinska `fold_0` is evaluated using an independent frozen Stage 1 UNI2+ABMIL model trained on Radboud `fold_0`. Under this held-out evaluator, weighted kappa improves from 25.84 to 50.69, accuracy from 44.47 to 50.50, AUC from 75.37 to 79.13, and macro-F1 from 35.52 to 39.29.

Broader cross-head and cross-backbone experiments in the appendix show mixed transfer: selected evidence remains useful to several unseen evaluators, but the effect is not uniform. We therefore interpret these experiments as evidence of *partial evaluator transfer*, rather than evaluator invariance.

Takeaway. RQ4. PathHunter produces evidence bags with greater tumor enrichment and lesion coverage while substantially suppressing false-positive evidence on negative slides. The selected evidence also transfers to independent evaluators in several settings, although this transfer is evaluator dependent. These results support model-based evidence localization, not yet clinical validation.

A.0.2 RQ5: What are the computational and evidence-budget trade-offs?

Because PathHunter operates on precomputed multi-scale representations, its computational benefit should not be interpreted as avoiding whole-slide feature extraction. We therefore report an end-to-end cost decomposition that includes tissue processing, high- and low-magnification embedding extraction, cross-scale assembly, Stage 1 scoring, hierarchical selection, and final MIL aggregation.

Under the Camelyon16 H-Optimus-1 implementation, full-bag MIL requires 129.09 s per slide, whereas PathHunter requires 154.15 s per slide. PathHunter therefore adds 25.07 s per slide, corresponding to a 19.4% increase (1.194×). We consequently do not claim an end-to-end inference speedup. Instead, PathHunter reduces the final aggregation bag from approximately 11,052.5 patches per slide to 233.9 selected patches while providing an explicit, auditable selection trace.

The required evidence budget is task dependent. The total acquired evidence is controlled by $B = Km$: increasing K increases spatial exploration, while increasing m increases within-ROI evidence collection. Increasing the candidate-pool size T alone does not increase the amount of acquired evidence. For tasks whose diagnostic morphology is more spatially distributed, larger or adaptive K, m, and consequently B can provide broader coverage. Our broader PathoBench results show that the framework can benefit grading and immune-classification tasks as well as sparse metastasis detection, while also indicating that the optimal evidence allocation is task and backbone dependent.

Table 7: End-to-end computational cost on Camelyon16 using H-Optimus-1. All values are amortized seconds per slide. PathHunter includes additional multi-scale preprocessing and hierarchical evidence selection.

Component	Full-bag MIL	PathHunter
WSI preprocessing and feature extraction		
Tissue segmentation	7.431	7.431
20× coordinates	2.554	2.554
5× coordinates	–	1.902
20× embedding	112.198	112.198
5× embedding	–	19.717
Cross-scale assembly/mapping	–	9.940
Evidence selection and prediction		
Feature loading	6.902	0.064
Stage-1 scoring/candidate construction	–	0.165
Manager/Worker selection	–	0.181
Final MIL aggregation	0.003	0.001
End-to-end total	**129.089**	**154.153**

Takeaway. RQ5. PathHunter trades additional preprocessing and selection cost for a bounded, structured evidence bag rather than providing an end-to-end speedup. Evidence coverage is controlled directly by $B = Km$, making the appropriate budget a task-dependent design choice, especially when diagnostic morphology is spatially distributed.

B Supplementary Material

C Limitations and Future Work.

Despite its effectiveness, PathHunter has several limitations that motivate future work. First, the patch budget B is currently treated as a dataset-dependent hyperparameter. While we observe that allocating a larger high-level spatial search space generally improves performance, the fixed-budget design limits adaptability across datasets with different tissue complexity and lesion sparsity. An important direction is to develop *adaptive budget allocation*, where the agent dynamically adjusts exploration depth and spatial coverage based on intermediate evidence or uncertainty. This would move the method closer to truly minimal evidence acquisition rather than fixed-budget evidence collection.

Second, the three-stage training pipeline (Stage 1–3) and the intensive reward shaping introduce a form of *self-evolving optimization*, where the policy is guided by signals derived from the evolving MIL backbone. Although our ablations suggest that performance gains are not driven by trivial feedback loops, such self-referential rewards may still bias exploration toward the model's current inductive preferences. Future work could investigate *decoupled or externally calibrated reward mechanisms*, such as uncertainty-aware objectives or auxiliary critics, to improve robustness and mitigate confirmation bias.

Third, the current framework relies on slide-level labels y under weak supervision. A promising direction is to move toward *self-supervised or label-efficient learning*, where the agent can learn effective search policies without explicit supervision. This may involve integrating contrastive objectives, generative modeling, or cross-slide consistency constraints to replace or complement likelihood-based rewards.

Overall, advancing adaptive computation, robust reward design, and self-supervised learning will be critical for improving the scalability and generalization of hierarchical RL-based WSI analysis.

D Datasets, Splits, and Evaluation Protocol

We evaluate PathHunter on Camelyon16, TCGA-NSCLC, and PANDA. Camelyon16 contains 270 training and 129 testing WSIs for metastasis detection. We extract non-overlapping 256×256 patches

at 20× and 5× after background removal, and use 90% of the official training set for training and 10% for validation. TCGA-NSCLC contains LUAD/LUSC WSIs with 759 training and 256 testing slides under the baseline split, using 60%/15%/25% train/validation/test and patient-level separation when multiple slides belong to the same subject. PANDA follows the official Patho-Bench `isup_grade` split used in our rebuttal-oriented analyses.

Unless otherwise specified, experiments are averaged over 5 random seeds. We report accuracy and area under the ROC curve (AUC). For binary tasks, AUC is computed from the positive-class probability; for multi-class tasks, we use a one-vs-rest strategy. We additionally report precision, recall, and weighted kappa when appropriate. Model checkpoints are selected primarily by validation AUC, with validation accuracy as a secondary criterion.

For the main-paper transfer matrix, we pair three pathology foundation models (CTransPath, UNI2-H, CONCH1.5) with four MIL heads (ABMIL, DSMIL, TransMIL, DTFD-MIL). Stage 1 trains the MIL baseline for 50 epochs with learning rate 10^{-4}, while Stage 2 and Stage 3 train the Worker and the joint Manager-Worker policy for 10 and 20 epochs, respectively, also with learning rate 10^{-4}. Detailed PPO settings, patch budgets, and manager/worker architecture choices are given in Appendix G, Appendix I, and Appendix J.

