Herald: Democratizing Compositional Reasoning for Visual Tasks without Any Training

Guan-Yuan Tan, Arghya Pal, Sailaja Rajanala, Raphaël C.-W. Phan, Chee-Ming Ting
School of Information Technology, Monash University
emails: {*guan.tan, arghya.pal, sailaja.rajanala, raphael.phan, cheeming.ting*}*@monash.edu*

***Abstract*—Premium large language models (LLMs) such as GPT-4 offer impressive multimodal performance, yet their pay-wall limits both accessibility and reproducibility. We ask whether a *coalition* of freely-accessible LLMs—each individually noisy and uncertain—can *collectively* rival or surpass their premium counterparts when treated as collaborative, on-the-fly programmers. We present Herald, a framework that (i) primes a diverse pool of zero-cost API LLMs with chain-of-thought cues, (ii) harvests their responses as human-readable Python fragments, control-flow branches, and live API calls, and (iii) employs a rank-and-fuse module to assemble the best fragments into a single executable script. The resulting program is executed by an *Executor* that produces the task output *and* a fully inspectable reasoning trace. Without any additional training, Herald tackles heterogeneous vision workloads—image editing, semantic tagging, and medical triage—and achieves state-of-the-art or better accuracy on both medical and non-medical benchmarks. By transforming latent model competence into legible artefacts, Herald enables a transparent interaction style that invites user scrutiny, iterative refinement, and accountable auditing. All code and reproducible workflows are released at https://github.com/tgy1221/Herald, offering an open, resource-efficient alternative to premium LLM services.**

I. Introduction

Large language models (LLMs) such as GPT–4 have captured the imagination of the HCI community because they appear to "just work" across modalities and domains. However, the reality of subscription fees, rate limits, and opaque update cycles places these premium models out of reach for many researchers, educators, and practitioners—especially in contexts where resources are scarce or data sovereignty is paramount. In effect, pay-walled models amplify the access and reproducibility gaps that HCI has long sought to close. This paper, therefore, asks a pointed question: *Can a coalition of freely accessible LLMs (free API calls), each individually noisy and unreliable, collectively rival or surpass their premium counterparts, while remaining transparent, auditable, and adaptable by end users?*

The question sits squarely at the intersection of interactive machine learning, explainable AI, and open-source tool building, and it invites us to rethink who can appropriate state-of-the-art AI and on what terms. To this end, we introduce Herald, a mixture-of-experts framework that turns a pool of zero-cost API LLMs into collaborative programmers. Each model receives the same high-level task description plus a generic *chain-of-thought* cue, prompting it to emit human-readable Python fragments, control-flow branches, and live API calls. An internal *Ranker* scores these fragments, while a *Fuser* stitches the best pieces into a single executable script. The script is run by an *Executor* that produces both the final task output and an inspectable reasoning trace—a crucial affordance for user trust and iterative refinement (Fig. 1).

Our approach builds on the emerging literature in *zero-shot compositional reasoning*, which seeks to orchestrate external tools without additional training data. Unlike prior pipelines that hinge on a single large expert, Herald treats the diversity of lightweight, open-source LLM "personalities" not as noise but as complementary expertise. With no fine-tuning whatsoever, the system tackles heterogeneous vision workloads—image editing, semantic tagging, and medical triage—and matches or exceeds the performance of subscription-based models.

The contributions of this work are threefold:

1) An *open-source* mixture-of-experts framework that transforms noisy, zero-cost LLMs into *code-generating collaborators*, lowering the financial and technical barriers to advanced AI.
2) A transparent *rank-and-fuse* pipeline that outputs legible, executable artefacts, enabling end users to audit and adapt the system's reasoning process.
3) An empirical evaluation across medical and non-medical benchmarks demonstrating that Herald achieves state-of-the-art accuracy while offering greater interpretability and user control.

By surfacing executable code instead of opaque logits, Herald opens a shared interaction space where users, developers, and auditors can see *why* the system acts as it does, correct errors, and repurpose the pipeline for new tasks. The remainder of the paper details the Herald architecture (Section III), presents quantitative and qualitative evaluations (Section IV), discusses design implications for interactive AI (Section IV-F), and concludes with paths forward for open, controllable, and human-centered LLM ecosystems.

