HIDDEN STATES, OPEN SECRETS: RECOGNIZING VLM FAILURES DURING VISUAL PREFILL

Anonymous authors
Paper under double-blind review

ABSTRACT

Vision-language models sometimes answer incorrectly despite retaining sufficient visual information in their internal representations. Such failures demonstrate the importance of how a model uses its representations, but also raise the question of when unsuccessful answers are preceded by distinguishable representations in the first place. We investigate this question across multimodal tasks by examining visual prefill, the image-token states formed even before access to any subsequent textual question. Prior work has shown that learned hidden states have geometric structures which reflect properties of the information they represent. As such, by collecting internal representations which usually lead to correct model responses, we obtain a reference for what favourable visual representations should geometrically be. By measuring the angular distance of other states to their nearest reference neighbors, we are able to find strong separation between usually correct and usually incorrect inputs on ScienceQA and ChartQA, with unsuccessful inputs tending to lie farther from successful references. However, on tasks like POPE and CV-Bench, the separation is much weaker. These results extend across distinct visual architectures and are visible within the first quarter of visual-token positions.

1 INTRODUCTION

In recent years, vision-language models (VLMs) have become increasingly capable of answering questions that combine visual information with natural language (Bai et al., 2025). In spite of this, their improving multimodal performance has revealed little about whether an incorrect answer stems from how they represent an image or how they use the information it contains. Consider a ScienceQA question that asks for the probability of an inherited trait given a Punnett square (Lu et al., 2022) (see Figure 2). An incorrect answer could follow from misrepresenting the allele combinations in the diagram or from mishandling the inheritance rule despite representing those combinations correctly. Prior work has shown that the apparent demands of a task provide an uncertain guide to this distinction. Confusions in visual embeddings are associated with downstream errors on basic visual judgments (Tong et al., 2024b), while failures on depth and correspondence tasks can occur even when sufficient task-relevant information remains accessible throughout the model (Fu et al., 2025). These findings motivate examining how the relationship between visual representations and successful answering varies across tasks.

To examine this relationship, we study VLMs' internal visual representations before the model has access to a textual question. This stage of processing, which we call visual prefill, produces the internal representations at image-token positions that can inform subsequent answering. Prior work has shown that hidden states have geometric structure that reflects properties of the information they represent (Khandelwal et al., 2020). In language models, representational directions and local neighborhood structure carry information about answer truthfulness (Marks & Tegmark, 2024; Yin et al., 2024). In VLMs, geometric analyses of spatial representations have similarly shown relations to success on spatial reasoning tasks (Min et al., 2026).

With these results, we expect that representations which capture sufficient task-relevant visual information exhibit some recurring geometric structure, and that departures from this structure may signal an unusually weaker representation. We search for this structure by collecting hidden states from correct responses. Because similar directions reflect similarities in the latent information, we

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

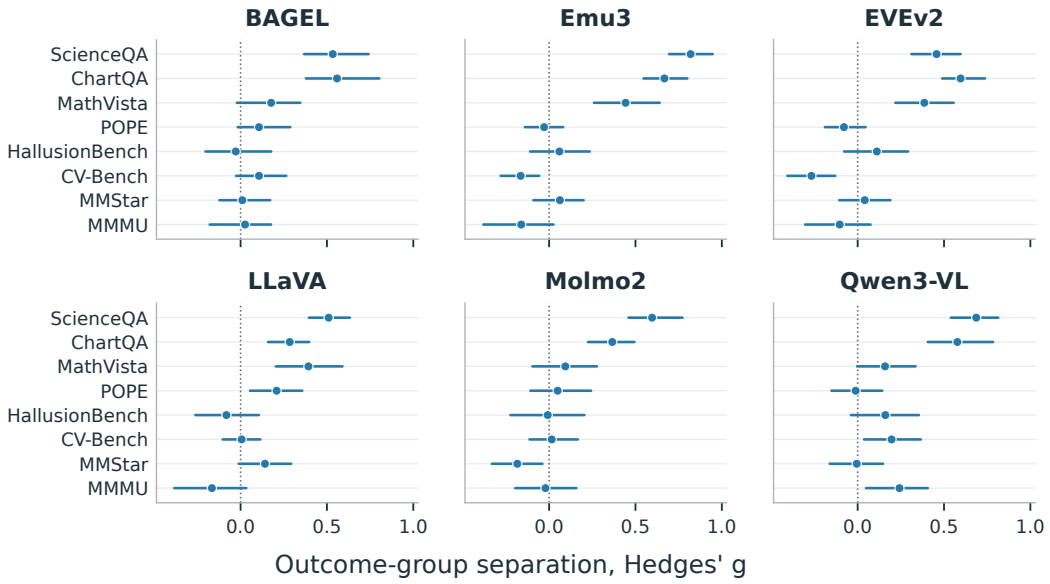

Figure 1: Visual-state separation depends on the task across different model architectures. ScienceQA and ChartQA show stronger separation than POPE in each model's primary descriptive cohort, while the other benchmarks vary. Points give late-region Hedges' g, and bars give 95% image-group intervals. Positive values mean that usually incorrect inputs lie farther from successful reference neighborhoods.

compare each visual state with its nearest reference directions to assess its resemblance to these successful examples (Sun et al., 2022; Modell et al., 2025). Representations which result in downstream failure should lie farther from successful reference neighborhoods, and vice versa. Our aim is to establish on which tasks successful and unsuccessful answers are preceded by distinguishable visual representations.

We find systematic differences across tasks in how visual representations distinguish later success from failure (Figure 1). Tasks requiring reasoning from diagrams and charts, i.e. ScienceQA and ChartQA (Lu et al., 2022; Masry et al., 2022), show stronger separation between successful and unsuccessful inputs than object-presence and spatial judgment tasks, i.e. POPE and CV-Bench (Li et al., 2023; Tong et al., 2024a). This behavioural distinction between task types recurs across encoder-based, encoder-free, and early-fusion models (Bai et al., 2025; Clark et al., 2026; Liu et al., 2024; Deng et al., 2025; Diao et al., 2025; Wang et al., 2024). In all six models we test, the behaviour is observable as early as within the first quarter of visual-token positions, as well as across decoder layers. For ScienceQA and ChartQA, training a small MLP to predict task-success from its visual-state angular distances can achieve performance comparable to or better than Kogilathota et al. (2026)'s VT method for predicting failure before the text prompt is even known.

2 Related Works

Visual representation and use. Visual-feature probes and end-to-end benchmarks assess different aspects of visual understanding. First, frozen-feature evaluations measure recoverable scene geometry and correspondence (Banani et al., 2024). Linking these representations to answering reveals encoder confusions associated with errors, alongside failures despite accessible visual information throughout the model (Tong et al., 2024b; Fu et al., 2025). Through their design, benchmarks implicitly or explicitly test these dynamics. Chart experiments further distinguish visual encoding from information extraction, finding benefits from improving both (Liu et al., 2025). To separate task demands, Cambrian and VisOnlyQA evaluate fundamental visual abilities with limited dependence on domain knowledge or complex reasoning (Tong et al., 2024a; Kamoi et al., 2025). MathVerse varies the information supplied through diagrams and text, while two-stage abstract-reasoning evaluations separate image description from rule induction (Zhang et al., 2024; Wang et al., 2026). And lastly,

HallusionBench uses controlled image-question comparisons to distinguish visual illusion from language hallucination (Guan et al., 2023). We complement these analyses by focusing on visual-state geometry in isolation, and its relation to later question correctness.

Representation geometry and reliability. Geometry provides a quantitative basis to interpret these internal differences. Hidden-state classifiers predict truthfulness, while directional and subspace analyses describe its representational organization (Azaria & Mitchell, 2023; Marks & Tegmark, 2024; Cho et al., 2026). In a more explicit direction, there are methods which detect hallucinations through local intrinsic dimension and spectral structure, including covariance across sampled responses and subspaces estimated from unlabeled generations (Yin et al., 2024; Chen et al., 2024a; Sriramanan et al., 2024; Du et al., 2024). In VLMs, contrastive spatial probes likewise connect errors to geometric entanglement of depth and vertical image position (Min et al., 2026). Reference-based approaches instead measure support from known examples. Deep k-nearest neighbors and trust scores assess agreement with labeled neighborhoods (Papernot & McDaniel, 2018; Jiang et al., 2018). Distances between normalized features support out-of-distribution detection, with related representation-distance methods enabling selective language generation (Sun et al., 2022; Ren et al., 2023). Reference geometry also guides in-context example selection for hallucination detection through class-aware prototypes (Vamshi et al., 2026). We use successful inputs as references and track their local directional similarity to visual states across positions and layers.

Hallucination detection and attribution. The choice of representation also determines when hallucinations can be detected. Span- and token-level detectors assess generated content (Gunjal et al., 2024; Park et al., 2025), while internal-state methods use hidden-state and attention signals (Miyazato et al., 2026; Zhang et al., 2026). Even before generation, HALP probes encoder features and decoder states at the final visual and query tokens, termed VT and QT, respectively (Kogilathota et al., 2026). Similarly, at the final prompt token, layer-wise probes have traced competition between visually supported answers and language priors (Zhou et al., 2026). More generally, attention analyses identify layer-specific patterns associated with hallucinated object tokens (Jiang et al., 2025; Cao et al., 2026). Related attribution work finds that final image-token states better predict recognition failures, while prompt representations better predict errors after recognition (Vo et al., 2026). We complement these endpoint analyses by examining at what position success and failure separate along the visual-token sequence, and how this depends on the task.

3 Background

3.1 Visual prefill and accessible context

Across the models we study, image inputs enter the decoder as sequences of visual tokens. Each decoder layer computes a hidden-state vector at every token position. We call the computation at these visual-token positions *visual prefill* and measure the resulting decoder states. The image context available at each position is model-dependent, since tokenization, visual encoding, and decoder attention determine which image regions can contribute to each state. For example, vision encoders mix information across image regions before their outputs reach the decoder, unlike the linear, sequential projections of many encoder-free models (Diao et al., 2025; 2024). Furthermore, instead of one-directional causal attention, BAGEL and Molmo2 use bidirectional attention within their visual-token blocks, allowing earlier positions to attend to later image tokens during decoder computation (Deng et al., 2025; Clark et al., 2026).

These differences in image context are separate from access to the textual question. With the exception of MMMU, image tokens precede the question on all the benchmarks we test. For all our models, the attention masks prevent visual-token states from attending to the later question text. This allows us to examine exactly how visual representations relate to subsequent correctness in isolation from the actual question. We group visual prefill states by their visual-token position and decoder layer. Since prefill can compute positions in parallel, distance measurements across visual-token positions describe where a signal appears within the input sequence rather than when it appears during computation. MMMU inputs can interleave images and question text, making the available question context depend on each visual token's position.

