PathNavigate: A Training-Free Pathology Agent with Surprise-Guided Scan and Shared Slide Memory for Whole-Slide Image VQA

Chunze Yang[1] **Qidong Liu**[1,✉] **Wenjie Zhao**[1] **Yue Tang**[1] **Jiusong Ge**[1] **Di Zhang**[1]
Jiashuai Liu[1] **Lei Wu**[1] **Junbo Lu**[1] **Ni Zhang**[1] **Xian Wu**[2] **Zeyu Gao**[3] **Chen Li**[1,✉]

[1]School of Computer Science and Technology, Xi'an Jiaotong University
[2]Tencent Jarvis Lab [3]University of Cambridge

{pureeeee, zhao_wenjie, 2206123832, JiusongGe, dzhang}@stu.xjtu.edu.cn
{ljs1007599414, 2216113083, lujunbo}@stu.xjtu.edu.cn
{liuqidong, nizhang, cli}@xjtu.edu.cn kevinxwu@tencent.com zg323@cam.ac.uk

Abstract

Whole-slide image visual question answering (WSI-VQA) frames pathology as an extreme-context search problem: to answer a free-form clinical query, a system must first navigate a gigapixel slide under a strict inspection budget to locate sparse, high-resolution evidence. Existing approaches largely fall into two paradigms: i) supervised pathology multimodal large language models (MLLMs) and agents can absorb localization and reasoning into learned modules, but they often couple navigation to task-specific supervision and retraining, limiting their practicality; ii) training-free pathology agents avoid this cost by keeping core models frozen, but often follow a question-first design, *i.e.,* constructing the initial candidate set mainly from query-conditioned relevance. This can miss decisive morphology that is not named in the question, and force heavier inference-time scaffolding. To address this challenge, we introduce **PathNavigate**, a training-free pathology agent built around a *scan-search-readout* routine. Before question matching, PathNavigate scans the current slide at low magnification with a shared online memory module over frozen pathology features, producing a slide-specific surprise field that marks an abnormal-region pool. It then applies question-conditioned PLIP relevance only within this pool to select high-magnification search targets. Finally, it extracts local high-magnification evidence and answers with a frozen perceptor-adjudicator stack, using the same online memory as slide-level context. Experiments on WSI-VQA and SlideBench-BCNB show that the proposed scan-search-readout design improves answer accuracy while exposing an inspectable evidence-selection trajectory, with fewer tokens and less auxiliary storage than a question-first agent. To ease reproducibility, we have released the code online[1].

1 Introduction

Whole-slide image (WSI) analysis has become a foundation of digital pathology: tissue sections can be digitized as gigapixel slides and used for diagnostic review, grading, and computational assessment [3, 26]. Yet a single slide is far too large to inspect exhaustively. WSI visual question answering (WSI-VQA) makes this challenge more explicit: given a slide and a free-form clinical question, the system must decide where to look and then answer in language [6]. More broadly, WSI-VQA is a

✉Corresponding authors: Qidong Liu and Chen Li.

[1]https://github.com/Pureeeee/PathNavigate_

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

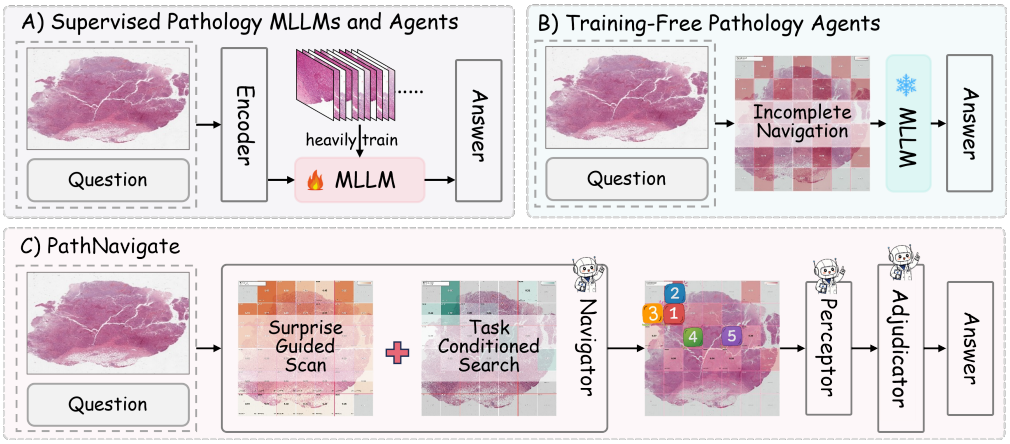

Figure 1: Existing WSI-VQA patterns versus the proposed **Scan-Search-Readout** pattern.

special form of extreme-context visual search, where the model must allocate a tiny inspection budget over a gigapixel visual field and reason from sparse, high-resolution evidence.

Recent progress in WSI-VQA generally follows two paradigms. (i) **Supervised Pathology Models and Agents**: slide-level multimodal large language models (MLLMs) such as SlideChat [8] and WSI-LLaVA [24], as well as trained agents such as CPathAgent [36], PathFinder [12], and SurvAgent [16], absorb evidence localization and reasoning into learned modules. Despite their effectiveness, they couple search and reasoning to task-specific supervision, requiring retraining when the task or workflow changes. This limits their flexibility in deployment. (ii) **Training-Free Pathology Agents**: by contrast, these agents keep the core MLLMs frozen and perform inference-time reasoning, with PathAgent as a representative example [4]. This line is increasingly popular because it preserves modularity and avoids task-specific retraining. Figure 1 reveals the difference between these two patterns: the former one localizes key ROI by MLLMs themselves, while the latter one uses reflected navigation to revise its route.

Since the training-free pattern relies primarily on reflected navigation for localization, the signal guiding the first pass over a gigapixel slide is crucial. Recent works often adopt the **Question-First Design**, *i.e.,* the initial candidate set is constructed mainly from question-conditioned relevance rather than from a slide-specific content prior [4, 17]. However, this design leaves two problems. (i) **Incomplete Navigation**: pathology questions often underspecify the visual cue that determines the answer. For example, a question about histological type may require finding invasive ductal architecture, and a grading question may hinge on a small high-grade focus; neither cue is necessarily named in the question itself. A query-dependent scan can therefore retrieve generally relevant tumor regions while missing decisive morphology. (ii) **Heavier Inference Machinery**: Due to weak scan signals, these agents rely more heavily on longer trajectories, reflection, caption caches, or external context [4, 23, 42], which can recover accuracy but increase latency, token consumption, and system complexity, undermining the practical appeal of training-free deployment.

To address the challenge mentioned above, we start from a pathology intuition: pathologists usually scan a slide broadly to establish what is typical in the current case before searching for the finding most relevant to the question. In digital pathology, broader scanning behavior is also more associated with diagnostic accuracy than zooming alone [10]. This motivates a three-stage **Scan-Search-Readout** routine. Before question matching, the agent should scan the current slide to build a slide-specific content prior, so diagnostically unusual regions are not discarded simply because they are not named by the question. During search, it should inject question relevance only within this content-aware candidate set, so the query refines rather than replaces slide-level exploration. During readout, it should reuse the same online state as slide-level memory for answering, rather than introducing a second store after navigation. Figure 1 visualizes the resulting shift: from directly retrieving question-matched regions to first scanning the current case, then searching within a compact candidate set, and finally answering with memory tied to the same scan.

In this paper, we instantiate this idea with the proposed **PathNavigate**, a training-free pathology agent for WSI-VQA. Our aim is to answer the design question above: how can a training-free agent establish a slide-specific search prior before question matching, while keeping answer-time slide

context coupled to the same state that supports navigation? PathNavigate first scans the slide at low magnification and builds a compact candidate set from a slide-specific surprise prior over frozen pathology features using a small online memory module that follows the Yaad instance of test-time memorization (a two-layer MLP with a Huber objective and a retention gate) [1, 2, 33]. It then applies question-conditioned PLIP relevance only within that set [18], so task information refines the scan rather than replacing it. Finally, it extracts regional evidence and answers with slide-level context read out from the same online state used during scanning. At a broader level, PathNavigate provides a test-time framework for extreme-context visual search under tight inspection budgets. Our contributions are three-fold:

1. We identify the limitation of question-first navigation in training-free WSI-VQA and propose a **Scan-Search-Readout** routine. The novelty lies in this WSI-specific control order rather than in any individual component: online memory, surprise-driven exploration, and question-conditioned retrieval each have precedents. The routine first builds a slide-specific content prior, then applies question relevance within that candidate set, and reuses the scan state for the answer.
2. We instantiate this routine as PathNavigate, a training-free pathology agent. PathNavigate uses a small online memory module over frozen pathology features to produce surprise-guided scan scores, applies PLIP only as within-pool reranking, and reads out the same online state as shared slide memory for answer aggregation.
3. We validate the design on WSI-VQA and SlideBench-BCNB. Overall comparisons and ablations, including a static mean-feature-deviation proposal baseline, an encoder swap, and a semantic re-judging of open-ended answers, show that the scan-search-readout design improves answer accuracy without task-specific training, and keeps every stage of evidence selection inspectable.

2 Problem Setup

Given a whole-slide image $\mathcal{W}$, a free-form question q, and frozen visual and language backbones, a training-free WSI-VQA system produces an answer a after observing only a small subset of ROIs in the slide. Let $\mathcal{X}_\ell(\mathcal{W}) = \{x_i^\ell\}_{i=1}^{N_\ell}$ and $\mathcal{X}_h(\mathcal{W}) = \{x_j^h\}_{j=1}^{N_h}$ denote the low- and high-magnification tiles extracted from $\mathcal{W}$, and let $z = \phi(x) \in \mathbb{R}^d$ be the frozen pathology feature of a tile. The system may inspect only a bounded evidence set before producing the final answer, so the central problem is not only answering, but also how to allocate a small visual budget over a gigapixel slide at test time.

Most training-free WSI-VQA pipelines can be described by four generic operations. (i) The slide is tiled at one or more magnifications and encoded by a frozen pathology model. (ii) A region-proposal step selects a compact region-of-interest (ROI) pool of candidate regions from the slide. (iii) The question prioritizes which candidate regions should receive the limited high-resolution evidence budget. (iv) A frozen pathology VLM, together with a frozen language model, aggregates regional evidence and slide-level context into the final answer.

PathNavigate keeps this same input-output setting and keeps all visual and language models frozen. Its difference lies in how the middle operations are instantiated. For operation (ii), PathNavigate proposes the ROI pool $\mathcal{R}$ from a slide-specific content prior built at ℓ, rather than constructing candidates directly from question relevance. In Section 3, this pool is written as $\mathcal{R}(q)$ because the question type changes the routing budget and spatial spacing; the surprise scores that rank the pool are computed without the question. For operation (iii), question relevance is used only to prioritize candidates within $\mathcal{R}$, producing the search targets $\mathcal{C}$. For operation (iv), local high-magnification evidence $\mathcal{E}$ is aggregated together with a readout of the same online state that produced the slide-specific content prior. In this terminology, our scan, search, and readout stages correspond to ROI proposal, question-based prioritization, and evidence aggregation, respectively.

3 PathNavigate

Overview. Given a slide $\mathcal{W}$ and a question q, PathNavigate instantiates scan-search-readout with four modules: *Shared Online Memory*, *Surprise-Guided Scan*, *Task-Conditioned Search*, and *Evidence Readout and QA*. The pipeline builds a low-magnification surprise field, a compact ROI pool $\mathcal{R}(q)$, selected search targets $\mathcal{C}(q)$, and a final evidence set $\mathcal{E}(q)$ before returning answer a. **Shared Online Memory** maintains a per-slide state over frozen CONCH v1.5 features and supplies both scan-time

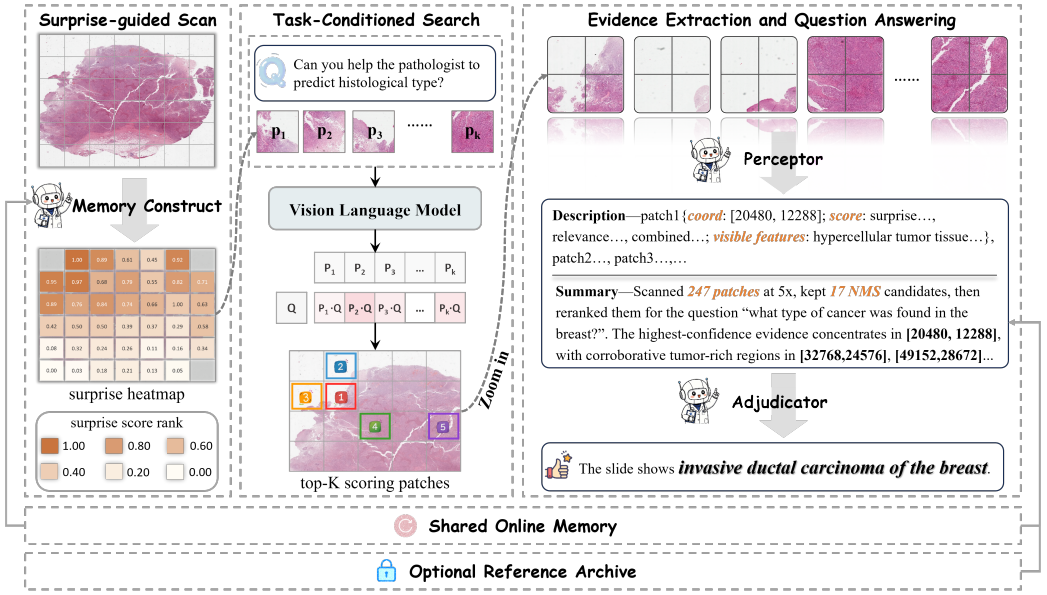

Figure 2: Overview of PathNavigate as a scan-search-readout pathology agent.

surprise scores and compact slide context. **Surprise-Guided Scan** uses these scores to propose the low-magnification ($5\times$) ROI pool; **Task-Conditioned Search** applies PLIP relevance only within that pool to select targets for high-magnification ($20\times$) readout; and **Evidence Readout and QA** converts selected patches into regional evidence for the final answer. Notably, the same scan state ties *where to look* to *what slide-level context is exposed*. We first define the memory, then describe the three operational stages.

