

APOLO: An Inspectable Vision-Language Workflow for Medical-Image Research

Sami Halawa Ribas

Independent Researcher, Madrid, Spain

ORCID: [0009-0002-2021-1983](https://orcid.org/0009-0002-2021-1983)

Technical report | 30 October 2026

Abstract

Medical-image assistants combine visual recognition, language generation and clinical interpretation in outputs that can be difficult to inspect or evaluate separately. APOLO explores a description-first workflow: record what a model observes, distinguish those observations from their interpretation, and preserve a human review point before an output informs a decision. This report specifies that architecture, documents the present DeepSeek-VL2-Tiny inference prototype, and defines an evaluation protocol for a separately deployed interpretation stage. The current prototype performs a single multimodal generation organized by prompting into findings, rationale and conclusions; it does not enforce a separate image-inaccessible diagnostic process. An accompanying low-rank adaptation example originates from an upstream COCO-captioning repository and is presented with attribution, not as an APOLO clinical experiment. No diagnostic accuracy, clinical validation, privacy certification or hardware-performance result is claimed. The contribution is an engineering specification that makes model boundaries, evidence provenance, failure states and evaluation obligations explicit. It provides a practical foundation for testing whether a reviewable textual representation can support useful downstream interpretation without obscuring information loss or treating generated explanations as verified reasoning.

Keywords: vision-language models; medical-image research; structured findings; model orchestration; human review; reproducibility; local inference.

1. Problem and contribution

A medical-image assistant has at least three different responsibilities: recognize visual evidence, interpret that evidence in context, and communicate uncertainty in a form a reviewer can assess. Combining them into one fluent answer makes implementation convenient, but makes failures difficult to locate. A conclusion can be wrong because an abnormality was not described, because a correct description was misinterpreted, or because a report expressed unwarranted certainty. These errors require different remedies.

APOLO investigates an explicit interface between observation and interpretation. Its intended workflow produces a structured description before an interpretation and keeps both available for review. The central engineering question is whether this representation makes errors easier to identify and measure while retaining the information required by a downstream task. This question is distinct from whether a language model can generate a convincing clinical narrative.

This report makes three contributions. First, it describes the implemented inference path and identifies the difference between prompt-level separation and process-level isolation. Second, it proposes a versioned observation contract, failure-handling rules and a clinician review boundary for a genuinely separated deployment. Third, it specifies a study design that can test descriptive fidelity, error propagation, downstream task performance and operational cost independently. The proposal remains testable even if a particular foundation model is replaced.

The intended audience is researchers and engineers building image-to-report prototypes. The report is not a diagnostic-model validation study. Its medical examples motivate requirements; they do not establish clinical efficacy, suitability for deployment or equivalence to specialist assessment.

2. Technical context

Modern vision-language models map visual inputs and text instructions to generated language. DeepSeek-VL2 provides the foundation-model family used by the present prototype [1]. Its published capabilities supply a starting point for implementation, not evidence of performance on APOLO's intended medical tasks. Model behavior depends on the selected checkpoint, processor, image preparation, prompt, decoding settings and evaluation population.

BLIP-2 illustrates a different approach to connecting pretrained image and language components [2]. It is relevant architectural background rather than a component of this implementation. APOLO's proposed observation interface is external to the internal cross-modal mechanisms of either model: it makes a generated representation available to software and reviewers between task stages.

Low-rank adaptation, or LoRA, is a parameter-efficient method for adapting pretrained models [3]. The existence of a LoRA adapter or a decreasing training loss does not establish the domain, authorship or diagnostic utility of a training run. Dataset provenance and task definitions remain essential. COCO is a general visual-recognition dataset [4]; a captioning example based on it cannot serve as evidence of medical-image validation.

These distinctions motivate a narrower, reproducible design claim. A readable intermediate report can support inspection and controlled experiments. Whether it preserves clinically significant evidence, reduces errors or improves reviewer performance must be measured.

3. Current prototype

The current Python inference prototype loads a processor, tokenizer and causal language model using a configurable base-model identifier, defaulting to `deepseek-ai/deepseek-vl2-tiny`. It converts available input images to RGB, constructs a multimodal conversation, and generates a response. An optional PEFT adapter can be loaded from a supplied directory. A base-model-only mode is also exposed.