II. Related Work

Compositional AI. Neural Module Networks[1] were proposed to dynamically compose collections of specialized neural modules to improve visual question answering. Furthermore, [2] utilized a standard LSTM sequence-to-sequence model as a program generator to tackle the compositional VQA task. Recently, following the advancement of LLMs, Pan *et al.*[3] proposes a compositional network to solve visual tasks without training. They leverage LLMs' in-context learning

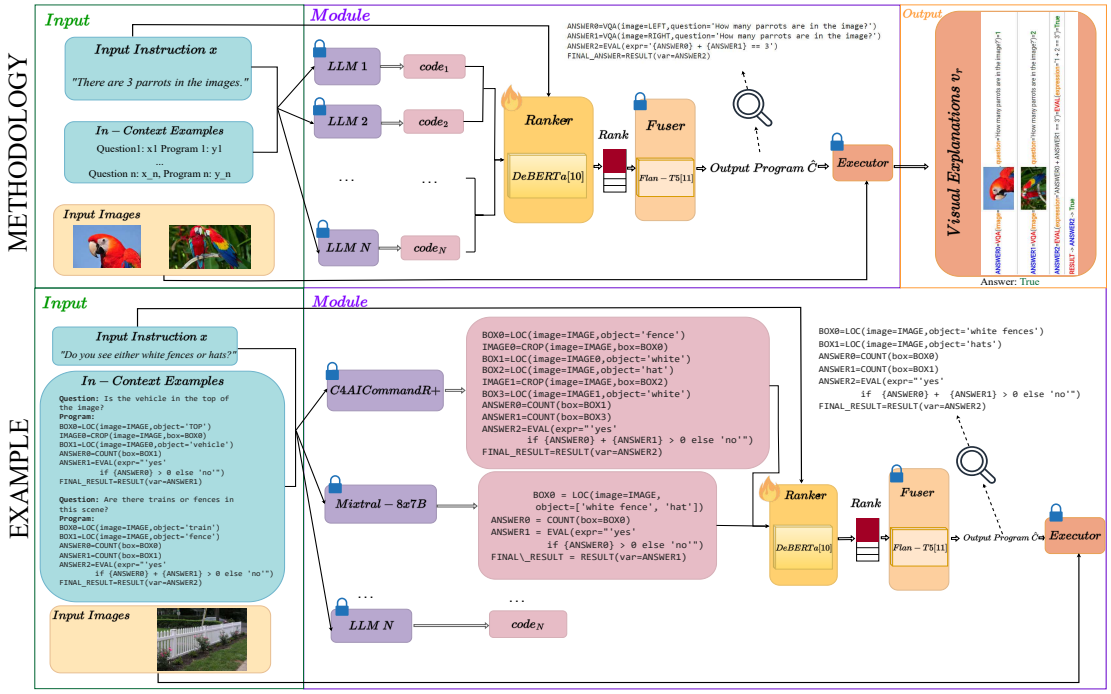

Fig. 1. Herald orchestrates lightweight, open-source LLMs to generate human-interpretable Python programs that solve diverse vision tasks *without additional training*. A Ranker–Fuser module assembles the best fragments, and an Executor delivers both the final output and a transparent reasoning trace.

capability to generate the program code [4] for automated module deployment and selection.

Mixture of Experts. While it is intuitive to use a single open-source LLM to replace closed-source LLMs, [5] shows that the performance of individual LLMs varies, due to different training environments. Thus, [6] has proposed an ensemble framework to tackle the hallucination and instability of LLMs. Similarly, [5] proposed LLM-Blender to rank and fuse the output from various LLMs. For a similar reason, GPT-3.5 or Codex used in [3] may also encounter instability issues. From [7], it is mentioned that an incorrect program is one of the root causes of the inferior results. Thus, it is motivated to perform ensemble learning to obtain a superior result in this current work.

III. Herald: MoE for Compositional Reasoning

In this section, we introduce our proposed framework, Herald, which integrates a *mixture-of-experts (MoE)* approach with *code-based reasoning* for zero-shot compositional visual tasks. The primary objective is to leverage each model's ability to generate executable Python subroutines and high-level API calls, addressing a wide range of tasks without any additional training.

