OpenMAIA: a Multimodal Automated Interpretability Agent based on open-source models

Josep Lopez Camuñas[1] **Christy Li**[2] **Tamar Rott Shaham**[2]
Antonio Torralba[2] **Agata Lapedriza**[3,1]

[1]Universitat Oberta de Catalunya, Barcelona, Spain
[2]Massachusetts Institute of Technology, Cambridge, MA, USA
[3]Northeastern University, Boston, MA, USA

Abstract

Understanding how large neural networks represent and transform information still remains a major obstacle to achieving transparent AI systems. Recent works such as *MAIA* (a Multimodal Automated Interpretability Agent) have shown that agent-based systems can iteratively generate and test hypotheses about neuron function without the need for human intervention, which offers a scalable solution for mechanistic interpretability. However, the existing agent-based systems rely on closed-source APIs, limiting reproducibility and access. To address this, we introduce *OpenMAIA*, an open-source implementation of *MAIA* that replaces its closed-source API-based components with open-source models. We experiment with two state-of-the-art multimodal Large Language Models (LLMs) (Gemma-3-27B, Mistral-Small-3.2-24B) as the *OpenMAIA* backbone models, and update the agent's interpretability toolset with open-source models. Following the neuron description evaluation protocol established in the original *MAIA* paper, which uses neurons from different vision backbones and also synthetic neurons, we show that *OpenMAIA*, when using an open-source backbone, achieves performance comparable to the same *OpenMAIA* configuration that employs Claude-Sonnet-4 as its backbone model. In addition, *OpenMAIA* converges more efficiently than its implementation with Claude-Sonnet-4. These results demonstrate that competitive, agent-based interpretability can be achieved with a fully open stack, providing a practical and reproducible foundation for community-driven research.

1 Introduction

Understanding the functional role of individual neurons within deep neural networks remains a central challenge in mechanistic interpretability. While a variety of tools and techniques have been developed to support these analyses, they often rely heavily on manual inspection and expert intuition, limiting scalability and reproducibility. To address these limitations, recent works such as *MAIA* (a Multimodal Automated Interpretability Agent) [1] have explored autonomous agents for interpreting vision systems and neuron-level behaviors. *MAIA* uses large vision-language models and specialized tools to automate interpretability tasks such as feature analysis, failure mode discovery, and model behavior explanation. It conducts experiments similar to those of human interpretability researchers by synthesizing inputs, identifying activating examples, and summarizing findings, to provide systematic, automated insights into how neural networks make decisions. Despite its effectiveness, *MAIA* depends on commercial platforms and closed-source APIs, which restrict transparency, limit customization, and add overhead costs, posing barriers to widespread use.

In this paper, we introduce *OpenMAIA*, an open-source implementation that retains the agentic experimentation loop of *MAIA* while replacing its closed components with state-of-the-art open-

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

source multimodal models. For the backbone model we experiment with Gemma-3-27B and Mistral-Small-3.2-24B [2, 3, 4] and provide updated open-source tools for image generation, editing, and summarization. Grounded entirely in openly available resources, *OpenMAIA* enables scalable, transparent, and community-driven research on mechanistic interpretability. Figure 1 illustrates a typical interpretability trace, from dataset exemplars to iterative hypothesis testing and final natural-language description.

Our evaluation of the neuron description tasks introduced in *MAIA* shows that open backbones achieve competitive explanations compared to Claude-Sonnet-4 [5], while converging more efficiently in fewer turns. These results empirically demonstrate that agent-based interpretability is feasible with a fully open stack, providing a foundation for reproducible and extensible research.

Code availability. *OpenMAIA* extends the publicly released MAIA framework by replacing closed-source APIs with open-weight models and tools. The implementation builds directly on the original MAIA repository, which we have updated and maintained: https://github.com/multimodal-interpretability/maia. This repository hosts all code, configuration files, and evaluation scripts required to reproduce our results and facilitate community contributions.

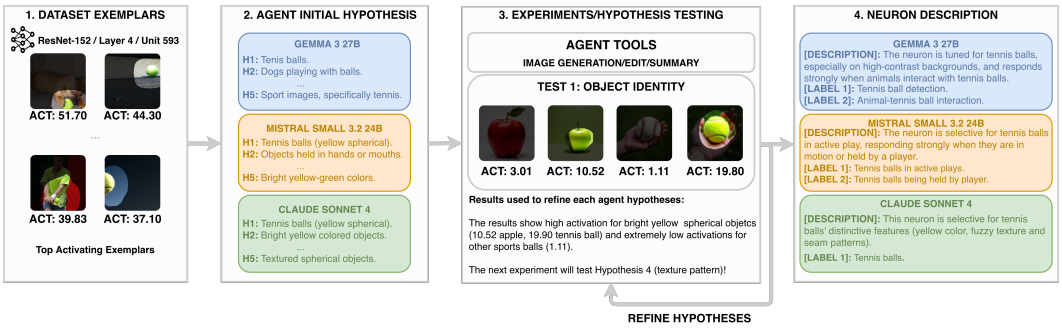


Figure 1: **Simplified experiment for a single neuron across agents.** Illustrative trace for *ResNet-152 (layer 4, unit 593)*: agents start from dataset exemplars, propose hypotheses, and run targeted image generation/editing experiments. Iterative refinement reveals selective features (e.g., tennis ball color, context, interaction), concluding with a natural-language description and concise labels.

