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.

Schema-Guided Tree-Structured Reasoning for Whole-Slide Pathology Workflows

Jihwan Kim[1,2,*], Ji-Hoon Jeong[1,3,4,*], Yoon-La Choi[1,2,5,6,*]

[1] Medical Imaging Insight Institute, Samsung Advanced Institute for Health Sciences & Technology, Sungkyunkwan University, Seoul, Korea
[2] Laboratory of Molecular and Digital Pathology for Theranostics, Samsung Medical Center, Seoul, Korea
[3] Department of Medical Device Management and Research, Samsung Advanced Institute for Health Sciences & Technology (SAIHST), Sungkyunkwan University, Seoul, Republic of Korea
[4] Research Institute for Future Medicine, Samsung Medical Center, Seoul, Republic of Korea
[5] Department of Pathology and Translational Genomics, Samsung Medical Center, Sungkyunkwan University School of Medicine, Seoul, Korea
[6] Department of Health Sciences and Technology, Samsung Advanced Institute for Health Sciences & Technology, Sungkyunkwan University, Seoul, Korea

[*] Corresponding authors.
J.H. Kim (kuchoco97@gmail.com), J.-H. Jeong (jihoon.jeong@skku.edu), Y.-L. Choi (ylachoi@skku.edu)

Abstract. We present a multimodal large language model that reproduces a pathologist's diagnostic workflow directly from a whole-slide image (WSI). Instead of emitting a single label or a free-form report, the model traverses a tree of clinical question-answer steps that terminates in a pathology report. An attention-based multiple-instance-learning (ABMIL) slide encoder is coupled to a quantized medical LLM through a lightweight visual projector. The central design choice is that the model learns only to answer, while the follow-up questions are generated deterministically by an engine that consults an organ- and procedure-specific decision schema derived automatically from the training annotations; thus, branching logic is never learned, and training mirrors inference. We pre-trained the slide encoder to predict the organ, procedure, abnormality, and severity, and co-trained slide-level reasoning with region-of-interest grounding in a single run. On two held-out validation sets, the system reaches a Workflow Ranking Score (WRS) of 0.846 on a random split and 0.796 on a harder stratified split. Removing the schema costs 0.130 WRS at the same backbone scale, which we attribute to the 4-8B model being unable to learn question generation and answering simultaneously without overfitting the former. This makes the schema a way of spending a constrained capacity budget, rather than claiming that structure should never be learned.

Keywords: Computational Pathology, Workflow Reasoning, Large Language Model, Multi-Modal

1 Introduction

Diagnostic pathology is a process rather than a single classification. A pathologist first identifies the organ and the specimen type, asks a cascade of increasingly specific questions, and finally dictates a report. Faithfully modeling this workflow from a whole-slide image (WSI) requires aggregating gigapixel imagery into a tractable representation and constraining the reasoning so that it follows a clinically valid path. We address these problems with a single system. ABMIL (Attention-Based Multiple Instance Learning) [1] encoder condenses tens of thousands of patch embeddings into a small set of high-salience, spatially diverse tokens; these tokens are projected into the embedding space of a quantized medical LLM. Crucially, we do not ask the LLM to invent the next diagnostic question. Instead, we externalize the branching logic into an explicit, organ/procedure-conditioned schema and let a deterministic engine expand the tree, so the network is only ever responsible for answering the current question from the pixels. This keeps the trainable behavior tightly aligned with the evaluation protocol and removes an entire class of hallucinated reasoning paths.

Code availability. The training, inference and evaluation code, the dockerfile, the decision schema and its derivation script, and the hyperparameters of **Table 1** are available at https://github.com/cuchoco/reg2026_submission. The repository also contains the containerized challenge entry point.

2 Methods

Fig. 1 gives an overview of the pipeline, from tissue segmentation and patch encoding through the two training stages to schema-driven inference.