3.2 Successful reference neighborhoods

For each image-question pair, we sample 15 responses. We label an input *usually correct* if at least eight responses are correct, and *usually incorrect* otherwise. We do this to try and understand whether the geometric structure of visual representations we study is correlated with a model's task difficulties. Furthermore, as mentioned previously, because visual prefill is deterministic and blind to the textual question, different questions about the same image can have identical visual prefill states while receiving different correctness labels. We retain these question-specific outcomes to examine how much visual representations alone distinguish subsequent success from failure. We construct reference clouds from usually correct inputs, matching the model, dataset, decoder layer, and visual-token position, and excluding reference states associated with conflicting labels.

The reference cloud allows us to test whether visual states preceding failure depart from the representation-geometric patterns associated with successful answers. We compare each held-out state with nearby references to capture local similarities within the sampled population. We focus on directional similarity, motivated by work relating normalized representation geometry to similarities in encoded information (Modell et al., 2025). Specifically, we normalize each hidden state to unit length, placing the reference cloud on a unit sphere, and define its *angular kNN distance* as the mean angle to its $k = 15$ nearest reference states (Sun et al., 2022). A larger distance indicates that a state's direction is less typical of nearby successful examples. Because different questions can share visual states, as discussed above, we exclude held-out states identical to those in the reference split in the kNN.

For more details on sampling, grading, reference selection, and the kNN estimator, see Appendix A.

3.3 Measuring separation across positions and layers

We compare the distance distributions of usually correct and usually incorrect inputs using Hedges' g, a small-sample-corrected version of Cohen's d that expresses the difference in group means relative to their pooled within-group variation (Hedges, 1981). In our case, Hedges' g measures the standardized difference in distance from the successful reference cloud between usually incorrect and usually correct inputs. Larger values indicate stronger separation, with usually incorrect inputs lying farther from successful references. Therefore, values near zero indicate weaker separation, i.e. less geometric disambiguation between usually correct and usually incorrect representations. We group hidden states based on their position in visual prefill, and the decoder layer they were sampled from. Because the number of tokens in visual prefill is variable—dependent on model and size of image—we split visual prefill into 16 position bins across 12 decoder layers. This allows us to examine around which visual tokens separation begins. We define the early and late visual regions as the first and last four bins respectively, i.e. the first quarter and last quarter of visual tokens in visual prefill. However, the Hedges' g separation we calculate is dependent on which visual states are sampled. To quantify this uncertainty, we recompute Hedges' g on 500 evaluation sets formed by sampling visual states with replacement, and report the 95% confidence intervals.

4 Task-dependent geometry during visual prefill

4.1 Separation is task-dependent

Multimodal tasks differ in both the knowledge needed to answer their questions and the information that must be preserved from their inputs. Returning to the opening problem, a task may challenge a model to build a sufficient visual representation, to use that representation successfully, or both. Existing work characterizes these demands through several overlapping distinctions. *Hidden in Plain Sight* defines *vision-centric* tasks as those answerable from visual evidence without domain expertise. On the tasks they study, visual representations support substantially better performance under direct readout than through the VLM's answers, showing failures to use information already available to the model (Fu et al., 2025). Cambrian-1 distinguishes vision-centric evaluation from Knowledge, OCR & Chart, and General benchmark groups, and operationalizes vision-centric evaluation through spatial relationships, counting, depth order, and relative distance (Tong et al., 2024a). These categories describe different task demands, but do not distinguish whether task difficulty lies in preserving sufficient information versus in using that information successfully.

ScienceQA

What is the probability that a cat produced by this cross will have long fur?

	F	f
F	FF	Ff
f	Ff	ff

(A) 1/4
(B) 4/4
(C) 3/4
(D) 2/4
(E) 0/4

Answer (A) 1/4

POPE

Is there a giraffe in the image?



Answer Yes

Figure 2: The visual input plays different roles in these tasks. The ScienceQA diagram specifies a genetic cross for a probability calculation (Lu et al., 2022). POPE asks about object presence (Li et al., 2023).

For example, consider a grid of coloured shapes and two questions about it. "Is there a red triangle?" could be answered by representing an inventory of the shape-colour combinations present. "What colour is the shape immediately left of the triangle in the third row?" additionally requires associating individual shapes with their locations. Both questions are vision-centric under the definition above, yet a representation sufficient for the first could discard information essential to the second. Furthermore, if questions can address arbitrary locations in an independently arranged grid, the representation must preserve all corresponding local distinctions to support those questions, which is a combinatorial task. Because visual prefill precedes access to the question, the model forms these representations before learning which distinctions the question requires.

We use *incompressibility* to describe the difficulty of preserving the distinctions needed to answer a family of questions in a compressed internal representation. Here, relevant information which should be preserved depends on what the "relevant family of questions" is. An inventory of the objects in an image may be sufficient for models or images whose relevant questions are presence judgments, while questions about individual elements require their attributes and relationships to remain distinguishable. This perspective cuts across benchmark categories and motivates examining their representational demands more specifically. It also preserves the distinction between sufficiency and use: even when the necessary information survives in a representation, answering still depends on the model's ability to recover and use it.

ScienceQA and ChartQA provide settings that may be especially representationally demanding. Consider the Punnett square in Figure 2. The specific ScienceQA question requires recovering allele combinations and applying an inheritance rule to calculate the probability of long fur (Lu et al., 2022). However, the same diagram supports many other relevant questions. "How many capital 'F's appear?" requires retaining individual character identities and their multiplicities. "How many squares are there?" requires retaining the geometry of the grid. "What fonts does the image use?" requires retaining typographic features that an abstract representation of the genetic cross could discard. ChartQA can impose related demands through associations among labels, marks, and values (Masry et al., 2022). In our experiments, both benchmarks exhibit separation between visual states preceding successful and unsuccessful answers. For Qwen3-VL, usually incorrect inputs lie farther from successful reference neighborhoods, with late-region Hedges' $g = 0.69$, 95% interval $[0.54, 0.81]$, on ScienceQA and $g = 0.58$, $[0.41, 0.78]$, on ChartQA.

The POPE photograph is very different. Much of its pixel-level variation, such as texture, shading, and background detail, may be incidental to ordinary questions about the depicted scene. A representation retaining objects, their salient attributes, and spatial relationships could support many such questions while discarding substantial image detail. In our measurements—despite nearly identical response accuracy, i.e. 89.04% on POPE and 89.06% on ScienceQA—Qwen3-VL's visual states on POPE exhibit very little separation between usually correct and usually incorrect inputs, with

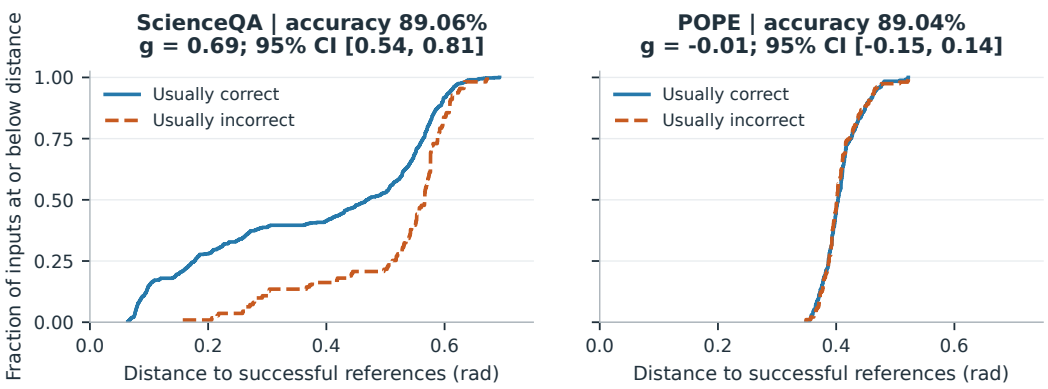

Figure 3: Qwen3-VL's usually correct and usually incorrect inputs separate on ScienceQA and largely overlap on POPE, despite nearly equal answer accuracy. Curves show the complete empirical distributions of late-region angular distance to successful references. Titles give Hedges' g and its 95% image-group interval. The shared axis describes this model and reference construction.

$g = -0.01$, 95% interval $[-0.15, 0.14]$. Figure 3 illustrates how similar answering performance conceals different relationships between visual-state geometry and subsequent failure.

This contrast recurs across models with different visual architectures and pretraining histories (Figure 1). In our main analysis, ScienceQA and ChartQA exhibit stronger separation than POPE in each of the six models. Late-region Hedges' g ranges from 0.46 to 0.82 on ScienceQA and from 0.29 to 0.67 on ChartQA, compared with -0.08 to 0.21 on POPE. The contrast in separation holds across all the models we test: encoder-based LLaVA and Molmo2 (Liu et al., 2024; Clark et al., 2026); encoder-free EVEv2 (Diao et al., 2025); multimodally pretrained Qwen3-VL and BAGEL[1] (Bai et al., 2025; Deng et al., 2025); and natively multimodal early-fusion Emu3 (Wang et al., 2024).

The remaining benchmarks show how this contrast extends beyond ScienceQA, ChartQA, and POPE. CV-Bench evaluates the spatial, counting, and depth judgments used to operationalize vision-centric tasks (Tong et al., 2024a). Its separation is weaker than ScienceQA and ChartQA in every model, with estimates close to zero or modestly positive in four models and reversed in EVEv2 and Emu3. MathVista, which combines mathematical reasoning with varied visual inputs, shows positive separation across all six models, although its magnitude varies substantially (Lu et al., 2024). HallusionBench exhibits little separation throughout, while MMStar and MMMU show more model-dependent patterns (Guan et al., 2023; Chen et al., 2024b; Yue et al., 2024).

Although established broad task categories describe what answering requires, they do not give convincing predictions about the relationship between visual representations and subsequent success. Instead, our incompressibility perspective directly concerns which distinctions a model must preserve to support the possible questions about an input. But whether a compact representation preserves those distinctions also depends on the abstractions the model has learned. Models may be trained to encode specific visual priors differently, making an input easier to represent adequately for one model than another. For example, Molmo2 receives explicit supervision for object pointing and counting, which may encourage representations that preserve individual object locations and multiplicities (Clark et al., 2026). This offers a possible interpretation of the model-dependent separation across benchmarks and motivates examining representational demands within them, rather than treating each benchmark as uniformly compressible or incompressible.