A. Overview of Herald

Let $L = \{L_1, L_2, \ldots, L_N\}$ be a set of N open-source large language models (LLMs). Although these LLMs differ in architecture, inference speed, and hyperparameters, they each aim to solve zero-shot compositional reasoning tasks $\{\tau_1, \tau_2, \ldots, \tau_m\}$. To harness their collective strengths, we allow every LLM L_i to generate human-interpretable API calls, referred to as *code*. Consider a downstream task τ_i requiring human face recognition with bounding boxes(bbox). An LLM might produce a sequence of API calls: i) image_loading() ii) image_reshaping() iii) face_detection() iv) bounding_box() v) knowledge_retrieval(). Each step could call different candidate APIs (e.g., Faster R-CNN vs. YOLO for bbox regression). We show an example in Fig. 1. Formally, given an input instruction x and in-context examples IC describing task τ, each model L_i generates $code_i^{\tau_i} = L_i(x, IC)$, leading to a set of candidate programs $C = \{code_1^{\tau}, \cdots, code_N^{\tau}\}$ generated by $L_i \in L$. Our approach relies on in-context learning and few-shot prompts to guide each LLM toward generating high-level Python calls. Crucially, each generated program remains entirely interpretable: users can audit and trace intermediate steps and logic.

B. Gated Mixture of Experts for Compositional Reasoning

Our MoE design draws inspiration from frameworks like Mixtral 8x7B [8] or DeepSeek, where gating functions select the best combination of experts. In contrast to those methods, which often employ homogeneous LLMs or train MoE modules within a single model, Herald operates over a heterogeneous set of independently trained LLMs. By exploiting each LLM's propensity to generate API-based code, the framework effectively orchestrates compositional reasoning for the target task τ_i.

Simplified Setting and Ranking with DeBERTa-v3 To illustrate our approach, suppose each L_i produces an initial

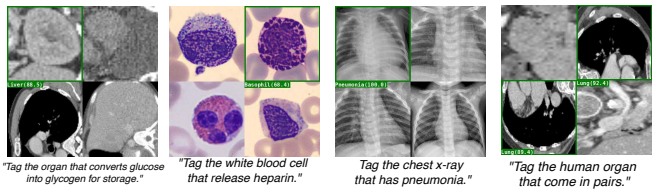

Fig. 2. **Knowledge Tagging (Medical)**. Best viewed while zoomed; each image shows the bounding box provided by Herald along with the tag reporting the detection accuracy.

Fig. 3. **Knowledge Tagging (Non-medical)** We show that the compositional nature of our method is capable of leveraging object or face detectors and LLMs as an external knowledge database to tackle the task with high accuracy without leveraging subscription-based LLMs as needed for [7], [9].

candidate code$_i$. We then aim to select the best candidate from among N possible LLMs. For this purpose, we employ a DeBERTa-v3 [10] model as a *ranker*, f, comparing candidates pairwise and returning a scalar score s_{ij} indicating whether code$_i$ is preferable to code$_j$. We chose DeBERTa-v3 due to its strong performance on natural language understanding tasks, making it well-suited for assessing the semantic alignment between the instruction x and the generated code. For this purpose we take inspiration from [5], the input x is encoded together with two candidate outputs $code_i$ and $code_j$ by concatenating them into a single sequence $[x; code_i; code_j]$. This sequence is then fed into a cross-attention Transformer, and a feature representation $f([x; code_i; code_j])$ is obtained, which is used to model the pairwise score s_{ij}. For each candidate pair, scores $s_i^{(i,j)}$ and $s_j^{(i,j)}$ for $code_i$ and $code_j$, respectively, are computed, and the pairwise difference score is defined as $s_{ij} = s_i^{(i,j)} - s_j^{(i,j)}$, such that a positive value indicates that $code_i$ is of higher quality than $code_j$.