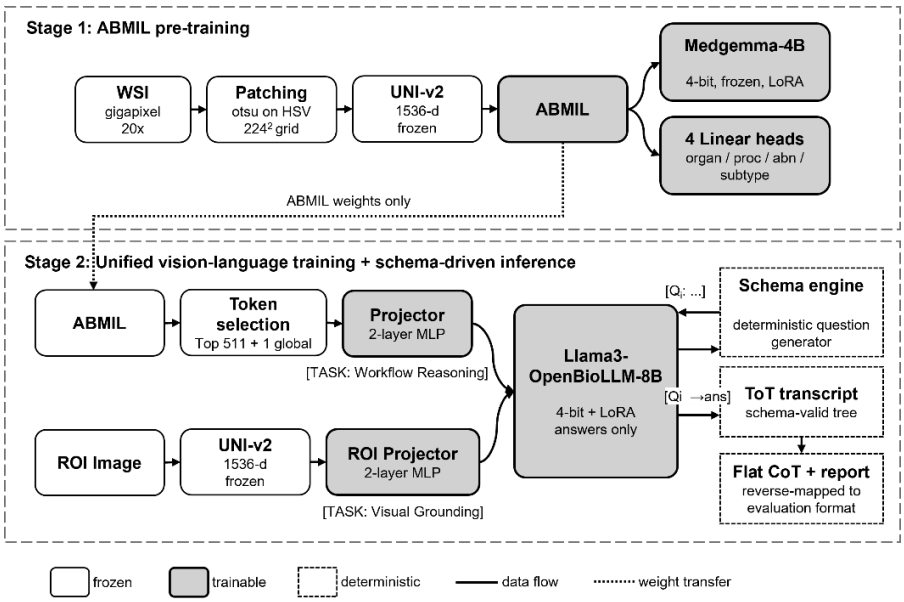

Fig. 1. Overview of the slide-to-report system. Stage 1 pre-trains the ABMIL slide encoder against a MedGemma-1.5-4B and four auxiliary heads, weighted equally; MedGemma supplies a training signal only and is then discarded, so only the ABMIL weights enter Stage 2. Stage 2 aligns that encoder with Llama3-OpenBioLLM-8B and interleaves slide-level reasoning with region-of-interest grounding. A deterministic schema engine emits every next question and the LLM produces only answers, so the same traversal runs in training, where answers are teacher-forced, and at inference, where the model supplies them.

2.1 Tissue Segmentation and Patching

Tissue is segmented at low resolution: we decimate the base level to a thumbnail, convert to HSV, median-blur the saturation channel (kernel 7), and apply Otsu thresholding followed by morphological closing (5x5, two iterations), with a degenerate path for zero-saturation slides that would break Otsu. A non-overlapping grid of 224x224 patches at 20x is generated in level-0 coordinates, and a patch is kept only if its projection onto the tissue mask contains foreground.

2.2 Feature Extraction

For every retained coordinate we crop the level-0 region, resize to 224×224, and encode it with a frozen pathology foundation model. We use UNI-v2 [2] exposed through the Trident [3] python package, producing a 1536-dimensional embedding per patch. Patches are streamed through a data loader with the encoder's native evaluation transforms and run under autocast at the encoder's precision. The per-slide tensor of patch features and their coordinates is saved as a single .pt file.

2.3 Slide Encoder and Generative MIL Pre-training

The slide encoder is a gated ABMIL network. Patch features (1536-d) pass through a pre-attention projection, an 8-head gated-attention pooler and a final feature projector to a 512-d pooled slide embedding. The attention weights additionally serve as a saliency map for token selection. Before language alignment we pre-train the slide encoder generatively. The pooled slide embedding is projected into a frozen, 4-bit MedGemma-1.5-4B [4] LLM, which is asked to produce the final pathology report for the slide; the language loss is applied to answer tokens only. In the same forward pass, four lightweight linear heads read the 512-d pooled feature and predict organ (9 classes), procedure (21 classes), abnormality present (2 classes) and an ordinal severity subtype (4 classes: non-neoplastic / benign / in-situ–dysplasia / invasive-malignant); missing labels are ignored per sample. The slide encoder is unfrozen and tuned (the LLM stays frozen with a LoRA [5] adapter), so the encoder learns representations that are simultaneously report-generative and diagnostically discriminative. We train with AdamW (lr 2e-4, cosine schedule, 10 epochs, batch 2 × 4 accumulation), capping each slide at 8000 patches by uniform subsampling. Only the slide-encoder weights are exported to

the next stage. Explicitly, the objective is $L = L_{\mathrm{LM}} + \frac{1}{4}\sum_k L_{\mathrm{CE}}^{(k)}$, where L_{LM} is the report language-modeling loss and the four cross-entropy terms cover organ, procedure, abnormality and severity. The four auxiliary terms are equally weighted by construction, and no relative weight between the language and the classification terms was tuned.