The default instruction asks the model to describe its observations, provide a rationale in a marked section, and then state conclusions. These sections are produced by the same model invocation. They are useful formatting targets, but they do not prevent the model from using the image while producing an interpretation. Nor do they establish that a later section is causally derived only from an earlier textual section.

The generation configuration uses sampling, with default temperature 0.4, top-p 0.9, a repetition penalty of 1.1 and a maximum of 1,024 new tokens. The code selects CUDA when available and otherwise uses the CPU. Those settings describe program configuration; they are not measured latency, throughput or memory results. Successful execution on a specified hardware/software combination must be demonstrated before claiming a deployment envelope.

Two implementation choices deserve particular attention. The loader catches an adapter-loading error and continues with the base model. That behavior is convenient during development but can obscure which system produced a response. A reproducible evaluation must either fail on an adapter error or clearly record that the intended configuration was not executed. The image loader can also skip images that fail to load. A study must check the expected and loaded image counts before generation, rather than accepting a response from a partial examination.

Capability	Present implementation	Separated-system specification
Image preparation	RGB image loading	Explicit modality, orientation and preprocessing checks
Findings and interpretation	Sections requested in one generation	Two independently invoked stages
Adapter selection	Optional; base-model fallback possible	Exact artifact identity and explicit failure policy

Capability	Present implementation	Separated-system specification
Text-only downstream access	Not enforced	Interpretation process receives only approved text
Structured validation	Formatting requested by prompt	Schema and content validation before progression
Human review	Intended workflow requirement	Recorded approval, edits and escalation
Clinical evaluation	Not reported	Prespecified independent study

4. Observation and interpretation contract

A separated APOLO workflow contains an observation service, a reviewable record and an interpretation service. The observation service receives the image and a task-specific instruction. Its output is a draft, not ground truth. A validator checks that required fields exist and that the output can be associated with the intended input. A reviewer can correct, reject or approve the description before further interpretation.

The proposed observation record includes a record identifier; modality and view when known; image-quality limitations; anatomical observations; missing or uncertain observations; preprocessing and prompt versions; model and adapter identifiers; and a generation identifier. An observation should distinguish an absent finding from a finding that could not be assessed. It should also distinguish visual estimates from measurements computed by a validated method. For example, an estimated anatomical ratio must not be presented as an instrument measurement.

The representation should avoid introducing patient identifiers that are unnecessary for the task. Nevertheless, removing pixels does not anonymize the resulting description. Text may contain identifying information, rare findings or contextual details. Both the image and the generated record require appropriate access, retention and transmission controls. The architecture changes a data interface; it does not by itself establish confidentiality or compliance.

The interpretation service receives an approved observation record and permitted contextual information. It must distinguish statements supported by that record from hypotheses requiring additional evidence. If a required observation is absent, the appropriate behavior is to state the gap or abstain rather than infer an unobserved feature. Any confidence value must have a defined estimation and calibration procedure; a number generated in prose is not a calibrated probability.

A final reviewer sees the source description, the interpretation, the model identities and the unresolved limitations together. Corrections should create a new version of the record so that the chain from original input to approved report can be reconstructed. The reviewer should be able to return to the image: a text-only downstream model does not imply that human review should lose access to the original visual evidence.

```

Proposed separated workflow
  image + acquisition context
    |
    v
  observation model -> draft observation record
    |
    v
  schema checks + review -> approved or rejected record
    |
    v
  text-only interpretation model -> draft interpretation
    |
    v
  human review -> accepted report, revision or abstention

```

The present single-generation prototype is a starting point for this workflow. It is not an implementation of all the boundaries in the diagram. A proposed separate language-model stage may use an appropriate locally deployable model; no particular Mistral checkpoint or quantized diagnostic adapter is claimed as implemented or evaluated here.

5. Control flow and failure handling

The following pseudocode specifies the intended evaluation behavior. It describes a design contract rather than executable code supplied with this report.

```
require expected_image_count == successfully_loaded_image_count
record image provenance, model revisions and generation settings
if requested_adapter_is_unavailable:
    stop and report configuration failure
observation = generate_observation(images, observation_prompt)
if observation_does_not_match_schema:
    return explicit validation failure
reviewed = review_and_version(observation)
if reviewed.status != approved:
    return revision request or abstention
interpretation = interpret_text(reviewed.record, permitted_context)
flag unsupported statements and unresolved missing observations
return interpretation together with reviewed record for final review
```

There should be no silent substitution between model configurations in a comparison. A retry policy should be prespecified and bounded by a meaningful failure condition. Repeated generation until a satisfactory answer appears changes the effective system and must be included in the reported cost and outcome analysis.