E Training and Inference Time Comparison

Compared to CPathAgent Sun et al. (2025b), which requires extensive computational resources involving 8 NVIDIA H800-80G GPUs and a total training time of 73 hours across three stages (9 hours for multimodal alignment, 25 hours for pathology task training, and 39 hours for agent capability development), our PathHunter model offers significant efficiency advantages. Utilizing a single NVIDIA H20 GPU, PathHunter completes the entire training pipeline within just 5 hours. Additionally, PathHunter demonstrates rapid inference capabilities, processing 20 slides in approximately 5.5 seconds (0.275 seconds per slide) with minimal GPU memory usage (peak of 0.083 GB). This substantial reduction in both hardware demands and training duration, combined with fast inference speed and low resource consumption, not only lowers the overall computational cost but also facilitates faster experimentation and deployment, making PathHunter a more practical and accessible solution for pathology-related tasks.

F WSI Environment Construction

We formulate each WSI as one episode of a two-level Markov decision process. Only the slide-level label $y \in \{0, ..., m\}$ (c is the class number) is used (weak supervision); no patch-level annotations are required at training time.

Input and hierarchy. For each WSI we assume pre-extracted multi-scale features: high-magnification patch features $\mathbf{F}_{\text{high}} \in \mathbb{R}^{M \times D}$ and low-magnification features (e.g., one vector per patch at low mag), plus patch coordinates. A mapping from low-magnification "cells" (ROIs) to high-magnification patches is precomputed and stored, each of the N cells corresponds to a list of patch indices. Thus the action hierarchy is: the *manager* selects one cell (ROI) among a restricted set of candidates; the *worker* then selects m patches without replacement from that cell. Within each cell, patches are grouped into a fixed 16×16 (can be modified according to different multi-scale) spatial grid (bins); the worker's candidates are these bins (each represented by the mean feature of unselected patches in that bin), so the low-level action space has size $M_{\text{low}} = 256$ per ROI.

Budget and rounds. Per episode we enforce a total patch budget B and a fixed number of rounds K, with $K \cdot m \leq B$. Initially the selected set S is empty. In each round $k = 1, \ldots, K$, the manager observes the current state (cell features, visit mask, and optional candidate mask) and picks one cell; the worker then selects m patches from that cell and adds them to S. No cell is visited more than once (enforced by the visit mask).

Observations. High-level (manager) cell features have dimension $D + 2$ when MIL scores are appended: for each cell we use the mean of the low-mag patch features in that cell, plus the max and top-k mean of the MIL instance scores over its patches. The manager's action space is fixed to

the top-N patches globally (by instance score); each action index maps to the cell containing that patch. Thus the manager effectively chooses among cells that contain the most suspicious patches. Optionally, the candidate set is further restricted to the top-T most suspicious cells. The visit mask marks which of these N positions have already been chosen; the candidate mask combines validity (e.g., tissue) and optional top-T restriction. Low-level (worker) given the chosen cell, the worker sees candidate features of size $M_{\text{low}} \times D$ (one vector per bin, mean of unselected patches in that bin), a binary candidate mask (which bins still have patches), and the cell feature vector (same $D + 2$ as above) as context.

Reward. Let S_k denote the set of patches selected after round k, and $P_\psi(y \mid S)$ the MIL model's probability of the slide label y given the bag S. The step reward at round k is

$$r_k = \log P_\psi(y \mid S_k) - \log P_\psi(y \mid S_{k-1}) - \lambda_{\text{cost}} + \eta \cdot \text{shaping}(S_k \setminus S_{k-1}), \tag{19}$$

where λ_{cost} is a small cost per round and the shaping term uses the MIL instance scores of the patches selected in this round (e.g., max or mean of top-q) to provide a dense learning signal when slide-level labels alone are weak. The episode terminal reward is $R_{\text{final}} = \log P_\psi(y \mid S)$. Returns are computed with discount γ and GAE for PPO.

Patch instance scores per MIL. The environment precomputes a scalar "instance score" for every patch in the WSI; these scores drive cell suspiciousness (for the manager's candidate set and cell features) and the reward shaping term. For Camelyon16, we always use the class-1 (tumor) logit as the local ranking target during warm-up. For NSCLC and PANDA, we use the Stage 1 MIL prediction for the relevant class, so the pseudo-target comes from the model rather than from a manually fixed tumor label. The way the score is obtained depends on the MIL backbone:

- **ABMIL:** Each patch is treated as a bag of size one. The shared encoder yields a single embedding z; the same classifier used for bag-level prediction is applied to z. The instance score is the classifier's logit for the target class.
- **DSMIL:** The instance branch (IClassifier) outputs per-patch logits. The instance score is the target-class logit from this IClassifier for each patch (same as the instance pathway used in the bag prediction $0.5 \cdot \text{bag} + 0.5 \cdot \max(\text{instance})$).
- **TransMIL:** Each patch is passed through the same first-layer projection fc_1 as in the bag forward, then through the same classifier fc_2 (used for bag logits). The instance score is the positive-class logit of this per-patch forward.
- **DFTD-MIL:** Each patch is passed through the dimension-reduction module and then the instance-level classifier. The instance score is the classifier's positive-class logit for that patch (the same classifier used on reduced patch features in the two-tier pipeline).

In all cases we take the selected class logit as the score so that higher values indicate higher suspicion. These scores are computed once per episode at `reset` and reused for cell aggregation (max and top-k mean per cell) and for the shaping term over the patches selected in each round.

Episode flow. `reset(slide+label)` loads the WSI features, computes patch and cell suspiciousness scores (from the MIL model's instance scores), and sets the top-N action-to-cell mapping. Then for K steps: the manager receives `get+high+obs()`, outputs an action (cell index in the top-K list), and the environment updates the visit mask; the worker receives `get+low+obs(cell+idx)` and selects N patches (e.g., via m categorical steps without replacement); `step+low+batch(patch+ids)` adds those patches to S, computes r_k, and returns. After K rounds, the episode ends and the final bag S is used for the terminal reward and for the supervised MIL loss on the slide label.