Ranker Training and Label Generation While the overall Herald framework operates zero-shot on the target various visual tasks, the DeBERTa-v3 Ranker itself undergoes separate, offline supervised training. This process equips the Ranker with a general ability to discern the instruction-code pair quality differences, and is applied zero-shot during inference. The supervision signal, the preference labels (z_i, z_j), for the training is derived automatically, without manual annotation specific to the target tasks. To optimize this capability, multiple quality functions Q (e.g., BERTScore, BARTScore) are assumed, and the learning problem is framed as a multi-task classification task. BERTScore and BARTScore assess semantic similarity between texts, providing a nuanced evaluation of the generated output's quality. Specifically, the loss is defined as:

$$L_Q = z_i \log\big(\sigma(s_i^{(i,j)})\big) + (1 - z_i) \log\big(\sigma(s_j^{(i,j)})\big), \quad (1)$$

$$(z_i, z_j) = \begin{cases} (1,0) & \text{if } Q(code_i, x) > Q(code_j, x), \\ (0,1) & \text{if } Q(code_i, x) < Q(code_j, x) \end{cases} \quad (2)$$

Here, the sigmoid function is denoted by $\sigma(\cdot)$, and the final multi-objective loss L is obtained by averaging L_Q over the multiple quality functions. During inference, pairwise comparisons among all candidates in the set L are performed to yield a matrix M with entries $M_{ij} = s_{ij}$. The final candidate ranking is then obtained by aggregating these scores. The best performing MaxLogits method proposed in [5] is used, and the final score for candidate $code_i$ is computed as $s_i = \langle M_{i\cdot}, M_{\cdot i} \rangle$.

C. Beyond Gating: Fusion of Top-K Codes

Selecting only one code$_i$ ignores potentially valuable insights from other high-ranking candidates. To address this, a *Fuser* model that synthesizes a final program $\hat{code}$ by merging the top-K ranked codes. Formally, the Fuser is a generative model (e.g., Flan-T5-XL [11]) that takes as input the instruction x and the top-K codes, $\{\text{code}_1, \ldots, \text{code}_K\}$, producing

$$\hat{code} = \text{Fuser}\big(\langle x \rangle, \langle \text{code}_1 \rangle, \ldots, \langle \text{code}_K \rangle\big). \quad (3)$$

We selected Flan-T5-XL[11] as the Fuser, for its robust instruction-following and text generation capabilities across various tasks, making it well-suited for synthesizing coherent code. This fusion step consolidates diverse approaches, potentially reconciling conflicting function calls or leveraging complementary logic in the top-ranked candidates, thus reducing the risk of missing crucial details in lower-ranked but still viable candidates.

D. Executing the Fused Code

Once the fused program $\hat{code}$ is obtained, we execute it via an *Executor* module. This component parses each line of $\hat{code}$, identifies the corresponding function or model to call (e.g., a Python-based API, a pre-trained neural network), and handles data flow among function calls. The Executor thus chains these subroutines to produce the final output.

Intermediate outputs are stored for debugging and visualization, providing transparency into the pipeline's internal reasoning. Ultimately, the Executor yields a final user-facing result r (e.g., an edited image or a textual answer) along with optional explanatory artifacts v_r.

E. In-Context Learning for Code Generation

A central challenge in zero-shot code generation is bridging the gap between natural language instructions and structured programmatic logic. Herald addresses this via *in-context learning* [12]. By providing each LLM with demonstration examples through few-shot prompts, we encourage it to produce well-structured Python code that integrates symbolic logic and external API calls. Crucially, no end-to-end fine-tuning is performed, allowing flexible adaptation to new tasks.

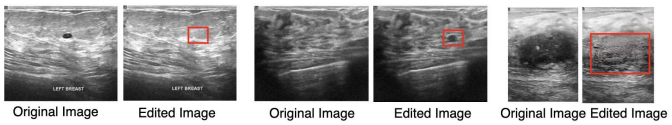

Fig. 4. **Image Editing (Medical).** (*left*) A benign tumor in a breast ultrasound image is edited to resemble normal breast tissue. (*middle*) A normal breast ultrasound image is modified to exhibit characteristics of a benign tumor. (*right*) A malignant tumor is edited to look like normal breast tissue.

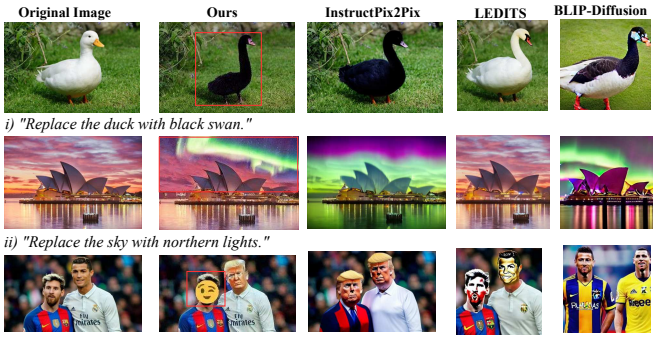

Fig. 5. **Image Editing (Non-medical).** The Herald first column, outperforms all the end-to-end editing frameworks such as Instruct Pix2Pix[13], LEDITS[14], and BLIP-Diffusion[15], i.e., second-last column, under complicated scenes, such as multiple subjects.