2 Related Work

Tool-Based / Automated Neuron Labeling. Network Dissection [6] addressed automatic CNN interpretability by aligning individual units with a predefined set of concepts using an activation–mask overlap criterion. While it provided a useful insight, its analysis was limited to a fixed, closed vocabulary. Building on the same idea, some works have taken advantage of CLIP's open vocabulary space to improve neuron labeling. CLIP-Dissect [7] used the pre-trained CLIP model to assign open-vocabulary labels to neurons without retraining or manual annotation, offering an efficient and scalable approach to interpretability. FALCON [8] extended this line of work by enhancing label fidelity through counterfactual analysis and spatial filtering. However, its outputs are short phrases rather than rich descriptions. In contrast, MILAN [9] moved beyond tag interpretability by producing open-ended, natural-language descriptions of individual neurons. It does so by training a captioning model on top-activating image regions (neuron exemplars), using crowd-sourced human annotations. More recently, Describe-and-Dissect [10] leveraged pretrained models (BLIP for captioning, GPT-3.5 for summarization, and synthetic image generation for validation)[11, 12] in a three-stage pipeline, providing concise natural-language explanations.

Agent-Based Interpretability. Early work by Bills et al. [13] used GPT-4 to generate natural-language explanations for neurons and test them through activation prediction, introducing a quantitative self-consistency measure. However, this framework was essentially single-pass and did not support iterative hypothesis refinement. Schwettmann et al. [14] advanced this paradigm with the FIND benchmark, where a language-based interpretability agent iteratively proposed, tested, and revised hypotheses on synthetic functions with known ground truth. Iteration improved descriptive accuracy but often failed to capture fine-grained behaviors. Most recently, *MAIA* [1] generalizes this agentic approach to the multimodal setting, equipping a pretrained multimodal LLM with tools for exemplar selection, synthetic image generation, targeted editing, and summarization. Through

iterative experimentation, *MAIA* not only produces expert-level neuron explanations and interventions on spurious features, but also extends to broader interpretability tasks such as bias detection and failure mode analysis.

Recent progress in Open Multimodal LLMs. Over the last two years, open-source multimodal large language models have made rapid progress. Early efforts such as BLIP-2 [15] and LLaVA [16] showed that linking vision encoders with LLMs could yield surprisingly capable visual assistants, while projects like OpenFlamingo [17, 18] and Otter [19] scaled this approach to multi-image and multi-turn reasoning. More recently, model families like Mistral 3 (Small, Medium, Large)[3, 4] and Gemma 3 (1B–27B)[2] have demonstrated strong multimodal reasoning and tool-use abilities, narrowing the gap with proprietary models like GPT-4o [20] and Claude-Sonnet-4[5] while remaining efficient enough for consumer-grade hardware. To our knowledge, previous interpretability frameworks have not yet taken advantage of these open multimodal backbones, and *OpenMAIA* is the first automated agent-based interpretability framework fully implemented on top of open-source models.

3 OpenMAIA Framework

3.1 MAIA overview

The original *MAIA* framework [1] formulates neuron interpretation as an agentic reasoning problem. An autonomous language–vision agent interacts with a target neuron by iteratively proposing hypotheses about its selectivity, testing them through synthetic image generation, and refining its description based on observed activations. Each iteration is triggered by a prompt that requests the agent to explain what the neuron *detects* or *responds to*, given a small set of maximally activating images.

The agent operates in a tool-augmented environment that enables multimodal reasoning. Its main tools are: (i) a **text-to-image generator** (DALL·E 3[21]) that produces candidate stimuli matching the current hypothesis; (ii) an **image editor** (InstructPix2Pix[22]) that locally modifies generated images to isolate or vary specific visual features; (iii) a **visual descriptor** based on a multimodal model (GPT-4V[23]) that summarizes what is depicted in each image; and (iv) a **language summarizer** that consolidates evidence across rounds into a concise neuron explanation. All these components are orchestrated by a closed, proprietary large multimodal model (GPT-4V[23]) that interprets prompts, selects tools, and integrates their outputs.

This design enables *MAIA* to autonomously converge toward human-interpretable neuron descriptions, but it also introduces several practical limitations: the reliance on commercial APIs restricts reproducibility, their internal training data and inference procedures are undisclosed, and the overall cost of large-scale evaluation is high. These factors motivate the development of an open, reproducible alternative, *OpenMAIA*, that preserves the same reasoning protocol while replacing every closed component with open-weight counterparts.

3.2 OpenMAIA implementation

OpenMAIA preserves *MAIA*'s iterative reasoning process but replaces all closed-source elements with publicly available open-weight models. The framework maintains the same agentic structure with prompt-driven reasoning, tool invocation, and evidence consolidation, but all language and vision modules are fully transparent and locally executable.

Multimodal backbone. We employ Gemma-3-27B and Mistral-Small-3.2-24B as vision–language backbones due to their strong performance in multimodal reasoning and structured tool use. In the LMSYS Vision Arena (Aug 2025), Gemma-3-27B-IT and Mistral-3.2 ranked #12 and #23 (Elo $\approx$ 1162 and 1135), outperforming larger VLMs [24]. On VLM@school, Mistral-Small-3.2-24B (42.9 %) and Gemma-3-27B (40.2 %) exceeded scale-based expectations [25], while Gemma 3 achieved the lowest mean error on the ORBIT object-property benchmark [26]. Both models natively support function calling for agentic control and have been externally verified for API reliability [27, 28], making them efficient, reproducible backbones for open multimodal agents.

Tool suite. All tools have been updated to rely exclusively on open-source models, as detailed below.