3.1 Shared Online Memory

Shared memory couples ROI proposal and answer-time context through one per-slide scan state. During scanning, tiles that would substantially change this state are treated as surprising; this process is denoted as **Memory Update**. At answer time, **Memory Readout** exposes a compact summary of the same surprise history and proposal process, so no separate learned store is introduced.

Construction and Initialization. Let x be a tile and let $\mathbf{z} = \phi(x) \in \mathbb{R}^d$ be its frozen CONCH v1.5 feature, the $d = 768$ patch embedding produced by the `conch_v15` encoder in TRIDENT [9, 25, 41]. The memory is a two-layer GELU reconstruction MLP,

$$\mathcal{M}_{\boldsymbol{\theta}}(\mathbf{z}) = \mathbf{W}_2\,\mathrm{GELU}(\mathbf{W}_1\mathbf{z} + \mathbf{b}_1) + \mathbf{b}_2, \qquad \mathcal{M}_{\boldsymbol{\theta}} : \mathbb{R}^d \to \mathbb{R}^d, \tag{1}$$

where $\boldsymbol{\theta} = \{\mathbf{W}_1, \mathbf{b}_1, \mathbf{W}_2, \mathbf{b}_2\}$ denotes the online parameters. The hidden width is $d_h = 768$, and a fresh memory $\mathcal{M}_{\boldsymbol{\theta}_0}$ is instantiated for every slide–question episode and no state is carried across questions, making surprise a within-slide signal rather than a comparison to a learned cohort. Initialization and optimization defaults are listed in Appendix B.1.

Memory Update. Following test-time memory methods [1, 33], and specifically the Yaad variant of Miras [2] (two-layer MLP memory, Huber attentional bias, multiplicative retention), the memory adapts to the tile stream with a self-supervised reconstruction objective. Tiles are streamed in a fixed raster order over level-0 coordinates. A tile is surprising when reconstructing it would strongly change the current memory, so we use the pre-update gradient norm as surprise. For tile x_i with feature $\mathbf{z}_i = \phi(x_i)$, the Huber reconstruction loss at state $\boldsymbol{\theta}_{i-1}$ is,

$$\mathcal{L}_\delta(\boldsymbol{\theta}_{i-1}; x_i) = \mathrm{Huber}_\delta\big(\mathcal{M}_{\boldsymbol{\theta}_{i-1}}(\mathbf{z}_i), \mathbf{z}_i\big)\,, \tag{2}$$

where δ is the Huber transition parameter. The surprise of x_i is the reconstruction-gradient norm before x_i updates the memory,

$$\sigma_i = \Big\| \nabla_{\boldsymbol{\theta}} \mathcal{L}_\delta(\boldsymbol{\theta}; x_i)\big|_{\boldsymbol{\theta}=\boldsymbol{\theta}_{i-1}} \Big\|_F . \tag{3}$$

Scoring before updating measures unexpectedness under earlier slide context. During the warm-up window T_w, every tile updates memory; after warm-up, we set $\tau_{\mathrm{sur}} = \mu_w + \lambda s_w$ from the first T_w scores. Tiles with $\sigma_i > \tau_{\mathrm{sur}}$ trigger one clipped SGD step on Eq. (2); other tiles apply multiplicative decay $\boldsymbol{\theta}_i \leftarrow \alpha_f \boldsymbol{\theta}_{i-1}$. Because the state is updated sequentially, surprise depends on traversal order; we fix raster order and quantify this dependence in Appendix D.5.

Memory Readout. At answer time, the adjudicator receives a short *Navigation Summary* rather than raw MLP weights or a long trace. It reports the running surprise mean, standard deviation, high-surprise fraction, and candidate count; Appendix B.2 gives the prompt.

3.2 Surprise-Guided Scan

The scan stage turns a gigapixel slide into a compact set of regions for closer inspection. Its goal is to build a slide-specific content prior before text-image reranking: online memory highlights regions atypical for the current slide, so we use the **Surprise Score** as a proposal prior rather than a pathology label. Here, the question acts only as a router for coverage and budget, not as a relevance scorer: the surprise field itself is computed without q, and q enters only through a keyword-based question type (morphology, clinical, or other) that sets the NMS spacing $\rho(q)$ and the budget $K_{\mathrm{search}}(q)$. We keep the argument in $\mathcal{R}(q)$ only to record this coarse dependence. **ROI Selection** then turns surprise scores into target regions.

Surprise Score. Let $\mathcal{X}_\ell(\mathcal{W}) = \{x_i^\ell\}_{i=1}^{N_\ell}$ denote the low-magnification tiles of slide $\mathcal{W}$, with grid coordinates $\mathbf{p}_i$ and frozen CONCH features $\mathbf{z}_i = \phi(x_i^\ell) \in \mathbb{R}^d$. Applying the memory rule in Section 3.1 gives each tile a surprise score: $\sigma_i = \sigma(x_i^\ell)$ for $i = 1, \ldots, N_\ell$, where $\sigma(\cdot)$ is the pre-update gradient norm in Eq. (3). Collecting these scores over all low-magnification tiles yields a slide-wide surprise field.

ROI Selection. After collecting a slide-wide surprise field, we apply thresholding and distance-based non-maximum suppression (NMS) to the whole field to obtain a spatially diverse ROI pool:

$$\mathcal{R}(q) = \mathrm{NMS}\Big(\{(\mathbf{p}_i, \sigma_i)\}_{i=1}^{N_\ell}; \rho(q)\Big), \tag{4}$$

where $\rho(q)$ is a minimum center distance in level-0 pixels rather than an IoU threshold. The operator sorts candidate centers by σ_i and greedily keeps a center only if it is at least $\rho(q)$ away from all previously kept centers. $\mathcal{R}(q)$ is the resulting set of kept ROI centers, the handoff to Section 3.3. As a result, morphology questions receive denser local coverage, while broader clinical questions receive larger spacing and a slightly larger first-pass search budget; Appendix B.1 lists the reported values. This stage is intentionally question-light: the question influences spatial coverage and budget, but it does not yet rank candidate semantics. *Example.* On a breast resection, the memory quickly fits the abundant adipose tissue and fibrous stroma, so a later nest of invasive tumor cells typically produces a large pre-update gradient; NMS then keeps one center per neighborhood, so a large tumor bed cannot crowd out a smaller, separate focus from the pool.

3.3 Task-Conditioned Search

Once the scan has produced $\mathcal{R}(q)$, the system only needs to decide which regions in this slide-specific pool deserve high-magnification budget. Task-Conditioned Search therefore uses two complementary signals: surprise measures whether an ROI is atypical for the slide, while PLIP relevance measures whether it matches the question, *i.e.,* **Question-conditioned Relevance**. A transparent **Within-pool Fusion** combines them. PLIP is applied at low magnification because this stage chooses which ROIs deserve follow-up.

Question-conditioned Relevance. For each ROI center $r \in \mathcal{R}(q)$, we extract its corresponding low-magnification patch x_r^ℓ and compute a PLIP similarity score:

$$r_{\mathrm{PLIP}}(r; q) = \cos\big(\psi_{\mathrm{PLIP}}(q), \phi_{\mathrm{PLIP}}(x_r^\ell)\big), \tag{5}$$

where ψ_{PLIP} and ϕ_{PLIP} are frozen PLIP text and image encoders [18].

Within-pool Fusion. Let $\sigma(r)$ denote the scan-time surprise score associated with ROI center r. Surprise and PLIP relevance have different numeric scales, and their ranges vary from slide to slide.

Because this stage only needs the ordering inside the current ROI pool, we normalize each score function s within $\mathcal{R}(q)$ to obtain a pool-relative score,

$$\hat{s}(r) = \frac{s(r) - m_s}{M_s - m_s + \varepsilon}, \tag{6}$$

where $s \in \{\sigma, r_{\mathrm{PLIP}}(\cdot; q)\}$, $m_s = \min_{r' \in \mathcal{R}(q)} s(r')$, and $M_s = \max_{r' \in \mathcal{R}(q)} s(r')$. The small ε prevents division by zero when all ROI scores are tied. This produces $\hat{\sigma}(r)$ for slide-specific surprise and $\hat{r}_{\mathrm{PLIP}}(r; q)$ for question relevance. We then combine the two normalized scores as:

$$f_\alpha(r; q) = \alpha\, \hat{r}_{\mathrm{PLIP}}(r; q) + (1 - \alpha)\, \hat{\sigma}(r), \tag{7}$$

where $\alpha \in [0, 1]$ controls the balance between task relevance and slide-specific surprise. We use $\alpha = 0.5$ as a fixed symmetric default and ablate the endpoints and sweep values in Section 4.3; the full numeric sweep is reported in Appendix Table 10. The final search targets are:

$$\mathcal{C}(q) = \mathrm{Top}\, K\big(\mathcal{R}(q), f_\alpha(\cdot; q); K_{\mathrm{search}}(q)\big), \tag{8}$$

where $K_{\mathrm{search}}(q)$ is a small question-aware first-pass budget. The same lightweight router that sets $\rho(q)$ allocates denser local follow-up to morphology questions and broader coverage to clinical questions; Appendix B.1 lists the values. The role split remains asymmetric: surprise builds the pool (at most $K_{\mathrm{pool}} = 30$ centers), and PLIP only reranks within that pool to choose $K_{\mathrm{search}}(q) = 10/12/13$ targets for other/morphology/clinical questions. *Example.* For "What is the nuclear grade of the invasive ductal carcinoma?" (a morphology question), PLIP tends to rank cellular tumor ROIs above an equally surprising but fat-rich ROI, while the denser morphology spacing keeps several tumor regions in play.

3.4 Evidence Readout and QA

Given search targets $\mathcal{C}(q)$ and the slide-level readout, the final stage collects answerable evidence: **Evidence Extraction** refines each ROI at high magnification, **Reference Archive** optionally retrieves context, and **Perceptor and Adjudicator** produce the answer.

Evidence Extraction. For each selected low-magnification target $r \in \mathcal{C}(q)$, we collect its local high-magnification neighborhood $\mathcal{P}_h(r)$. When high-magnification features are available, we instantiate a fresh local memory for each ROI, so the stage ranks micro-features inside the selected region without inheriting or overwriting the global slide state. It applies the same surprise rule inside $\mathcal{P}_h(r)$ with a shorter ROI-size-adaptive warm-up schedule and keeps the top-T patches,

$$\mathcal{E}(q) = \bigcup_{r \in \mathcal{C}(q)} \mathrm{Top}\, T\big(\mathcal{P}_h(r), \sigma_h(\cdot\,; r)\big), \tag{9}$$

where $\sigma_h(\cdot\,; r)$ is the local surprise score inside ROI r. Here, T is a small per-ROI evidence budget, and the total number of perceptor calls is capped by a separate global budget; Appendix B.1 lists the launch values. If high-magnification features are unavailable or disabled, the low-magnification target serves as fallback evidence. In the reported configuration $T = 2$ and at most $V_{\max} = 15$ perceptor calls are made per question.

Reference Archive. The optional archive aids final adjudication, not navigation. Before adjudication, we may retrieve k reference cases by cosine similarity between mean-pooled low-magnification CONCH slide features and append their short summaries as *Reference Context*. The archive is built from 100 training-split WSIs, disjoint from the test slides; the reported main results use $k = 3$ unless an ablation disables the archive. Table 3 reports the system without it.

Perceptor and Adjudicator. We use the frozen pathology VLM as *perceptor* to turn patches into evidence strings, and the frozen language model as *adjudicator* to produce the final answer from regional evidence, the Navigation Summary, and optional Reference Context. *Example.* In the nuclear-grade case of Figure 4, the perceptor describes some 20× regions as well differentiated with small uniform nuclei and others as moderately pleomorphic; the adjudicator reconciles these mixed regional grades into the slide-level answer "moderate". Concrete backbones, serving stack, and adjudication settings are reported in Section 4.1 and Appendix Table 4.

Inference Summary. At inference time, PathNavigate scans at low magnification to build $\mathcal{R}(q)$, reranks with PLIP to obtain $\mathcal{C}(q)$, reads high-magnification evidence $\mathcal{E}(q)$, and answers from regional evidence plus the shared memory readout. Appendix B.3 provides pseudocode.