IV. Experiments

Herald demonstrates substantial flexibility for complex visual tasks involving natural language prompts and image inputs. We evaluate Herald on four medical and non-medical tasks: i) Knowledge Tagging, ii) Image Editing, iii) Compositional Visual Question Answering, iv) Image Pair Reasoning. A primary objective of our work is to support open-source development and democratize the mixture-of-experts (MoE) paradigm. Consequently, we exclude subscription-based GPT models such as GPT-3.5 (`text-davinci-003`), Codex, GPT-4o-mini and GPT-4 (even though our approach achieves performance on par with them), opting instead for an ensemble of five open-source large language models (LLMs) selected for their diverse reasoning capabilities and public accessibility, namely C4AI Command R+, Llama 3 70B Instruct, Nous-Hermes-2-Mixtral-8x7B-DPO, Yi-1.5, Mixtral-8x7B-Instruct-v0.1. All five models are available on HuggingFace, making them easily adoptable by various research groups.

Additionally, to compare with [7], the self-annotated datasets they used for evaluation are **not publicly available**. Thus, we collect our own collection of datasets for evaluation. We have incorporated an error-handling technique that utilizes regular expressions to remove unnecessary system messages from the generated program.

A. Task 1: Knowledge Tagging

We have considered knowledge tagging on medical and non-medical data. For the medical domain, the task involves tagging images with relevant medical information, such as organ identification or disease understanding and classification.

TABLE I
Image Editing Evaluation Result.

Model	Herald (Ours)	Instruct Pix2Pix[13]
Accuracy(↑)	**64.00**	54.39
Model	LEDITS[14]	BLIP-Diffusion[15]
Accuracy(↑)	17.50	29.83

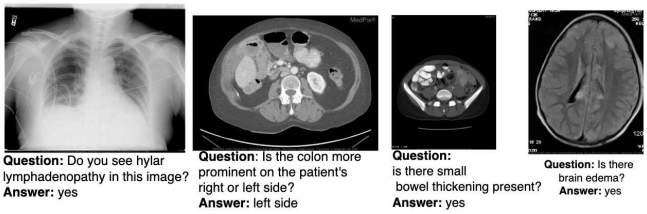

Fig. 6. **VQA (Medical)**. On VQA-RAD [21] Herald accurately answers questions about radiology images, such as identifying hylar lymphadenopathy and the prominent side of the colon, detects small bowel thickening and brain edema, demonstrating its ability to reason about complex medical questions.

We collected images from the MedMNIST dataset [22] for the medical domain. From Fig. 2, Herald successfully identifies the function and characteristics of the organ and white blood cell, and the correct disease from the chest x-ray. For the non-medical domain, there is no standard public benchmark available online. Similar to [7], we have created our own dataset, which includes the images annotated with tagging instructions, bounding boxes, and labels to evaluate the performance of Herald. The dataset is designed so the framework needs external knowledge to answer the questions. We evaluate the performance by computing the F1 Score. Herald obtains a 0.66 F1 score on the testing set, showing promising results qualitatively, as shown in Fig. 3.

B. Task 2: Instruction-based Image Editing

There is no available benchmark for the Image Editing task in both the medical and non-medical domains. Thus, we have gathered a collection of images and annotated them with editing instructions from online. We collected from MedMNIST [22] and the breast ultrasound dataset [27] for medical image editing. For non-medical image editing, we gathered the dataset online. We compare with the most recent state-of-the-art, end-to-end image editing works, namely Instruct Pix2Pix[13], LEDITS[14] and BLIP-Diffusion[15]. For medical image editing, we evaluated Herald on tasks such as converting a benign breast ultrasound image to a normal breast ultrasound image, normal to benign, and malignant to normal as shown in Fig. 4. Herald removed benign lesions while preserving the overall structure of the image, added synthetic lesions to simulate benign conditions, and reconstructed tissue to remove malignant tumors. We also evaluate Herald in the non-medical domain. From Fig. 5, the InstructPix2Pix generates a black creature while creating ambiguity, as swans have far longer necks than ducks. BLIP-Diffusion generates an unrealistic lighting effect on the building, and LEDITS failed to replace and add the subjects. Lastly, all the methods cannot hide the face of the correct target, while Herald can follow the instructions and generate the desired image. From Table I, Herald obtained the highest accuracy.