G Experimental Hyperparameters

Table 8 summarizes the main hyperparameters used in our experiments. Budget B is the total number of patches selected per slide per episode; K is the number of manager steps; m is the number of patches the worker selects per cell ($K \cdot m \leq B$). Patch budget is the maximum patches per slide in Stage 1 MIL. Cost weight is the coefficient for patch-count penalty; shaping weight scales the

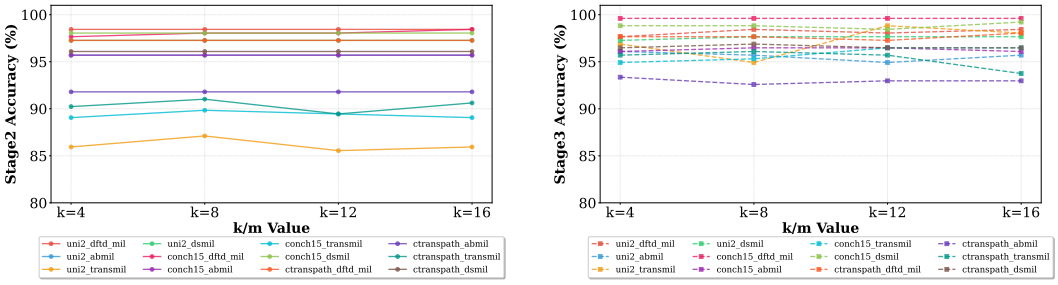
MIL-based shaping term (max or mean-top-q of instance scores). Top-T restricts the manager to the top-T suspicious cells; Top-N ranks cells by global patch suspicion; cell MIL top-k is used for cell-level features. Stage 1 uses Adam with $\beta = (0.5, 0.9)$ and optional early stopping on validation AUC. Stage 3 entropy coefficient is applied in PPO updates. Feature dimension 1536 corresponds to backbones such as UNI2; patch size is dataset-specific.

Table 8: Experimental hyperparameters used for training and evaluation.

Category	Parameter	Value
WSI & RL budget		
	Budget B (patches per WSI)	256
	Rounds K	16
	Patches per ROI m	16
Reward & shaping		
	Cost weight λ_{cost}	0.01
	Shaping weight	0.5
	Shaping type	max; mean+top+q ($q = 3$)
Manager action space		
	Top-T cells	20
	Top-N patches	25
	Cell MIL top-k	5
Stage 1: MIL baseline		
	Epochs	50
	Learning rate	1×10^{-4}
	Batch size	32
	Early stopping patience	10
Stage 2: Worker-only		
	Epochs	10
	Learning rate	1×10^{-4}
Stage 3: Joint PPO		
	Joint epochs	20
	Manager learning rate	1×10^{-4}
	Worker learning rate	1×10^{-4}
	Base model learning rate	1×10^{-4}
	Discount γ	0.99
	GAE λ	0.95
	PPO clip range	0.2
	Value loss coefficient	0.5
	Entropy coefficient	0.01
Feature & backbone		
	Feature dimension	1536(UNI2)/768(others)
	Patch size	256(Camelyon16) or 512(NSCLC)
	Magnification (high / low)	20× / 5×
	Low-level grid per ROI	16×16

H Stage-2 and Stage-3 Hyperparameter Sensitivity

We additionally analyze how PathHunter behaves under different high-level and low-level patch allocation choices. Figure 3 compares Stage 2 and Stage 3 accuracy trends across different combinations of high-level rounds K and per-round low-level patch selections m, with $K, m \in \{4, 8, 12, 16\}$.



(a) Stage 2 accuracy trends. (b) Stage 3 accuracy trends.

Figure 3: Accuracy sensitivity to the hierarchical patch-allocation hyperparameters (K, m). The left panel shows Stage 2 pseudo-expert warm-up behavior, while the right panel shows Stage 3 joint optimization behavior.

Two trends are worth noting. First, Stage 2 is comparatively stable across different (K, m) settings, which is consistent with its role as a local pseudo-expert warm-up stage rather than the main source of final performance. Second, Stage 3 is more sensitive to the patch-allocation budget, indicating that the final acquisition policy depends more strongly on how global exploration and local refinement are balanced. In practice, this supports our design choice to treat (K, m) as task-dependent hyperparameters rather than fixed universal defaults.

These trends also help explain why stronger Stage 3 gains require more care than Stage 2 warm-up alone: Stage 2 mainly learns a restricted local ranking policy, whereas Stage 3 must coordinate high-level ROI acquisition with low-level evidence refinement under slide-level feedback.

I RL MANAGER NETWORK ARCHITECTURE

The manager selects one low-magnification "cell" (ROI) at each step. Table 9 gives the architecture; descriptions follow below.

Table 9: RL Manager (high-level policy) network architecture.

Component	Specification	Value
Input	Cell features	$\mathbf{x} \in \mathbb{R}^{B\times N\times D}$
	Visit mask	$\mathbf{m}_{\text{visit}} \in \mathbb{R}^{B\times N}$
	Optional masks	tissue, candidate-cell (top-T)
Cell encoder	Layer 1	$\text{Linear}(D_{\text{in}} \to 256)$, ReLU, Dropout(0.1)
	Layer 2	$\text{Linear}(256 \to 256)$, ReLU
Policy head		$\text{Linear}(256 \to 1)$
Critic	Input	$\bar{\mathbf{h}} = \frac{1}{N}\sum_{n=1}^{N} \mathbf{h}_{:,n,:}$
	Layers	$\text{Linear}(256 \to 256)$, ReLU; $\text{Linear}(256 \to 1)$
Output		Logits (per cell), scalar value

Where B is batch size; N is the number of low-mag cells; D is cell feature dim ($D_{\text{feat}} + 1$ if visit mask appended). The visit mask is binary over already-visited cells; their logits are set to $-\infty$. Tissue and candidate-cell masks can further restrict the action space. $D_{\text{in}} = D$. The cell encoder is shared over cells and outputs $\mathbf{h} \in \mathbb{R}^{B\times N\times 256}$. The policy head yields one logit per cell, then masking and softmax give $\pi_H(c \mid s)$. The critic uses mean pooling over cells (so variable N per WSI) and a two-layer MLP to output $V(s)$.

J RL WORKER NETWORK ARCHITECTURE

The worker selects m patches without replacement from the candidate patches in the manager-chosen cell. Table 10 gives the architecture; descriptions follow below.

Table 10: RL Worker (low-level policy) network architecture.