4.2 Separation across visual positions and decoder depth

The preceding subsection's comparisons use visual states from the final quarter of visual prefill. However, for ScienceQA and ChartQA, the divergence signal is already present within the first quarter of visual positions in all six models (Figure 4). Early-region Hedges' g ranges from 0.472

[1]Qwen3-VL and BAGEL use pretrained language model backbones, so they are not *natively* multimodal pretrained

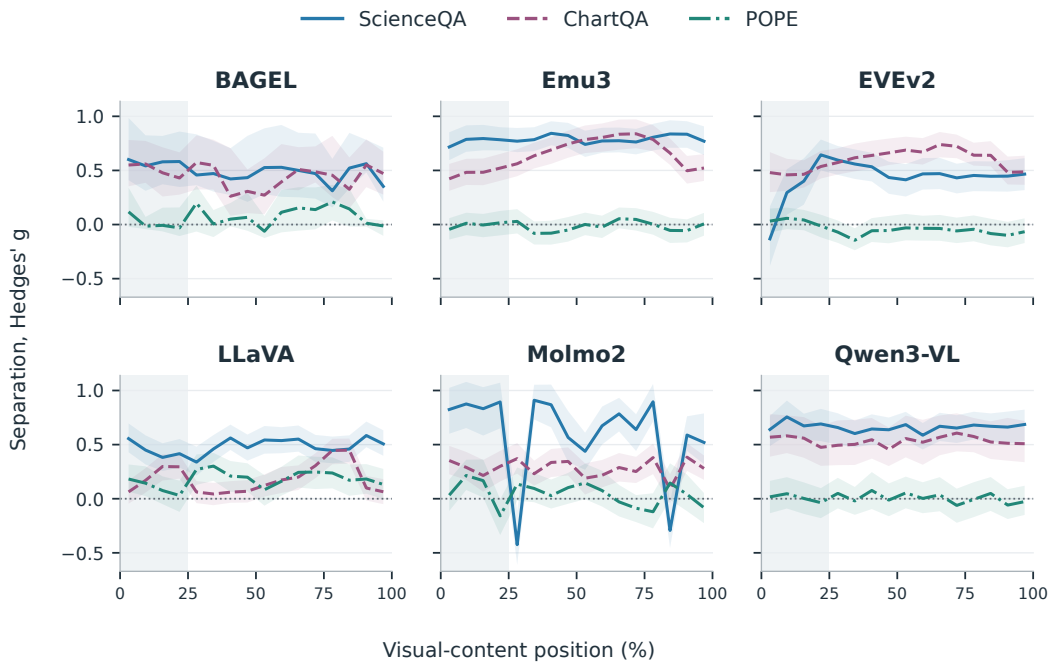

Figure 4: The ScienceQA and ChartQA signature is already visible in the first quarter of visual positions. Gray shading marks that quarter in the primary descriptive analysis. Position indexes the visual sequence. Bidirectional attention and visual encoding can provide image-wide context at early positions. At each bin, we average an input's angular distances across retained sampled depths and compute Hedges' g. Bands show pointwise 95% image-group intervals. Supplementary figures give per-position availability and raw distances.

to 0.987 on ScienceQA, from 0.219 to 0.581 on ChartQA, and from -0.004 to 0.154 on POPE. As described in Section 3, these positions can already incorporate information from across the image, depending on visual encoding and attention, so an early visual-token position does not necessarily imply limited image context. Nevertheless, the signal also appears in EVEv2, which on ScienceQA has $g = 0.611$, with a 95% interval of $[0.476, 0.735]$, within the early visual region. Notably, EVEv2 orders image patches from left to right across each row, proceeding from the top of the image to the bottom. Under causal attention, these states cannot incorporate later image patches or the subsequent question (Diao et al., 2025). Thus, for EVEv2, early-region divergence is visible in representations whose image content is restricted to approximately the upper quarter of the image.

We also examine whether the divergence signal persists across decoder depth or if it is concentrated in particular layers. Previous position comparisons average distances across sampled layers. However, the results still hold when measuring separation at each layer and visual position separately. Across all six models, ScienceQA and ChartQA exhibit separation over broad regions of decoder depth and visual position, while POPE generally shows weaker separation (Appendix B.2). The task-dependent signal is therefore visible in both shallow and deep decoder states, rather than arising from a narrow set of strongly separating layers.

Appendices C and B.5 examine sensitivity to the distance measure and to reference/evaluation duplicate-state exclusions.

5 Predicting Answer Errors from Visual Geometry

The separation between successful and unsuccessful inputs suggests that visual geometry may also help predict whether an individual input will lead to an error. We test this on Qwen3-VL with ScienceQA and ChartQA, where we observe consistent separation. For each input, we collect its angular distances to successful reference neighborhoods at five decoder depths in the final visual-

position bin. These five distances form the input to an MLP trained to predict whether the model will answer the question usually correctly or usually incorrectly, using the majority label from 15 sampled responses. Training and evaluation sets are separated by image group, and the reference neighborhoods remain fixed. We report AUROC to evaluate how well the predictor ranks usually incorrect inputs above usually correct inputs, without selecting a classification threshold.

We compare our method against HALP, which predicts answer errors from internal representations before generation (Kogilathota et al., 2026). Its visual-token method, HALP VT, trains an MLP on the decoder hidden state at the final visual token. This provides a comparison between predicting errors directly from hidden states and predicting them through their distances to successful references. We evaluate both methods on the same held-out ScienceQA and ChartQA inputs using the same correctness labels. For HALP, we select the decoder layer with the highest validation AUROC. For both methods, we average predicted probabilities across three training seeds. Appendix C.3 gives the training and evaluation details.

Distances to successful references carry predictive information about answer errors on both tasks (Table 1). Our geometry MLP achieves an AUROC of 0.792 on ScienceQA and 0.730 on ChartQA, compared with 0.789 and 0.680 for the adapted HALP baseline. These results suggest that proximity to successful reference neighborhoods captures information useful for error prediction, with observed performance similar to a probe trained directly on hidden states. The relationship between visual geometry and answering success therefore extends to individual held-out inputs. Before the question arrives, an image's resemblance to successfully represented examples already helps predict whether the model will answer correctly.

Table 1: Geometric summaries support held-out error prediction on both Qwen3-VL tasks. Entries give AUROC for the geometry MLP with 167,425 parameters and the adapted HALP baseline. Differences subtract the baseline from the geometry MLP, with paired 95% image-group intervals. The baseline uses archived image-end delimiter states rather than HALP's exact final visual token. Both intervals include zero.

Task	Geometry MLP	HALP adaptation	Difference [95% interval]
ScienceQA	0.792	0.789	0.003 $[-0.086, 0.092]$
ChartQA	0.730	0.680	0.051 $[-0.049, 0.152]$

We hold the evaluation inputs, correctness labels, and three-seed probability averaging fixed across methods. However, the methods differ in how they use representations and supervision. Our five distance features summarize states across decoder depths and incorporate additional labeled examples through the successful reference cloud, whereas the HALP adaptation reads a full hidden state at one validation-selected layer. The reference cloud therefore supplies exogenous supervision beyond the examples used to train the predictors, complicating a direct comparison with HALP. At the same time, our geometry MLP has 167,425 trainable parameters, compared with 2,263,809 for each HALP probe, approximately 13.5 times less. The results demonstrate predictive usefulness with a smaller classifier, while leaving open how much the additional reference supervision contributes. Appendix C.3 gives the configurations and supervision details.

Angular distance measures direction while discarding vector length. In Appendix C, we separately compare logistic predictors using angular distance, state magnitude, or both, finding that angular distance adds predictive information beyond magnitude on several model–task pairs.

6 Discussion

The question raised in the introduction concerns how a VLM's final answer reflects both how it represents an image and how it uses that representation to respond to a question. We give a geometric method which provides a way to examine that relationship during visual prefill. We hypothesize that if successful inputs exhibit recurring representational structure, a neighborhood of such representations with downstream success can be used to identify unusual representations. On ScienceQA and ChartQA, comparing a model's visual representations to the reference cloud distinguishes downstream success from failure across different VLM architectures. For POPE and CV-Bench however,

there is a much weaker separation. Despite nearly identical answer accuracy on ScienceQA and POPE, Qwen3-VL's visual representations relate differently to its eventual errors.

In Section 4.1, we suggest interpreting the task-dependent contrast through the information that a representation must preserve. Before seeing the question the model is intended to answer, the model must decide which distinctions to preserve for its subsequent use. A task requiring substantial knowledge or calculation may admit a compact representation of its visual premises, and conversely, a simple question about one element of a crowded image may require retaining details that a summary would discard. Incompressibility describes this difficulty relative to the questions an image could reasonably support. It also depends on the model's learned abstractions, since the same visual relationships may be easier for one model to encode compactly than another.

However, our methods and interpretations have a number of notable limitations. First, in this paper, we do not operationalize a measure of incompressibility. The examples given in Section 4.1 illustrate this hypothesis in a heuristic manner. But a controlled test could vary the number of independently relevant visual details while holding the question operation fixed, then ask whether the geometric signature strengthens as more information must be retained. The new GridVQA benchmark may be very suitable for this kind of controlled test, but we were not able to experiment with it in this paper (Belsare et al., 2026).

Furthermore, a reference cloud is not an infallible instrument for studying in/out-of-distribution behaviour (Sun et al., 2022). Selecting successful answers arbitrarily does not control for familiarity from training, language priors, or guessing, which all can also contribute to success. Conversely, sampling successful answers is not a perfect operation to find all relevant representational directions. A sufficient representation may lie far from the cloud because its visual content or organization is poorly represented among the selected references. The interpretation of distance then changes based on the biases in the reference cloud. Identical visual states can also precede different outcomes for different questions about the same image. We explore this further in Appendix B.1, and find weak and inconsistent correlations within mixed-outcome inputs, which further limits a general interpretation of the relationship between distance and difficulty.

These limitations leave questions about the causal relationships of information represented by distant states. Our geometric comparisons remain entirely non-causal, and do not give an intervention that tests whether changing the representation changes the answer. In a manner similar to *Hidden in Plain Sight*, for a chart, are its label-value associations still recoverable when the model answers incorrectly (Fu et al., 2025)? A readout that recovers them would support a failure to use available information, but a failed readout does not automatically give the converse. The failed readout could instead reflect limitations of the readout—whether it be due to non-linearity, dimensionality, or something else—so establishing insufficiency requires further evidence.

