This MICCAI paper is the Open Access version, provided by the MICCAI Society. It is identical to the accepted version, except for the format and this watermark; the final published version is available on SpringerLink.

Supplementary Material: Method Description

REG2026 — Pathologist Reasoning-Guided Report Generation Challenge

Overview

Given a whole-slide image (WSI), the task requires (i) a step-by-step chain-of-thought (CoT) reasoning trace ending in a structured pathology report (*Interface 1, Workflow Reasoning*, Metric A, 70% of the overall score), and (ii) free-text answers to questions about a given region of interest (ROI), including correctly rejecting non-informative background regions (*Interface 0, Visual Grounding*, Metric B, 30%). This document summarizes the design decisions behind our submission and the reasoning behind them; quantitative results are reported separately.

1 Data Acquisition and Preprocessing

1.1 WSI download

Training WSIs (approx. 11,220 cases across 7 organs) and the CoT question–answer corpus were obtained from the organizers' CDN as a JSON manifest listing per-file URLs. A single-connection download of gigapixel TIFFs was I/O-bound and the dominant cost of the whole pipeline; we switched to `aria2c` with multiple parallel connections per file, which reduced average per-file download time by roughly an order of magnitude and made encoding GPU-bound rather than I/O-bound. WSIs are deleted immediately after encoding to bound local disk usage, since the raw pixel data is not needed once patch embeddings are computed.

1.2 Patch selection

A WSI is several gigapixels and cannot be encoded in full. Patch selection depends on whether the TIFF exposes a low-resolution pyramid overview level: a random 42-WSI sample found 41/42 (97.6%) did not, so blind random (y, x) sampling is the dominant path in practice, not a rare fallback (main paper, Fig. 2c). When an overview level is available, we instead use a stratified approach: tissue regions are located from the pyramid overview, a coarse 8×8 grid (in downsampled overview-pixel coordinates) is imposed over the tissue mask, and candidate patches (224×224 px, 0.5 μm/px, matching the foundation model's native resolution) are drawn evenly from every non-empty grid cell. This avoids clustering all sampled patches in a single region and guarantees coverage of the whole slide whenever the overview is available. Under either path, roughly $3\times$ the target number of candidates are read in parallel and ranked by pixel-intensity variance, used as an inexpensive proxy for structural complexity (empty stroma has low variance; glandular/atypical regions have high variance); the top-scoring patches are kept. WSIs for which this procedure still yields too few tissue patches fall back further to a wider, purely random candidate pool merged with whatever was already found, so that very sparse slides still produce a usable patch set instead of an empty one.

1.3 Stain normalization

The data originates from multiple institutions, countries and scanner models, which introduces H&E color variation unrelated to tissue morphology and adds noise to the embedding space. Normalization is applied only to patches that survive the artifact-rejection filter (main paper,

Section 2.1), not before, so the filter's thresholds see raw, un-normalized color. We apply Macenko stain normalization on GPU (`torchstain`), fitting a single reference from the first 5 WSIs processed: candidate patches are pooled across these slides and the patch whose pixel-intensity variance is closest to the pool's median is used as the fit target, rather than an arbitrary single WSI (which the literature suggests is not representative). The same reference is serialized and reused at inference time, so that training and deployment see color-normalized inputs consistently — an earlier version of the encoder used at inference time skipped stain normalization entirely, producing a train/inference distribution mismatch that was identified and corrected.

2 Slide-Level Feature Encoder

We use **H-Optimus-1** (Bioptimus, ViT-Giant/14, gated release) as a frozen patch encoder, run entirely offline from local weights (the deployment environment has no internet access, so any model relying on a live hub download was excluded regardless of quality).

The choice was informed primarily by the previous edition of this challenge: the top-ranked team benchmarked five foundation models and found H-Optimus-1 to be both the strongest performer and the most stable across institutions and scanners, which we took as a domain-specific signal more relevant than generic pathology foundation-model leaderboards. Two alternatives were considered and rejected:

- **Virchow2** as a second encoder in an ensemble: measured GPU memory was already close to the deployment limit with a single encoder, and we found no evidence that Virchow2's advantages on generic classification/subtyping benchmarks would transfer to a workflow-reasoning and report-generation task, which is qualitatively different from what those benchmarks measure.
- **TITAN** as a ready-made slide-level encoder: it is co-trained with a different tile encoder (CONCH) and is therefore not compatible with H-Optimus/Virchow2 patch embeddings — adopting it would have meant standing up a third, unvalidated encoding pipeline from scratch this close to a single, irrevocable submission deadline, for a model that is optimized for retrieval/captioning rather than the organ- and template-selection task we need.