2.4 Unified Vision–Language Model

The main model aligns the pre-trained slide encoder with LLM (Llama3-OpenBioLLM-8B [6] , MedGemma-1.5-4B [4]). The LLM is loaded in 4-bit and adapted with LoRA (rank 16, α 32, dropout 0.05) [7] on all attention and MLP projections; token embeddings are resized for the visual soft token and frozen. The frozen ABMIL encoder scores every patch by its mean attention. From these we select top-k = 511 patches using attention-guided spatial non-maximum suppression: patches are visited in descending attention order and any patch within 448 level-0 pixels (two patch strides) distance of an already-selected patch is suppressed, yielding a compact set of salient yet spatially non-redundant tokens. The selected patches are re-ordered into a raster (y-then-x) sequence, and a learnable 2-D coordinate encoding (an MLP mapping normalized $(x,y)\in[0,1]^2$ to the feature dimension, min–max normalized over the whole slide) is added so the LLM perceives slide geometry. A single aggregated slide token (the ABMIL pooled embedding) is appended as a global summary. The resulting 1536-d visual sequence is mapped by a two-layer GELU MLP projector to the LLM hidden size and substituted in place of the <image> or <image_soft_token> placeholder [8] in the prompt; the corresponding label positions are masked. The two language models play distinct roles: the MedGemma-1.5-4B of Sec. 2.3 stays frozen, supplies a report-generation gradient to the slide encoder during Stage 1 and is then discarded, whereas the backbone described here is the one trained and deployed.

2.5 Tree-of-Thought and Schema-Constrained Training

Each training slide is annotated with a flat chain-of-thought [9]: an ordered list of (question, answer, next_question) steps that together form a directed reasoning graph. We convert this into a tree-of-thought [10] dialogue. Repeated and noisy steps are cleaned and merged, edges are reconnected to the nearest consistent successor to preserve locality, and every question is assigned a dynamic integer ID. Each dialogue turn presents the currently answerable questions to the model as [Qi: question text] and expects terse answers of the form [Qi -> answer]; binary questions are normalized to yes/no. Converging questions are deferred until all of their parent branches have been resolved, so the tree closes correctly. The model trained in this section is the unified vision-language model of Sec. 2.4; it is not a separate network. Each turn conditions on both modalities at once: the projected slide token sequence, which stays fixed for the whole dialogue, and the questions the schema engine has rendered for that turn. The loss is applied to the answer tokens only, so the questions are context, never a prediction target.

The key design choice is that the model is trained to produce only the answers. The questions that appear in the next turn are not learned; they are produced

deterministically by a Python engine that consults an explicit decision schema. Given the current (organ, procedure), the just-answered question, its answer, and the answer history, the engine returns the set of allowed follow-up questions and prunes those already asked. Training data is therefore produced by a teacher-forced schema traversal that mirrors the inference loop step for step: we walk the schema, fill each answer from the ground-truth answer pool, and expand children exactly as inference would. A faithfulness gate then reverse-maps the generated tree back to flat edges and accepts this schema-forced construction only if the reconstructed edge set equals the ground-truth edge set; otherwise the dialogue is built by traversing the ground-truth tree instead, so no annotated case is dropped. This aligns the reasoning structure seen in training with the one produced at inference, and confines learning to the genuinely visual sub-problem of answering each question from the slide.

Schema derivation. The schema is obtained deterministically from the training annotations. Because every annotated step is a (question, answer, next question) triple, the union of all steps for a given organ and procedure already defines a directed reasoning graph. We aggregate these triples and emit, for each question, either a single list of successors when every observed answer leads to the same ones, or an answer-conditioned mapping otherwise; binary answers are collapsed to yes/no. This yields 27 organ/procedure schemas covering 101 unique questions and 480 branches; a slide activates 15.8 questions on average, and the 25-turn inference cap is never reached in practice. Supplementary **Table S1** reports the full schema statistics. The derivation script is released with the code, so the schema can be regenerated for a newly annotated corpus without manual authoring.