Image generation. We use FLUX.1[dev], a 12 B rectified-flow transformer released as open-weights and deployed through Diffusers with 4-bit NF4 quantization. This configuration maintains useful

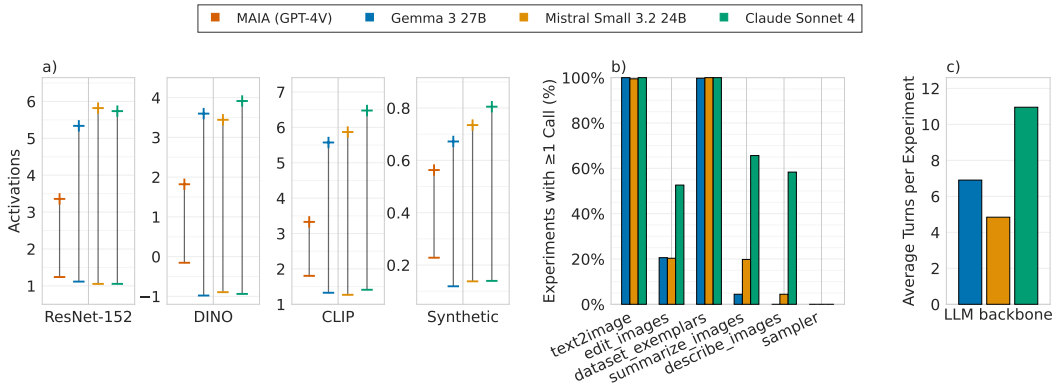


Figure 2: **Comparison of LLM backbones. (a) Predictive activation evaluation.** For each *vision backbone* (ResNet-152, DINO ViT-S/8, CLIP RN50, Synthetic neurons) we aggregate activations produced under each *OpenMAIA LLM backbone* evaluation. The vertical segment per axis spans the mean of the positive and neutral activations; the "+" marker denotes the mean positive activation and the "−" marker the mean neutral activation. *MAIA* score is used only as a *reference*. **(b) Tool usage.** For each LLM backbone, bars show the percentage of experiments that invoked each tool at least once. **(c) Conversation length.** Average number of turns per experiment for each LLM backbone.

visual fidelity while reducing VRAM requirements, enabling interactive reasoning loops on a single high-memory GPU [29, 30, 31].

Image editing. For text-guided modifications, *OpenMAIA* employs FLUX.1 Kontext[dev], an open-weight model unifying generation and editing. Kontext is optimized for local edits, multi-turn consistency, and reference conditioning, and achieves strong results on KontextBench [32, 33]. Its stability across sequential edits aligns well with the causal-intervention reasoning loop.

Image summarization. For visual description and evidence consolidation, we reuse our multimodal backbones (Gemma-3-27B, Mistral Small 3.2). This ensures semantic consistency within the agent while maintaining strong performance on established vision–language benchmarks.

Together, these components provide a fully open, cost-efficient, and reproducible implementation of the *MAIA* framework.

4 Evaluation

To evaluate *OpenMAIA* in this short paper, we adopt the quantitative *neuron description* evaluation protocol introduced in *MAIA* [1]. In this task, the agent must produce natural-language descriptions of individual neuron behavior from both real and synthetic neurons. We then assess these descriptions using *MAIA*'s predictive activation framework, which measures whether a proposed description can be used to generate images that elicit maximally the neuron as predicted.

More concretely, the evaluation protocol works as follows. For each neuron: **(i)** We give the agent's description to a different language model that generates two sets of candidate text prompts: *positive prompts*, intended to maximally activate the neuron, and *neutral prompts*, intended to elicit baseline activation; **(ii)** a text-to-image model converts the selected prompts into images, and we record the neuron's average activation produced by the positive versus the neutral generated images; **(iii)** a description is considered predictive when its positive generations elicit higher activations relative to the neutral baseline.

We apply this procedure across neurons of four vision backbones (ResNet-152, DINO ViT-S/8, CLIP RN50) and synthetic neurons with known ground truth, using two open-source multimodal models (Gemma-3-27B and Mistral-Small-3.2-24B) as the describing agents of *OpenMAIA*. For each condition, we uniformly sample neurons across layers (384 evaluations per backbone) and compute *predictive activation consistency*, the primary *MAIA* metric. Since real neurons lack ground truth, this activation-based protocol serves as a reproducible and model-agnostic benchmark. We also report Claude-Sonnet-4 as a closed-source baseline and include the original *MAIA* (GPT-4V) scores for continuity.

Figure 2a shows that both Gemma-3-27B and Mistral-Small-3.2-24B achieve strong predictive gaps between positive and neutral generations, on par with Claude-Sonnet-4 and clearly outperforming the original *MAIA* baseline (GPT-4V). This indicates that fully open backbones and tools can produce competitive explanations without relying on proprietary APIs.

Tool usage. As shown in Figure 2b, Gemma-3-27B and Mistral-Small-3.2-24B rely primarily on dataset exemplars and image generation, with occasional use of image editing. However, they make little use of image summarization or description tools. In contrast, Claude-Sonnet-4 employs a more diverse tool mix, including frequent calls to summarization modules. Interestingly, this diversity in tool usage correlates with overall performance: Claude Sonnet achieves the strongest predictive scores, followed by Mistral and then Gemma, mirroring the ordering of tool diversity. This suggests that a broader use of the available tools may contribute to more accurate neuron explanations.

Efficiency. Figure 2c reports conversation length (number of turns). Mistral-Small-3.2-24B reached conclusions in the fewest experimental iterations ($\sim$5 on average), while Gemma-3-27B required slightly longer dialogues ($\sim$7 turns). Claude-Sonnet-4, by contrast, often exceeded 10 turns. This suggests that *OpenMAIA*'s open backbones tend to converge more quickly, which is desirable for efficiency. At the same time, the longer trajectories of Claude-Sonnet-4 may reflect a more cautious style, repeatedly validating hypotheses before committing to a final description.

4.1 Discussion

While *OpenMAIA* follows the evaluation protocol of the original *MAIA*, a completely fair comparison between the two systems is not possible. The original framework relied on proprietary models and APIs, such as GPT-4V and DALL·E 3, whose architectures, training data, and inference parameters remain undisclosed. In contrast, *OpenMAIA* employs open-weight models (Gemma-3-27B and Mistral-Small-3.2-24B) and an independent toolchain for image generation and editing. These differences naturally affect visual fidelity, prompt adherence, and linguistic behavior, which can influence evaluation outcomes.

Furthermore, *MAIA*'s closed APIs could not be fully reproduced, as several components originally used in the framework (including early versions of GPT-4V and DALL·E 3) are no longer accessible or have changed substantially since their first release. Our results should be interpreted as an empirical comparison of general performance and convergence behavior rather than a strict replication.