Component	Specification	Value
Input	Candidate patch features	$\mathbf{p} \in \mathbb{R}^{B\times M\times D_p}$
	Candidate mask	$\mathbf{m}_{\text{cand}} \in \mathbb{R}^{B\times M}$
	Cell feature	$\mathbf{c} \in \mathbb{R}^{B\times D_c}$
Patch encoder	Input	$[\mathbf{p}; \mathbf{c}]$ (concat; $\mathbf{c}$ broadcast to $[B, M, D_c]$)
	Layer 1	$\text{Linear}(D_p + D_c \to 256)$, ReLU, Dropout(0.1)
	Layer 2	$\text{Linear}(256 \to 256)$, ReLU
Policy head		$\text{Linear}(256 \to 1)$
Critic	Input	$\bar{\mathbf{h}} = \frac{1}{M}\sum_{j=1}^{M} \mathbf{h}_{:,j,:}$
	Layers	$\text{Linear}(256 \to 256)$, ReLU; $\text{Linear}(256 \to 1)$
Output		Logits (per patch), scalar value

Where M is the number of candidate patches in the cell; D_p is patch feature dim; D_c is cell feature dim (e.g., $D_p + 2$ with MIL scores, 2 is the datasets' class number). The candidate mask marks valid patches; invalid positions get logits $-\infty$. Patch and cell features are concatenated per patch and fed to a two-layer MLP (hidden 256, dropout 0.1), giving $\mathbf{h} \in \mathbb{R}^{B\times M\times 256}$. The policy head gives one logit per patch; after masking, softmax yields $\pi_L(p \mid s, c)$. The worker selects m patches without

replacement per step; after each choice the patch is masked out. The critic uses mean-pooled patch embeddings and a two-layer MLP to output $V(s, c)$.

K TransMIL Scoring, Stage-2 Warm-Up, and Reproducibility

Patch or instance scores in PathHunter are backbone-dependent and are used only as a *local prior* for cell ranking, pseudo-expert construction, and optional reward shaping; they are not assumed to be faithful attribution maps. This distinction is especially important for TransMIL, where no native per-patch diagnostic score is produced by the bag-level architecture.

For TransMIL, we therefore use a proxy instance score: each patch is passed through the same first-layer projection fc_1 as in the bag forward and then through the same classifier fc_2 used for bag logits. The resulting per-patch logit is used only to rank candidate evidence during Stage 2 and to build manager-side cell features; the final Stage 3 policy is still optimized against the slide-level objective $\log P_\psi(y \mid S)$ and can deviate from this proxy. This is why TransMIL is relatively weak in Stage 2-only ablations yet improves substantially after Stage 3 joint optimization.

This clarification is also the reason we describe Stage 2 as *pseudo-expert warm-up* rather than curriculum learning. Top-m in Stage 2 defines only the within-ROI warm-up target; it does not determine the final patch budget, nor does it constrain Stage 3 to replay a fixed ranking. On Camelyon16, the warm-up uses the class-1 tumor logit explicitly, whereas on NSCLC and PANDA the pseudo-target is derived from the Stage 1 MIL prediction rather than manually fixed to a tumor class.

At inference time, Stage 1 MIL instance scores are used to construct and characterize the Manager candidate space. They do not directly determine the final patch set: the final evidence bag is selected by the learned Manager and Worker policies rather than by deterministic top-score ranking. The Worker operates on a fixed 16×16 grid inside each ROI, selecting bins sequentially and then sampling one patch from each chosen bin without replacement. As a result, reproducibility depends on the learned acquisition policy and the selected patch budget, not on direct access to patch labels or a hand-crafted tumor detector.

Overall, the TransMIL case illustrates the intended role of Stage 2 scores in PathHunter: they provide a weakly informative local ranking signal that stabilizes early acquisition, while Stage 3 performs the actual task-aligned optimization under slide-level feedback.

L More Ablation Study

L.1 Ablation Study with Manager and Worker

Table 11 reports an ablation study on the Camelyon16 dataset using the UNI2-H backbone, evaluating the individual and combined contributions of the Manager and Worker modules in our PathHunter framework. The high level Manager action space is in the top N patch instance. The full model, "Full PathHunter (Manager + Worker)," which employs both components, achieves the best results with an accuracy of 95.35 and an AUC of 98.83, demonstrating robust and reliable performance. When the Manager module is disabled and ROIs are randomly selected across the WSI ("Manager disabled (random ROI in top N, Worker)"), there is a substantial degradation in performance, reflected by a drop in accuracy to 63.57 and AUC to 72.14. This underscores the critical role of the Manager in meaningful ROI localization. The "Worker disabled (Manager only, random patch)" setting, which applies random patch selection at the lower level without the Worker module, with a slightly reduced AUC (97.88). This result demonstrates the effectiveness of worker patch selection. Finally, when both the Manager and Worker are disabled ("Manager and worker disabled"), resulting in random patch selection without any learned guidance, the model suffers poor performance, with an accuracy of 66.67 and AUC of 67.55. This confirms the necessity of both components working synergistically for optimal classification and localization.

Overall, the experiments validate that the collaborative functioning of Manager and Worker modules is essential for high-performance whole-slide image analysis.

Table 11: Ablation study on Camelyon16 based on UNI2-H evaluating the impact of Manager and Worker modules in the PathHunter framework.

Methods	Accuracy	AUC
Full PathHunter (Manager + Worker)	**95.35**	**98.83**
Manager disabled (random ROI in top N, Worker)	63.57	72.14
Worker disabled (Manager only, random patch)	95.35	97.88
Manager and worker disabled	66.67	67.55

L.2 Ablation Study with Top m methods

Table 12 compares the classification performance between the traditional two-level top-m m patch selection strategy and our proposed PathHunter method on the Camelyon16 dataset. All models utilize the UNI2-H backbone combined with different MIL architectures. We report accuracy, recall, precision, AUC, and the mean tumor cell ratio (MeanRatio) in selected patches. The evaluation is conducted over 129 whole slide images.

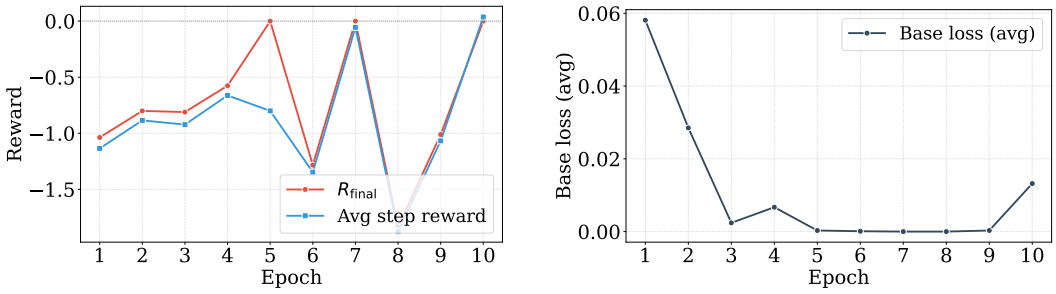
Our RL method consistently achieves substantially higher accuracy and precision than the two-level baseline across all MIL variants, demonstrating its ability to better distinguish cancerous slides. For example, accuracy improves markedly from 38.76% to 99.99% in DSMIL and from 44.19% to 99.99% in DFTD MIL. While recall remains comparably high for both methods due to the imbalanced dataset, RL yields greatly enhanced AUC values, reflecting more reliable classification confidence.