2.6 Visual-Grounding Co-Training

To give the LLM a consistent visual grounding at the patch/ROI scale, we merge a region-of-interest visual-grounding (VG) corpus into the same training run. Each VG sample is a labelled ROI image with a question/answer pair; the ROI is encoded on the fly by the frozen UNI-v2 patch encoder and projected by a dedicated ROI projector into the same LLM space via the same <image> mechanism. WSI and ROI samples are interleaved and prefixed with an explicit task tag ([TASK: Workflow Reasoning] vs. [TASK: Visual Grounding]). ROI samples are undersampled to an 8:2 WSI:ROI ratio so that slide-level reasoning remains the primary objective while the model still learns to read local morphology. Training uses TRL's SFTTrainer with a custom collator (batch 2 × 8 accumulation, 15 epochs, lr 1e-4, warmup 0.02, weight decay 0.01, max sequence length 3072, paged 8-bit AdamW); only projectors, the coordinate MLP, and LoRA adapters are trainable. Each visual-grounding sample presents the model with both modalities at once: the projected ROI token sequence substituted for the image placeholder, together with the accompanying question text. As for slide-level turns, the loss is applied to the answer tokens only.

2.7 Schema-Driven Inference

At test time we reproduce the training traversal, but the answers now come from the model. Preprocessing (Sec. 2.1) and UNI-v2 feature extraction (Sec. 2.2) are run on the query slide, and inference starts from the two root questions organ and procedure. In each turn the engine gathers all questions whose schema parents are already answered, renders them as [Qi: ...], and the model answers greedily. Answers update the (organ, procedure) context and are fed to the schema engine, which emits the next layer of questions; converging nodes are held back until their branches complete. The loop runs until the final pathology report is produced (capped at 25 turns). Finally, a reverse mapping deterministically convertsthe ToT transcript back into the flat format required by the evaluation platform, mapping normalized yes/no answers back to their canonical clinical phrasings. Because the same schema and the same tree bookkeeping drive both training and inference, the model is queried in the regime it was optimized for. The final report is generated freely by the model rather than assembled from the schema answers, so it may state findings that no schema question explicitly queried; the schema constrains the reasoning trace, not the content of the report.

3 Experiments and Results

3.1 Datasets and Metrics

The full annotated corpus of 11,220 WSIs [11] is first partitioned 9:1 into training and validation folds at the slide level; all model training and selection use the 90% training fold, and every slide in the two evaluation sets is held out from training. Both 100-slide evaluation sets are sampled from the same 10% validation fold, so they share the identical data distribution and differ only in how the 100 slides are drawn. Val-Random is a uniform random draw of 100 slides from the validation fold and reflects the natural (organ- and diagnosis-imbalanced) test distribution. Val-Stratified instead draws 100 slides with stratification across organs, procedures and diagnostic categories: by upweighting rare classes it deliberately over-represents the long-tail branches of the schema and is therefore substantially harder.

Performance follows the challenge protocol. Workflow reasoning is scored by a Workflow Ranking Score, a weighted combination of four sub-metrics: Binary Path Validity (5%), which checks that binary routing follows a valid path; Edge F1 (30%), the F1 over the reasoning-tree edges against the reference; MESS (25%), the correctness of non-final intermediate answers; and Final Report Score (40%), the quality of the terminal pathology report. The overall challenge score linearly combines workflow (70%) and a visual-grounding component (30%); the two validation sets contain workflow cases only, so the visual component is not exercised here, and we report the Workflow Ranking Score and its sub-metrics.

3.2 Implementation Details

Table 1 summarizes the architecture and the two training stages. All models are implemented in PyTorch with HuggingFace Transformers/PEFT/TRL and the Trident foundation-model wrappers. Quantization uses bitsandbytes NF4 with double quantization and a bf16 compute dtype; flash-attention-2 is enabled when present. Our final submitted system uses a Llama-family instruction-tuned backbone (Llama3-OpenBioLLM-8B); we additionally train an otherwise identical MedGemma variant for comparison. The two settings reported here differ only in the backbone and in whether the schema is active: Test Phase 1 used MedGemma-1.5-4B with free-form generation of both questions and answers, whereas the final system uses Llama3-OpenBioLLM-8B with schema-constrained, answer-only supervision. All other components are identical.