4 Experiments

4.1 Setup

Benchmarks. We evaluate on WSI-VQA [6], SlideBench-BCNB [8], and PathMMU [34]. WSI-VQA and BCNB test whole-slide question answering, while PathMMU is used only as a patch-level transfer check. Dataset statistics, splits, and protocol details are deferred to Appendix A.1.

Baselines and protocol. We organize baselines into three groups aligned with the abstract taxonomy. **Pretrained MLLM Models** include GPT-4o [19], GPT-5.4 [27], Gemini-3.1-Pro-preview [13], and Qwen3.5-4B/9B [28], evaluated as off-the-shelf single-thumbnail systems. **Supervised Pathology Models** include the WSI-VQA model [6], SlideChat [8], WSI-LLaVA [24], MedDr [14], TITAN [9], Quilt-LLaVA [31], LLaVA-Med [22] where available, MedGemma-1.5-4B [30], and Patho-R1 only [43]; these use pathology-specific training, medical/pathology adaptation, or WSI foundation-model pretraining, but not a training-free WSI navigation loop. **Training-Free Pathology Agents** include PathAgent [4] and PathNavigate. TITAN is evaluated zero-shot: each answer option is embedded with TITAN's text encoder and the option closest to the slide embedding (built from all $20\times$ CONCH v1.5 features) is chosen, so it yields MCQ accuracy only; PRISM [32], evaluated the same way, is reported in Appendix D.2. Appendix A.2 gives baseline settings and model identifiers.

Implementation Details. PathNavigate uses Patho-R1-7B [43] as frozen perceptor, Qwen3.5-4B [28] as adjudicator, and CONCH v1.5 [9, 25] for patch features; Qwen endpoints use vLLM [21] where required. The scan runs over $5\times$ tiles (512 px, i.e., 4096 px at level 0) and evidence is read from $20\times$ tiles. The launch configuration is: ROI pool $K_{\text{pool}} = 30$; first-pass targets $K_{\text{search}} = 10/12/13$ for other/morphology/clinical questions; $T = 2$ patches per ROI; at most 15 perceptor calls per question; $\alpha = 0.5$; warm-up $T_w = 100$; learning rate 0.05; archive $k = 3$. These values come from implementation defaults and development-stage choices rather than a held-out search (Appendix D.8). Appendix A.1 records hardware/software, and Appendix B.1 lists routing budgets and caps.

Evaluation Metrics. For WSI-VQA, we report Total accuracy, MCQ accuracy, open-ended accuracy, BLEU-1, and ROUGE-L in the main table, with BLEU-4 in Appendix C.1. All rows are scored on the 735 WSI-VQA questions (390 MCQ, 345 open-ended); for PathNavigate and its variants, 9 questions that failed because of a pipeline error are counted as incorrect. Open-ended answers are scored by string matching after medical-synonym normalization, and Appendix D.4 adds a blinded LLM-judge check. For BCNB and PathMMU, we use letter-exact accuracy; BCNB Overall follows the official weighted aggregation over seven diagnostic sub-tasks. Efficiency reports latency, token count, and offline footprint, where lower values are preferred.

4.2 Main results

Overall comparison on SlideBench-BCNB (Table 1). BCNB is fully multiple-choice and reports diagnostic subscores. PathNavigate is best on Overall, ER, HER2, and Molecular Subtype, runner-up on Tumor and PR, and competitive but not dominant on Grading. The largest gains over its own frozen perceptor (Patho-R1 only) are on Tumor, ER, PR, and HER2, where locating invasive regions directly improves the answer context; HER2 stays compressed for all models because expression level is only weakly recoverable from H&E.

Overall comparison on WSI-VQA (Table 2). WSI-VQA complements BCNB with multiple-choice and open-ended answers. Thumbnail and patch-only baselines can solve some MCQs but lack reliable evidence recovery for free-form responses. Under the same Patho-R1-7B/Qwen3.5-4B stack, PathAgent matches PathNavigate on MCQ (201 of 390 correct for both) yet drops sharply on Open accuracy and ROUGE-L, indicating that PathNavigate's gains come from scan-derived context and high-magnification readout rather than a decoder swap. A blinded semantic re-judging of 80 shared open-ended questions gives 75.0% versus 15.0% strict semantic accuracy, and in a separate pairwise preference judgment PathNavigate wins 54, ties 25, and loses 1 (Appendix D.4), so the Open gap is not an artifact of substring matching.

Appendix Table 4 gives settings; Appendix Figures 5 and 6 summarize controlled comparisons; Appendix A.3 documents the PathAgent reproduction protocol and Appendix D.9 compares the two designs side by side.

Table 1: Primary SlideBench-BCNB results with full sub-task breakdown. The boldface refers to the highest score and the underline indicates the next best result of the models.

Method	Input	Overall	Tumor	ER	PR	HER2	Grading	Mol. Sub
Pretrained MLLM Models								
GPT-4o [19]	Slide-T	43.77	56.81	50.00	49.43	42.11	38.44	26.84
GPT-5.4 [27]	Slide-T	45.85	60.40	50.38	49.91	44.52	42.33	28.45
Gemini-3.1-Pro-preview [13]	Slide-T	44.91	58.51	50.19	49.62	43.57	40.82	27.60
Qwen3.5-4B [28]	Slide-T	39.48	49.43	33.36	27.32	48.02	52.70	19.19
Qwen3.5-9B [28]	Slide-T	42.00	53.88	36.39	30.15	47.54	**59.18**	21.46
Supervised Pathology Models								
SlideChat [8]	WSI	54.22	**89.89**	72.97	**74.01**	47.54	23.11	20.60
WSI-LLaVA [24]	WSI	52.86	85.35	60.78	62.00	42.77	46.33	29.21
MedDr [14]	Slide-T	35.48	59.64	43.76	39.22	27.41	29.37	20.79
TITAN [9]	Slide	44.39	85.16	42.44	41.68	36.86	45.36	22.50
Quilt-LLaVA [31]	Patch	34.11	33.46	49.43	44.05	25.85	34.99	25.24
MedGemma-1.5-4B [30]	Slide-T	50.40	79.68	52.36	60.40	47.87	42.33	21.27
Patho-R1 only [43]	Patch	37.26	42.16	54.63	50.38	25.61	35.53	26.65
Training-Free Pathology Agents								
PathAgent [4]	WSI	56.06	87.52	63.71	65.97	44.52	55.94	30.25
PathNavigate (Ours)	WSI	**59.42**	88.28	**76.09**	69.00	**48.39**	52.59	**32.33**

Table 2: Compact WSI-VQA results. TITAN reports only MCQ because this zero-shot setting can only be mapped to answer options; boldface/underline mark best/runner-up.

Method	Input	Total	MCQ	Open	BLEU-1	ROUGE-L
Pretrained MLLM Models						
GPT-4o [19]	Slide-T	27.62	32.56	22.03	0.085	0.236
GPT-5.4 [27]	Slide-T	35.65	42.05	28.41	0.142	0.288
Gemini-3.1-Pro-preview [13]	Slide-T	34.42	40.51	27.54	0.128	0.275
Qwen3.5-4B [28]	Slide-T	26.39	42.56	8.12	0.092	0.241
Qwen3.5-9B [28]	Slide-T	38.50	49.23	26.38	0.573	0.480
Supervised Pathology Models						
WSI-VQA model [6]	WSI	42.18	46.92	36.81	0.108	0.318
MedDr [14]	Slide-T	45.17	43.59	46.96	0.368	0.452
TITAN [9]	Slide	—	34.10	—	—	—
Quilt-LLaVA [31]	Patch	30.07	29.23	31.01	0.002	0.251
WSI-LLaVA [24]	Patch	33.20	42.82	22.32	0.057	0.312
MedGemma-1.5-4B [30]	Slide-T	42.99	**53.08**	31.59	0.180	0.349
Training-Free Pathology Agents						
PathAgent [4]	WSI	33.88	51.54	13.91	0.427	0.335
PathNavigate (Ours)	WSI	**55.65**	51.54	**60.29**	**0.609**	**0.635**

PathMMU transfer. Because PathMMU is image-level, Appendix Table 9 reports it only as patch-level validation (53.60 letter-exact accuracy).

4.3 Core component analysis

The ablations map to scan-search-readout: *w/o* PLIP removes question-aware reranking, PLIP-only removes surprise, Random ROI removes structured proposal, and archive/Navigation Summary variants test answer-time context. Table 3 supports a division of labor: surprise proposes, PLIP refines, high-magnification evidence grounds the answer, and compact context stabilizes adjudication. The largest drop comes from replacing the proposal mechanism with random ROIs (-6.80 Total), showing that the first-pass pool is the primary failure point. A static proposal that ranks tiles by cosine distance to the slide-mean feature (Mean-dist.) recovers part of this gap (51.97) but stays

Table 3: Structural WSI-VQA ablations.

Variant	Total	MCQ	Open
PathNavigate	**55.65**	**51.54**	60.29
w/o PLIP	53.47	47.18	**60.58**
PLIP-only	52.24	46.92	58.26
Mean-dist. pool	51.97	47.95	56.52
w/ Random ROI	48.84	44.62	53.62
w/o archive	53.47	48.97	58.55
w/o Nav. Sum.	54.15	50.00	58.84

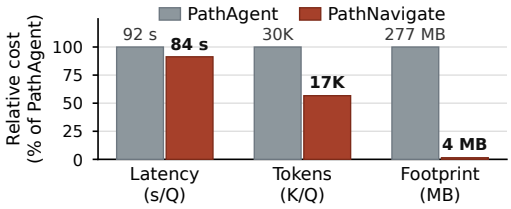

Figure 3: Relative WSI-VQA cost.

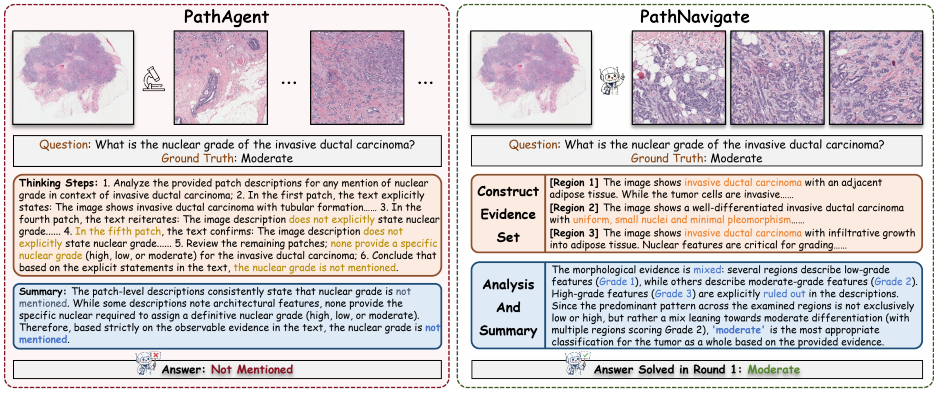

Figure 4: Same-slide comparison on a nuclear-grade question. PathAgent's patch descriptions never state a grade and it answers "not mentioned"; PathNavigate aggregates mixed regional grades from $20\times$ evidence and answers "moderate" (ground truth: moderate).

3.67 points below the online-memory pool, and the two scores rank tiles very differently (Spearman 0.014; Appendix D.6); removing PLIP costs less but still hurts MCQ, where task wording helps discriminate among plausible regions. Archive and Navigation Summary ablations are smaller but consistent, suggesting that answer-time context mainly stabilizes borderline cases rather than creating the evidence signal. Swapping the scan encoder from CONCH v1.5 to UNI2-h [7] at both magnifications changes Total by only $+0.27$ (55.92 vs. 55.65; Appendix D.1), so the routine is not tied to one encoder. Additional BCNB, architecture, high-magnification readout (Appendix Table 13), negative-result, and sensitivity details are deferred to Appendix C.4.

4.4 System-level efficiency

We measure per-question cost for PathNavigate and PathAgent on the same hardware, WSI-VQA set, and evaluation pipeline. Figure 3 normalizes PathAgent to 100%: PathNavigate keeps latency similar while reducing prompt tokens and footprint (on-disk auxiliary artefacts, not model weights). Fewer prompt tokens reduce evidence passed into the adjudicator, lowering context pressure without removing high-magnification readout. The claim is scoped to storage and tokens against training-free agents, not peak GPU memory or parity with trained slide MLLMs (breakdown in Appendix C.5).

4.5 Qualitative case studies

Figure 4 compares the two agents on the same slide and question ("What is the nuclear grade of the invasive ductal carcinoma?"). PathAgent's retrieved patches are described as ductal carcinoma, but none of the descriptions commits to a grade, so it answers "not mentioned". PathNavigate's evidence set covers several tumor-rich $20\times$ regions whose descriptions mix low- and moderate-grade features, and the adjudicator resolves them to "moderate", matching the ground truth. The case shows that answer quality depends on gathering enough diagnostic tissue before adjudication. Appendix C.3 gives the full workflow and intermediate evidence trace.

5 Related Work

Supervised Pathology MLLMs and Agents. Pathology VQA requires gigapixel navigation with morphological and diagnostic reasoning. Earlier datasets and benchmarks, including PathVQA [15], Quilt-LLaVA [31], and WsiCaption [5], advanced medical VQA but only partially stress slide-level evidence search. Recent supervised pathology MLLMs such as SlideChat [8] and WSI-LLaVA [24], and trained agents such as CPathAgent [36], PathFinder [12], and SurvAgent [16], absorb localization and reasoning into learned modules but depend on task-specific training or medical image-text data. These methods are useful upper references for pathology-specific adaptation, but they answer a different question from ours: how much slide reasoning can be learned with supervision, rather than how a frozen system should allocate inspection at test time.