Crucially, RL selects patches with higher tumor cell ratios compared to the two-level approach, indicating more effective and focused exploration of relevant regions. This improved patch selection suggests that RL learns to prioritize diagnostically meaningful areas, which may contribute to its superior overall performance.

In summary, these results validate the effectiveness of our RL framework in jointly optimizing patch selection and slide-level classification, surpassing the conventional two-level top-m strategy under various MIL backbones.

Table 12: Comparison of Two-level top m and RL methods on Camelyon16 using UNI2-H backbone with different MIL architectures. Metrics include accuracy (Acc), recall, precision, AUC, and the mean tumor cell ratio in selected patches (MeanRatio). The number of tested slides is 129 in all cases. Numbers higher between two level and PathHunter are **bolded**.

Model	Method	Acc	Recall	Precision	AUC	MeanRatio
uni2+abmil	two+level	44.19	**100.0**	41.18	78.98	0.1980
uni2+abmil	PathHunter	**96.90**	91.84	**100.0**	**91.84**	**0.2318**
uni2+dsmil	two+level	38.76	**100.0**	38.28	50.00	0.2392
uni2+dsmil	PathHunter	**100.0**	100.0	**100.0**	**100.0**	**0.2808**
uni2+transmil	two+level	39.53	**100.0**	38.59	60.43	**0.0061**
uni2+transmil	PathHunter	**100.0**	100.0	**100.0**	**100.0**	0.0060
uni2+dftd+mil	two+level	44.19	**100.0**	40.50	96.20	0.2158
uni2+dftd+mil	PathHunter	**100.0**	100.0	**100.0**	**100.0**	**0.2534**



(a) Reward: R_{final} and mean step reward $\bar{r}_k$ vs. epoch. (b) Average base-model (CE) loss vs. epoch.

Figure 4: **Stage3 training (10 epochs, UNI2 + Camelyon16 + ABMIL, Stage1 frozen).** (**a**) Reward curves: final episode reward R_{final} and mean per-step reward $\bar{r}_k$; both fluctuate and trend toward zero by epoch 10. (**b**) Average base-model cross-entropy loss drops quickly and remains near zero.

M Training Process Analysis of Stage 3

We report Stage 3 joint training with a frozen Stage 1 model: UNI2 backbone, Camelyon16 dataset, and ABMIL. The test set has 129 slides; we train for 10 epochs and record reward, test metrics, and base loss per epoch. Figure 4 shows two panels in one row.

Panel (a) Reward. The final episode reward R_{final} and the mean per-step reward $\bar{r}_k$ (over round rewards r_k) are plotted vs. epoch. Both vary across epochs; R_{final} ranges from about -1.84 to near 0, and $\bar{r}_k$ reaches slightly above 0 by epoch 10, reflecting improved policy returns as training progresses.

Panel (b) Base loss. The average base-model (CE) loss over training bags drops sharply in the first few epochs (from $\sim$0.058 to $\sim$0.002) and then stays near zero, consistent with the base classifier fitting the selected patches while the RL agents refine patch selection.

N Transfer Study on Different Datasets

Table 13: Cross-domain performance on different OOD tasks (mean±std). Camelyon16 models are trained on TCGA-NSCLC, while TCGA-NSCLC models are trained on Camelyon16. All metrics are reported in %. Δ denotes the absolute change in percentage points (PathHunter − baseline). Positive changes are shown in green and negative changes in gray.