3.3 Results on the Validation Splits

Table 2 reports the WRS and its sub-metrics on both validation sets. The schema-guided model attains the best WRS on both and degrades gracefully under the stratified shift, keeping the reasoning path valid on long-tail cases. With the structure fixed by the schema the backbone is secondary, the two variants differing by 0.004 WRS on Val-Random, so we adopt Llama for submission. Removing the schema and generating the tree freely costs 0.130 WRS under the same backbone, with the largest drops in Binary Path Validity and Final Report Score: schema-constrained, answer-only supervision improves not just structural validity but downstream report quality**.** Supplementary **Tables S2** and **S3** give a representative successful and a representative failed case, each with the complete reasoning path, every intermediate answer and the final report. The failure mode is characteristic of the design: a single incorrect early binary answer commits the schema engine to the wrong branch, so the trace stays structurally valid while becoming diagnostically wrong, and Edge F1 and MESS therefore degrade together rather than independently.

Table 1. Configuration of the slide-to-report system.

Component	Setting
Magnification / patch	20×, 224×224, non-overlapping
Tissue segmentation	Otsu on HSV saturation + morphological closing; integral-image filtering
Patch encoder	UNI-v2 (ViT-H), 1536-d, frozen
Slide encoder	ABMIL, 8 heads, head-dim 256, dropout 0.1
Visual Token selection	top-k = 511, 1 aggregated token
LLM	Llama3-OpenBioLLM-8B & MedGemma-1.5-4B-it (Gemma-3), QLoRA (r 16, α 32, dropout 0.05)
Stage 1 (generative MIL)	Report LM + 4 aux heads (organ 9 / proc 21 / abn 2 / subtype 4); AdamW 2e-4, 10 epoch, bf16

Stage 2 (unified SFT)	Answer-only loss; AdamW 1e-4, 15 epoch, bs 2×8, seq 3072; 8:2 WSI:ROI
Reasoning control	External schema; questions generated, answers learned
Inference	Greedy, max 25 turns, schema-driven; ToT → flat CoT reverse mapping

Table 2. Workflow reasoning results on the two validation sets. Proposed is the schema-guided model with a Llama-family backbone (our submission); MedGemma is the same pipeline with a MedGemma backbone; No-schema (free) removes the schema and lets the model generate the full tree. Higher is better; best per column in bold.

Dataset	Method	Ranking	Path Val. (5%)	Edge F1 (30%)	MESS (25%)	Report (40%)
Val-Random	Llama (schema)	**0.846**	0.600	**0.912**	**0.868**	**0.814**
Val-Random	Medgemma (schema)	0.842	**0.640**	0.908	0.861	0.805
Val-Random	Medgemma (free)	0.712	0.520	0.797	0.715	0.669
Val-Stratified	Llama (schema)	**0.796**	**0.630**	**0.882**	**0.816**	**0.740**
Val-Stratified	Medgemma (schema)	0.760	0.600	0.849	0.771	0.705
Val-Stratified	Medgemma (free)	0.661	0.500	0.766	0.678	0.593

3.4 Results on Official Test Phase

Table 3 reports the performance of the models on the official Test Phase data, evaluated through the challenge platform under the containerized submission setting. The Test Phase 1 submission was made before our final configuration was selected and therefore evaluates the no-schema MedGemma variant; the schema-guided Llama system reported in **Table 2** was submitted to Test Phase 2.

Table 3. Results of the submitted model on the official Test Phase data

Dataset	N	Method	Ranking	Path Val. (5%)	Edge F1 (30%)	MESS (25%)	Report (40%)
Test Phase-1	350	Medgemma (free)	0.660	0.349	0.807	0.740	0.538
Test Phase-2	70	Llama (schema)	0.569	0.329	0.717	0.636	0.445