AI USE STATEMENT

In this work, we used generative AI tools for proposing or refining hypotheses, designing or providing feedback on research methodology or experiments, implementing methods, supporting qualitative and thematic data analysis, and interpreting results. We have not used generative AI tools for assisting with translation nor cleaning and reformatting datasets. Generating synthetic data sets, helping develop theoretical models or conceptual frameworks, formulating mathematical claims, providing critical ingredients for proving mathematical claims, and assisting in the writing of proofs are not applicable to this work. Additionally, we used generative AI tools for creating or editing software code, drafting parts of a research paper, summarizing or analysing existing literature, discovering research topics or identifying gaps, brainstorming, sourcing/searching for information, editing a research paper to improve readability, and identifying relevant literature. We have reviewed all AI-assisted work. The final resolutions for all ideas, drafts, theses, literature review/positioning were done entirely at the manual discretion of the authors. LLM-generated code was tested for correctness with Hypothesis (MacIver et al., 2019), and verified by 3 authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Amos Azaria and Tom Mitchell. The internal state of an llm knows when it's lying. (arXiv:2304.13734), 2023. doi: 10.48550/arXiv.2304.13734. URL https://arxiv.org/abs/2304.13734.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, Mei Li, Kaixin Li, Zicheng Lin, Junyang Lin, Xuejing Liu, Jiawei Liu, Chenglong Liu, Yang Liu, Dayiheng Liu, Shixuan Liu, Dunjie Lu, Ruilin Luo, Chenxu Lv, Rui Men, Lingchen Meng, Xuancheng Ren, Xingzhang Ren, Sibo Song, Yuchong Sun, Jun Tang, Jianhong Tu, Jianqiang Wan, Peng Wang, Pengfei Wang, Qiuyue Wang, Yuxuan Wang, Tianbao Xie, Yiheng Xu, Haiyang Xu, Jin Xu, Zhibo Yang, Mingkun Yang, Jianxin Yang, An Yang, Bowen Yu, Fei Zhang, Hang Zhang, Xi Zhang, Bo Zheng, Humen Zhong, Jingren Zhou, Fan Zhou, Jing Zhou, Yuanzhi Zhu, and Ke Zhu. Qwen3-vl technical report. (arXiv:2511.21631), 2025. doi: 10.48550/arXiv.2511.21631. URL http://arxiv.org/abs/2511.21631.

Mohamed El Banani, Amit Raj, Kevis-Kokitsi Maninis, Abhishek Kar, Yuanzhen Li, Michael Rubinstein, Deqing Sun, Leonidas Guibas, Justin Johnson, and Varun Jampani. Probing the 3d awareness of visual foundation models. (arXiv:2404.08636), 2024. doi: 10.48550/arXiv.2404.08636. URL http://arxiv.org/abs/2404.08636.

Sujay Belsare, Sudarshan Nikhil, Sushant Kumar, Ponnurangam Kumaraguru, and Chirag Agarwal. Gridvqa-x: A framework for evaluating multimodal explainability methods. (arXiv:2606.14740), 2026. doi: 10.48550/arXiv.2606.14740. URL http://arxiv.org/abs/2606.14740.

Fanpu Cao, Xin Zou, Xuming Hu, and Hui Xiong. When looking is not enough: Visual attention structure reveals hallucination in mllms. (arXiv:2605.11559), 2026. doi: 10.48550/arXiv.2605.11559. URL http://arxiv.org/abs/2605.11559.

Chao Chen, Kai Liu, Ze Chen, Yi Gu, Yue Wu, Mingyuan Tao, Zhihang Fu, and Jieping Ye. Inside: Llms' internal states retain the power of hallucination detection. (arXiv:2402.03744), 2024a. doi: 10.48550/arXiv.2402.03744. URL http://arxiv.org/abs/2402.03744.

Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Jiaqi Wang, Yu Qiao, Dahua Lin, and Feng Zhao. Are we on the right way for evaluating large vision-language models? (arXiv:2403.20330), 2024b. doi: 10.48550/arXiv.2403.20330. URL https://arxiv.org/abs/2403.20330.

Seonglae Cho, Zekun Wu, Kleyton Da Costa, and Adriano Koshiyama. The confidence manifold: Geometric structure of correctness representations in language models. (arXiv:2602.08159), 2026. doi: 10.48550/arXiv.2602.08159. URL https://arxiv.org/abs/2602.08159.

Christopher Clark, Jieyu Zhang, Zixian Ma, Jae Sung Park, Mohammadreza Salehi, Rohun Tripathi, Sangho Lee, Zhongzheng Ren, Chris Dongjoo Kim, Yinuo Yang, Vincent Shao, Yue Yang, Weikai Huang, Ziqi Gao, Taira Anderson, Jianrui Zhang, Jitesh Jain, George Stoica, Winson Han, Ali Farhadi, and Ranjay Krishna. Molmo2: Open weights and data for vision-language models with video understanding and grounding. (arXiv:2601.10611), 2026. doi: 10.48550/arXiv.2601.10611. URL https://arxiv.org/abs/2601.10611.

Chaorui Deng, Deyao Zhu, Kunchang Li, Chenhui Gou, Feng Li, Zeyu Wang, Shu Zhong, Weihao Yu, Xiaonan Nie, Ziang Song, Guang Shi, and Haoqi Fan. Emerging properties in unified multimodal pretraining. (arXiv:2505.14683), 2025. doi: 10.48550/arXiv.2505.14683. URL https://arxiv.org/abs/2505.14683.

Haiwen Diao, Yufeng Cui, Xiaotong Li, Yueze Wang, Huchuan Lu, and Xinlong Wang. Unveiling encoder-free vision-language models. (arXiv:2406.11832), 2024. doi: 10.48550/arXiv.2406.11832. URL http://arxiv.org/abs/2406.11832.

Haiwen Diao, Xiaotong Li, Yufeng Cui, Yueze Wang, Haoge Deng, Ting Pan, Wenxuan Wang, Huchuan Lu, and Xinlong Wang. Evev2: Improved baselines for encoder-free vision-language models. (arXiv:2502.06788), 2025. doi: 10.48550/arXiv.2502.06788. URL https://arxiv.org/abs/2502.06788.

Xuefeng Du, Chaowei Xiao, and Yixuan Li. Haloscope: Harnessing unlabeled llm generations for hallucination detection. (arXiv:2409.17504), 2024. doi: 10.48550/arXiv.2409.17504. URL https://arxiv.org/abs/2409.17504.

Stephanie Fu, Tyler Bonnen, Devin Guillory, and Trevor Darrell. Hidden in plain sight: Vlms overlook their visual representations. (arXiv:2506.08008), 2025. doi: 10.48550/arXiv.2506.08008. URL https://arxiv.org/abs/2506.08008.

Tianrui Guan, Fuxiao Liu, Xiyang Wu, Ruiqi Xian, Zongxia Li, Xiaoyu Liu, Xijun Wang, Lichang Chen, Furong Huang, Yaser Yacoob, Dinesh Manocha, and Tianyi Zhou. Hallusionbench: An advanced diagnostic suite for entangled language hallucination and visual illusion in large vision-language models, 2023. URL https://arxiv.org/abs/2310.14566v5.

Anisha Gunjal, Jihan Yin, and Erhan Bas. Detecting and preventing hallucinations in large vision language models. (arXiv:2308.06394), 2024. doi: 10.48550/arXiv.2308.06394. URL https://arxiv.org/abs/2308.06394.

Larry V. Hedges. Distribution theory for glass's estimator of effect size and related estimators. *Journal of Educational Statistics*, 6(2):107–128, 1981. doi: 10.2307/1164588. URL https://www.jstor.org/stable/1164588.

Heinrich Jiang, Been Kim, Melody Y. Guan, and Maya Gupta. To trust or not to trust a classifier. (arXiv:1805.11783), 2018. doi: 10.48550/arXiv.1805.11783. URL http://arxiv.org/abs/1805.11783.

Zhangqi Jiang, Junkai Chen, Beier Zhu, Tingjin Luo, Yankun Shen, and Xu Yang. Devils in middle layers of large vision-language models: Interpreting, detecting and mitigating object hallucinations via attention lens. (arXiv:2411.16724), 2025. doi: 10.48550/arXiv.2411.16724. URL http://arxiv.org/abs/2411.16724.

Ryo Kamoi, Yusen Zhang, Sarkar Snigdha Sarathi Das, Ranran Haoran Zhang, and Rui Zhang. Visonlyqa: Large vision language models still struggle with visual perception of geometric information. (arXiv:2412.00947), 2025. doi: 10.48550/arXiv.2412.00947. URL http://arxiv.org/abs/2412.00947.

Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. Generalization through memorization: Nearest neighbor language models. (arXiv:1911.00172), 2020. doi: 10.48550/arXiv.1911.00172. URL http://arxiv.org/abs/1911.00172.

Sai Akhil Kogilathota, Sripadha Vallabha E. G, Luzhe Sun, and Jiawei Zhou. Halp: Detecting hallucinations in vision-language models without generating a single token. (arXiv:2603.05465), 2026. doi: 10.48550/arXiv.2603.05465. URL https://arxiv.org/abs/2603.05465.

Elizaveta Levina and Peter Bickel. Maximum likelihood estimation of intrinsic dimension. In *Advances in Neural Information Processing Systems*, volume 17. MIT Press, 2004. URL https://papers.nips.cc/paper_files/paper/2004/hash/74934548253bcab8490ebd74afed7031-Abstract.html.

Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne Xin Zhao, and Ji-Rong Wen. Evaluating object hallucination in large vision-language models. (arXiv:2305.10355), 2023. doi: 10.48550/arXiv.2305.10355. URL https://arxiv.org/abs/2305.10355.

Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. Llava-next: Improved reasoning, ocr, and world knowledge, 2024. URL https://llava-vl.github.io/blog/2024-01-30-llava-next/.

Junteng Liu, Weihao Zeng, Xiwen Zhang, Yijun Wang, Zifei Shan, and Junxian He. On the perception bottleneck of vlms for chart understanding. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 10829–10841, Suzhou, China, 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.findings-emnlp.573. URL https://aclanthology.org/2025.findings-emnlp.573/.

Pan Lu, Swaroop Mishra, Tony Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering. (arXiv:2209.09513), 2022. doi: 10.48550/arXiv.2209.09513. URL http://arxiv.org/abs/2209.09513.

Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts. (arXiv:2310.02255), 2024. doi: 10.48550/arXiv.2310.02255. URL http://arxiv.org/abs/2310.02255.

David R. MacIver, Zac Hatfield-Dodds, and Many Other Contributors. Hypothesis: A new approach to property-based testing. *Journal of Open Source Software*, 4(43):1891, 2019. doi: 10.21105/joss.01891. URL https://joss.theoj.org/papers/10.21105/joss.01891.

Samuel Marks and Max Tegmark. The geometry of truth: Emergent linear structure in large language model representations of true/false datasets. (arXiv:2310.06824), 2024. doi: 10.48550/arXiv.2310.06824. URL https://arxiv.org/abs/2310.06824.