Even under these evaluation constraints, we observe that *OpenMAIA* and *MAIA* obtain comparable results, and on some occasions *OpenMAIA* outperforms *MAIA*. These results empirically demonstrate that open implementations can deliver comparable interpretive quality while ensuring full transparency and reproducibility.

Beyond establishing reproducibility, our findings suggest that a substantial portion of *MAIA*'s interpretive ability arises from its structured reasoning loop rather than from proprietary model capacity. The iterative process of hypothesis generation, visual testing, and result refinement appears to play a central role in interpretive quality across models. This suggests that open multimodal systems can also support reliable, large-scale neuron-level interpretation.

5 Conclusion

This paper introduces *OpenMAIA*, the first fully open-source interpretability agent derived from *MAIA*. It integrates two open-source multimodal backbones, Gemma-3-27B and Mistral-Small-3.2-24B, and an updated tool suite built entirely on open-weight models. Our evaluation shows that these open-source backbones produce neuron descriptions comparable to Claude-Sonnet-4 while converging more efficiently with fewer experimental iterations.

By removing reliance on proprietary APIs, *OpenMAIA* enables transparent and reproducible analysis of model behavior. The updated codebase is publicly available at https://github.com/multimodal-interpretability/maia and offers a solid foundation for scalable, community-driven research in multimodal interpretability. This work advances the broader goal of developing open, reproducible, and accessible interpretability frameworks [34].

Acknowledgements

We are grateful for the support of ARL grant #W911NF-24-2-0069, MIT-IBM Watson AI Lab grant #W1771646, Hyundai NGV, ONR MURI grant #033697-00007, and PID2022-138721NB-I00, funded by MCIN/AEI/10.13039/501100011033 and FEDER, EU.

References

[1] Tamar Rott Shaham, Sarah Schwettmann, Franklin Wang, Achyuta Rajaram, Evan Hernandez, Jacob Andreas, and Antonio Torralba. A multimodal automated interpretability agent. In *Forty-first International Conference on Machine Learning*, 2024.

[2] Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, Louis Rouillard, Thomas Mesnard, Geoffrey Cideron, Jean bastien Grill, Sabela Ramos, Edouard Yvinec, Michelle Casbon, Etienne Pot, Ivo Penchev, Gaël Liu, Francesco Visin, Kathleen Kenealy, Lucas Beyer, Xiaohai Zhai, Anton Tsitsulin, Robert Busa-Fekete, Alex Feng, Noveen Sachdeva, Benjamin Coleman, Yi Gao, Basil Mustafa, Iain Barr, Emilio Parisotto, David Tian, Matan Eyal, Colin Cherry, Jan-Thorsten Peter, Danila Sinopalnikov, Surya Bhupatiraju, Rishabh Agarwal, Mehran Kazemi, Dan Malkin, Ravin Kumar, David Vilar, Idan Brusilovsky, Jiaming Luo, Andreas Steiner, Abe Friesen, Abhanshu Sharma, Abheesht Sharma, Adi Mayrav Gilady, Adrian Goedeckemeyer, Alaa Saade, Alex Feng, Alexander Kolesnikov, Alexei Bendebury, Alvin Abdagic, Amit Vadi, András György, André Susano Pinto, Anil Das, Ankur Bapna, Antoine Miech, Antoine Yang, Antonia Paterson, Ashish Shenoy, Ayan Chakrabarti, Bilal Piot, Bo Wu, Bobak Shahriari, Bryce Petrini, Charlie Chen, Charline Le Lan, Christopher A. Choquette-Choo, CJ Carey, Cormac Brick, Daniel Deutsch, Danielle Eisenbud, Dee Cattle, Derek Cheng, Dimitris Paparas, Divyashree Shivakumar Sreepathihalli, Doug Reid, Dustin Tran, Dustin Zelle, Eric Noland, Erwin Huizenga, Eugene Kharitonov, Frederick Liu, Gagik Amirkhanyan, Glenn Cameron, Hadi Hashemi, Hanna Klimczak-Plucińska, Harman Singh, Harsh Mehta, Harshal Tushar Lehri, Hussein Hazimeh, Ian Ballantyne, Idan Szpektor, Ivan Nardini, Jean Pouget-Abadie, Jetha Chan, Joe Stanton, John Wieting, Jonathan Lai, Jordi Orbay, Joseph Fernandez, Josh Newlan, Ju yeong Ji, Jyotinder Singh, Kat Black, Kathy Yu, Kevin Hui, Kiran Vodrahalli, Klaus Greff, Linhai Qiu, Marcella Valentine, Marina Coelho, Marvin Ritter, Matt Hoffman, Matthew Watson, Mayank Chaturvedi, Michael Moynihan, Min Ma, Nabila Babar, Natasha Noy, Nathan Byrd, Nick Roy, Nikola Momchev, Nilay Chauhan, Noveen Sachdeva, Oskar Bunyan, Pankil Botarda, Paul Caron, Paul Kishan Rubenstein, Phil Culliton, Philipp Schmid, Pier Giuseppe Sessa, Pingmei Xu, Piotr Stanczyk, Pouya Tafti, Rakesh Shivanna, Renjie Wu, Renke Pan, Reza Rokni, Rob Willoughby, Rohith Vallu, Ryan Mullins, Sammy Jerome, Sara Smoot, Sertan Girgin, Shariq Iqbal, Shashir Reddy, Shruti Sheth, Siim Põder, Sijal Bhatnagar, Sindhu Raghuram Panyam, Sivan Eiger, Susan Zhang, Tianqi Liu, Trevor Yacovone, Tyler Liechty, Uday Kalra, Utku Evci, Vedant Misra, Vincent Roseberry, Vlad Feinberg, Vlad Kolesnikov, Woohyun Han, Woosuk Kwon, Xi Chen, Yinlam Chow, Yuvein Zhu, Zichuan Wei, Zoltan Egyed, Victor Cotruta, Minh Giang, Phoebe Kirk, Anand Rao, Kat Black, Nabila Babar, Jessica Lo, Erica Moreira, Luiz Gustavo Martins, Omar Sanseviero, Lucas Gonzalez, Zach Gleicher, Tris Warkentin, Vahab Mirrokni, Evan Senter, Eli Collins, Joelle Barral, Zoubin Ghahramani, Raia Hadsell, Yossi Matias, D. Sculley, Slav Petrov, Noah Fiedel, Noam Shazeer, Oriol Vinyals, Jeff Dean, Demis Hassabis, Koray Kavukcuoglu, Clement Farabet, Elena Buchatskaya, Jean-Baptiste Alayrac, Rohan Anil, Dmitry, Lepikhin, Sebastian Borgeaud, Olivier Bachem, Armand Joulin, Alek Andreev, Cassidy Hardin, Robert Dadashi, and Léonard Hussenot. Gemma 3 technical report, 2025. URL `https://arxiv.org/abs/2503.19786`.