	Camelyon16				TCGA-NSCLC			
Method	**Prec.**	**Rec.**	**Acc.**	**AUC**	**Prec.**	**Rec.**	**Acc.**	**AUC**
				UNI2				
ABMIL	64.68±0.00	51.42±0.00	62.79±0.00	53.06±0.00	37.51±0.00	47.95±0.00	46.88±0.00	37.04±0.00
+PathHunter	66.67±0.00	4.08±0.00	62.79±0.00	53.45±0.13	47.33±0.13	92.16±0.32	46.09±0.25	37.72±0.12
Δ (pp)	+1.99	-47.34	+0.00	+0.39	+9.82	+44.21	-0.79	+0.68
DSMIL	56.10±0.00	50.40±0.00	62.02±0.00	49.74±0.00	52.45±0.00	51.69±0.00	52.34±0.00	55.38±0.00
+PathHunter	100.00±0.00	61.63±2.71	85.43±1.03	99.98±0.01	95.41±0.49	76.48±1.20	86.72±0.65	96.63±0.13
Δ (pp)	+43.90	+11.23	+23.41	+50.24	+42.96	+24.79	+34.38	+41.25
TransMIL	56.10±0.00	50.40±0.00	62.02±0.00	45.56±0.00	45.31±0.00	48.96±0.00	50.00±0.00	48.47±0.00
+PathHunter	89.97±2.60	87.35±3.00	91.47±1.63	96.85±0.23	72.22±6.53	20.48±2.80	57.34±2.03	66.70±1.28
Δ (pp)	+33.87	+36.95	+29.45	+51.29	+26.91	-28.48	+7.34	+18.23
DTFD-MIL (MaxS)	34.50±5.02	47.91±0.50	59.22±0.38	43.13±0.56	55.03±1.50	52.20±0.64	51.33±0.63	45.83±0.28
+PathHunter	97.78±1.38	88.98±2.08	95.04±0.93	99.71±0.11	67.70±1.53	93.44±0.32	75.00±1.52	83.27±1.04
Δ (pp)	+63.28	+41.07	+35.82	+56.58	+12.67	+41.24	+23.67	+37.44
				CTransPath				
ABMIL	41.31±0.00	44.23±0.00	37.21±0.00	41.28±0.00	55.35±0.00	51.40±0.00	50.39±0.00	51.82±0.00
+PathHunter	34.62±0.00	73.47±0.00	37.21±0.00	41.28±0.00	49.84±0.08	98.40±0.00	50.86±0.16	53.83±0.14
Δ (pp)	-6.69	+29.24	+0.00	+0.00	-5.51	+47.00	+0.47	+2.01
DSMIL	51.03±0.00	50.94±0.00	46.51±0.00	46.84±0.00	47.36±0.00	49.49±0.00	48.44±0.00	56.87±0.00
+PathHunter	94.97±0.91	100.00±0.00	97.98±0.38	99.94±0.01	59.39±0.21	99.20±0.00	66.48±0.29	89.94±0.08
Δ (pp)	+43.94	+49.06	+51.47	+53.10	+12.03	+49.71	+18.04	+33.07
TransMIL	42.09±0.00	42.16±0.00	39.53±0.00	38.24±0.00	50.21±0.00	50.21±0.00	50.39±0.00	52.60±0.00
+PathHunter	56.27±1.42	88.16±2.00	69.46±1.60	85.94±0.99	66.34±1.57	63.36±1.92	66.41±1.42	78.37±0.19
Δ (pp)	+14.18	+46.00	+29.93	+47.70	+16.13	+13.15	+16.02	+25.77
DTFD-MIL (MaxS)	49.56±0.66	49.54±0.68	50.85±0.62	44.38±0.22	74.51±0.00	50.38±0.00	49.22±0.00	56.80±0.04
+PathHunter	89.46±1.31	89.80±1.29	92.09±0.58	97.14±0.44	52.43±0.27	100.00±0.00	55.70±0.47	84.64±0.35
Δ (pp)	+39.90	+40.26	+41.24	+52.76	-22.08	+49.62	+6.48	+27.84
				CONCH1.5				
ABMIL	64.68±0.00	51.42±0.00	62.79±0.00	43.88±0.00	57.93±0.00	50.73±0.00	49.61±0.00	49.89±0.00
+PathHunter	29.15±2.32	14.29±1.29	54.26±0.85	45.62±0.21	49.21±0.00	100.00±0.00	49.61±0.00	50.40±0.06
Δ (pp)	-35.53	-37.13	-8.53	+1.74	-8.72	+49.27	+0.00	+0.51
DSMIL	31.01±0.00	50.00±0.00	62.02±0.00	50.13±0.00	54.59±0.00	50.69±0.00	49.61±0.00	54.80±0.00
+PathHunter	100.00±0.00	63.27±2.24	86.05±0.85	100.00±0.00	57.52±0.43	99.20±0.00	63.83±0.63	86.33±0.34
Δ (pp)	+68.99	+13.27	+24.03	+49.87	+2.93	+48.51	+14.22	+31.53
TransMIL	81.25±0.00	51.02±0.00	62.79±0.00	33.01±0.00	44.23±0.00	44.35±0.00	44.53±0.00	40.26±0.00
+PathHunter	63.00±21.82	3.67±1.53	62.33±0.79	41.19±2.59	62.58±0.89	52.96±1.18	61.56±0.63	65.46±0.61
Δ (pp)	-18.25	-47.35	-0.46	+8.18	+18.35	+8.61	+17.03	+25.20
DTFD-MIL (MaxS)	33.87±6.87	48.08±0.10	59.53±1.14	33.80±0.96	24.41±0.00	50.00±0.00	48.83±0.00	53.68±0.74
+PathHunter	100.00±0.00	77.55±4.83	91.47±1.83	98.79±0.65	50.40±0.22	100.00±0.00	51.95±0.43	75.06±0.58
Δ (pp)	+66.13	+29.47	+31.94	+64.99	+25.99	+50.00	+3.12	+21.38

We evaluate the effectiveness of the proposed PathHunter framework on cross-domain same-task and different-task out-of-distribution (OOD) generalization, using three different feature extractors (UNI2-H, CTransPath, CONCH15) and four MIL baselines (ABMIL, DSMIL, TransMIL, DTFD-MIL).

Different-task OOD Results (Table 13): Across all feature backbones, integrating PathHunter consistently boosts overall performance metrics (Accuracy and AUC) on OOD datasets, often substantially. For instance, DSMIL and DTFD-MIL show remarkable improvements of 20-50%+ in Accuracy and AUC across both Camelyon16 and TCGA-NSCLC datasets, demonstrating PathHunter's robustness in domain shifts. Precision and recall gains vary by method and dataset; however, notable increases in recall indicate enhanced sensitivity to positive instances under distributional changes.

Table 14: Cross-domain performance on the same-task OOD setting, where models are trained on Camelyon16 and evaluated on Camelyon17 over five seeds. Values are reported as mean±std in %. Δ denotes the absolute change in percentage points (PathHunter − Baseline).

Method	Model	N	Acc.	AUC	Prec.	Rec.
ABMIL	Baseline	5	74.55±0.00	78.32±0.00	60.44±0.00	74.98±0.00
ABMIL	PathHunter	5	74.35±0.00	77.98±0.06	24.84±0.00	79.59±0.00
Δ (pp)			-0.20	-0.34	-35.60	+4.61
DSMIL	Baseline	5	78.76±0.00	74.23±0.00	59.67±0.00	69.13±0.00
DSMIL	PathHunter	5	98.12±0.30	98.26±0.13	85.63±1.76	97.14±1.00
Δ (pp)			+19.36	+24.03	+25.96	+28.01
TransMIL	Baseline	5	70.74±0.00	64.40±0.00	53.67±0.00	58.32±0.00
TransMIL	PathHunter	5	95.47±0.37	99.44±0.04	69.02±1.91	97.96±0.00
Δ (pp)			+24.73	+35.04	+15.35	+39.64
DTFD-MIL	Baseline	5	63.33±0.00	76.70±0.00	58.39±0.00	73.30±0.00
DTFD-MIL	PathHunter	5	96.67±0.37	98.68±0.24	78.35±2.05	91.43±1.53
Δ (pp)			+33.34	+21.98	+19.96	+18.13

While ABMIL sometimes exhibits minor decreases or unstable precision, the overall trend confirms PathHunter's positive impact on cross-domain generalization for most models.

Same-task OOD Results (Table 14): We evaluate on Camelyon17, an OOD dataset defined by patient IDs 1-100 but sharing the same task as Camelyon16. PathHunter markedly enhances the performance of DSMIL, TransMIL, and DTFD-MIL baselines, with accuracy and AUC improvements often exceeding 20%. Both precision and recall show consistent gains, indicating improved discrimination and higher true positive rates. In contrast, ABMIL exhibits minimal improvement or even a decline in precision, suggesting that certain architectures may require additional adaptation to fully leverage PathHunter's benefits.

PathHunter systematically enhances multiple state-of-the-art MIL methods across diverse feature backbones and challenging distribution shifts. It delivers especially pronounced improvements in recall and AUC, reflecting a more robust and reliable identification of relevant pathological features under OOD settings. This generalizability highlights PathHunter's potential as a versatile framework for improving WSI classification in real-world heterogeneous scenarios.