Ahmed Masry, Do Xuan Long, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. Chartqa: A benchmark for question answering about charts with visual and logical reasoning. (arXiv:2203.10244), 2022. doi: 10.48550/arXiv.2203.10244. URL https://arxiv.org/abs/2203.10244.

Cheolhong Min, Jaeyun Jung, Daeun Lee, Hyeonseong Jeon, Yu Su, Jonathan Tremblay, Chan Hee Song, and Jaesik Park. Why far looks up: Probing spatial representation in vision-language models. (arXiv:2605.30161), 2026. doi: 10.48550/arXiv.2605.30161. URL http://arxiv.org/abs/2605.30161.

Ryuhei Miyazato, Shunsuke Kitada, and Kei Harada. Ensemhaldet: Robust vlm hallucination detection via ensemble of internal state detectors. (arXiv:2604.02784), 2026. doi: 10.48550/arXiv.2604.02784. URL https://arxiv.org/abs/2604.02784.

Alexander Modell, Patrick Rubin-Delanchy, and Nick Whiteley. The origins of representation manifolds in large language models. (arXiv:2505.18235), 2025. doi: 10.48550/arXiv.2505.18235. URL http://arxiv.org/abs/2505.18235.

Nicolas Papernot and Patrick McDaniel. Deep k-nearest neighbors: Towards confident, interpretable and robust deep learning. (arXiv:1803.04765), 2018. doi: 10.48550/arXiv.1803.04765. URL http://arxiv.org/abs/1803.04765.

Eunkyu Park, Minyeong Kim, and Gunhee Kim. Halloc: Token-level localization of hallucinations for vision language models. (arXiv:2506.10286), 2025. doi: 10.48550/arXiv.2506.10286. URL https://arxiv.org/abs/2506.10286.

Kiho Park, Yo Joong Choe, and Victor Veitch. The linear representation hypothesis and the geometry of large language models. (arXiv:2311.03658), 2024. doi: 10.48550/arXiv.2311.03658. URL http://arxiv.org/abs/2311.03658.

Jie Ren, Jiaming Luo, Yao Zhao, Kundan Krishna, Mohammad Saleh, Balaji Lakshminarayanan, and Peter J. Liu. Out-of-distribution detection and selective generation for conditional language models. (arXiv:2209.15558), 2023. doi: 10.48550/arXiv.2209.15558. URL http://arxiv.org/abs/2209.15558.

Gaurang Sriramanan, Siddhant Bharti, Vinu Sankar Sadasivan, Shoumik Saha, Priyatham Kattakinda, and Soheil Feizi. Llm-check: Investigating detection of hallucinations in large language models. In *Advances in Neural Information Processing Systems*, volume 37, pp. 34188–34216. Curran Associates, Inc., 2024. doi: 10.52202/079017-1077. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/3c1e1fdf305195cd620c118aaa9717ad-Abstract-Conference.html.

Yiyou Sun, Yifei Ming, Xiaojin Zhu, and Yixuan Li. Out-of-distribution detection with deep nearest neighbors. (arXiv:2204.06507), 2022. doi: 10.48550/arXiv.2204.06507. URL https://arxiv.org/abs/2204.06507.

Shengbang Tong, Ellis Brown, Penghao Wu, Sanghyun Woo, Manoj Middepogu, Sai Charitha Akula, Jihan Yang, Shusheng Yang, Adithya Iyer, Xichen Pan, Ziteng Wang, Rob Fergus, Yann LeCun, and Saining Xie. Cambrian-1: A fully open, vision-centric exploration of multimodal llms. (arXiv:2406.16860), 2024a. doi: 10.48550/arXiv.2406.16860. URL https://arxiv.org/abs/2406.16860.

Shengbang Tong, Zhuang Liu, Yuexiang Zhai, Yi Ma, Yann LeCun, and Saining Xie. Eyes wide shut? exploring the visual shortcomings of multimodal llms, 2024b. URL https://arxiv.org/abs/2401.06209v2.

Bodla Krishna Vamshi, Rohan Bhatnagar, and Haizhao Yang. Manifold-based sampling for in-context hallucination detection in large language models. (arXiv:2601.06196), 2026. doi: 10.48550/arXiv.2601.06196. URL http://arxiv.org/abs/2601.06196.

Khang Nhat Hoang Vo, Artem Vazhentsev, Artem Shelmanov, Timothy Baldwin, and Yova Kementchedjhieva. Failing to see or failing to know? attributing errors in vision-language models. (arXiv:2607.04683), 2026. doi: 10.48550/arXiv.2607.04683. URL http://arxiv.org/abs/2607.04683.

Xinhe Wang, Jin Huang, Xingjian Zhang, Tianhao Wang, and Jiaqi W. Ma. Your reasoning benchmark may not test reasoning: Revealing perception bottleneck in abstract reasoning benchmarks. (arXiv:2512.21329), 2026. doi: 10.48550/arXiv.2512.21329. URL https://arxiv.org/abs/2512.21329.

Xinlong Wang, Xiaosong Zhang, Zhengxiong Luo, Quan Sun, Yufeng Cui, Jinsheng Wang, Fan Zhang, Yueze Wang, Zhen Li, Qiying Yu, Yingli Zhao, Yulong Ao, Xuebin Min, Tao Li, Boya Wu, Bo Zhao, Bowen Zhang, Liangdong Wang, Guang Liu, Zheqi He, Xi Yang, Jingjing Liu, Yonghua Lin, Tiejun Huang, and Zhongyuan Wang. Emu3: Next-token prediction is all you need. (arXiv:2409.18869), 2024. doi: 10.48550/arXiv.2409.18869. URL https://arxiv.org/abs/2409.18869.

Fan Yin, Jayanth Srinivasa, and Kai-Wei Chang. Characterizing truthfulness in large language model generations with local intrinsic dimension. (arXiv:2402.18048), 2024. doi: 10.48550/arXiv.2402.18048. URL http://arxiv.org/abs/2402.18048.

Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, Cong Wei, Botao Yu, Ruibin Yuan, Renliang Sun, Ming Yin, Boyuan Zheng, Zhenzhu Yang, Yibo Liu, Wenhao Huang, Huan Sun, Yu Su, and Wenhu Chen. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. (arXiv:2311.16502), 2024. doi: 10.48550/arXiv.2311.16502. URL https://arxiv.org/abs/2311.16502.

Feiran Zhang, Yixin Wu, Zhenghua Wang, Xiaohua Wang, Changze Lv, Xuanjing Huang, and Xiaoqing Zheng. Vib-probe: Detecting and mitigating hallucinations in vision-language models via variational information bottleneck. (arXiv:2601.05547), 2026. doi: 10.48550/arXiv.2601.05547. URL https://arxiv.org/abs/2601.05547.

Renrui Zhang, Dongzhi Jiang, Yichi Zhang, Haokun Lin, Ziyu Guo, Pengshuo Qiu, Aojun Zhou, Pan Lu, Kai-Wei Chang, Peng Gao, and Hongsheng Li. Mathverse: Does your multi-modal llm truly see the diagrams in visual math problems? (arXiv:2403.14624), 2024. doi: 10.48550/arXiv.2403.14624. URL http://arxiv.org/abs/2403.14624.

Runyu Zhou, Qi Zhang, Qixun Wang, and Yisen Wang. Diagnosing visual ignorance in vision-language models. (arXiv:2606.06890), 2026. doi: 10.48550/arXiv.2606.06890. URL http://arxiv.org/abs/2606.06890.

A Experimental protocol

A.1 Models, inputs, and repeated responses

We compare six models using their supplied image processing and prompt templates, with the checkpoints and decoder dimensions listed in Table 2. LLaVA, Molmo2, and Qwen3-VL use visual encoders, as does BAGEL's image-understanding path, which also uses a connector. Emu3 instead represents images through discrete visual tokens, while EVEv2 uses a shallow visual tokenizer. Because the models also differ in multimodal pretraining, these comparisons cannot isolate architecture under a shared training procedure.

Table 2: Model checkpoints and decoder dimensions. Revision prefixes identify the weights used for collection.

Model	Layers	Width	Checkpoint	Revision
BAGEL (Deng et al., 2025)	28	3584	BAGEL-7B-MoT	`5019f57d`
Emu3 (Wang et al., 2024)	32	4096	Emu3-Chat	`d9cb6ffe`
EVEv2 (Diao et al., 2025)	28	3584	EVE-7B-HD-v2.0	`a6892aee`
LLaVA (Liu et al., 2024)	32	4096	v1.6-Mistral-7B	`2424fdd4`
Molmo2 (Clark et al., 2026)	36	4096	molmo2-8b	`e28fa285`
Qwen3-VL (Bai et al., 2025)	36	4096	Qwen3-VL-8B-Instruct	`0c351dd0`

Image and question context. As discussed in Section 3, the image information available at a visual position depends on both the initial image processing and the decoder's attention mask. BAGEL and Molmo2 allow bidirectional attention within image blocks, while the other four models use causal decoder attention. Even under causal decoder attention, an early state can contain image-wide information already mixed by a visual encoder. In EVEv2, by contrast, the tokenizer uses local convolutions and orders patches across successive image rows, restricting early decoder states to the corresponding image prefix.

Despite these differences in image context, both attention masks prevent visual states from attending to later question text on image-first inputs. MMMU can instead interleave images and question text, so a visual state may have access to text preceding its image.

Benchmarks and responses. For each benchmark, we separate inputs used to construct successful reference clouds from those used to evaluate distance and correctness. Table 3 gives the source data, evaluation-set sizes, and reference-cloud sizes. Each model's evaluation set contains 9,992 image-question pairs across the eight benchmarks, giving 59,952 question-model inputs across the six models and a total of 899,280 sampled responses.

For Emu3 on MMMU, we use a replacement collection in which states were recorded consistently, excluding all earlier captures from the reported results. The earlier collection combined three different capture procedures and representation spaces, which prevented us from pooling its states into a common reference cloud.

To measure how consistently a model answers each input, we sample 15 responses at temperature 0.7, allowing up to 128 tokens per response unless the model produces its end token earlier. For BAGEL, a beginning-of-sequence token counts towards this limit, although we remove it from the stored response. We otherwise use the generation settings supplied with the recorded model revision. These settings apply to the descriptive experiments, whereas the intervention experiments use the separate prompts and generation limits described in Appendix A.3.

Answer evaluation. We evaluate each response by extracting an answer and comparing it with the benchmark's gold answer using deterministic rules. We extract the selected option for multiple-choice questions and a yes or no answer for binary questions, counting a response as incorrect if extraction fails.