Training-Free Pathology Agents. To avoid task-specific retraining [35, 44], training-free agents formulate WSI-VQA as evidence gathering with frozen models. Existing systems often depend on engineered tools, caption caches, or external knowledge bases [29, 37, 39, 40], and many follow a question-first retrieval prior [4, 20]. This can work when query terms align with visible tissue, but it makes early search inherit the user's wording and can miss morphology that is diagnostic but not named. PathNavigate changes the order instead of enlarging the tool inventory: it scans first, builds the pool from slide-specific surprise, and applies the question only within the pool. Its online memory follows test-time memorization models such as Titans and Yaad [1, 2]; what is specific to WSI-VQA is using that state both to propose regions before the question is read and as answer-time slide context. Appendix D.9 contrasts PathNavigate and PathAgent along each design axis.

6 Limitations

Surprise is a proposal signal, not a lesion detector. On CAMELYON16 [11] held-out slides, surprise ranks tumor tiles only slightly better than chance (ROC-AUC 0.537; Appendix D.3), and folds, pen marks, or stain shifts can also score high; reranking and $20\times$ readout reduce but do not remove such proposals. **Order dependence.** Sequential updates and a threshold fixed after warm-up make proposals depend on traversal order (Spearman -0.42 between raster and reverse-raster rankings); end-to-end accuracy on 100 questions did not change significantly, but the method is not order-invariant (Appendix D.5). **Fixed budgets and low-magnification blind spots.** Cues invisible at $5\times$ may never enter the pool, and the budget does not grow when the surprise field is flat (enlarging it from $30/10/15$ to $60/20/25$ on 100 low-dispersion questions gave $+2$ answers at $+38\%$ time; Appendix D.7). Hyperparameters are heuristic rather than selected on a held-out split. **Evaluation scope.** All WSI-level results are on breast cancer (TCGA-BRCA and BCNB). Open-ended scoring relies on synonym-normalized matching plus a single automatic judge, without pathologist review, and we swapped the scan encoder but not the PLIP reranker. **Broader impacts.** A training-free agent that shows where it looked could lower the cost of adapting slide-level question answering to new tasks and sites, and make model outputs easier for pathologists to audit. The main risks are over-reliance on wrong answers that come with plausible-looking evidence and uneven behavior beyond the breast-cancer data studied here; PathNavigate is a research system, is not validated for clinical diagnosis, and should support rather than replace expert review.

7 Conclusion

We presented PathNavigate, a training-free WSI-VQA agent whose shared online memory builds a slide-specific surprise prior before question matching and is reused as answer-time context. On WSI-VQA and SlideBench-BCNB, this ordering improves answer accuracy over pretrained MLLMs, supervised pathology models, and a question-first training-free agent, and it remains effective when the scan encoder is swapped. Test-time navigation is thus a first-class design variable even with frozen backbones; H&E-weak labels and the limits above call for modality-aware evidence and adaptive inspection budgets in future agents.

Acknowledgments and Disclosure of Funding

This work was supported by Noncommunicable Chronic Diseases-National Science and Technology Major Project (Grant No. 2025ZD0544802), National Natural Science Foundation of China (No. 62676310, 62602509, 62506291), the Key Research and Development Program of the Shaanxi Province (Grant No. 2024SF-GJHX-32), Natural Science Basic Research Program of Shaanxi Province (Program No. 2026JC-YBQN-0783), the Key Research and Development Program of the Ningxia Hui Autonomous Region (Grant No. 2023BEG02023), China Postdoctoral Science Foundation under Grant Number 2026M791685, the Fundamental Research Funds for the Central Universities (No. xzy012026032, xzy012026033), the XJTU Research Fund for AI Science (No. 2025YXYC004), the Post-Doctoral Research Project of Shaanxi Province (No. 2025BSHSDZZ047), Xi'an Jiaotong University Kunpeng & Ascend Center of Cultivation, the project "Research on Key Technologies for Full-Chain Intelligent Pathological Diagnosis" (Grant No. HX202440) of The First Affiliated Hospital of Xi'an Jiaotong University, the Royal Society under Grant [RGS\R2\252688] and GE HealthCare.

The authors declare no competing interests.

References

[1] Ali Behrouz, Peilin Zhong, and Vahab Mirrokni. Titans: Learning to memorize at test time. *arXiv preprint arXiv:2501.00663*, 2024.

[2] Ali Behrouz, Meisam Razaviyayn, Peilin Zhong, and Vahab Mirrokni. It's all connected: A journey through test-time memorization, attentional bias, retention, and online optimization. *arXiv preprint arXiv:2504.13173*, 2025.

[3] Gabriele Campanella, Matthew G Hanna, Luke Geneslaw, Allen Miraflor, Vitor Werneck Krauss Silva, Klaus J Busam, Edi Brogi, Victor E Reuter, David S Klimstra, and Thomas J Fuchs. Clinical-grade computational pathology using weakly supervised deep learning on whole slide images. *Nature Medicine*, 25(8):1301–1309, 2019.

[4] Jingyun Chen, Linghan Cai, Zhikang Wang, Yi Huang, Songhan Jiang, Shenjin Huang, Hongpeng Wang, and Yongbing Zhang. PathAgent: Toward interpretable analysis of whole-slide pathology images via large language model-based agentic reasoning. *arXiv preprint arXiv:2511.17052*, 2025.

[5] Pingyi Chen, Honglin Li, Chenglu Zhu, Sunyi Zheng, Zhongyi Shui, and Lin Yang. WsiCaption: Multiple instance generation of pathology reports for gigapixel whole-slide images. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 546–556. Springer, 2024.

[6] Pingyi Chen, Chenglu Zhu, Sunyi Zheng, Honglin Li, and Lin Yang. WSI-VQA: Interpreting whole slide images by generative visual question answering. In *European Conference on Computer Vision*, pages 401–417. Springer, 2024.

[7] Richard J Chen, Tong Ding, Ming Y Lu, Drew FK Williamson, Guillaume Jaume, et al. Towards a general-purpose foundation model for computational pathology. *Nature Medicine*, 30(3): 850–862, 2024.

[8] Ying Chen, Guoan Wang, Yuanfeng Ji, Yanjun Li, Jin Ye, Tianbin Li, Ming Hu, Rongshan Yu, Yu Qiao, and Junjun He. SlideChat: A large vision-language assistant for whole-slide pathology image understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5134–5143, 2025.

[9] Tong Ding, Sophia J Wagner, Andrew H Song, Richard J Chen, Ming Y Lu, Andrew Zhang, Anurag J Vaidya, Guillaume Jaume, Muhammad Shaban, Ahrong Kim, et al. A multimodal whole-slide foundation model for pathology. *Nature Medicine*, 2025.

[10] Trafton Drew, Mark Lavelle, Kathleen F Kerr, Hannah Shucard, Tad T Brunyé, Donald L Weaver, and Joann G Elmore. More scanning, but not zooming, is associated with diagnostic accuracy in evaluating digital breast pathology slides. *Journal of Vision*, 21(11):7–7, 2021.

[11] Babak Ehteshami Bejnordi, Mitko Veta, Paul Johannes Van Diest, Bram Van Ginneken, Nico Karssemeijer, et al. Diagnostic assessment of deep learning algorithms for detection of lymph node metastases in women with breast cancer. *JAMA*, 318(22):2199–2210, 2017.

[12] 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*, pages 23431–23441, 2025.

[13] Google. Gemini 3.1 Pro preview. Gemini API documentation, `https://ai.google.dev/gemini-api/docs/models/gemini-3.1-pro-preview`, 2026.

[14] Sunan He, Yuxiang Nie, Hongmei Wang, Shu Yang, Yihui Wang, Zhiyuan Cai, Zhixuan Chen, Yingxue Xu, Luyang Luo, Huiling Xiang, et al. GSCo: Towards generalizable AI in medicine via generalist-specialist collaboration. *arXiv preprint arXiv:2404.15127*, 2024.

[15] Xuehai He, Yichen Zhang, Luntian Mou, Eric Xing, and Pengtao Xie. PathVQA: 30000+ questions for medical visual question answering. *arXiv preprint arXiv:2003.10286*, 2020.

[16] Guolin Huang, Wenting Chen, Jiaqi Yang, Xinheng Lyu, Xiaoling Luo, Sen Yang, Xiaohan Xing, and Linlin Shen. SurvAgent: Hierarchical CoT-enhanced case banking and dichotomy-based multi-agent system for multimodal survival prediction. *arXiv preprint arXiv:2511.16635*, 2025.

[17] Wentao Huang, Weimin Lyu, Peiliang Lou, Qingqiao Hu, Xiaoling Hu, Shahira Abousamra, Wenchao Han, Ruifeng Guo, Jiawei Zhou, Chao Chen, et al. Act like a pathologist: Tissue-aware whole slide image reasoning. *arXiv preprint arXiv:2603.00667*, 2026.

[18] Zhi Huang, Federico Bianchi, Mert Yuksekgonul, Thomas J Montine, and James Zou. A visual–language foundation model for pathology image analysis using medical Twitter. *Nature Medicine*, 29(9):2307–2316, 2023.

[19] Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. GPT-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.

[20] AHM Karim and Ozlem Uzuner. Multimodal retrieval-augmented generation with large language models for medical VQA. *arXiv preprint arXiv:2510.13856*, 2025.

[21] Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with PagedAttention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, pages 611–626, 2023.

[22] Chunyuan Li, Cliff Wong, Sheng Zhang, Naoto Usuyama, Haotian Liu, Jianwei Yang, Tristan Naumann, Hoifung Poon, and Jianfeng Gao. LLaVA-Med: Training a large language-and-vision assistant for biomedicine in one day. *Advances in Neural Information Processing Systems*, 36: 28541–28564, 2023.

[23] Songhao Li, Jonathan Xu, Tiancheng Bao, Yuxuan Liu, Yuchen Liu, Yihang Liu, Lilin Wang, Wenhui Lei, Sheng Wang, Yinuo Xu, et al. A co-evolving agentic AI system for medical imaging analysis. *arXiv preprint arXiv:2509.20279*, 2025.

[24] Yuci Liang, Xinheng Lyu, Wenting Chen, Meidan Ding, Jipeng Zhang, Xiangjian He, Song Wu, Xiaohan Xing, Sen Yang, Xiyue Wang, et al. WSI-LLaVA: A multimodal large language model for whole slide image. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22718–22727, 2025.

[25] Ming Y Lu, Bowen Chen, Drew FK Williamson, Richard J Chen, Ivy Liang, Tong Ding, Guillaume Jaume, Igor Odintsov, Long Phi Le, Georg Gerber, et al. A visual-language foundation model for computational pathology. *Nature Medicine*, 30(3):863–874, 2024.

[26] Muhammad Khalid Khan Niazi, Anil V Parwani, and Metin N Gurcan. Digital pathology and artificial intelligence. *The Lancet Oncology*, 20(5):e253–e261, 2019.

[27] OpenAI. GPT-5.4 model. OpenAI API documentation, `https://developers.openai.com/api/docs/models/gpt-5.4`, 2026.

[28] Qwen Team. Qwen3.5: Towards native multimodal agents. `https://qwen.ai/blog?id=qwen3.5`, February 2026.

[29] Samuel Schmidgall, Rojin Ziaei, Carl Harris, Eduardo Reis, Jeffrey Jopling, and Michael Moor. AgentClinic: A multimodal agent benchmark to evaluate AI in simulated clinical environments. *arXiv preprint arXiv:2405.07960*, 2024.

[30] Andrew Sellergren, Sahar Kazemzadeh, Tiam Jaroensri, Atilla Kiraly, Madeleine Traverse, Timo Kohlberger, Shawn Xu, Fayaz Jamil, Cían Hughes, Charles Lau, et al. MedGemma technical report. *arXiv preprint arXiv:2507.05201*, 2025.

[31] Mehmet Saygin Seyfioglu, Wisdom O Ikezogwo, Fatemeh Ghezloo, Ranjay Krishna, and Linda Shapiro. Quilt-LLaVA: Visual instruction tuning by extracting localized narratives from open-source histopathology videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13183–13192, 2024.

[32] George Shaikovski, Adam Casson, Kristen Severson, Eric Zimmermann, Yi Kan Wang, Jeremy D Kunz, et al. PRISM: A multi-modal generative foundation model for slide-level histopathology. *arXiv preprint arXiv:2405.10254*, 2024.

[33] Yu Sun, Xiaolong Wang, Zhuang Liu, John Miller, Alexei Efros, and Moritz Hardt. Test-time training with self-supervision for generalization under distribution shifts. In *International Conference on Machine Learning*, pages 9229–9248. PMLR, 2020.

[34] Yuxuan Sun, Hao Wu, Chenglu Zhu, Sunyi Zheng, Qizi Chen, Kai Zhang, Yunlong Zhang, Dan Wan, Xiaoxiao Lan, Mengyue Zheng, et al. PathMMU: A massive multimodal expert-level benchmark for understanding and reasoning in pathology. In *European Conference on Computer Vision*, pages 56–73. Springer, 2024.