TABLE II
COMPOSITIONAL IMAGE QUESTION ANSWERING AND IMAGE PAIR REASONING RESULTS.

Model	Pre-trained	#Pre-train Images	Fine-tuned	OK-VQA	GQA	VQA-RAD	NLVRv2	VQA-RAD (Pair)
VILT-NLVR [16]	Yes	4M	Yes	45	23	46	**76**	14
BLIP [17]	Yes	129M	Yes	57	37	46	43	29
ALBEF [18]	Yes	14M	Yes	24	18	39	51	29
PNP-VQA [19]	Yes	129M	Yes	25	28	31	–	–
Herald (Mixtral-8x7B) [20]	No	–	No	22	5	15	43	14
Herald	No	–	No	**74**	**44**	**69**	46	**57**

TABLE III
PERFORMANCE COMPARISON (AUC, ACC) ON VARIOUS MEDICAL IMAGE DATASETS FOR DIFFERENT METHODS. ALL THE NUMBERS ARE TAKEN FROM THE MEDMNIST [22] PAPER, WHILE THE LAST ROW SHOWS HERALD PERFORMANCES.

Methods	PathMNIST		ChestMNIST		DermaMNIST		OCTMNIST		PneumoniaMNIST		RetinaMNIST	
	AUC	ACC	AUC	ACC	AUC	ACC	AUC	ACC	AUC	ACC	AUC	ACC
ResNet-18(28) [23]	0.983	0.907	0.768	0.947	0.917	0.735	0.943	0.743	0.944	0.854	0.717	0.524
ResNet-18 (224) [23]	0.989	0.909	0.773	0.947	0.920	0.754	0.958	0.763	0.956	0.864	0.710	0.493
ResNet-50 (28) [23]	**0.990**	**0.911**	0.769	0.947	0.913	0.735	0.952	0.762	0.948	0.854	0.726	0.528
ResNet-50 (224) [23]	0.989	0.892	0.773	0.948	0.912	0.731	0.958	0.776	0.962	0.884	0.716	0.511
auto-sklearn [24]	0.934	0.716	0.649	0.779	0.902	0.719	0.887	0.601	0.942	0.855	0.690	0.515
AutoKeras [25]	0.959	0.834	0.742	0.937	0.915	0.749	0.955	0.763	0.947	0.878	0.719	0.503
Google AutoML Vision	0.944	0.728	0.778	0.948	0.914	0.768	0.963	0.771	0.991	0.946	0.750	0.531
Herald	0.954	0.751	**0.799**	**0.959**	**0.927**	**0.789**	**0.998**	**0.781**	**0.998**	**0.966**	**0.762**	**0.554**

TABLE IV
ABLATION STUDIES: #IC: #IN-CONTEXT EXAMPLES, M:MIXTRAL-8X7B [20], L:LLAMA-3 [26], M+L: ENSEMBLE OF M&L.

#IC	Compositional Image Question Answering										Image Editing									
	Program Accuracy (%)					Task Accuracy (%)					Program Accuracy (%)					Task Accuracy (%)				
	Ours	GPT-4o	M	L	M+L	**Ours**	GPT-4o	M	L	M+L	**Ours**	GPT-4o	M	L	M+L	**Ours**	GPT-4o	M	L	M+L
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	67	100	0	33	33	67	100	0	33	33	67	67	0	33	33	67	0	0	33	0
10	100	100	0	0	67	100	100	0	0	67	100	33	67	100	0	100	0	33	100	0

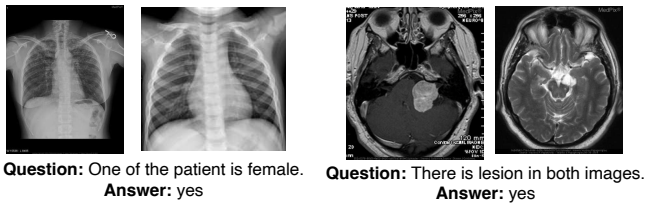

Fig. 7. **Image pair reasoning results in the medical domain.** Herald successfully reasons about pairs of medical images, such as identifying whether a patient's chest X-rays are of a female patient, and detecting lesions in brain scans.