O Five-Seed Statistical Significance Analysis

To complement the main-paper aggregate comparisons, Table 15 reports a five-seed significance analysis for the ABMIL setting under the Camelyon16-trained protocol across the three backbones and the two evaluation datasets. In this rerun, each baseline score was deterministic, so we evaluated the PathHunter results with a two-sided one-sample t-test over the five RL seeds using the corresponding baseline score as a fixed reference. We report uncorrected significance markers ($^{***}p < 0.001$, $^{**}p < 0.01$, $^{*}p < 0.05$). When the five RL runs also had zero variance, we mark the entry with † and report only the descriptive difference.

Table 15: Five-seed statistical comparison for the ABMIL setting under the Camelyon16-trained protocol. Each PathHunter cell reports mean±std over 5 seeds together with the absolute change from the deterministic baseline. Significance markers denote two-sided one-sample t-tests over the five PathHunter seeds against the corresponding baseline score treated as a fixed reference: $^{***}p < 0.001$, $^{**}p < 0.01$, $^{*}p < 0.05$. Entries marked with † had zero variance across the five PathHunter runs, so no variance-based p-value is reported. All values are in %.

Backbone	Method	Camelyon16					TCGA-NSCLC				
		Prec.	Rec.	Acc.	AUC	WK	Prec.	Rec.	Acc.	AUC	WK
CTransPath											
CTransPath	ABMIL	87.09	83.21	86.05	85.94	69.16	93.60	93.46	93.36	98.01	86.74
	PathHunter + ABMIL	87.70±0.10 +0.61***	87.35±0.82 +4.14***	90.54±0.31 +4.49***	96.28±0.26 +10.34***	79.91±0.69 +10.75***	89.47±0.00 -4.13†	95.20±0.00 +1.74†	92.19±0.00 -1.17†	97.92±0.02 -0.09***	84.39±0.00 -2.35†
UNI2-H											
UNI2-H	ABMIL	91.67	83.67	87.60	98.65	71.90	94.14	94.16	94.14	98.47	88.28
	PathHunter + ABMIL	100.00±0.00 +8.33†	89.80±1.83 +6.13**	96.12±0.69 +8.52***	99.82±0.02 +1.17***	91.60±1.53 +19.70***	93.70±0.29 -0.44*	97.60±0.00 +3.44†	95.63±0.16 +1.49***	99.07±0.02 +0.60***	91.25±0.31 +2.97***
CONCH1.5											
CONCH1.5	ABMIL	93.01	86.73	89.92	92.73	77.45	96.49	96.47	96.48	98.60	92.96
	PathHunter + ABMIL	100.00±0.00 +6.99†	81.22±0.82 -5.51***	92.87±0.31 +2.95***	94.65±0.47 +1.92***	84.29±0.71 +6.84***	96.79±0.01 +0.30***	96.64±0.32 +0.17	96.80±0.16 +0.32*	98.45±0.04 -0.15**	93.59±0.31 +0.63*

Across Camelyon16, most improvements remain statistically reliable for all three backbones, with especially strong gains in accuracy, AUC, and weighted kappa. On TCGA-NSCLC, the pattern is more backbone-dependent: UNI2-H shows significant gains in accuracy, AUC, and weighted kappa, whereas CTransPath exhibits mixed changes together with several zero-variance entries, and CONCH1.5 shows smaller but still significant gains in accuracy and weighted kappa alongside a slight

Table 16: Contextual comparison on TCGA-NSCLC with heterogeneous multimodal and agentic pathology systems. Because architectures, supervision signals, and evaluation protocols differ, this table is included for qualitative positioning only.

Method	TCGA-NSCLC
GPT-4.1-mini-2025-04-14Open (2025)	65.2
GPT-4.1-2025-04-14Open (2025)	65.8
Gemini-2.5-Flash-Preview-04-17Kavukcuoglu et al. (2025)	84.5
Gemini-2.5-Pro-Preview-03-25Kavukcuoglu et al. (2025)	89.2
Quilt-LLaVAIkezogwo et al. (2023)	54.0
PathGen-LLaVASun et al. (2024)	53.5
CPath-OmniSun et al. (2025a)	86.7
CPathAgentSun et al. (2025b)	90.8
CTranspath+ABMIL+PathHunter (ours)	**92.5**
Uni2-H+ABMIL+PathHunter (ours)	**95.7**
CONCH1.5+ABMIL+PathHunter (ours)	**96.8**

AUC decrease. Because the number of seeds is small ($n = 5$) and the baseline scores are deterministic in this rerun, these tests should be interpreted as exploratory rather than fully confirmatory.

P Baseline Methods Comparison

Table 16 provides contextual comparison against recent multimodal and agentic pathology systems on TCGA-NSCLC. Because these methods differ in supervision, architecture, and evaluation protocol, we treat this table as qualitative context rather than as primary evidence for our claims. The main evidence remains the controlled baseline-to-PathHunter comparisons in Table 1, where the backbone, MIL head, and split protocol are matched.

With that caveat, PathHunter remains competitive with the strongest systems listed here, while staying within a purely visual weak-supervision pipeline. This comparison is most useful for positioning: it suggests that carefully learned evidence acquisition can close part of the gap between static MIL pipelines and more complex multimodal or agentic systems, without relying on language supervision at inference time.

Q Additional RL Ablations on PANDA

We thank the reviewer for the suggestion. We performed additional ablations on the official Patho-Bench PANDA `isup_grade` split, focusing on the Karolinska domain with UNI2 + ABMIL. In all experiments below, we reused the same Stage-1 and Stage-2 seed models and only varied the Stage-3 training/evaluation setting. Unless otherwise specified, we used a patch budget of $B = 2048$ and the same search configuration $(K, m, t) = (8, 8, 60)$.

Relation to the main paper. The main paper now reports held-out evaluator validation on PANDA to address the self-referential reward concern. Here we retain three complementary controls that further probe why the sequential policy works: matched RL-vs-static-vs-greedy comparison, a reward-shaping sweep, and a frozen-baseline ablation.

RL vs. static and greedy selection. Table 17 compares PathHunter against two score-based alternatives under the same experimental protocol. First, a static top-m selection does not improve over the Stage-1 baseline. Second, a greedy sequential policy performs substantially worse than both the baseline and PathHunter. In contrast, PathHunter with RL improves weighted kappa from 91.80 to 94.68 without reward shaping. This indicates that the gain is not explained by simple ranking or greedy score following, but instead comes from learned sequential patch selection.