[35] 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*, pages 10360–10371, 2025.

[36] 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*, 2025.

[37] Xiangru Tang, Anni Zou, Zhuosheng Zhang, Ziming Li, Yilun Zhao, Xingyao Zhang, Arman Cohan, and Mark Gerstein. MedAgents: Large language models as collaborators for zero-shot medical reasoning. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 599–621, 2024.

[38] Eugene Vorontsov, Alican Bozkurt, Adam Casson, George Shaikovski, Michal Zelechowski, et al. A foundation model for clinical-grade computational pathology and rare cancers detection. *Nature Medicine*, 30:2924–2935, 2024.

[39] Sheng Wang, Zihao Zhao, Xi Ouyang, Qian Wang, and Dinggang Shen. ChatCAD: Interactive computer-aided diagnosis on medical image using large language models. *arXiv preprint arXiv:2302.07257*, 2023.

[40] Deshui Yu, Yizhi Wang, Saihui Jin, Taojie Zhu, Fanyi Zeng, Wen Qian, Zirui Huang, Jingli Ouyang, Jiameng Li, Zhen Song, et al. YpathRAG: A retrieval-augmented generation framework and benchmark for pathology. *arXiv preprint arXiv:2510.08603*, 2025.

[41] Andrew Zhang, Guillaume Jaume, Anurag Vaidya, Tong Ding, and Faisal Mahmood. Accelerating data processing and benchmarking of AI models for pathology. *arXiv preprint arXiv:2502.06750*, 2025.

[42] Wenchuan Zhang, Jingru Guo, Hengzhe Zhang, Penghao Zhang, Jie Chen, Shuwan Zhang, Zhang Zhang, Yuhao Yi, and Hong Bu. Patho-AgenticRAG: Towards multimodal agentic retrieval-augmented generation for pathology VLMs via reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 29921–29929, 2026.

[43] Wenchuan Zhang, Penghao Zhang, Jingru Guo, Tao Cheng, Jie Chen, Shuwan Zhang, Zhang Zhang, Yuhao Yi, and Hong Bu. Patho-R1: A multimodal reinforcement learning-based pathology expert reasoner. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 28418–28426, 2026.

[44] Zhenfeng Zhuang, Fangyu Zhou, and Liansheng Wang. Libra-MIL: Multimodal prototypes stereoscopic infused with task-specific language priors for few-shot whole slide image classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 14040–14048, 2026.

PathNavigate: A Training-Free Pathology Agent with Surprise-Guided Scan and Shared Slide Memory for Whole-Slide Image VQA

Appendix

Table of Contents

A Experimental Settings and Reproduction Protocols **16**

A.1 Experimental Setting Details 16

A.2 Baseline Settings, Model Names, and API Scope 16

A.3 PathAgent Reproduction Protocol 17

A.4 Deployment on Kunpeng and Ascend Platforms 17

B Method Details **18**

B.1 Question-aware Routing and Budget Allocation 18

B.2 Navigation Summary Prompt Example 18

B.3 Inference Pseudocode 19

C Experimental Results and Additional Analyses **19**

C.1 Full Result Tables 19

C.2 Supplementary Comparison Plots 21

C.3 Workflow Example 22

C.4 Sensitivity and Component Figures 23

C.5 Efficiency Visualization 24

C.6 Negative Findings 25

D Additional Analyses **25**

D.1 Scan Encoder Swap 25

D.2 TITAN and PRISM Zero-Shot Protocol 25

D.3 Region-Level Alignment on CAMELYON16 26

D.4 Semantic Re-Judging of Open-Ended Answers 26

D.5 Traversal-Order Sensitivity 27

D.6 Static Mean-Feature-Deviation Baseline 27

D.7 Budget Expansion on Low-Dispersion Slides 27

D.8 Source of Hyperparameters 28

D.9 Design Comparison with PathAgent 28

A Experimental Settings and Reproduction Protocols

This section collects the material needed to reproduce the experimental stack: benchmark definitions, baseline categories, implementation details, metric definitions, model/API scope, and the controlled PathAgent protocol.

A.1 Experimental Setting Details

This subsection expands the shortened experimental-setting paragraphs in the main paper.

Benchmarks. WSI-VQA [6] contains 735 TCGA-BRCA questions, including 390 multiple-choice and 345 open-ended questions. We use the same test records and answer join for PathNavigate, the PathAgent row, and the reproduced baselines. SlideBench-BCNB [8] contains 7,274 BCNB multiple-choice questions over Tumor, ER, PR, HER2 status, HER2 expression, Grading, and Molecular Subtype; compact BCNB tables follow the main table and merge HER2 status with HER2 expression by averaging the two scores, while Overall keeps the official seven-task weighting. All local and reproduced rows use the full test set and the SlideChat letter-exact protocol. PathMMU [34] is image-level rather than WSI-level, so it is reported only as a patch-level transfer validation for the frozen perception and answering modules; the complete transfer table is collected in Table 9.

Baseline categories.

- **Pretrained MLLM Models.** GPT-4o, GPT-5.4, Gemini-3.1-Pro-preview, and Qwen3.5-4B/9B are treated as off-the-shelf MLLMs. They use one low-magnification slide thumbnail in the main comparison and do not add pathology-specific WSI navigation.
- **Supervised Pathology Models.** This group covers the WSI-VQA model, SlideChat, WSI-LLaVA, MedDr, TITAN, Quilt-LLaVA, LLaVA-Med where available, MedGemma-1.5-4B, and Patho-R1 only. These rows use pathology-specific training, medical/pathology adaptation, or WSI foundation-model pretraining, but they are not training-free pathology agents in our comparison.
- **Training-Free Pathology Agents.** PathAgent and PathNavigate use frozen core models with explicit WSI evidence-gathering loops. Direct agent-to-agent claims use the controlled PathAgent protocol rather than the original paper-reported MCQ-only number.

Implementation stack. PathNavigate uses Patho-R1-7B [43] as the frozen perceptor, Qwen3.5-4B [28] as the frozen adjudicator, CONCH v1.5 [25] for patch features, and vLLM [21] for Qwen endpoint serving where required. Local runs were executed on a dual Intel Xeon Platinum 8352V CPU host with NVIDIA RTX 4090 24GB GPUs. The software stack is Python 3.9.25, PyTorch 2.7.0+cu126, CUDA 12.6, Transformers 4.57.6, and vLLM 0.19.0 for the Qwen3.5 endpoint used in the VLM reproduction pipeline. The shared online memory is a two-layer GELU MLP ($\mathbb{R}^{768} \to \mathbb{R}^{768}$) updated only at test time; PLIP fusion uses the fixed default $\alpha{=}0.5$; main results enable the optional $k{=}3$ case archive over the $\sim$4 MB in-domain index; no reflection round is used unless explicitly stated. Routing overrides and budget caps are listed in Appendix B.1. We have also deployed the pipeline on Kunpeng CPU and Ascend NPU platforms (Appendix A.4).

Metric details. The WSI-VQA test set has 735 questions (390 multiple-choice, 345 open-ended), and every row uses these denominators. For PathNavigate and its variants, 9 questions failed because of a pipeline error and are counted as incorrect; open-ended answers are matched after medical-synonym normalization. We also report compact text-generation scores. The main table keeps BLEU-1 and ROUGE-L, while Table 7 also reports BLEU-4. BCNB uses letter-exact scoring for each diagnostic sub-task and the official weighted Overall score. PathMMU uses letter-exact validation accuracy. Efficiency metrics are per-question median latency, peak adjudicator-token count, peak GPU memory when tabulated, and offline footprint; lower is better.

A.2 Baseline Settings, Model Names, and API Scope

This subsection combines the reproducibility-facing baseline table with the exact closed-source and open-source model identifiers used by the main comparison.

Pretrained MLLM rows in the main tables include closed-source API endpoints and local Qwen thumbnail runs, all using the same single-thumbnail (Slide-T) visual input. We cite provider

Table 4: Detailed settings behind the shortened method names used in the main result tables. All local reproduced rows use Qwen3.5-4B as the common final adjudicator / answer-normalization layer.

Name in main tables	Source	Visual input	Backbone / setting
Pretrained MLLM Models			
GPT-4o	local API run	slide thumbnail	OpenAI endpoint `gpt-4o`
GPT-5.4	local API run	slide thumbnail	OpenAI endpoint `gpt-5.4`
Gemini-3.1-Pro-preview	local API run	slide thumbnail	Google endpoint `gemini-3.1-pro-preview`
Qwen3.5	reproduced	slide thumbnail	Qwen/Qwen3.5-4B and Qwen/Qwen3.5-9B
Supervised Pathology Models			
WSI-VQA model	reproduced	WSI-level	official WSI-VQA model
SlideChat	reproduced	slide-level	trained slide MLLM
WSI-LLaVA	reproduced	WSI (BCNB) / patches (WSI-VQA)	official WSI setting on BCNB; local patch input on WSI-VQA
MedDr [14]	reproduced	slide thumbnail	Slide-T setting
TITAN [9]	reproduced	all 20× CONCH v1.5 features	zero-shot option matching (Appendix D.2)
Quilt-LLaVA [31]	reproduced	patch input	official checkpoint
LLaVA-Med [22]	reproduced	patch input	official checkpoint
MedGemma [30]	reproduced	slide thumbnail	MedGemma 1.5-4B
Patho-R1 only	reproduced	selected patches	Patho-R1-7B [43] evidence + Qwen3.5-4B [28] adjudicator
Training-Free Pathology Agents			
PathAgent	reproduced	patch trajectory	official pipeline + Patho-R1-7B [43] / Qwen3.5-4B [28] join
PathNavigate	reproduced	scan-search-readout	Patho-R1-7B [43] / Qwen3.5-4B [28], Appendix B.1

documentation for endpoint families rather than treating API labels as standalone paper titles. For OpenAI we call the `gpt-5.4` model alias. The Google Gemini documentation lists the standard Gemini 3.1 Pro endpoint as `gemini-3.1-pro-preview`; we did not use the separate `gemini-3.1-pro-preview-customtools` endpoint. The Qwen rows use the public Qwen3.5 model-card names `Qwen/Qwen3.5-4B` and `Qwen/Qwen3.5-9B`; we therefore cite the Qwen3.5 release recommended by the model card rather than the older Qwen2.5-VL technical report. For MedGemma, the main tables keep MedGemma-1.5-4B as the size-comparable medical VLM baseline and omit the much larger 27B variant from the primary comparison.

A.3 PathAgent Reproduction Protocol

The PathAgent row in the main paper uses the official codebase configured with Patho-R1-7B [43], Qwen3.5-4B [28], and our evaluation join. This keeps the direct agent-to-agent comparison on equal footing rather than anchoring it to the original paper-reported MCQ-only number. We keep the detailed protocol note here, rather than in the introduction, so the main paper stays focused on PathNavigate itself.

Table 5: Protocol note for the PathAgent row. The row is used as a controlled agent-loop comparison under our evaluation stack, not as an exact reproduction of the original PathAgent paper.

Item	Setting in this paper
Code path	Official PathAgent implementation configured locally.
Question set	Same WSI-VQA test split and question records as Table 2.
Answer stack	Same frozen Patho-R1-7B perceptor and Qwen3.5-4B adjudicator used for PathNavigate; all local reproduced rows use Qwen3.5-4B as the final answer-normalization layer.
Case archive	PathAgent has no case-archive component, so it is run as released, without PathNavigate's archive.
Evaluation join	Same prediction join, answer parsing, SeqMatcher-based MCQ scoring, and BLEU/ROUGE scripts used for PathNavigate.
Reported use	Direct comparison to PathNavigate uses this controlled protocol. The original paper-reported MCQ value is cited only as context because it used a different release/evaluation setting.

A.4 Deployment on Kunpeng and Ascend Platforms

Because PathNavigate needs no training, moving it to a new compute platform only requires running its frozen components and the small online memory at inference time. Beyond the NVIDIA GPU environment used for the reported experiments (Appendix A.1), we have also deployed the full scan–search–readout pipeline on servers built on Huawei Kunpeng CPUs and Ascend NPUs. In this deployment, the frozen pathology encoders, the PLIP reranker, the perceptor, and the adjudicator are served on Ascend NPUs through the Ascend software stack, while CPU-side stages such as slide

tiling, feature I/O, distance-based NMS, and evidence bookkeeping run on Kunpeng processors. The per-episode online memory is a two-layer MLP updated with single SGD steps, so it adds little load on either device. The method, routing budgets, and prompts are unchanged across platforms, and no component requires platform-specific retraining. All numbers reported in this paper are from the GPU environment.

B Method Details

After the dataset, baseline, and reproduction context, this section records the deferred method-level material: the question-aware routing and budget heuristics, the Navigation Summary prompt, and the full inference pseudocode.

B.1 Question-aware Routing and Budget Allocation

The main paper refers to the scan controller as a lightweight dynamic routing and budget-allocation mechanism. This controller does not change backbone weights or introduce question-specific model parameters. It only adjusts spatial coverage and evidence budget using a few pre-scan heuristics derived from the current question.

The released code exposes global parser defaults, but the paper reports the launch configuration used for the main experiments. Table 6 records the reported overrides in lookup form.

Table 6: Reported routing, memory, and evidence-budget settings used by the main experiments.