For numerical ChartQA answers, we allow a relative error of up to 5%, except when the gold answer is zero, which requires an exact match. Other free-form numerical answers are evaluated at the

Table 3: Evaluation sets and reference-cloud sizes for each benchmark. Questions that use the same ordered images belong to one group. Each reference cloud contains the listed number of distinct groups at each layer and position, for each reference resample. Source split names identify the benchmark data from which we form our reference and evaluation partitions. Revision prefixes identify the dataset versions.

Dataset	Source split	Questions	Groups	References	Revision
ScienceQA	Image validation	1,035	926	373	`69dd4d6b`
ChartQA	Test	1,211	740	152	`9e63b7df`
MathVista	Test-mini	487	479	36	`2b6ad694`
POPE	Test	4,302	239	72	`4db12766`
HallusionBench	Image	443	166	31	`cd417161`
CV-Bench	Test, 2D and 3D	1,316	1,285	512	`bc284db5`
MMStar	Validation	740	706	200	`bc98d668`
MMMU	Validation	458	443	115	`98e6ac0c`

precision specified by the example, while text answers, including nonnumerical ChartQA answers, are compared for equality after normalization.

After scoring all 15 responses, we label an input usually correct if at least eight are correct, and usually incorrect otherwise.

For MathVista, these labels incorporate a correction to an earlier error in mapping options to gold answers. Because MathVista stores the content of the correct option, we match that content to the selected option instead of interpreting the gold answer as an option index. We use the corrected labels both to select reference clouds and to group inputs in all reported MathVista distance comparisons.

A.2 Reference geometry and uncertainty

State aggregation. Because the number of visual tokens varies across inputs and models, we compare states at relative positions within each visual sequence. We divide the sequence into 16 position bins and, at each of 12 sampled decoder layers, average the visual-content states within each bin. The resulting vector is then compared with reference vectors from the same model, dataset, layer, and bin.

To assign a token to a bin, we divide its native visual index by one less than the visual-token count, using a denominator of at least one. The resulting relative position determines which of 16 equal bins contains the token, with the bin index capped at 15. Structural tokens such as separators retain their native positions and contribute to this token count, but token-role annotations exclude their states from the content average. We accumulate the bin means in float32 before storing them in the original capture precision.

We similarly divide each model's decoder depth into 12 equally spaced bins, selecting the layer nearest each bin's centre and choosing the shallower layer in a tie.

Grouping and reference selection. Different questions about the same image can share visual states, so we keep them together when dividing inputs between the reference and evaluation partitions. We identify each image group by hashing its normalized images and retaining the ordered sequence of hashes, which also distinguishes different image orders in multi-image questions. A deterministic hash split with seed 2401 then assigns each group to the reference or evaluation partition with equal probability.

Within the reference partition, we select states from usually correct inputs, excluding vectors associated with conflicting majority labels because their relationship to successful answering is ambiguous. We also exclude identical vectors shared by the reference and evaluation partitions without consulting evaluation labels. Within the evaluation partition itself, questions retain their separate outcomes even when their visual states are identical, allowing us to examine how far visual states alone distinguish question outcomes, as discussed in Section 3.

From the remaining candidates, deterministic bipartite matching selects at most one state per image group and one copy of any state vector, separately for each model, dataset, layer, and position bin. Within each dataset, we match cloud sizes to the smallest feasible size across the compatible model and layer-position combinations, while allowing the selected images to differ. We cap this size at 512 and require at least 15 references, giving the dataset-specific sizes in Table 3.

A particular selection of references need not represent all eligible successful inputs equally well, so we repeat reference selection 16 times using seeds 240100 through 240115.

Angular distance. After normalizing each evaluation vector and its matching references to unit length, we clip their dot products to $[-1, 1]$ and take the arccosine to obtain angles in radians. We take the mean angle to the 15 nearest references within each cloud, then average over the 16 reference selections to obtain the reported distance. Here, 15 neighbours determines the size of the local geometric comparison, independently of the choice to sample 15 answers when assigning correctness labels.

We exclude zero-norm states, which cannot be normalized, as well as measurements with fewer than 15 eligible references. After these exclusions, we summarize each input's early region by averaging its available distances across sampled layers and bins 0–3, and its late region using bins 12–15. Computing the distances before averaging across layers keeps each state's comparison within its own layer's reference space.

The descriptive comparisons require an input to have a distance in both the early and late regions, a requirement met by 59,591 of the 59,952 question-model inputs. The supplementary distribution figures report the usually correct and usually incorrect counts for each model and dataset, while Appendix B.5 examines how these exclusions affect which inputs contribute to EVEv2's ScienceQA results.

Effect sizes and intervals. To obtain Hedges' g, we subtract the usually correct group's mean distance from the usually incorrect group's mean, divide by their pooled within-group standard deviation, and apply the small-sample correction (Hedges, 1981).

To quantify how the estimate depends on the evaluated images, we form 500 bootstrap datasets by drawing the original number of image groups with replacement. Each selected group contributes all its questions, including repeated copies when a group is selected more than once, preserving the dependence among questions about the same image. We recompute the statistic on each dataset and take the 2.5th and 97.5th percentiles of the defined estimates as the bounds of a 95% confidence interval.

With the fitted reference clouds and observed response labels held fixed, these intervals describe uncertainty from the evaluated images rather than from selecting references or sampling new responses. For curves across visual positions, the intervals apply separately to each position and provide neither a simultaneous confidence band nor an exact onset of failure.

Choice of distance. Before choosing angular kNN, we considered Euclidean kNN, local intrinsic dimensionality, PCA-subspace displacement, local PCA residuals, and local outlier factor. Unlike a proximity measure, local intrinsic dimensionality estimates the dimensions needed to describe a neighbourhood, with one common estimator using ratios of neighbour distances,

$$\widehat{m} = \left[\frac{1}{k-1} \sum_{j=1}^{k-1} \log \frac{r_k}{r_j} \right]^{-1},$$

where r_j is the distance to the jth neighbour (Levina & Bickel, 2004; Yin et al., 2024). The estimator is sensitive to repeated states, which give zero distances and undefined logarithmic ratios, as well as to neighbours whose distances are nearly equal. In the latter case, the ratios approach one and their average logarithm approaches zero, so taking its reciprocal can produce a large and unstable estimate. These difficulties motivated measuring proximity directly. To examine whether the task contrast depends on how that proximity is defined, Appendix C compares angular kNN, whitened angular kNN, centroid angle, and diagonal Mahalanobis displacement on matched inputs.

A.3 GEOMETRY-CHANGING INTERVENTIONS

The relationship between visual geometry and later correctness raises a further question of whether moving a representation towards successful states changes the model's answer. We explored this through a reconstruction of a historical steering procedure, followed by an experiment in which the direction was fitted on separate training inputs. We then tested whether an intervention that directly reduces angular distance would affect correctness. Each experiment uses its own input set and generation settings, as described below.

Reconstructing the historical steering experiment. The original execution logs and input set are unavailable, so we reconstruct the historical LLaVA intervention from its saved implementation on a fresh set of 352 ScienceQA inputs. Using a prompt that requests step-by-step reasoning followed by an answer letter, the unmodified model answers 178 inputs correctly under greedy decoding and the historical answer scorer.

We use these outcomes to construct a direction from unsuccessful towards successful states, taking the final prompt-token states from the output of decoder block 18, counting blocks from zero. These states are stored in float16 and remain unnormalized when we compute each correctness group's mean in float32. All 352 inputs contribute to the means, and subtracting the incorrect group's mean from the correct group's mean gives the steering direction.

We evaluate steering on the first 120 incorrect inputs in a fixed order, all of which also contributed to the fit, so fitting and evaluation overlap. Both conditions use the same prompt-end direction, scaled by the specified multiplier before any edit. The visual-end condition adds it at the last visual-content token, while the prompt-end condition adds it at the final token of the fully rendered prompt. Both continue to add the scaled direction on every subsequent cached decoding call.

Answer-parser sensitivity. Because the evaluation inputs were selected using the historical scorer, all 120 unmodified responses count as incorrect under that scorer. The current parser instead scores nine as correct, leaving 111 baseline errors eligible for recovery and nine successes eligible for degradation.

At multipliers 1, 2, and 4, visual-end steering recovers 10, 14, and 20 of the 111 errors, respectively, while degrading 0, 0, and 1 of the nine successes. Prompt-end steering recovers 21, 19, and 22 errors and degrades one success at each multiplier. Although accuracy improves under either parser, both the number of baseline errors and the estimated size of the improvement depend on the scoring rule. The historical scorer also leaves the degradation rate undefined because it identifies no correct baseline responses to degrade. A semantic review of the changed answers would be needed to distinguish improved reasoning from changes that make an answer easier to parse.

Train-only short-answer follow-up. To separate fitting from evaluation, we conducted a short-answer follow-up in which the direction was fitted on 256 training image groups and evaluated on 64 other groups, although those evaluation inputs had appeared in earlier exploration. We fix the multiplier at 4 and intervene at decoder block 18. Sampling 15 responses per evaluation group at temperature 0.7 with a 256-token limit gives 960 responses in each condition.

Each intervention response is paired with a response from the unmodified model for the same input and sample index, using the same deterministic generation seed and resetting the random-number generators before generation. Recovery and degradation describe changes within these pairs, with recovery counted when an incorrect baseline response becomes correct and degradation when a correct response becomes incorrect.

Of the 960 baseline responses, 578 are correct and 382 are incorrect. Steering at visual end and during decoding recovers 4 errors and degrades 2 successes, giving a net gain of 0.21 percentage points with a pointwise 95% image-group bootstrap interval of $[-0.31, 0.83]$. Steering at prompt end and during decoding changes more answers in both directions, recovering 102 errors and degrading 84 successes. The difference between these recoveries and degradations gives a net gain of 1.88 percentage points, with interval $[-5.32, 8.44]$.

To examine which parts of the intervention affect answering, Table 4 includes conditions that change its timing and conditions that change its geometry. The timing conditions distinguish edits made

during prefill, during decoding, or during both, with the prefill edits applied at visual end, at prompt end, or across all visual positions.

To distinguish direction from magnitude, the angular control keeps the edited state's direction while restoring its original norm, whereas the radial control keeps the original direction and adopts the edited norm. The random-direction controls test specificity by choosing a different direction while matching the primary edit's angular displacement and target norm for each state. The reversed-direction control instead subtracts the fitted direction. All these geometric controls apply at visual end and during decoding. Across the full set of ten conditions in Table 4, we estimate intervals from 1,000 resamples of whole image groups while keeping paired responses fixed.

Table 4: Visual-end steering changes few answers in the short-answer follow-up. Recoveries are counted among 382 baseline errors and degradations among 578 baseline successes. Net accuracy changes and their pointwise 95% intervals are expressed in percentage points over all 960 responses. The direction is fitted on separate training groups.