4 Discussion

Four limitations should temper these results. First, the decision schema is derived programmatically from the chain-of-thought annotations of the training corpus, so its coverage is inherited from that corpus rather than authored by an expert. Extending the system to a new organ system or a new institution's reporting conventions therefore does not require hand-authoring a schema, but it does require a comparably annotated corpus from which one can be re-derived. For an organ/procedure pair absent from the schema, inference degrades gracefully, first to any schema sharing the same organ and then to a corpus-wide global schema, rather than failing, although such traversals are no longer specialized to that specimen type. We therefore trade coverage for reliability and do not claim zero-shot transfer. Second, the schema is a hard constraint. When the teacher-forced schema traversal fails to reproduce the reference edge set exactly, the sample is not discarded: the builder falls back to a traversal of the ground-truth tree, so every annotated case contributes to training. The cost is instead a residual train/inference mismatch, since these cases are supervised in ground-truth order while inference always follows the schema; this affects 8.4% of training samples and is concentrated in slides carrying multiple independent diagnoses (20.1%, versus 2.5% for single-instance slides), predominantly breast (28.2%). Third, our central ablation is run at a single backbone scale. We attribute the failure of free-form generation to the limited capacity of a 4-8B model asked to learn question generation and answering simultaneously, but establishing that this is a capacity effect rather than a property of the task would require a scale sweep, or a separately parameterized questioner, which our compute budget did not permit. The conclusion 'the schema helps' should therefore be read as scoped to this resource regime. Fourth, each slide is annotated with a single reasoning path, so clinically equivalent alternative orderings are scored as error rather than as variation.

For future work, the schema is best read as a workaround for a capacity constraint rather than an end state. Instead of scaling to a larger monolith, we envision decomposing the workflow into three cooperating agents with separate parameters: a questioner proposing the next diagnostic question, an answerer resolving it, and a vision agent re-examining the slide each turn at a chosen region and magnification, which removes the fixed 512-token input selected in advance. The schema would not disappear but be demoted from generator to validity oracle that accepts, rejects or repairs each proposal, preserving the clinical-validity guarantee while giving the questioner training signal that answer-only supervision cannot provide.

5 Conclusion

We present a reproducible slide-to-report system that models diagnostic pathology as a tree-of-thought reasoning process over WSIs. Decoupling question generation from question answering proved critical at a modest model scale: removing the schema degraded the WRS by 0.130 and the Final Report Score by 0.136, consistent with the question-generation objective competing for capacity with answering, whereas swapping the backbone under a fixed schema changed the WRS by only 0.004. The schema

is derived automatically from the training annotations rather than hand-authored, but its coverage stays bounded by that corpus, which caps the reasoning paths the system can represent. This framework serves as a reproducible benchmark for evaluating agentic decomposition in computational pathology.

Declarations

Ethical approval and informed consent. This work did not involve any new collection of human data or biological material by the authors. All experiments were performed on the publicly available, de-identified whole-slide image corpus released for the REG2026 challenge [11] under a CC BY 4.0 license; ethical oversight and participant consent for the original data collection were the responsibility of the data providers. All procedures were consistent with the principles of the Declaration of Helsinki. No identifiable patient information is presented in this article.

Funding. The authors declare that no funds, grants, or other support were received during the preparation of this manuscript.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Ilse, M., J. Tomczak, and M. Welling. Attention-based deep multiple instance learning. in International conference on machine learning. 2018. PMLR.
2. Chen, R.J., et al., Towards a general-purpose foundation model for computational pathology. Nature medicine, 2024. 30(3): p. 850–862.
3. Zhang, A., et al., Accelerating data processing and benchmarking of ai models for pathology. arXiv preprint arXiv:2502.06750, 2025.
4. Sellergren, A., et al., Medgemma 1.5 technical report. arXiv preprint arXiv:2604.05081, 2026.
5. Hu, E.J., et al., Lora: Low-rank adaptation of large language models. Iclr, 2022. 1(2): p. 3.
6. Grattafiori, A., et al., The llama 3 herd of models. arXiv preprint arXiv:2407.21783, 2024.
7. Dettmers, T., et al., Qlora: Efficient finetuning of quantized llms. Advances in neural information processing systems, 2023. 36: p. 10088–10115.
8. Liu, H., et al., Visual instruction tuning. Advances in neural information processing systems, 2023. 36: p. 34892–34916.
9. Wei, J., et al., Chain-of-thought prompting elicits reasoning in large language models. Advances in neural information processing systems, 2022. 35: p. 24824–24837.
10. Yao, S., et al., Tree of thoughts: Deliberate problem solving with large language models. Advances in neural information processing systems, 2023. 36: p. 11809–11822.
11. Song, E., et al., Pathologist Reasoning-Guided Report Generation Challenge. 2026: Zenodo.