Component	Reported setting
Magnifications	Scan and PLIP reranking at $5\times$ (512 px tiles, 4096 px at level 0 for $40\times$ slides); evidence readout at $20\times$.
Scan order	Raster order over level-0 coordinates; warm-up uses the first T_w tiles in this order.
Question router	Keyword heuristic mapping each question to morphology, clinical, or other.
Scan-time coverage	NMS distance is $\max(d_{\min}, 4096)$ for morphology, $\max(d_{\min}, 20480)$ for clinical, and $\max(d_{\min}, 0.08\,\mathrm{diag}(\mathcal{W}))$ otherwise.
First-pass search budget	Base $K_0 = 10$; $K_{\mathrm{search}}(q) = K_0 + 2$ for morphology, $K_0 + 3$ for clinical, and K_0 otherwise. Pre-rerank pool $K_{\mathrm{pool}} = \max(30, K_{\mathrm{search}}(q) R_{\max})$ with $R_{\max} = 1$.
Shared online memory	Reinitialized with PyTorch defaults for every slide–question episode (no state carried across questions); two-layer GELU MLP, hidden width 768, learning rate 0.05, gradient clipping 5.0, forgetting factor $\alpha_f = 0.999$, warm-up $T_w = 100$, threshold scale $\lambda = 1.0$.
Local high-mag memory	Reinitialized for each ROI with warm-up $\min(\max(T_w/5, 5), \lvert\mathcal{P}_h(r)\rvert)$.
Evidence and adjudication budget	$T = 2$ local patches per ROI, global perceptor cap $V_{\max} = 15$, and one adjudication round. Reflection is disabled in main results; exploratory reflection runs split remaining targets evenly across remaining rounds.

B.2 Navigation Summary Prompt Example

The Navigation Summary is a short prompt field inserted into the adjudicator input. It is not a learned memory module and does not expose the full scan trace. We separate the reusable prompt slot from the filled qualitative example so the reader can distinguish format from case-specific evidence.

Prompt slot template.

> **Navigation Summary.** Low-magnification scan statistics: mean `<mean>`, standard deviation `<std>`, high-surprise fraction `<fraction>`, and `<count>` candidate regions after thresholding and NMS. Use this as slide-level context, but base the final answer on the regional evidence below.

Filled qualitative example (numeric statistics paraphrased for readability). For the nuclear-grade case in Figure 4, the adjudicator receives the Navigation Summary together with the selected regional evidence shown below:

> **Question.** What is the nuclear grade of the invasive ductal carcinoma?
> **Navigation Summary.** The scan produced a compact set of high-surprise candidates after distance-based NMS. The selected targets repeatedly concentrate on tumor-rich high-magnification regions rather than coarse local fields.

Regional Evidence. Evidence patches describe irregular nests, moderate nuclear pleomorphism, hyperchromasia, sparse mitoses, and partial tubule loss.
Reference Context. Optional retrieved cases use similar wording for moderate nuclear grade.
Adjudication instruction. Answer the question directly and use the evidence above to choose the most compatible label.

B.3 Inference Pseudocode

Algorithm 1 Training-free inference in PathNavigate

Input: slide $\mathcal{W}$, question q, frozen feature extractors and frozen answer stack
Extract low-magnification tiles $\mathcal{X}_\ell(\mathcal{W})$ and initialize fresh online memory $\mathcal{M}_{\boldsymbol{\theta}_0}$
Scan low-magnification tiles; before each update, compute surprise σ_i from the reconstruction-gradient norm
After warm-up, update memory only for tiles with $\sigma_i > \tau_{\mathrm{sur}}$ and record the surprise history
Apply thresholding and distance-based NMS with spacing $\rho(q)$ to obtain ROI pool $\mathcal{R}(q)$
Compute PLIP relevance inside $\mathcal{R}(q)$ and combine it with surprise to rank candidate regions
Select $K_{\mathrm{search}}(q)$ targets as $\mathcal{C}(q)$
For each selected target, build a local high-magnification memory and keep the top-T surprise patches
Aggregate selected patches into evidence set $\mathcal{E}(q)$ and construct the Navigation Summary from scan statistics
Optionally retrieve Reference Context from the in-domain case archive
Use the perceptor to convert selected patches into regional evidence strings
Use the adjudicator to answer from regional evidence, Navigation Summary, and optional Reference Context
Output: final answer a

C Experimental Results and Additional Analyses

This section groups all supplementary evidence after the experimental setup. We place the complete numeric tables first, then provide compact visual summaries, workflow illustrations, ablations, efficiency details, and negative findings.

C.1 Full Result Tables

Table 7: Full WSI-VQA table including BLEU-4. TITAN reports only MCQ because this zero-shot setting can only be mapped to answer options.

Method	Total	MCQ	Open	BLEU-1	BLEU-4	ROUGE-L
Pretrained MLLM Models						
GPT-4o [19]	27.62	32.56	22.03	0.085	0.038	0.236
GPT-5.4 [27]	35.65	42.05	28.41	0.142	0.071	0.288
Gemini-3.1-Pro-preview [13]	34.42	40.51	27.54	0.128	0.064	0.275
Qwen3.5-4B [28] (Slide-T)	26.39	42.56	8.12	0.092	0.045	0.241
Qwen3.5-9B [28] (Slide-T)	38.50	49.23	26.38	0.573	**0.571**	0.480
Supervised Pathology Models						
WSI-VQA model [6]	42.18	46.92	36.81	0.108	0.052	0.318
MedDr [14] (Slide-T)	45.17	43.59	46.96	0.368	0.178	0.452
TITAN [9] (zero-shot, CONCH v1.5 feats; MCQ-only)	—	34.10	—	—	—	—
Quilt-LLaVA [31] (Patch)	30.07	29.23	31.01	0.002	0.001	0.251
LLaVA-Med [22] (Patch)	21.22	35.64	4.93	0.003	0.001	0.227
WSI-LLaVA [24] (Patch)	33.20	42.82	22.32	0.057	0.027	0.312
MedGemma 1.5-4B [30] (Slide-T)	42.99	**53.08**	31.59	0.180	0.101	0.349
Training-Free Pathology Agents						
PathAgent [4]	33.88	51.54	13.91	0.427	0.389	0.335
PathNavigate (Ours)	**55.65**	51.54	**60.29**	**0.609**	0.507	**0.635**

SlideChat is listed for offline-footprint context, but its token accounting is not directly comparable: it is a trained single-forward slide MLLM rather than a training-free agent with per-question navigation, evidence selection, and adjudication calls.

Table 8: Full SlideBench-BCNB breakdown with merged HER2 and delta rows. HER2 averages the HER2-status and HER2-expression sub-tasks as in Table 1; Overall keeps the official seven-task weighting. All deltas are absolute percentage points relative to Patho-R1 alone.

Method	Overall	Tumor	ER	PR	HER2	Grading	Mol.Sub
Pretrained MLLM Models							
GPT-4o [19]	43.77	56.81	50.00	49.43	42.11	38.44	26.84
GPT-5.4 [27]	45.85	60.40	50.38	49.91	44.52	42.33	28.45
Gemini-3.1-Pro-preview [13]	44.91	58.51	50.19	49.62	43.57	40.82	27.60
Qwen3.5-4B [28]	39.48	49.43	33.36	27.32	48.02	52.70	19.19
Qwen3.5-9B [28]	42.00	53.88	36.39	30.15	47.54	**59.18**	21.46
Supervised Pathology Models							
SlideChat [8]	54.22	**89.89**	72.97	**74.01**	47.54	23.11	20.60
WSI-LLaVA [24]	52.86	85.35	60.78	62.00	42.77	46.33	29.21
MedDr [14]	35.48	59.64	43.76	39.22	27.41	29.37	20.79
TITAN [9] (zero-shot, CONCH v1.5 feats)	44.39	85.16	42.44	41.68	36.86	45.36	22.50
Quilt-LLaVA [31]	34.11	33.46	49.43	44.05	25.85	34.99	25.24
MedGemma 1.5-4B [30]	50.40	79.68	52.36	60.40	47.87	42.33	21.27
Patho-R1 alone (PathNavigate's perceptor, no agent)	37.26	42.16	54.63	50.38	25.61	35.53	26.65
Training-Free Pathology Agents							
PathAgent [4]	56.06	87.52	63.71	65.97	44.52	55.94	30.25
PathNavigate (Ours)	**59.42**	88.28	**76.09**	69.00	**48.39**	52.59	**32.33**
Δ vs Patho-R1 alone (pp)	**+22.16**	**+46.12**	**+21.46**	**+18.62**	**+22.78**	+17.06	**+5.68**

Table 9: Patch-level pathology MCQ transfer on PathMMU. Reported as a single accuracy column with letter-exact scoring. The PathNavigate row is the final local available-set run; most other rows are local available-set runs, while the Quilt-LLaVA row is taken from the PathMMU paper. We restrict the comparison to open-source pathology / general-purpose VLMs or paper-reported open-source baselines; closed-source frontier APIs are omitted because their PathMMU numbers are not reproducible under our protocol.

Method	PathMMU val Acc
Random choice	25.00
Pretrained MLLM Models	
Qwen3.5-4B [28]	38.16
Qwen3.5-9B [28]	45.84
Supervised Pathology Models	
Quilt-LLaVA [31]	26.50
TITAN [9]	32.18
WSI-LLaVA [24]	36.18
SlideChat [8]	37.42
Patho-R1 only [43]	42.18
MedGemma-1.5-4B [30]	50.62
Training-Free Pathology Agents	
PathAgent [4]	52.84
PathNavigate (Ours)	**53.60**

Table 10: Full numeric α sweep on WSI-VQA. The main paper summarizes the endpoint and default rows in Table 3. The α=0 row is the no-PLIP setting because the ranking is driven only by the surprise prior, and α=1 is the PLIP-only endpoint.

α	Total	MCQ	Open	BLEU-1
0.00 (surprise-only / no PLIP)	53.47	47.18	**60.58**	0.587
0.25	54.29	49.23	60.00	0.594
0.50 *(default, PathNavigate)*	**55.65**	**51.54**	60.29	**0.609**
0.75	54.97	50.51	60.00	0.598
1.00 (PLIP-only)	52.24	46.92	58.26	0.580

Table 11: Candidate-pool constructor ablation on BCNB. Full uses Table 1; Receptor is avg(ER, PR); other rows replace only candidate-pool construction.

Pool constructor	Total	Tumor	Receptor	Grading	Mol.Sub
Full PathNavigate (surprise pool)	**59.42**	**88.28**	**72.54**	**52.59**	**32.33**
PLIP-built	57.62	88.00	70.89	44.60	29.58
random	54.17	87.62	68.81	40.17	23.16
Δ PLIP $-$ Full PathNavigate	-1.80	-0.28	-1.65	-7.99	-2.75
Δ random $-$ Full PathNavigate	-5.25	-0.66	-3.73	-12.42	-9.17

Table 12: Memory architecture comparison. This table is deferred from the main paper because the main claim is structural; the appendix keeps the full footprint details.

System	Offline artefacts	Runtime substrate	Tokens/Q
PathAgent (training-free, query-driven; measured)	277 MB caption cache	per-Q trajectory	$\sim$30K
Patho-AgenticRAG (RL-enhanced agentic RAG; reported)	multi-GB ColQwen2 KB	router + KB rows	not reported
TissueLab (co-evolving tool agent; reported)	curated tool factories	HDF5 intermediates	not reported
SlideChat (trained slide MLLM; reported)	176K-VQA training corpus	single forward	not comparable
PathNavigate (training-free, ours; measured)	**4 MB in-domain case archive**	$\sim$4.5 **MB online-memory MLP**	$\sim$**17K**

C.2 Supplementary Comparison Plots

The main paper reports full numeric tables and the same-slide qualitative PathAgent comparison. Here we keep compact visual summaries that are useful for quick scanning after the full result tables.

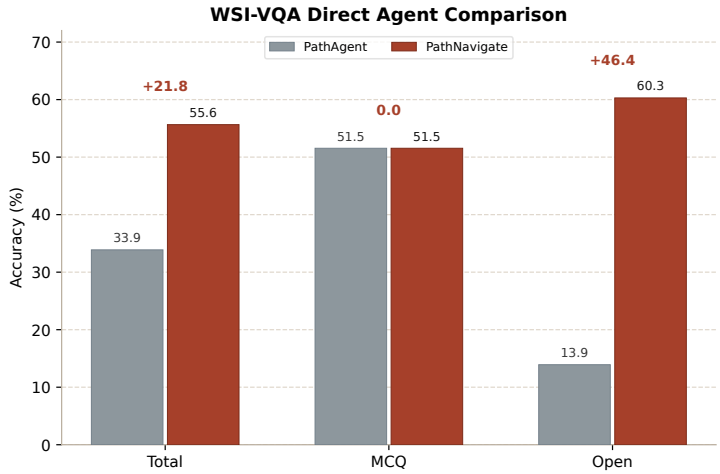

Figure 5: Agent-level WSI-VQA comparison between the PathAgent row and PathNavigate under the same Patho-R1/Qwen3.5 evaluation join. Bars report Total, MCQ, and open-ended accuracy; the largest separation is on open-ended answering, while MCQ is tied.