Operational tests should distinguish input rejection, generation failure, schema failure, reviewer rejection and downstream abstention. A single aggregate success rate hides these different mechanisms. Logs should record enough configuration and timing information to reconstruct a run while excluding unnecessary image content or identifying text. Configuration records and sensitive clinical payloads should have separate access policies.

6. Training-example provenance

The accompanying development material includes a DeepSeek-VL2-Tiny supervised LoRA example using COCO-2014 captioning. Its trainer-state artifact matches the public upstream DeepSeek-VL2-LORA-Finetune repository [5]. The example is attributed to that upstream project and is not reported as a training experiment performed by the APOLO author.

The recorded validation values illustrate how training artifacts can be described without changing their meaning:

Recorded step	Evaluation loss	Evaluation token accuracy
2,000	1.6408386	0.6041866
4,000	1.6160347	0.6084486
6,000	1.6107216	0.6089608

These are upstream example values, not new results of this report. The final recorded state is at epoch 0.75 and step 6,000, while the configuration specifies 8,000 maximum steps. The values are not diagnostic accuracy, clinical agreement or evidence of a completed full training schedule. They are included to demonstrate provenance-aware reporting of a concrete development artifact.

A medical adaptation study would require its own licensed dataset, annotation protocol, training configuration, artifact hashes and independent evaluation. Preference optimization would additionally require traceable preference pairs and a documented collection procedure. Such a study is part of the proposed evaluation programme, not a completed result presented here.

7. Reproducible evaluation protocol

7.1 Questions and system comparisons

A first study should test whether the observation interface preserves task-relevant evidence and whether its errors can be localized. Four comparisons would be informative: a direct multimodal answer; the same model asked for findings and interpretation in one generation; an interpretation model given an automatically generated description; and the same interpretation model given a reference description. A reviewer-corrected description is a useful additional condition when human workload is measured.

These comparisons separate the cost of a textual bottleneck from the quality of the observation model. A strong interpretation model cannot recover visual evidence omitted from its input. Conversely, a poor result from generated descriptions should not be attributed solely to interpretation if reference descriptions perform differently. Model size, prompting, context and data access should be held consistent wherever the comparison permits.

7.2 Data and annotation

The study protocol should name the target modality, task, population and reference standard before data collection. Training, validation and test partitions should be separated at the patient level and, where possible, evaluated across an external institution or acquisition setting. Duplicate images and related examinations must not cross partitions. Dataset identifiers, permitted uses and exclusion rules should accompany a manifest that can be inspected without exposing patient identities.

An annotation rubric should cover required findings, unsupported findings, clinically important omissions, uncertainty and image quality. Independent reviewers should be blinded to model condition when practicable. Disagreements should be retained as data, with an adjudication procedure reported separately. A language-similarity score can describe wording overlap but cannot replace assessment of the correctness and clinical significance of findings.

7.3 Endpoints and analysis

The primary descriptive endpoint should be prespecified, for example the rate of clinically important unsupported or omitted findings under a defined annotation rubric. Secondary endpoints can include completeness, uncertainty handling, downstream task discrimination and calibration, reviewer editing time and final-report acceptance. Abstentions and invalid outputs must be counted, with both conditional performance and overall coverage reported.

For classification tasks, report class prevalence, decision thresholds, sensitivity, specificity and uncertainty intervals alongside discrimination metrics. Use patient-level resampling when examinations are correlated within patients. Comparisons should use matched cases where possible. Report subgroup definitions and sample sizes before interpreting apparent performance differences. Determine sample size from the precision or detectable difference required by the primary endpoint; this report supplies no invented cohort size or statistical significance claim.

7.4 Operational and information-flow tests

Measure loading time, generation time, peak memory, output length and failure rates on named hardware with pinned software and model revisions. Report cold and warm execution separately, include unsuccessful cases, and state batch size and decoding policy. A single machine's measurements should not be generalized to a hospital deployment without testing the relevant workload.