[3] Mistral AI. Mistral small 3.1 (mistral-small-2503). `https://mistral.ai/news/mistral-small-3-1`, 2025. Apache 2.0 license; released March 17, 2025.

[4] Mistral AI. Mistral small 3.2 (mistral-small-3.2-24b-instruct-2506). `https://huggingface.co/mistralai/Mistral-Small-3.2-24B-Instruct-2506`, 2025. Apache 2.0 license; released June 2025. Minor update of Mistral Small 3.1.

[5] Anthropic. System card for claude opus 4 and claude sonnet 4. Technical report, Anthropic, 2025. Introduces Claude Sonnet 4, describes model capabilities, safety testing, and deployment under AI Safety Level 2.

[6] David Bau, Bolei Zhou, Aditya Khosla, Aude Oliva, and Antonio Torralba. Network dissection: Quantifying interpretability of deep visual representations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6541–6549, 2017.

[7] Tuomas Oikarinen and Tsui-Wei Weng. CLIP-dissect: Automatic description of neuron representations in deep vision networks. In *The Eleventh International Conference on Learning Representations*, 2023.

[8] Neha Kalibhat, Shweta Bhardwaj, Bayan Bruss, Hamed Firooz, Maziar Sanjabi, and Soheil Feizi. Identifying interpretable subspaces in image representations. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023.

[9] Evan Hernandez, Sarah Schwettmann, David Bau, Teona Bagashvili, Antonio Torralba, and Jacob Andreas. Natural language descriptions of deep visual features. In *International Conference on Learning Representations*, 2021.

[10] Nicholas Bai, Rahul A Iyer, Tuomas Oikarinen, Akshay Kulkarni, and Tsui-Wei Weng. Interpreting neurons in deep vision networks with language models. *TMLR*, 2025.

[11] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. *arXiv preprint arXiv:2201.12086*, 2022. preprint; published January 28, 2022.

[12] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, ..., and Dario Amodei. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020. Introduces GPT-3, released May 2020.

[13] Steven Bills, Nick Cammarata, Dan Mossing, Henk Tillman, Leo Gao, Gabriel Goh, Ilya Sutskever, Jan Leike, Jeff Wu, and William Saunders. Language models can explain neurons in language models. 2023.

[14] Sarah Schwettmann, Tamar Shaham, Joanna Materzynska, Neil Chowdhury, Shuang Li, Jacob Andreas, David Bau, and Antonio Torralba. Find: A function description benchmark for evaluating interpretability methods. *Advances in Neural Information Processing Systems*, 36: 75688–75715, 2023.

[15] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: bootstrapping language-image pre-training with frozen image encoders and large language models. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023.

[16] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL `https://openreview.net/forum?id=w0H2xGHlkw`.

[17] Anas Awadalla, Irena Gao, Josh Gardner, Jack Hessel, Yusuf Hanafy, Wanrong Zhu, Kalyani Marathe, Yonatan Bitton, Samir Gadre, Shiori Sagawa, Jenia Jitsev, Simon Kornblith, Pang Wei Koh, Gabriel Ilharco, Mitchell Wortsman, and Ludwig Schmidt. Openflamingo: An open-source framework for training large autoregressive vision-language models. *arXiv preprint arXiv:2308.01390*, 2023.

[18] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob Menick, Sebastian Borgeaud, Andy Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikolaj Binkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karen Simonyan. Flamingo: a visual language model for few-shot learning. *ArXiv*, abs/2204.14198, 2022.

[19] Bo Li, Yuanhan Zhang, Liangyu Chen, Jinghao Wang, Jingkang Yang, and Ziwei Liu. Otter: A multi-modal model with in-context instruction tuning. *arXiv preprint arXiv:2305.03726*, 2023.

[20] OpenAI. Gpt-4o technical report, 2024. `https://openai.com/index/gpt-4o`.

[21] OpenAI. Dall·e 3, 2023. `https://openai.com/dall-e-3`.

[22] Thomas Brooks et al. Instructpix2pix: Learning to follow image editing instructions. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. Text-guided image editing model; often struggles with localized compositional edits.

[23] OpenAI. GPT-4V(ision) System Card. `https://openai.com/index/gpt-4v-system-card/`, September 2023.

[24] Christopher Chou, Lisa Dunlap, Koki Mashita, Krishna Mandal, Trevor Darrell, Ion Stoica, Joseph E. Gonzalez, and Wei-Lin Chiang. Visionarena: 230k real world user-vlm conversations with preference labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3877–3887, June 2025.

[25] René Peinl and Vincent Tischler. Vlm@school – evaluation of ai image understanding on german middle-school knowledge. arXiv preprint arXiv:2506.11604, 2025. Evaluates 13 open-weight VLMs on German-language school-level visual QA.