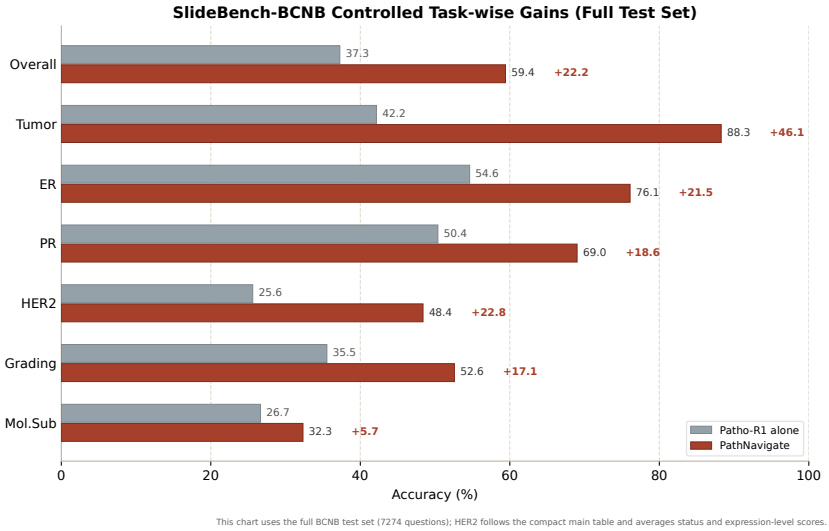

Figure 6: Task-wise SlideBench-BCNB comparison between Patho-R1 alone and PathNavigate under the unified protocol. The largest gains appear on Tumor, ER, PR, and the merged HER2 score, matching the compact main-table reporting.

C.3 Workflow Example

Workflow panels use a fixed layout so that cases can be compared on equal visual footing. Each panel includes seven blocks in a constant order: slide overview, surprise prior, PLIP relevance, combined ranking, top low-magnification candidates, selected evidence patches, and the final answer summary. The main paper uses the same-slide comparison in Figure 4; the single-system workflow panel below shows the full internal trace for PathNavigate.

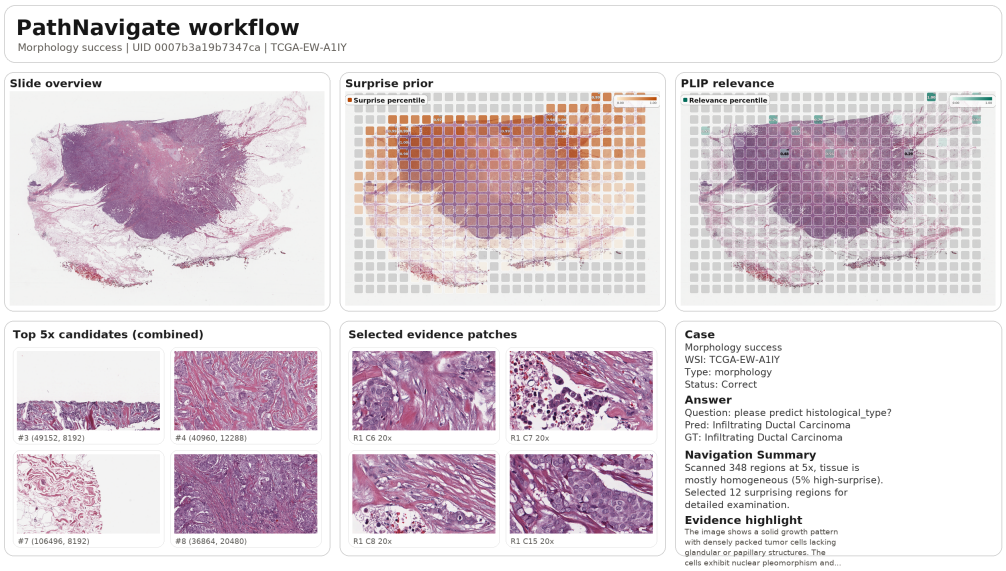

Figure 7: Representative workflow case for PathNavigate on WSI-VQA. From left to right, the panel shows the slide overview, surprise prior, PLIP relevance, combined within-pool ranking, top low-magnification candidate regions, selected evidence patches, and the final answer summary. Heatmaps are normalized within the slide, so stronger saturation denotes higher surprise or relevance.

C.4 Sensitivity and Component Figures

We group the remaining claim-bearing plots here so the main paper can stay table-led while the appendix still exposes the main shape of the ablation evidence. The corresponding numeric rows are collected in Tables 10, 11, 13, and 14, with negative-result checks in Appendix C.6.

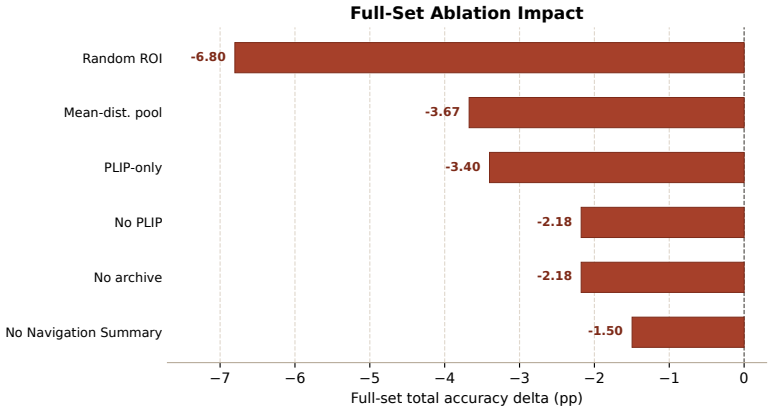

Figure 8: Compact full-set ablation impact summary. Each bar shows the total-accuracy delta against the full-set PathNavigate reference for that ablation. The archive-off and Navigation-Summary-off rows quantify the answer-time context paths, the PLIP endpoint and no-PLIP setting show that question-conditioned reranking is useful but not a standalone replacement for the scan prior, and Mean-dist. replaces the online surprise with a static feature-distance score.

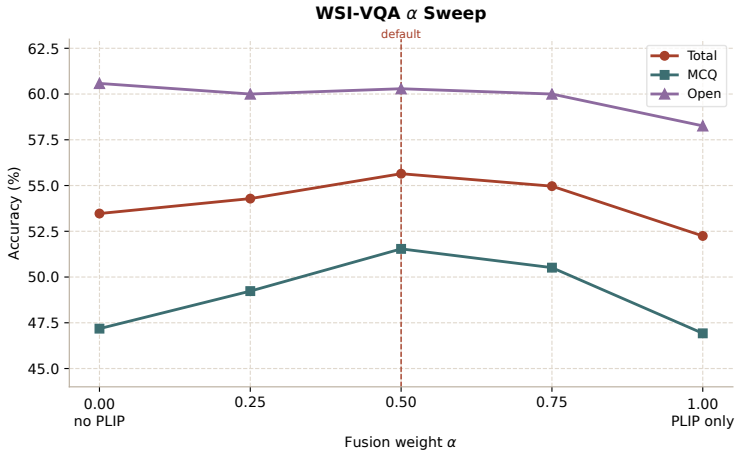

Figure 9: α sweep on WSI-VQA. The default α=0.5 gives the highest Total and MCQ accuracy; open-ended accuracy stays nearly flat until the PLIP-only endpoint, so the endpoint drops come mainly from MCQ. This supports the interpretation that PLIP is most useful as a within-pool refinement signal rather than as the primary proposal rule.

Table 13: High-magnification evidence readout ablation. The same selected low-magnification targets are held fixed while the evidence readout is changed. BCNB reports the aggregate Overall score.

Readout	Total	MCQ	Open	BCNB
Full	**55.65**	**51.54**	**60.29**	**59.42**
Global mem.	52.65	48.21	57.68	56.32
Random local	53.74	49.49	58.55	57.51
Low-mag only	50.48	45.38	56.23	52.36

This ablation isolates readout after target selection. Fresh local high-magnification memory outperforms global reuse, random local sampling, and low-magnification-only evidence, showing that selected ROIs still need nucleus-resolution morphology.

Table 14: Shared online-memory MLP architecture sensitivity on WSI-VQA. All five variants were rerun under the same sweep protocol and each reports 726 answered questions with 9 skips, so the numbers are read as a within-sweep comparison rather than mixed directly with the main-result row (its L2H768 row answers 407 questions correctly versus 409 in the main run). The default two-layer equal-width MLP remains best.

Variant	Total	Morphology	Clinical	Other
L1H768	50.96	58.40	49.79	45.10
L2H384	52.62	61.76	51.93	44.71
L2H768 (default)	**56.06**	**65.97**	**54.08**	**48.63**
L2H1536	53.44	61.76	51.93	47.06
L3H768	54.68	65.55	50.64	48.24

The sweep shows a simple pattern: collapsing the substrate to a single linear layer hurts most, while adding width or one extra layer does not improve over the default. This supports the main-paper choice of a shallow but non-degenerate two-layer online-memory MLP.

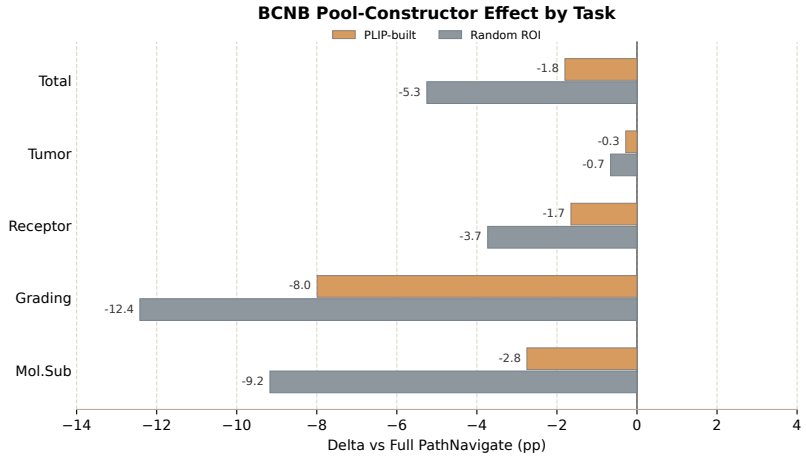

Figure 10: Candidate-pool constructor effect on BCNB. Bars show task-wise deltas relative to the Full PathNavigate main-result row, making receptor and grading-sensitive changes visible without over-emphasizing tumor-saturated categories where all constructors are close.

C.5 Efficiency Visualization

The main paper reports the training-free agent cost comparison as the bar chart in Figure 3. This appendix keeps the internal breakdown so the archive and PLIP costs remain visible.

Table 15: Internal PathNavigate efficiency breakdown on WSI-VQA under the same hardware and evaluation stack as Figure 3. These rows validate where the runtime and token costs enter, but they are not separate systems in the main comparison.

System	Latency (s/Q, median)	Peak GPU mem	Tokens/Q (peak)
PathNavigate (reported default)	$\sim$84	$\sim$22 GB	$\sim$17K
PathNavigate − archive (`abl-no-rag`)	$\sim$28	$\sim$22 GB	$\sim$11K
PathNavigate − PLIP (`abl-no-plip`)	$\sim$74	$\sim$21 GB	$\sim$17K

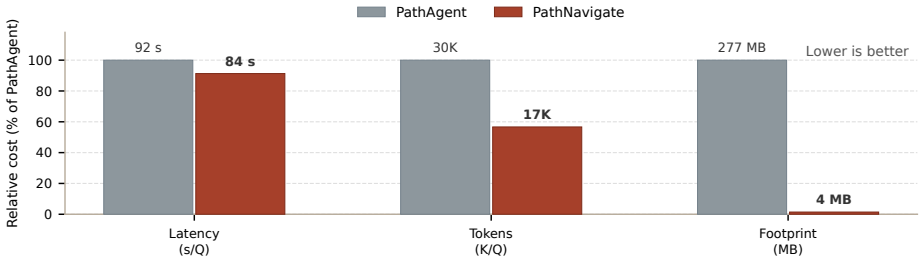

Figure 11: Detailed per-question cost profile for training-free agents, complementing the main-paper comparison in Figure 3. The breakdown keeps the direct PathAgent–PathNavigate comparison visible: latency is similar, token use is lower, and the offline footprint is much smaller.

C.6 Negative Findings

Two negative findings are worth recording even though they do not belong in the main story. We also fixed two implementation defaults from these checks before reporting the main experiments: a single adjudication pass matched or exceeded multi-round variants, and free-text perceptor evidence matched or exceeded a strict JSON schema.

- **Norm-preserving consolidation.** We tested an inverted update rule that only consolidated low-surprise patches. The idea was to protect a stable notion of normal tissue on tumor-heavy slides. In our setting, the change did not produce a reliable accuracy gain, so we keep it as a documented negative result rather than a main contribution.
- **Episodic trace memory.** We also tested an extra memory layer that prepended the full per-step trace to the adjudicator prompt. This consistently made the prompts longer and did not help final accuracy. We therefore removed it from the reported system.

D Additional Analyses

This section collects additional robustness, baseline, and evaluation analyses. Unless noted otherwise, every run keeps the questions, PLIP reranker, routing budgets, random seed (42), and language-model endpoints of the main configuration, and changes exactly one factor. Open-ended accuracy uses the same medical-synonym normalization as Table 2, and the 9 failed WSI-VQA questions are counted as incorrect, as in the main results.