Intervention	Recovered	Degraded	Net change [95% interval]
Visual end and decoding	4	2	$0.21\ [-0.31, 0.83]$
Visual end, prefill only	4	2	$0.21\ [-0.31, 0.83]$
Decoding only	0	0	$0.00\ [0.00, 0.00]$
All visual positions, prefill only	4	3	$0.10\ [-0.52, 0.62]$
Prompt end and decoding	102	84	$1.88\ [-5.32, 8.44]$
Angular component	3	1	$0.21\ [-0.21, 0.73]$
Radial component	2	0	$0.21\ [0.00, 0.52]$
Random direction, seed 24101	0	1	$-0.10\ [-0.31, 0.00]$
Random direction, seed 24102	3	1	$0.21\ [-0.21, 0.62]$
Reversed direction	6	2	$0.42\ [-0.10, 1.04]$

Direct angular descent. The mean-difference direction used above is derived from correctness groups, but it does not directly optimize the angular distance used in our descriptive analysis. We therefore also tested an intervention on Qwen3-VL with ScienceQA that reduces this distance directly, using a separate reference bank constructed from successful states in the study's training inputs. The edit acts during prefill at decoder block 21 and visual-position bin 3, both numbered from zero.

We use the gradient of the selected bin mean's angular kNN distance to choose how to rotate its visual-content states while preserving each state's norm. The maximum per-token angular changes are 0.02, 0.05, and 0.1 radians, with individual steps scaled by their tangent-gradient magnitudes relative to the largest magnitude in the bin. These limits specify how far a state may rotate, rather than how much its distance to the references must decrease.

The experiment includes 64 inputs with 15 responses each, of which 160 are incorrect and 800 are correct before intervention. Despite reducing the targeted distance for every input, all three conditions recover zero errors and degrade zero successes.

B SUPPLEMENTARY RESULTS

Unless stated otherwise, we use the main analysis population and majority-correctness labels, requiring each input to have a distance in both the early and late visual regions. An input contributes at an individual visual position only if its distance is available there. Following Appendix A, we estimate pointwise confidence intervals from 500 resamples of whole image groups, keeping observed response counts and reference clouds fixed.

B.1 DISTANCE TRACKS REPEATED-RESPONSE SUCCESS UNEVENLY ACROSS TASKS

The majority label discards differences in how consistently a model succeeds, giving the same label to an input with eight correct responses and one with 15. To examine this variation, we calculate the Spearman correlation between visual distance and the number of correct responses separately for each model and dataset. Negative values indicate that inputs farther from successful references tend to be answered correctly less often.

When we include all analysed inputs, the late-region correlations are negative in every model on ScienceQA and ChartQA, ranging from -0.382 to -0.161 and from -0.386 to -0.160, respectively. POPE, in contrast, has much weaker associations, ranging from -0.044 to 0.022. The difference between these tasks therefore also appears when correctness is represented by the number of successful responses rather than the majority label alone, as shown for the early and late visual regions in Figures 6 and 7.

We also repeat the calculation on mixed-outcome inputs, defined by one through 14 correct responses and comprising 16,292 of the 59,591 analysed inputs. Within this subset, the relationship between distance and success frequency is much less consistent across models, with 14 of the 18 late-region confidence intervals for ScienceQA, ChartQA, and POPE including zero.

Of the remaining four intervals, three lie entirely below zero. BAGEL has a correlation of -0.254 on ChartQA, with interval $[-0.385, -0.113]$. LLaVA also has negative correlations on ChartQA, at -0.141 with interval $[-0.210, -0.069]$, and on POPE, at -0.073 with interval $[-0.132, -0.006]$. EVEv2 instead shows a positive relationship among its 64 mixed-outcome ScienceQA inputs, with correlation 0.399 and interval $[0.150, 0.602]$, compared with estimates from -0.141 to 0.006 in the other five models.

These results make distance difficult to interpret as a general measure of question difficulty. Since the negative correlations on ScienceQA and ChartQA often weaken after removing inputs with zero or 15 correct responses, differences involving these extreme outcomes may account for much of the full-population relationship. Restricting the range of outcomes can itself weaken a correlation, however, which prevents this comparison from isolating the contribution of the extremes. Figure 5 shows their prevalence alongside that of mixed outcomes for each model and dataset.

Since 15 responses only imprecisely estimate an input's success probability under a particular decoding policy, we interpret these correlations as relationships with observed answering frequency under our collection settings.

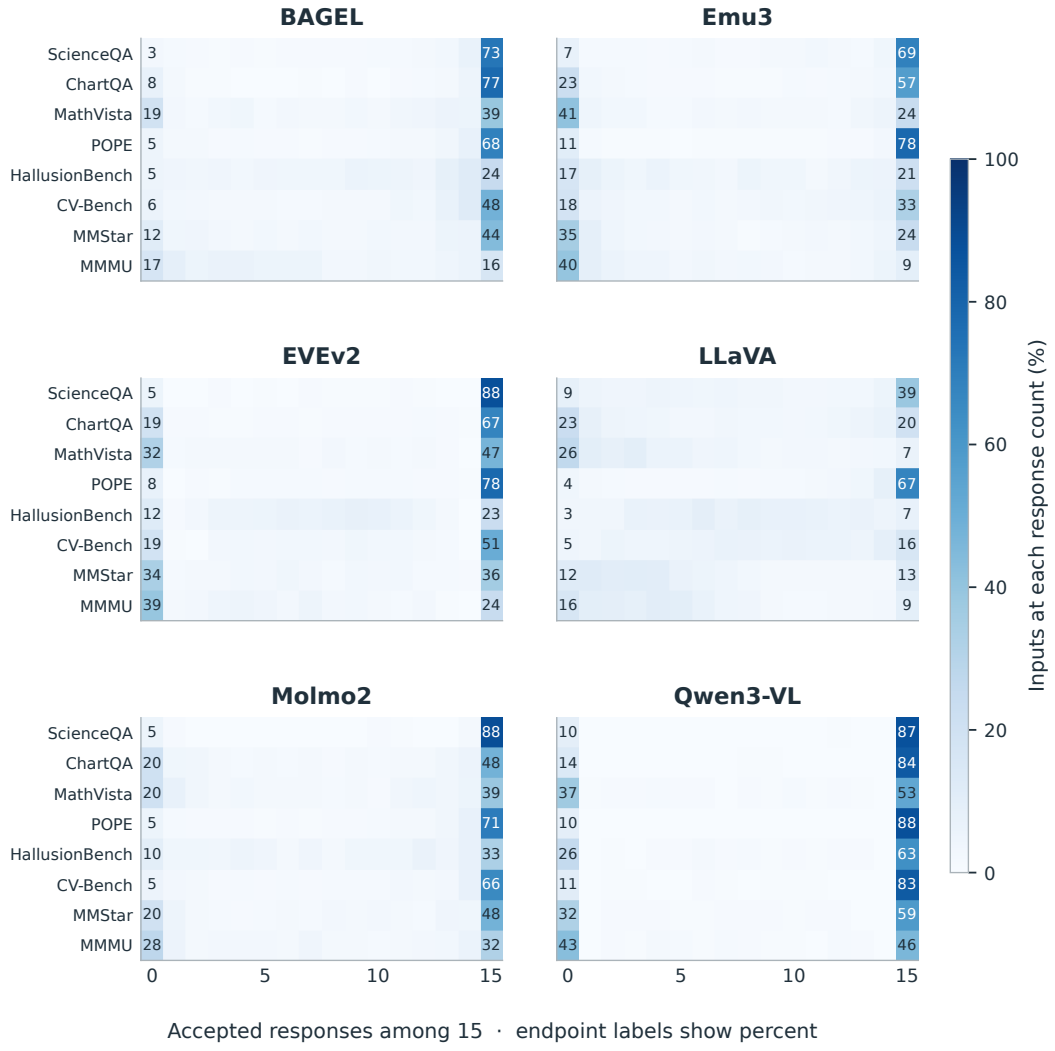

Figure 5: Qwen3-VL has fewer mixed-outcome inputs than LLaVA as a fraction of the evaluated inputs on every dataset. Each row gives the percentages with zero through 15 correct responses and sums to 100%. Inputs with 1 through 14 correct responses have mixed outcomes. All inputs have early- and late-region distances. Colours share one scale, and the endpoint cells are labelled with rounded percentages.

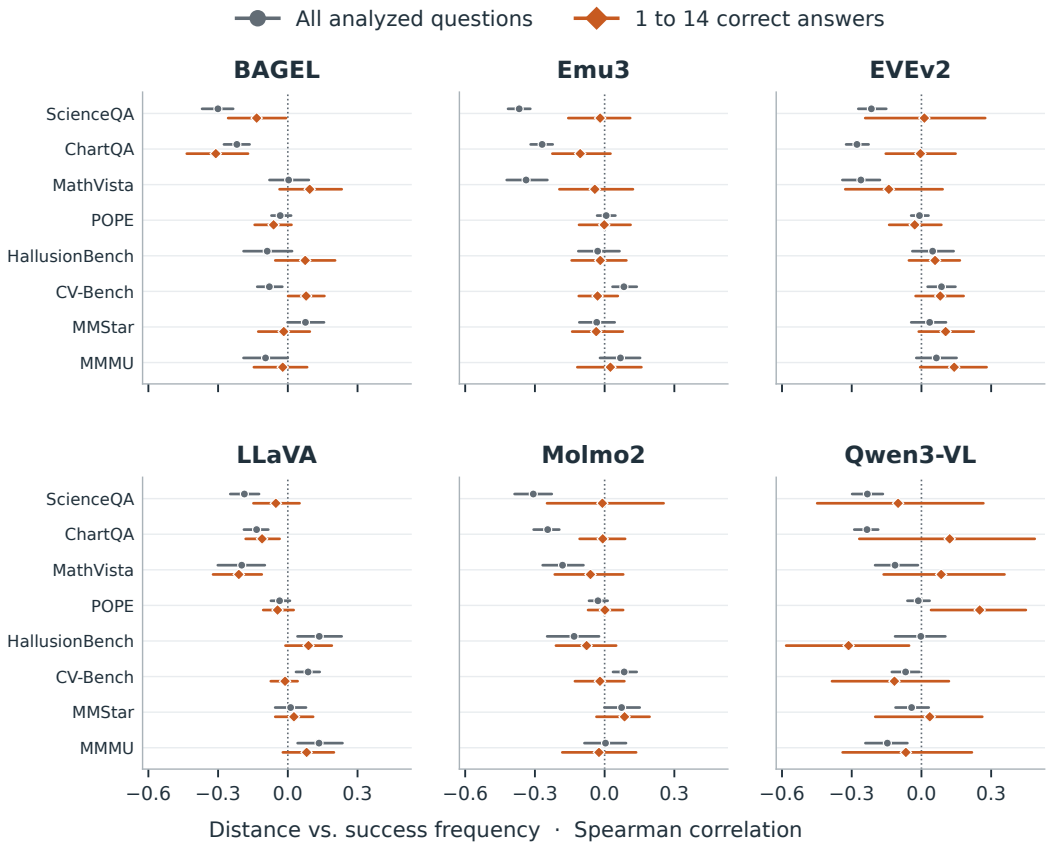

Figure 6: Greater early-region distance accompanies lower response accuracy on ScienceQA and ChartQA in every model. Points show Spearman correlations between first-quarter distance and success frequency over 15 responses for all 48 model–dataset pairs. Grey circles include all inputs with early- and late-region distances, while orange diamonds include only those with 1 through 14 correct responses. Bars give pointwise 95% intervals from 500 image-group bootstrap resamples.