For a separated implementation, verify that the interpretation process cannot access image files or image-bearing messages. Inspect permitted network traffic, logging, crash reports and retained caches. These tests can demonstrate a particular implementation boundary. They cannot establish that generated text is anonymous or that an entire deployment satisfies every legal or organizational requirement.

8. Limitations and intended interpretation

The principal limitation is the absence of a clinical evaluation in this report. Neither the prototype's format nor its foundation model establishes diagnostic reliability. The textual representation can omit spatial relationships, introduce unsupported findings and carry sensitive information. A downstream model can repeat or amplify those errors. Long, fluent rationales can also be inconsistent with the evidence used to produce an answer; their presence should not be treated as a validated explanation of model behavior.

The present implementation does not enforce its proposed process boundaries and can continue after some configuration failures. Those behaviors must be addressed before a controlled comparison. Public access to the underlying APOLO implementation and a pinned execution environment is also incomplete, limiting independent reproduction of the prototype as a package.

The report therefore offers an architecture and an evaluation contract. Its value is that each claim can be tested at the appropriate layer: generation, observation fidelity, interpretation, human review and information flow. Clinical utility remains an empirical question.

9. Conclusion

APOLO organizes medical-image research around an inspectable distinction between visual observations and their interpretation. The current DeepSeek-VL2-Tiny prototype expresses that distinction within one generated response. The proposed next architecture makes it an explicit software interface with versioned records, failure handling and human review. Separating those implementation states, attributing the upstream training example correctly and defining measurable endpoints provides a firmer basis for further work than an undifferentiated claim of model performance. A reproducible clinical study is the next research contribution to establish.

Declarations

Author contributions. Sami Halawa Ribas: conceptualization, software prototyping, architectural specification, investigation, and manuscript preparation. Upstream foundation models, adaptation methods and the COCO training example are credited to their respective authors.

Funding and interests. This report was prepared independently. The author develops APOLO-related software and has an interest in its further development. No clinical sponsorship or external grant is claimed for the work reported here.

Ethics. This report describes software and a proposed evaluation protocol. It reports no newly conducted human-participant study, clinical trial, expert-panel experiment or institutional ethics approval. Future studies require the permissions and review appropriate to their data and design.

Data and code availability. No new clinical dataset, diagnostic predictions or medically fine-tuned checkpoint is released with this report. The upstream foundation-model and training-example sources are public [1,5]. The APOLO implementation described here has not been released as a complete public reproducibility package. The public [APOLO project model card](#) is descriptive project material; at the time of this report it does not distribute model weights. The observation contract and evaluation protocol in this report are available for independent implementation.

AI assistance. An AI assistant helped restructure and edit the manuscript, inspect implementation material and check bibliographic references. The text does not attribute authorship to the assistant or present generated numerical values as experiments.

References

1. Wu, Z., Chen, X., Pan, Z., et al. (2024). *DeepSeek-VL2: Mixture-of-Experts Vision-Language Models for Advanced Multimodal Understanding*. arXiv:2412.10302. <https://doi.org/10.48550/arXiv.2412.10302>

2. Li, J., Li, D., Savarese, S., and Hoi, S. (2023). *BLIP-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models*. arXiv:2301.12597. <https://doi.org/10.48550/arXiv.2301.12597>
3. Hu, E. J., Shen, Y., Wallis, P., et al. (2021). *LoRA: Low-Rank Adaptation of Large Language Models*. arXiv:2106.09685. <https://doi.org/10.48550/arXiv.2106.09685>
4. Lin, T.-Y., Maire, M., Belongie, S., et al. (2014; revised 2015). *Microsoft COCO: Common Objects in Context*. arXiv:1405.0312. <https://doi.org/10.48550/arXiv.1405.0312>
5. wanyuhan7 / tjy (2025). *DeepSeek-VL2-LORA-Finetune*: public software and COCO-captioning training example. Repository: <https://github.com/wanyuhan7/DeepSeek-VL2-LORA-Finetune>. Trainer-state source: https://github.com/wanyuhan7/DeepSeek-VL2-LORA-Finetune/blob/main/checkpoint-6000/trainer_state.json. Accessed 30 October 2026.