[26] Abhishek Kolari, Mohammadhossein Khojasteh, Yifan Jiang, Floris den Hengst, and Filip Ilievski. Orbit: An object property reasoning benchmark for visual inference tasks. *arXiv preprint*, arXiv:2508.10956, 2025. Systematic benchmark of object-property reasoning.

[27] Amazon Web Services. Mistral small 3.2 24b instruct (v2506) on amazon bedrock and sagemaker jumpstart. AWS ML Blog (July 29, 2025), 2025. Independent report of improved function-calling reliability and image-text reasoning capabilities in Mistral 3.2.

[28] Berkeley Function-Calling Leaderboard Team. Berkeley function-calling leaderboard (bfcl). BFCL online benchmark (ICML 2025), 2025. Standardized benchmark for structured tool/function-calling ability in LLMs.

[29] Black Forest Labs. Flux. `https://github.com/black-forest-labs/flux`, 2024.

[30] Hugging Face. Quantizing diffusers models — 4-bit nf4 inference. Hugging Face blog, 2025. Guidelines and evaluation of quantization strategies (NF4) for diffusion models in Diffusers framework.

[31] Black Forest Labs. 4-bit nf4 quantization of flux.1 [dev]. Hugging Face model card documentation, 2025. Demonstrates effective deployment of FLUX.1 [dev] under 4-bit NF4 quantization while maintaining visual quality.

[32] Black Forest Labs, Stephen Batifol, Andreas Blattmann, Frederic Boesel, Saksham Consul, Cyril Diagne, Tim Dockhorn, Jack English, Zion English, Patrick Esser, Sumith Kulal, Kyle Lacey, Yam Levi, Cheng Li, Dominik Lorenz, Jonas Müller, Dustin Podell, Robin Rombach, Harry Saini, Axel Sauer, and Luke Smith. Flux.1 kontext: Flow matching for in-context image generation and editing in latent space, 2025. URL `https://arxiv.org/abs/2506.15742`.

[33] Black Forest Labs. Flux.1 kontext [dev] — 12 b open-weight model for image editing. Hugging Face model card, 2025. Developer-focused open-weight version with local/global editing and multi-turn robustness.

[34] Zihao Lin, Samyadeep Basu, Mohammad Beigi, Varun Manjunatha, Ryan A Rossi, Zichao Wang, Yufan Zhou, Sriram Balasubramanian, Arman Zarei, Keivan Rezaei, et al. A survey on mechanistic interpretability for multi-modal foundation models. *arXiv preprint arXiv:2502.17516*, 2025.

Appendix A: Qualitative Neuron Descriptions

Your overall task is to describe the visual concepts that maximally activate a neuron inside a deep network for computer vision. You are provided with a Python library containing functions to run experiments on a specific neuron (located inside the "System" class) using tools provided in the "Tools" class. Make sure to utilize a variety of tools from the library to maximize the scope and depth of your experimentation. Neurons may be selective for specific visual concepts, multiple unrelated concepts, or broader, general concepts. Therefore, creatively design experiments to test both general and highly specific hypotheses. If a neuron shows selectivity for multiple concepts, ensure that each of these is clearly described in your final report. To characterize the neuron's visual selectivity clearly, actively utilize all available functions (dataset_exemplars, edit_images, text2image, summarize_images, describe_images, display) from the Tools class throughout your experiments.

Follow this structured approach strictly:

1. At each experimental step, you must write only one experiment implementation using Python code in the following format:

   `[CODE]:`

   ```
   # e.g.:
   # Write exactly one experiment implementation using Python,
   # the System class, and Tools class.
   # Obtain preliminary results from dataset_exemplars
   # Display values and images explicitly using tools.display()
   # if you wish to view them.
   ```

2. After the code execution, wait for the results before proceeding further.
3. Based on the displayed neuron activations, first summarize common features in inputs that highly activate the neuron. Do this independently of previous hypotheses.
4. Then write multiple hypotheses explaining potential visual concepts activating the neuron. Hypotheses should:
   - Include multiple possibilities if necessary (e.g., dogs OR cars OR birds).
   - Explore different abstraction levels or alternative explanations.
   - Be specific and general to cover all possibilities.
5. Format your hypotheses clearly:

   ```
   [HYPOTHESIS LIST]:
   Hypothesis_1:  <hypothesis_1>
   ...
   Hypothesis_n:  <hypothesis_n>
   ```

6. After analyzing the results of the executed experiment, revise your hypothesis list accordingly, but only if new activations are different than previous ones.
7. Next, propose additional experiments to test all current hypotheses thoroughly, again using only one experiment at a time, clearly formatted as above.

Proceed in a step-by-step manner, strictly adhering to this flow:

- MAIA: Propose one clearly defined experiment (`[CODE]`).
- Experiment Execution: Await and carefully analyze the results.
- MAIA: Update your hypothesis list (`[HYPOTHESIS LIST]`) based on analysis.
- Experiment Execution: Continue iteratively.

Repeat these cycles until each hypothesis is conclusively proven or disproven through multiple experiments. When fully confident in the neuron's selectivity, provide a final comprehensive and specific description:

`[DESCRIPTION]: <final detailed description>`

The description must be:

- Highly specific (e.g., “dogs running on grass”, not merely “dog”).
- Complete, covering all relevant selective concepts.
- Clearly indicate multiple concepts separated by logical “OR” if applicable.

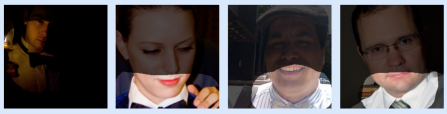
Finally, summarize your findings concisely into one or more (only if needed) labels derived from the hypotheses you are most confident about:
`[LABEL 1]: <label describing first confirmed concept>`
`[LABEL 2]: <label describing second confirmed concept>` (if applicable)

Now, begin your first experiment clearly and wait for results.