Effect of reward shaping. We also evaluated the sensitivity to the reward shaping weight. Table 18 shows that the no-shaping variant ($\lambda_{\text{shape}} = 0$) achieves the best weighted kappa among all tested settings. Moderate positive shaping weights remain competitive in some cases, but none surpass the no-shaping result in this experiment. This reduces the concern that our improvement is driven by a feedback loop induced by the shaping term. In this setting, the core RL objective is sufficient to produce the strongest result.

Method	Acc	AUC	F1	WK
Stage-1 baseline	80.89	96.03	69.07	91.80
Top-m static	80.89	96.03	69.07	91.80
Greedy sequential	54.93	78.20	39.12	55.05
PathHunter (joint, full reward)	79.88	93.88	69.55	92.27
PathHunter (joint, w/o shaping)	**82.29**	95.02	**70.99**	**94.68**

Table 17: Comparison of PathHunter against static and greedy patch selection on the official PANDA split. WK denotes weighted kappa. All values are reported in %.

λ_{shape}	Acc	AUC	F1	WK
0.0	**82.29**	**95.02**	**70.99**	**94.68**
0.1	75.86	92.15	67.98	90.70
0.2	80.89	93.88	70.85	92.16
0.4	74.85	94.02	67.66	90.69
0.5	79.88	93.88	69.55	92.27
0.6	79.68	93.77	68.28	92.81
0.8	80.68	94.32	68.34	91.54

Table 18: Reward shaping sweep on the official PANDA split. All values are reported in %.

Frozen baseline vs. joint training. To test whether the gain comes purely from policy learning on top of fixed baseline scores, we froze the Stage-1 model and trained only the RL component. This frozen variant performs poorly (WK 90.31), below both the Stage-1 baseline (WK 91.80) and the jointly trained PathHunter model (WK 94.68 without shaping). We therefore do *not* claim that policy-only optimization with a frozen classifier is sufficient. Instead, this ablation suggests that the improvement comes from the interaction between learned patch selection and downstream MIL adaptation, rather than from a trivial reuse of the baseline score map.

Transferability of selected evidence. We next test whether the learned evidence-selection behavior remains useful when the downstream evaluator or pathology backbone changes. For frozen-evaluator transfer, a source Stage 3 policy selects a fixed $B = 256$ bag, while a different frozen Stage 1 MIL model evaluates both the full bag and the selected bag without participating in selection or retraining. Table 19(a) shows positive transfer for several source–evaluator pairs, including substantial Macro-F1 gains for DTFD-MIL and DSMIL. However, transfer is not uniformly favorable; in particular, the H-Optimus-1 to TransMIL evaluation decreases both AUC and Macro-F1. We therefore interpret these results as evidence of *partial evaluator transfer*, rather than evaluator invariance.

We further transfer only the frozen Manager and Worker between H-Optimus-1 and UNI2 while retaining the target backbone's own frozen scorer and evaluator. As shown in Table 19(b), H-Optimus-1→UNI2 produces small positive changes across all reported metrics, whereas the reverse direction slightly improves fixed-threshold metrics but decreases AUC. Thus, the learned policy is not completely tied to its source representation, although transfer is neither symmetric nor uniformly metric-preserving.

Beyond sparse metastasis evidence. We additionally evaluate PathHunter on broader morphology-driven tasks including histologic grading, immune classification, lymphovascular invasion, and fine-grained breast pathology. As shown in Table 20, the grading and immune-class tasks generally benefit across all three pathology backbones, indicating that the gains are not restricted to binary metastasis detection. However, the results become mixed for lymphovascular invasion and fine-grained BRACS classification.

These findings suggest that the relevant distinction is not simply binary versus multi-class prediction, but the amount and spatial diversity of evidence required for a reliable decision. Tasks dominated by sparse or locally decisive evidence can often be represented by a compact selected bag, whereas tasks requiring distributed architecture, stromal or immune context, heterogeneity, or meaningful negative regions may require a larger or adaptive evidence budget. In particular, increasing the candidate pool T alone does not increase the acquired evidence budget $B = Km$ and therefore does not guarantee sufficient coverage.

Table 19: Post-hoc transfer analysis on Camelyon16. (a) A source Stage 3 policy selects a fixed $B = 256$ evidence bag, which is evaluated by a different frozen Stage 1 MIL model using the same backbone. (b) The frozen Manager and Worker are transferred across pathology backbones, while the target backbone retains its own frozen AttentionMIL scorer and evaluator. No target-side policy training or evaluator adaptation is performed. Values are absolute changes in percentage points relative to full-bag evaluation.

(a) Transfer to unseen frozen MIL evaluators

Source Policy	Frozen Evaluator	Δ AUC	Δ Macro-F1
CONCH1.5 + AttentionMIL	CONCH1.5 + TransMIL	+7.72	0.00
CONCH1.5 + AttentionMIL	CONCH1.5 + DTFD-MIL	+1.05	+25.76
UNI2 + AttentionMIL	UNI2 + DSMIL	+0.38	+10.05
H-Optimus-1 + AttentionMIL	H-Optimus-1 + DTFD-MIL	-3.19	+12.76
H-Optimus-1 + AttentionMIL	H-Optimus-1 + TransMIL	-6.51	-37.86

(b) Transfer of frozen Manager/Worker across backbones

Policy Transfer	Δ AUC	Δ Prec.	Δ Rec.	Δ Macro-F1	Δ Acc.
H-Optimus-1 → UNI2	+0.28	+1.64	+3.06	+2.59	+2.33
UNI2 → H-Optimus-1	-2.37	+0.53	+1.02	+0.87	+0.78

Table 20: Scope analysis on broader morphology-driven PathoBench tasks. Each entry reports the absolute Stage 3 minus Stage 1 change in the task-specific official metric, in percentage points. Results use one official outer fold per task with validation-only model selection and one final outer-test evaluation.

Task (Official Metric)	CONCH1.5	H-Optimus-1	UNI
HNSC grade (WK)	+5.56	+5.79	+7.06
PDA immune class (BAcc.)	+3.68	+2.85	+3.05
ccRCC immune class (BAcc.)	+3.32	+0.65	+12.77
CRC lymphovascular invasion (AUC)	-4.86	+2.78	+4.17
BRACS fine-grained class (BAcc.)	-7.33	-8.66	+0.85

Summary. Overall, these additional results support three points. First, the observed gain is not recovered by either static top-m ranking or greedy sequential selection. Second, reward shaping is not required for improvement; in fact, removing it yields the best result in our current PANDA experiment. Third, the frozen-model ablation shows that competitive performance requires joint adaptation of the selector and the downstream classifier, which is consistent with our interpretation that PathHunter improves via learned selection-and-aggregation rather than by a simple score-ranking heuristic.