3 Multiple-Instance Learning Aggregation

Patch embeddings are aggregated into a single slide-level representation with an attention-based MIL (ABMIL) head, trained rather than replaced by simple mean-pooling, since attention pooling lets the model down-weight uninformative patches instead of diluting the diagnostic signal.

Multi-task training. The aggregator is trained jointly on two objectives: organ classification and diagnostic-template classification, rather than on organ labels alone. Ablating this against several alternative MIL architectures (DSMIL, TransMIL) trained only on organ labels showed that *what* the embedding is optimized for mattered more than the attention architecture itself: architectures trained only on organ classification produced embeddings that were harder to separate for template selection downstream, even when their organ accuracy was equal or better. This is the reason we did not adopt TransMIL despite it yielding the best standalone organ classifier.

Ensemble of five folds. Rather than a single model trained on the full dataset, we train five independent ABMIL instances on five cross-validation folds and average softmax organ probabilities across them at inference time. This mirrors a random-forest-style variance-reduction argument (multiple models trained on overlapping 80% subsets, averaged, versus one model trained on 100% of the data) at zero additional inference cost, and — importantly for a project

with a single, non-revocable final submission — it is the only architecture that still allows any form of local held-out validation, since a single model trained on 100% of the data would leave nothing locally unseen to validate against.

4 Diagnostic Template Selection (Graph Head)

Each CoT trace follows one of several dozen organ-specific question/answer graphs ("templates"), pre-mined from the training corpus. Rather than generating this graph structure token-by-token, we treat template selection as a classification problem: a small head predicts, per organ, which pre-mined template the case follows, and the graph's edges are read off directly from a lookup table. This trades template novelty for reliability on graph-structure metrics, and lets the generative model focus entirely on producing answer text rather than also having to reproduce exact question wording.

We compared a plain MLP head against a nearest-prototype (cosine-similarity) classifier with class re-weighting, initially motivated by the fact that many templates have only a handful of training examples. Once evaluated directly on the final multi-task-ABMIL embeddings (rather than on the embeddings used in an earlier, now-superseded ablation), the plain MLP consistently outperformed the prototype-based classifier, including a variant where the prototypes were properly frozen as originally intended. We adopted the plain MLP as the final architecture on this basis; class re-weighting was kept as an available option but not used in the final configuration, since it did not provide a consistent additional benefit once the embedding itself was multi-task-trained. As with the encoder ensemble, five folds are trained and averaged rather than a single model, for the same variance/validation argument.

5 Answer and Report Generation

Free-text answers (for questions not covered by a high-confidence lookup of the most common training answer) and the final structured report are generated by **MedGemma 1.5 4B** with a **LoRA** adapter, rather than the frozen base model or a full fine-tune. LoRA was chosen over full fine-tuning to keep the frozen base model reusable for both interfaces and to keep the adapter small enough to package alongside the rest of the model weights under the platform's size and load-time constraints; a low-rank adapter over the attention projection matrices was judged sufficient given that the target task is stylistic/domain adaptation (matching the reporting conventions and terminology of the training corpus) rather than acquiring new visual capability. The adapter is trained on the full training corpus rather than a held-out split, since (a) the evaluation text distribution on the true test set is disjoint from anything the adapter could see regardless, so there is no leakage risk analogous to the classification heads, and (b) more training text directly improves report style/terminology coverage.

An early version of this adapter was trained on text-only sequences while the deployed inference path conditions generation on the slide thumbnail image for several steps; this train/inference distribution mismatch was identified and the adapter retrained to remove it. A high-confidence lookup path (returning the most common training answer when consensus for a given (organ, question) pair exceeds a fixed threshold) is used ahead of the generative model for lower-variance questions, reserving the LoRA-adapted model for questions where a simple majority answer is not representative.

6 Retrieval-Augmented Generation (RAG)

For the diagnosis step and for report-style conditioning, we retrieve the nearest training cases by cosine similarity in the slide-embedding space (the same embedding space produced by the

MIL encoder) against an index built from the full training corpus, rather than relying purely on the generative model. Retrieval excludes the query case itself by exact filename match against the index, so a query WSI can never retrieve its own report – a no-op at deployment (the test WSI is never in the training index) but required for any local evaluation that reuses a training WSI as a query, as our own ablation does (main paper, Section 3.2). The rationale is that a large share of the scoring criteria for the final report reward near-literal overlap with reference text (keyword/entity overlap, n-gram overlap), which favors retrieving and adapting a genuinely similar, human-written case over paraphrasing from a language model. Retrieval is restricted to slide-embedding space (rather than, e.g., a separate text encoder over questions) so that the retrieved neighbors are tissue-morphology–similar, not just organ-matched.