Appendix B: Qualitative Neuron Descriptions

GEMMA 3 27B

```
# Obtain preliminar results from dataset_exemplars
exemplar_data = tools.dataset_exemplars(system)
for activation, image in exemplar_data:
    tools.display(image, f"Activation: {activation}")
```



Act: 36.96 Act: 35.61 ... Act: 34.16 Act: 30.56

The top activations from the `dataset_exemplars` function all show images of people, and a very strong commonality is the presence of a bow tie. The images are also generally close-up shots of faces or upper bodies. The lighting is often dim or dramatic.

[HYPOTHESIS LIST]:
Hypothesis_1: The neuron is activated by images of people wearing bow ties.
Hypothesis_2: The neuron is activated by close-up images of faces.
Hypothesis_3: The neuron is activated by images of people in formal wear (including but not limited to bow ties).
Hypothesis_4: The neuron is activated by images with low-key lighting (dark and dramatic).
Hypothesis_5: The neuron is activated by images of men.

```
# Test hypothesis 1: Formal wear and bow ties
prompts = ["a man wearing a bow tie", "a woman wearing a bow tie",
           "a child wearing a bow tie", "a dog wearing a bow tie",
           "a cat wearing a bow tie"]
images = tools.text2image(prompts)
act_list, act_map = system.call_neuron(images)
for activation, image in zip(act_list, act_map):
   tools.display(image, f"Activation: {activation}")
```

Act: 29.01 Act: 29.76 Act: 25.96 Act: 24.77

Act: 28.40

The results of the "bow tie" experiment show that the neuron *is* strongly activated by images of people (and to a lesser extent, animals) wearing bow ties. The activations for the people with bow ties (29.01, 29.76) are similar to the activations from the `dataset_exemplars` function (around 30-36). The dog and cat with bow ties have lower activations (24.77, 28.4), but are still non-zero. This suggests the bow tie is a key feature.

Hypothesis 1 (bow ties) is strongly supported.
Hypotheses 2 (close-up faces) and **3** (formal wear) are also likely relevant.
Hypothesis 4 (low-key lighting) may be a factor as the generated images are also quite dark.
Hypothesis 5 (men) is not supported as the woman with a bow tie also has a high activation.

```
# Test hypothesis 3: people in formal wear
prompts = ["a man in a tuxedo", "a woman in a ball gown",
           "a man in a suit and tie", "a child in a suit",
           "a person in formal wear"]
images = tools.text2image(prompts)
act_list, map_list = system.call_neuron(images)
for activation, image in zip(act_list, map_list):
    tools.display(image, f"Activation: {activation}")
```

MISTRAL SMALL 3.2 24B

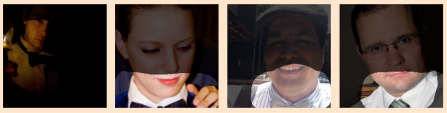
I'll begin by proposing the first experiment to understand the neuron selectivity. This initial experiment will use the `dataset_exemplars` function to identify the most activating images from the ImageNet dataset.

```
# First experiment: Identify the top activating images from ImageNet
#dataset
exemplar_data = tools.dataset_exemplars(system)

# Display the top 5 exemplars with their
# activation values
for act, img in exemplar_data[:5]:
   tools.display(img, f"Activation: {act:.2f}")
```

After running this experiment, we'll be able to:
- See the actual images that maximally activate the neuron.
- Observe common visual patterns across these images.
- Form initial hypotheses about what the neuron might be detecting.

Please execute this code and show me the results so we can proceed with analysis and hypothesis formation.



Act: 36.96 Act: 35.61 ... Act: 34.16 Act: 30.56

Based on the displayed neuron activations and the corresponding exemplar images, I can summarize the common features that highly activate this neuron:

[SUMMARY]:
The neuron appears to be strongly activated by images featuring:
1. Men wearing bow ties
2. Close-up facial features with bow ties
3. Formal attire, particularly the bow tie element
4. Dark backgrounds that highlight the bow tie
5. The bow tie itself as a prominent visual feature

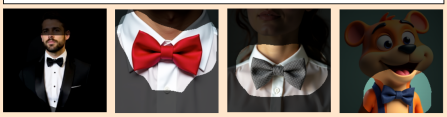
[HYPOTHESIS LIST]:
Hypothesis_1: The neuron is selective for bow ties, particularly when worn by men.
Hypothesis_2: The neuron responds to formal attire, with a specific focus on bow ties.
Hypothesis_3: The neuron detects dark backgrounds with highlighted facial features and accessories.
Hypothesis_4: The neuron is sensitive to the contrast between dark backgrounds and light-colored bow ties.
Hypothesis_5: The neuron responds to the specific shape and pattern of bow ties.

To test these hypotheses more thoroughly, I'll propose the next experiment:

```
# Second experiment: Test specific bow tie concepts
# Generate images specifically targeting bow tie
# concepts
prompts = ["a man wearing a black bow tie on a dark background",
   "a close-up of a red bow tie on a white shirt",
   "a woman wearing a patterned bow tie",
   "a cartoon character with an oversized bow tie",
   "a man in formal attire without a bow tie (wearing a necktie)"]

# Generate images and test neuron activations
images = tools.text2image(prompts)
act_list, map_list = system.call_neuron(images)

# Display results
for activation, image, prompt in zip(act_list, map_list, prompts):
tools.display(image, f"Prompt: {prompt}
              Activation: {activation:.2f}")
```

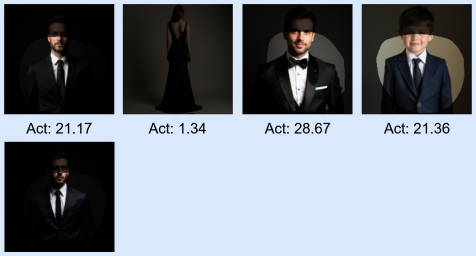


Act: 29.41 Act: 16.28 Act: 21.90 Act: 25.79

Act: 19.23

Figure 3: Full experimental trace of *OpenMAIA* (page 1 of 2).