C. Task 3: Compositional Image Question Answering

Compositional Question Answering is a VQA task that requires decomposing the task into multiple steps. We have used OK-VQA [28] and GQA [29] dataset model evaluation in the non-medical domain. For the medical domain, we gathered radiology images from VQA-RAD [21]. We compared Herald with a few end-to-end frameworks, which are capable of the Visual Question Answering task, namely VILT-NLVR[16], BLIP[17], ALBEF[18], and PNP-VQA [19]. From Fig. 6, Herald accurately answers various challenging disease identification problems. For example, it successfully identifies hilar lymphadenopathy, small bowel thickening, and detects evidence of brain edema. These results demonstrate Herald's ability to reason about medical concepts and provide accurate answers, making it a valuable tool for radiology applications. From Table II, we show that Herald can leverage its chain-of-thought reasoning[30] capability to perform zero-shot prediction and achieve the highest accuracy amongst all the baseline frameworks in OK-VQA, GQA, and VQA-RAD datasets. Furthermore, all the baseline frameworks, such as BLIP, ALBEF, and PNP-VQA, require pre-training with large amounts of images and fine-tuning on the VQA datasets, adding up to a much longer time and effort. This poses difficulties for most of the researchers who are in a low-resource environment.

D. Task 4: Image Pair Reasoning

In this task, we evaluate the performance on the NLVRv2 dataset [31], a benchmark that requires reasoning on a pair of images and justifying whether the statement about the images is accurate. In the medical domain, we gathered the data from VQA-RAD [21]. Similar to [7], we have created a subset of the NLVRv2 and VQA-RAD test dataset by sampling random examples to form the evaluation dataset. We compare our result with BLIP[17], ALBEF[18], and ViLT-NLVR[16]. ViLT-NLVR is a ViLT model that is fine-tuned on the NLVRv2 dataset, which incurs roughly an extra 96 GPU hours using 64 V100 GPUs. For medical image pair reasoning, Herald is tasked with determining similarities or differences between diagnostic images, and changes between scans (e.g., tumor growth or reduction). From Table II, while ViLT-NLVR achieves the highest accuracy on the NLVRv2 dataset, Herald showed a solid zero-shot performance on the NLVRv2 and VQA-RAD datasets.

E. Zero-Shot Medical Image Classification

We evaluated Herald's zero-shot medical image classification capabilities using the MedMNIST v2 dataset [22]. Table III demonstrates this comparative analysis. The results clearly

demonstrate Herald's strength. Without any task-specific training, Herald achieves state-of-the-art or highly competitive Area Under the Curve (AUC) and Accuracy (ACC) scores on the majority (five out of six) of the diverse medical datasets.

F. Ablation studies

In Image Question Answering and Image Pair Reasoning Task, we picked Mixtral-8x7b[8], a Mixture of Experts(MoE) framework for comparison. From the last two rows of Table II, Herald obtains a higher accuracy in both tasks. This is due to the incorrect program generated by MoE, which harms the performance of various visual tasks. From Table IV, in each model, following the increment of in-context examples, the performance of each model is improved for both the program and task accuracy.

V. Conclusions

In this paper, we propose a cost-effective compositional visual reasoning framework, Herald that bridges the gap of utilizing LLMs to guide visual tasks and shows capabilities beyond what an individual model can do. Herald can generate interpretable visual explanations, looking into the intermediate output of each step, providing flexibility to inspect the rationality of the program and, in turn, increasing the interpretability and the reliability of the final result.

References

[1] Andreas, et al., "Neural module networks," in *Proceedings of the IEEE/CVF CVPR*, 2016, pp. 39–48.

[2] Johnson et al., "Inferring and executing programs for visual reasoning," in *CVPR*, 2017, pp. 2989–2998.

[3] P. e. a. Lu, "Chameleon: Plug-and-play compositional reasoning with large language models," *NeurIPS*, vol. 36, 2024.

[4] Madaan et al., "Language models of code are few-shot commonsense learners," *arXiv preprint arXiv:2210.07128*, 2022.