A behavior-consistency step constrains style-example retrieval to training cases whose diagnosis behavior (benign / malignant / in-situ) agrees with what has already been established earlier in the same case's reasoning chain, falling back to plain nearest- neighbor retrieval when no consistent candidate is found. This was added after observing that the free-generation and pure-retrieval components of the pipeline could independently reach different conclusions for the same case (e.g. the model establishing "benign" early in the chain, then retrieval independently substituting a malignant example downstream purely because it was the nearest neighbor by embedding). Because retrieval always has a safe fallback to its original behavior, this constraint can only improve or maintain consistency, never make it worse.

7 Visual Grounding Interface (Metric B)

Interface 0 receives a single ROI thumbnail and a question, and must (a) recognize non-informative background regions and reject them without attempting a diagnostic claim, and (b) otherwise answer using the tissue actually visible in the crop.

Background rejection is implemented as a lightweight heuristic (fraction of near-white pixels, supplemented by a mean-saturation signal) rather than a learned classifier: the official background ROIs in this task are visually unambiguous (near-pure white), and a heuristic is inexpensive, fully deterministic, and avoids adding another trained component whose failure mode would be silent. When the heuristic fires, a fixed, non-diagnostic string is returned directly, which by construction cannot make an incorrect diagnostic claim on background. For tissue regions, the frozen MedGemma model answers directly from the ROI image and question text. We considered routing tissue-ROI answers through the same LoRA adapter used for Interface 1 (for architectural and prompt consistency across both interfaces) but this was not implemented in the submitted version; sharing the same underlying model family and, where practical, the same tissue/background logic between both interfaces was nonetheless treated as a design goal, since the two interfaces are graded as components of the same overall system rather than independent submissions.

8 Deployment Constraints and Their Effect on the Design

Several platform constraints shaped the choices above rather than being incidental:

- **No internet access at runtime** ruled out any component that fetches weights or indices on demand (e.g. hub-hosted model strings); all weights, including the foundation encoder, LoRA adapter and RAG index, are packaged into the submitted model artifact and loaded from local disk.
- **One container invocation per case**, with every model loaded from scratch each time, put a premium on keeping cold-start weight loading fast and ruled out designs that assume any warm state between cases (e.g. a persistent cache), since none is available in production.

- **A fixed per-case time budget and bounded GPU/RAM** motivated patch-count subsampling instead of exhaustive whole-slide encoding, and influenced the decision to keep the foundation encoder and LLM frozen (only lightweight heads and a low-rank adapter are trained), keeping both the packaged weight footprint and per-case compute predictable.
- **A single, non-revocable final submission** was the main argument, in several places above, for preferring designs that preserve some form of local validation (k-fold ensembling for the trained heads) over designs that would use 100% of the available data with no way to sanity-check locally before the final, irreversible submission.

9 Variance-Ranked vs. Random Patch Retention (Reviewer Follow-Up)

Full numbers behind the summary in the main paper's Discussion. Responding to a reviewer request, we compared the spatial spread of variance-ranked vs. randomly-retained patches from the same, already artifact-filtered candidate pool, on two independent samples: variance-ranking touches a mean 9.6–10.1 of 64 spatial grid cells vs. 20.7–22.1/64 for random retention (56-WSI and 105-WSI samples respectively), a consistent $> 2\times$ narrowing. Since tissue itself does not always fill the grid (mean 49% of cells contain any tissue across both samples), we also normalized to each WSI's own tissue extent: variance-ranking still uses only 29% of available tissue-bearing cells vs. 67% for random retention, so the narrowing is not an artifact of sparse slides. This isolates a *spatial-coverage* bias from the separately-measured *artifact* question (main paper, artifact-filtering paragraph): both selections draw only from candidates that already passed the artifact filter, so kept patches are mostly genuine tissue either way – what differs is how much of that tissue each strategy actually samples. Whether this narrower coverage costs anything in downstream BPV was not established here; that would need re-training `graph_head` on a randomly-sampled variant, out of scope this close to the camera-ready deadline.