GEMMA 3 27B



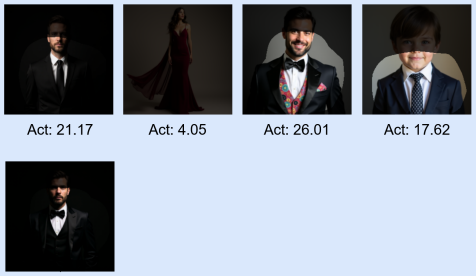

```
The results of the "formal wear without bow ties" experiment are
very telling. The images of people in formal wear *without* bow ties
(activations of 21.17, 1.34, 21.36, and 16.77) have significantly
*lower* activations than the images with bow ties from previous
experiments (which were consistently in the 25-35 range). The woman
in a dress with no accessories has a very low activation (1.34),
similar to the previous results with a woman in a ball gown.

This strongly suggests that the *bow tie* is a critical feature for
activating this neuron. The other elements (formal wear, dark
lighting, close-up face) are likely contributing factors, but the bow
tie seems to be the most important trigger.

[HYPOTHESIS LIST]:
Hypothesis_1: The neuron is activated by images of people wearing bow
ties. (Extremely strongly supported)
Hypothesis_2: The neuron is activated by close-up images of faces.
(Likely relevant, but not primary)
Hypothesis_3: The neuron is activated by images of people in a
specific style of formal wear (bow ties, classic portraiture style).
(Strongly supported - but bow tie is the key element)
Hypothesis_4: The neuron is activated by images with low-key lighting
(dark and dramatic). (Likely relevant, often co-occurs with the key
features)

Now I will test a final experiment to confirm the importance of the
bow tie:

# Test with other types of neckwear
prompts = ["a man in a black suit and necktie, studio lighting",
           "a woman in a long evening gown and a scarf",
           "a man in a tuxedo and a colorful tie",
           "a child in a suit and a patterned tie",
           "a man in a dark suit and a cravat, low key lighting"]
images = tools.text2image(prompts)
act_list, map_list = system.call_neuron(images)
for activation, image in zip(act_list, map_list):
    tools.display(image, f"Activation: {activation}")
```




```
The final experiment confirms the critical role of the bow tie: images
featuring bow ties show high activations (26.01 and 32.91), while
those with other neckwear such as ties or scarves have significantly
lower activations (21.17, 4.05, and 17.62), clearly indicating that
the neuron is highly selective for images of people wearing bow ties,
especially in formal or classic portraiture with dark or dramatic
lighting.

[DESCRIPTION]: The neuron is highly selective for images of people
(and to a lesser extent, other animals) wearing bow ties. The images
that most strongly activate the neuron are typically close-up
portraits of people in formal wear (such as tuxedos), with low-key or
dramatic lighting. The presence of a bow tie is a necessary condition
for high activation.
[LABEL 1]: Bow Tie Detection
[LABEL 2]: Formal Portraiture
```

MISTRAL SMALL 3.2 24B

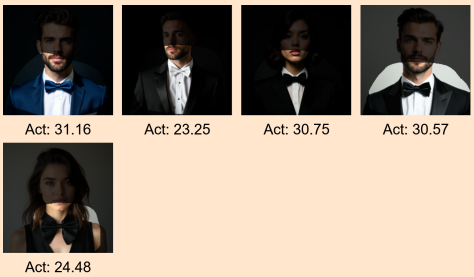
```
[HYPOTHESIS LIST UPDATE]:
Hypothesis_1: The neuron is highly selective for black bow ties
(highest activation for black bow tie, lower for other colors).
Hypothesis_2: The neuron responds strongly to bow ties worn by men
(higher activation for male subjects).
Hypothesis_3: The neuron prefers dark backgrounds with highlighted
bow ties (consistent high activation for dark backgrounds).
Hypothesis_4: The neuron shows moderate activation for patterned bow
ties but less than solid black.
Hypothesis_5: The neuron responds to the concept of formal attire but
with stronger preference for bow ties over neckties.
Hypothesis_6: The neuron may have some sensitivity to the size of the
bow tie (higher activation for oversized bow tie in cartoon character)

I'll propose the next experiment focusing on variations of bow tie
color and gender:

# Third experiment: Test bow tie color and gender variations
prompts = [
   "a man wearing a blue bow tie on a  dark background",
   "a man wearing a white bow tie on a dark background",
   "a woman wearing a black bow tie on a dark background",
   "a man wearing a black bow tie on a light background",
   "a woman wearing a black bow tie on a light background"
]

# Generate images and test neuron activations
images = tools.text2image(prompts)
act_list, map_list = system.call_neuron(images)

# Display results
for activation, image, prompt in zip(act_list, map_list, prompts):
    tools.display(image, f"Prompt: {prompt}
                  Activation: {activation:.2f}")
```




```
[DESCRIPTION]:
This neuron is highly selective for bow ties, showing strong
activation regardless of color (black, blue, white) or gender of the
wearer. It responds most strongly to black bow ties on dark
backgrounds but can still detect bow ties on light backgrounds. The
neuron appears to be sensitive to the presence of a bow tie as a
distinct visual feature, with slightly higher activation for male
subjects. It shows consistent responses to different bow tie textures
and patterns, suggesting its selectivity is focused on the general
concept of a bow tie rather than specific details.

[LABEL 1]: Bow tie detector
[LABEL 2]: Formal attire accessory detector (with bow tie preference)
```

Figure 4: Full experimental trace of *OpenMAIA* (page 2 of 2).

We provide full-page qualitative traces illustrating the complete experimental workflow of *OpenMAIA* across both open-source models. These examples demonstrate the agent's iterative reasoning process, visual interventions, and convergence toward interpretable neuron-level explanations