D.1 Scan Encoder Swap

We replace CONCH v1.5 with UNI2-h [7] for both the $5\times$ scan features and the $20\times$ evidence features, and rebuild the case archive from the same 100 training WSIs with the matching encoder. The online-memory width follows the feature dimension. The 0.27-point difference in Total is not statistically significant.

Table 16: Scan-encoder swap on all 735 WSI-VQA questions.

	UNI2-h	CONCH v1.5 (main)
Feature dimension	1536	768
Total	411/735 (55.92)	409/735 (55.65)
MCQ	199/390 (51.03)	201/390 (51.54)
Open	212/345 (61.45)	208/345 (60.29)

We did not run the complementary swap of the PLIP reranker for CONCH or MUSK text–image embeddings; PLIP is used only as a lightweight within-pool reranker, and this swap remains open.

D.2 TITAN and PRISM Zero-Shot Protocol

TITAN. For each WSI we pass all $20\times$ CONCH v1.5 patch features, their level-0 coordinates, and the patch size to TITAN's `encode_slide_from_patch_features`, and reuse the

slide embedding across that slide's questions. Each answer option is treated as a class; four fixed templates (`CLASSNAME.`, `a pathology image of CLASSNAME.`, `a histopathology image of CLASSNAME.`, `an H&E-stained whole-slide image of CLASSNAME.`) are embedded with `zero_shot_classifier`, and `zero_shot` returns the option with the highest probability. This choice-only protocol never sees the question text and uses no Qwen, Patho-R1, online memory, PLIP, or archive. **PRISM** [32]. We extract Virchow [38] features for all tissue tiles at $20\times$ (224×224, no overlap, no top-k sampling), encode the slide with the official PRISM pipeline (checked against the official example), and apply the same choice-only templates.

Table 17: Slide foundation models on the 385 WSI-VQA MCQs that both can evaluate. Paired outcomes (choice-only): both correct 65, TITAN only 65, PRISM only 65, both wrong 190; option agreement 36.62%; exact sign test $p = 1.0$.

Prompting	TITAN	PRISM
Choice-only, four templates (main)	130/385 (33.77)	130/385 (33.77)
Question-aware, one template	119/385 (30.91)	126/385 (32.73)
Question-aware, four templates	—	119/385 (30.91)

Both models reach the same accuracy but agree on barely a third of the answers, so their errors are largely complementary rather than shared. Tables 2 and 7 report the same TITAN choice-only run on all 390 WSI-VQA MCQs (133/390, i.e., 34.10; TITAN answers 3 of the 5 MCQs outside the common set correctly); the 385-question restriction is needed only for pairing with PRISM.

D.3 Region-Level Alignment on CAMELYON16

To test whether high surprise falls on diagnostic tissue, we score tiles of CAMELYON16 [11] slides, which carry lesion annotations. The 96 slides are split by source into 48 development and 48 held-out slides; settings were chosen on the development half only, and Table 18 reports the held-out half.

Table 18: Surprise as a lesion ranker on 48 held-out CAMELYON16 slides.

ROC-AUC	AP	Precision@10	Recall@10	Hit@10
0.537	0.199	0.179	0.052	0.458

Surprise is only weakly aligned with lesions over the whole ranking, with stronger enrichment among the top-ranked tiles. Enlarging the candidate set mainly improves coverage: over all 96 slides, lesion recall rises from 0.041 at 10 candidates to 0.366 at 100, and the fraction of slides with at least one lesion hit rises from 0.354 to 0.698. We therefore treat surprise as a slide-specific proposal prior whose precision depends on the downstream reranking and $20\times$ readout, not as a lesion probability.

D.4 Semantic Re-Judging of Open-Ended Answers

We take a fixed set of 80 open-ended questions for which both PathNavigate and PathAgent have saved predictions, hide which system produced each answer, and randomize the A/B order. A Qwen3.5-9B judge reads the question, the reference answer, and both predictions, and labels each prediction as correct, partially correct, or incorrect. It also gives a separate pairwise preference (A better, B better, or tie); this preference is a judgment of its own and is not derived from the three labels. No judge output failed to parse.

Table 19: Blinded semantic judging of 80 shared open-ended questions. Separate pairwise preference judgment from PathNavigate's side: 54 wins, 25 ties, 1 loss.

	PathNavigate	PathAgent
Strictly correct	60/80 (75.00)	12/80 (15.00)
Correct or partially correct	61/80 (76.25)	22/80 (27.50)

The gap persists under semantic judging, so it is not produced by substring matching. A single automatic judge is not a substitute for blinded pathologist review.

D.5 Traversal-Order Sensitivity

The online memory is updated sequentially, so the scan order can change the surprise field. On 60 WSIs with fixed feature caches, we re-run the scan under different orders and compare the proposal ranking and top-30 pool with raster order (Table 20). We then run the full pipeline on 100 pre-selected questions whose answers depend on image evidence, changing only the $5\times$ traversal order (Table 21).

Table 20: Proposal-level change relative to raster order (60 WSIs).

Order	Mean Spearman	Top-30 Jaccard
Serpentine	0.996	0.778
Column-major	0.309	0.356
Reverse raster	-0.419	0.208
Five fixed random orders (mean)	0.267	0.250

Table 21: End-to-end accuracy across traversal orders (100 questions). p: paired test against raster.

	Raster	Reverse raster	Fixed random
Total	44/100	48/100	44/100
MCQ	28/72	32/72	28/72
Open	16/28	16/28	16/28
Mean perceptor calls	12.89	13.26	14.07
p vs. raster	—	0.503	1.000

Proposals are clearly order-dependent: only the near-identical serpentine order preserves them, and reversing the order yields a negative rank correlation, which is consistent with part of the score reflecting position in the stream as well as content. Final accuracy did not differ significantly on this subset, presumably because the fixed budget, PLIP reranking, and $20\times$ verification absorb part of the variation, but the method is not order-invariant. A spatially uniform two-stage threshold estimate did not consistently improve agreement. Scoring all tiles with the state frozen after a full fitting pass would remove the order effect from the scores and is left to future work.

D.6 Static Mean-Feature-Deviation Baseline

We replace the online surprise with the cosine distance between each tile feature and the slide-mean feature, keeping NMS, PLIP reranking, and all budgets. The two scores are nearly unrelated at the proposal level (mean Spearman 0.014, top-30 Jaccard 0.094).

Table 22: Mean-distance proposals on all 735 WSI-VQA questions (Random ROI: 48.84 Total).

	PathNavigate	Mean-dist. pool	Difference
Total	409/735 (55.65)	382/735 (51.97)	+3.67
MCQ	201/390 (51.54)	187/390 (47.95)	+3.59
Open	208/345 (60.29)	195/345 (56.52)	+3.77

Mean deviation is an informative deterministic baseline (above Random ROI), and online surprise improves on it. Because the static score is order-invariant, it is also a natural fallback when order robustness matters more than accuracy.

D.7 Budget Expansion on Low-Dispersion Slides

We pre-select 100 morphology questions whose slides have the lowest coefficient of variation of surprise and enlarge the ROI pool / base ROI budget K_0 / perceptor-call cap from $30/10/15$ to $60/20/25$ (morphology questions add $+2$ to K_0 in both settings, Table 6), keeping everything else fixed.

A larger budget gives a small, non-significant gain at a clear cost, so we keep the default and list adaptive budgeting as a limitation.

Table 23: Default versus enlarged budget on 100 low-dispersion questions. Paired outcomes (both correct / default only / enlarged only / both wrong): $59/3/5/33$; exact sign test $p = 0.727$.

	$30/10/15$ (default)	$60/20/25$	Change
Total	$62/100$	$64/100$	$+2$
MCQ	$32/63$	$33/63$	$+1$
Open	$30/37$	$31/37$	$+1$
Mean perceptor calls	14.42	21.50	$+49.1\%$
Mean time per question (s)	119.8	165.1	$+37.8\%$
Mean $20\times$ tiles scanned	396.9	637.8	$+60.7\%$

D.8 Source of Hyperparameters

Table 24: Final values and how they were chosen.

Parameter	Value	Source
Memory input / hidden width	$768/768$	matches stored CONCH v1.5 features
Memory depth	2	implementation default; Table 14
Warm-up tiles	100	development-stage choice
Learning rate / threshold scale	$0.05/1.0$	implementation default
ROI pool size K_{pool}	30	compute budget
Planned ROIs (other/morph./clin.)	$10/12/13$	question-type heuristic
Min. spacing (morph./clin.)	$4096/20480$ px	question-type heuristic
Min. spacing (other)	8% of slide diagonal	scale-adaptive heuristic
PLIP weight α	0.5	symmetric default; Table 10
Patches per ROI / perceptor-call cap	$2/15$	compute budget
Adjudication rounds / seed	$1/42$	fixed configuration

D.9 Design Comparison with PathAgent

Table 25: PathNavigate versus PathAgent (official implementation, as reproduced in this paper). We do not claim that online memory, surprise, retrieval, or local readout is individually new; the difference lies in when the question enters and how the slide state is reused.

Axis	PathNavigate	PathAgent
Slide state	online memory built by a full low-magnification scan, re-initialized per slide–question episode	no slide-level state; retrieval starts from the question
Initial candidates	question-independent surprise pool	question–patch relevance retrieval
When the question enters	after the pool is built: routing and within-pool reranking	from the first retrieval step
Follow-up search	bounded $20\times$ readout of selected ROIs	further retrieval driven by missing information, up to 5 rounds
Evidence budget	≤ 2 patches per ROI, ≤ 15 perceptor calls	iterative patch retrieval without a fixed call cap
Case archive	$k = 3$ retrieved training cases	not part of its design

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and Section 1 state the scope as training-free WSI-VQA with frozen backbones and a scan-search-readout routine. The claims are supported by the experiments in Section 4, with protocol, ablation, and efficiency details in the appendix.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Section 6 is a dedicated Limitations section covering surprise versus lesion alignment, traversal-order dependence, fixed budgets, heuristic hyperparameters, evaluation scope, and non-clinical use; Appendix D reports the supporting analyses and Appendix C.6 records negative findings.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [N/A]

Justification: The paper does not present theoretical theorems or proofs. The mathematical expressions define the proposed test-time memory, surprise scoring, routing, and evidence readout procedure.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: Section 4.1 and Appendix A.1 specify datasets, splits, metrics, model identifiers, software, hardware, and evaluation joins. Appendix B.1, Appendix B.3, and Appendix A.3 provide routing, pseudocode, and the controlled PathAgent reproduction protocol.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The abstract provides the public code URL, and the experiments use public benchmark datasets or documented API/local model endpoints. Appendix A.1 and Appendix A.2 describe the data sources, preprocessing assumptions, model names, and evaluation settings needed to reproduce the reported rows.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The method is training-free, and the test-time settings are specified in Section 4.1 and Appendix A.1. Hyperparameters such as the online-memory architecture, learning rate, PLIP fusion weight, case-archive size, routing budgets, and baseline model identifiers are listed in the appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

Justification: The main results are reported on fixed public test sets with deterministic evaluation joins and controlled ablations, but we do not report confidence intervals for the main tables. Paired exact tests are reported for the traversal-order, budget, and TITAN/PRISM analyses (Appendix D), and we disclose sample sizes, metrics, protocols, and full result tables; repeated API/model runs would be substantially more expensive and may introduce endpoint-version variability.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Appendix A.1 specifies the CPU/GPU host, GPU memory class, software stack, and vLLM serving setup. Section 4.4 and Appendix C.5 report per-question latency, token count, GPU memory, and offline footprint for the main training-free agent comparisons.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The work uses public or previously released pathology benchmarks and does not collect new identifiable patient data or involve new human-subject experiments. The paper reports limitations, reproducibility details, and responsible-use context for a non-deployed research prototype.

   Guidelines:

- The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Section 6 (Broader impacts) discusses both sides. The potential positive impact is more efficient and auditable whole-slide evidence selection for pathology VQA research. The paper also limits clinical claims: incorrect answers, H&E-weak labels, dataset bias, and non-deployment risk mean the system should not be used for autonomous diagnosis without expert review and validation.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper does not release a new high-risk pretrained language model, image generator, or scraped dataset. The released asset is code for a research WSI-VQA agent that uses existing public benchmarks and existing model endpoints/checkpoints under their respective terms.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.

- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: Section 4.1 and Appendix A.2 cite the datasets, baselines, foundation models, APIs, and toolchains used in the experiments, and list exact model or endpoint identifiers where relevant. Each asset is used under the license or terms of use stated on its original release page, and we do not redistribute restricted model weights or private data.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The new asset is the public code release for PathNavigate, not a new dataset or pretrained checkpoint. The paper and appendix document the method, model dependencies, evaluation joins, hyperparameters, and reproduction protocol that accompany the released code.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

Justification: The paper does not conduct crowdsourcing experiments or recruit human participants. It evaluates models on existing pathology VQA and benchmark datasets.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: The study does not involve newly recruited human subjects, interventions, or new patient-data collection. It uses existing public or released benchmark datasets and therefore does not require a new IRB protocol for human-subject recruitment in this work.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: LLM/VLM use is central to the method and is described in Sections 1–4: PathNavigate uses frozen Patho-R1-7B as the pathology perceptor and Qwen3.5-4B as the adjudicator, with additional pretrained MLLMs used as baselines. Appendix A.2 lists the exact model names and endpoints used in evaluation.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.