[5] Jiang Dongfu et al., "Llm-blender: Ensembling large language models with pairwise ranking and generative fusion," *arXiv preprint arXiv:2306.02561*, 2023.

[6] Zhang Chenrui et al., "Prefer: Prompt ensemble learning via feedback-reflect-refine," *AAAI*, vol. 38, no. 17, pp. 19 525–19 532, 2024.

[7] Gupta Tanmay et al., "Visual programming: Compositional visual reasoning without training. 2023 ieee," in *CVPR*, 2022.

[8] Jiang Albert Q. et al., "Mixtral of experts," *arXiv:2401.04088*, 2024.

[9] Gao Zhi et al., "Clova: A closed-loop visual assistant with tool usage and update," in *CVPR*, 2024, pp. 13 258–13 268.

[10] Pengcheng He et al., "Deberta: Decoding-enhanced bert with disentangled attention," in *ICLR*, 2021.

[11] H. W. e. a. Chung, "Scaling instruction-finetuned language models," *Journal of Machine Learning Research*, vol. 25, no. 70, pp. 1–53, 2024.

[12] Brown Tom B et al., "Language models are few-shot learners," in *NeurIPS*, Dec. 2020.

[13] B. et al., "Instructpix2pix: Learning to follow image editing instructions," in *CVPR*, 2023, pp. 18 392–18 402.

[14] L. Tsaban and A. Passos, "Ledits: Real image editing with ddpm inversion and semantic guidance," *arXiv preprint arXiv:2307.00522*, 2023.

[15] Li, Dongxu et al., "Blip-diffusion: Pre-trained subject representation for controllable text-to-image generation and editing," *NeurIPS*, vol. 36, 2024.

[16] Kim, Wonjae et al., "Vilt: Vision-and-language transformer without convolution or region supervision," in *ICML*, PMLR, 2021, pp. 5583–5594.

[17] Li Junnan et al., "Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation," in *ICML*, PMLR, 2022, pp. 12 888–12 900.

[18] Li, Junnan et al., "Align before fuse: Vision and language representation learning with momentum distillation," *NeurIPS*, vol. 34, pp. 9694–9705, 2021.

[19] Anthony Meng Huat et al., "Plug-and-play vqa: Zero-shot vqa by conjoining large pretrained models with zero training," *arXiv preprint arXiv:2210.08773*, 2022.

[20] mistralai, *Mixtral-8x7b*, https : / / huggingface . co / mistralai/Mixtral-8x7B-Instruct-v0.1, 2023.

[21] Lau, Jason J et al., "A dataset of clinically generated visual questions and answers about radiology images," *Scientific data*, vol. 5, no. 1, pp. 1–10, 2018.

[22] Yang, Jiancheng et al., "Medmnist v2-a large-scale lightweight benchmark for 2d and 3d biomedical image classification," *Scientific Data*, vol. 10, no. 1, p. 41, 2023.

[23] He, Kaiming et al., "Deep residual learning for image recognition," in *CVPR*, 2016, pp. 770–778.

[24] M. Feurer and F. Hutter, "Automated machine learning," *Cham: Springer*, pp. 113–134, 2019.

[25] Jin, Haifeng et al., "Auto-keras: An efficient neural architecture search system," in *SIGKDD*, 2019, pp. 1946–1956.

[26] Meta-Llama, *Llama 3*, https://huggingface.co/meta-llama/Meta-Llama-3-70B, 2024.

[27] Al-Dhabyani et al., "Dataset of breast ultrasound images," *Data in brief*, vol. 28, p. 104 863, 2020.

[28] Marino, Kenneth et al., "Ok-vqa: A visual question answering benchmark requiring external knowledge," in *CVPR*, 2019, pp. 3195–3204.

[29] D. A. Hudson and C. D. Manning, "Gqa: A new dataset for real-world visual reasoning and compositional question answering," in *CVPR*, 2019, pp. 6700–6709.

[30] J. Wei, X. Wang, D. Schuurmans, *et al.*, "Chain-of-thought prompting elicits reasoning in large language models," *NeurIPS*, vol. 35, pp. 24 824–24 837, 2022.

[31] Suhr, Alane et al., "A corpus for reasoning about natural language grounded in photographs," *arXiv preprint arXiv:1811.00491*, 2018.