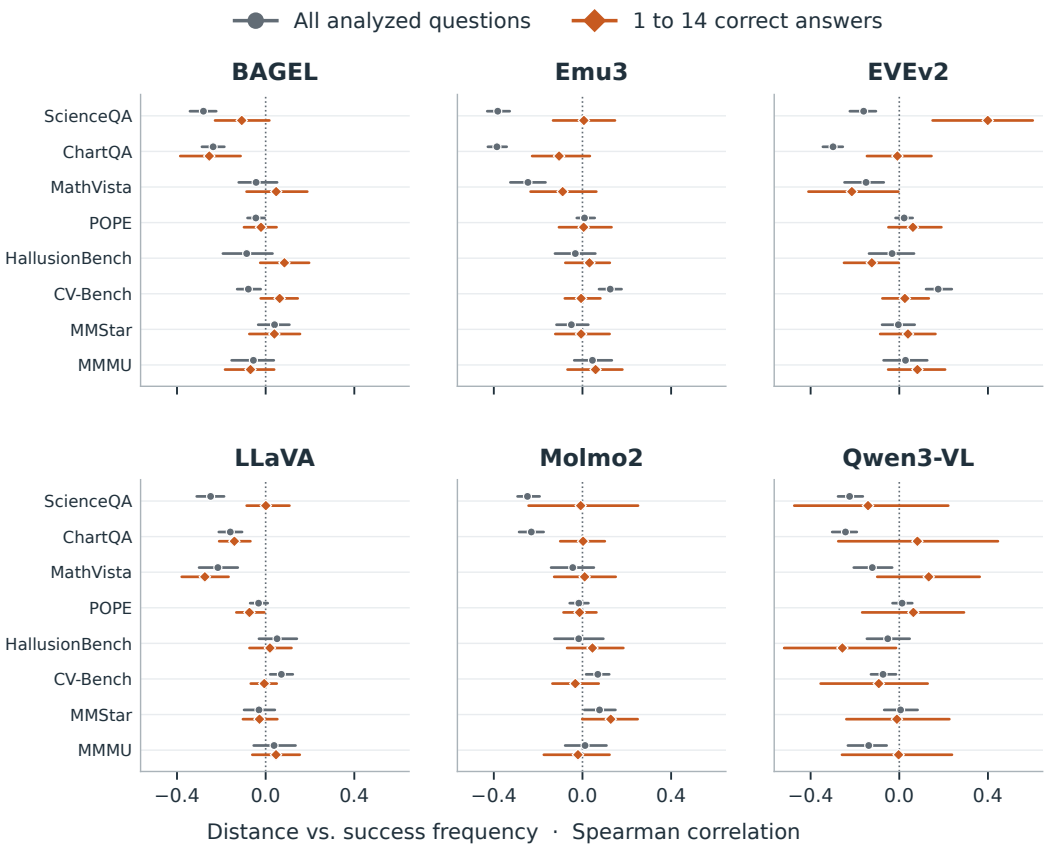

Figure 7: The late-region correlations on ScienceQA and ChartQA usually weaken among mixed-outcome inputs, with EVEv2's ScienceQA estimate reversing sign. Points show Spearman correlations between last-quarter distance and success frequency over 15 responses for all 48 model–dataset pairs. Grey circles use all inputs with early- and late-region distances. Orange diamonds restrict the comparison to those with 1 through 14 correct responses, changing both the population and the outcome range. Bars give pointwise 95% intervals from 500 image-group bootstrap resamples.

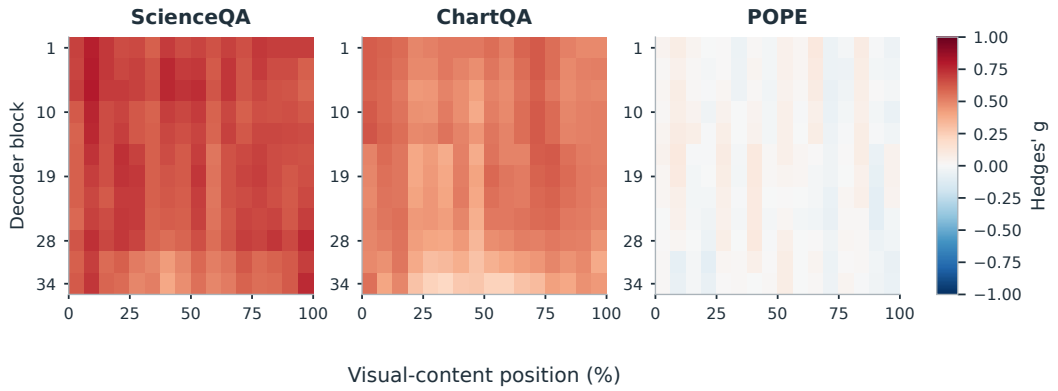

Figure 8: Qwen3-VL's task contrast extends across decoder depth and visual position. ScienceQA and ChartQA have positive separation throughout the displayed layer-position combinations, while POPE remains near zero. Colour gives Hedges' g for the inputs with a measurement at each cell. Decoder blocks are numbered from zero, with shallower layers at the top.

B.2 Layer-position maps across architectures

Averaging distances across decoder layers could conceal a signal concentrated at only a few depths. We therefore calculate Hedges' g separately at each sampled layer and visual position, showing Qwen3-VL in Figure 8 and all six models in Figure 9.

Each cell uses inputs from the main analysis with a distance at that layer and position, so exclusions can change the contributing population between cells. Colours represent Hedges' g point estimates, without a separate significance test at every cell.

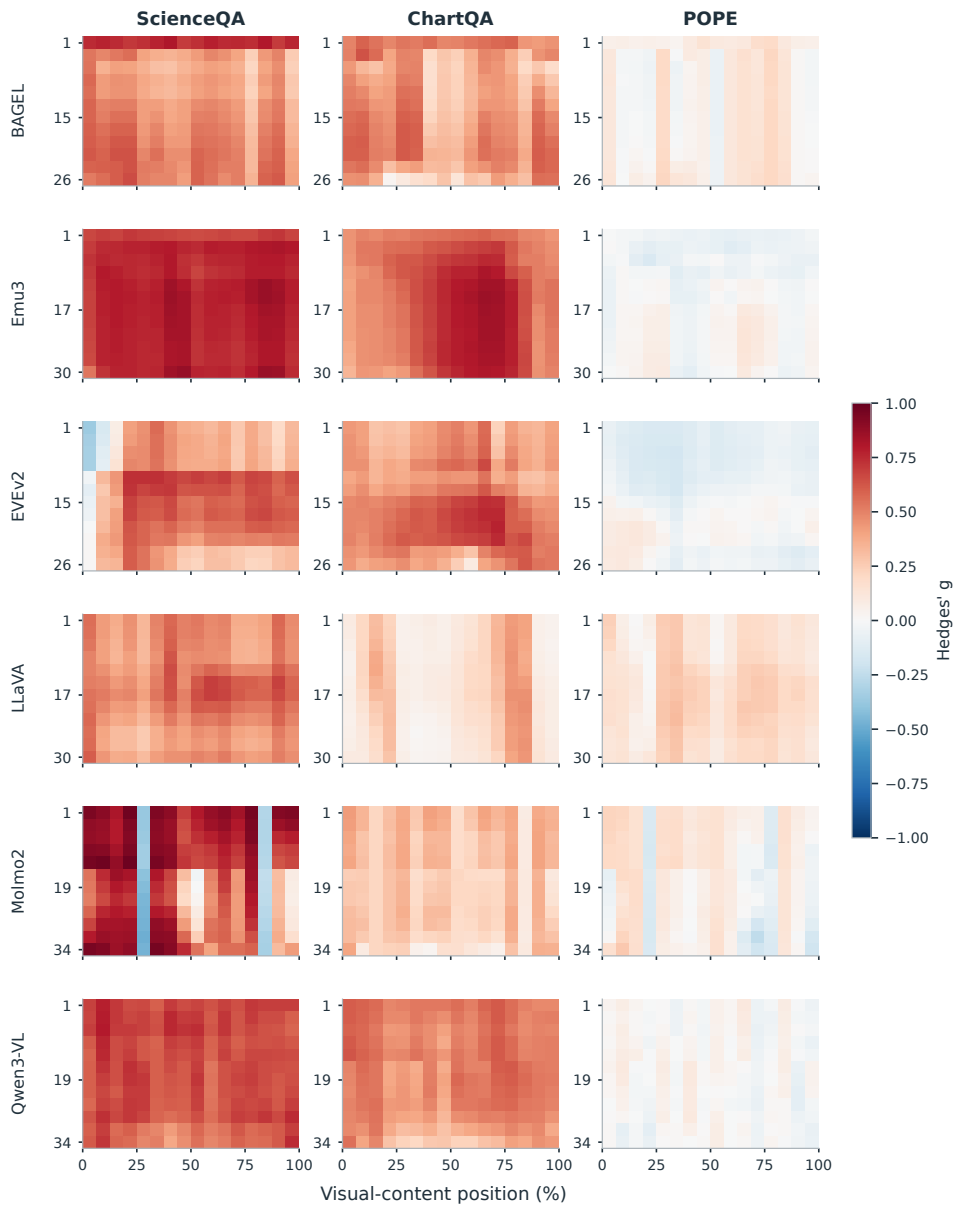

Figure 9: Separation extends across depth, although some models have local reversals. ScienceQA's estimates remain positive throughout the BAGEL, Emu3, LLaVA, and Qwen3-VL panels, while EVEv2 and Molmo2 have some negative cells. ChartQA remains positive in every panel. Rows distinguish models and columns distinguish tasks. Colour gives Hedges' g on a shared scale, with zero-based decoder blocks ordered from shallow to deep. EVEv2's contributing inputs vary with duplicate-state exclusions.

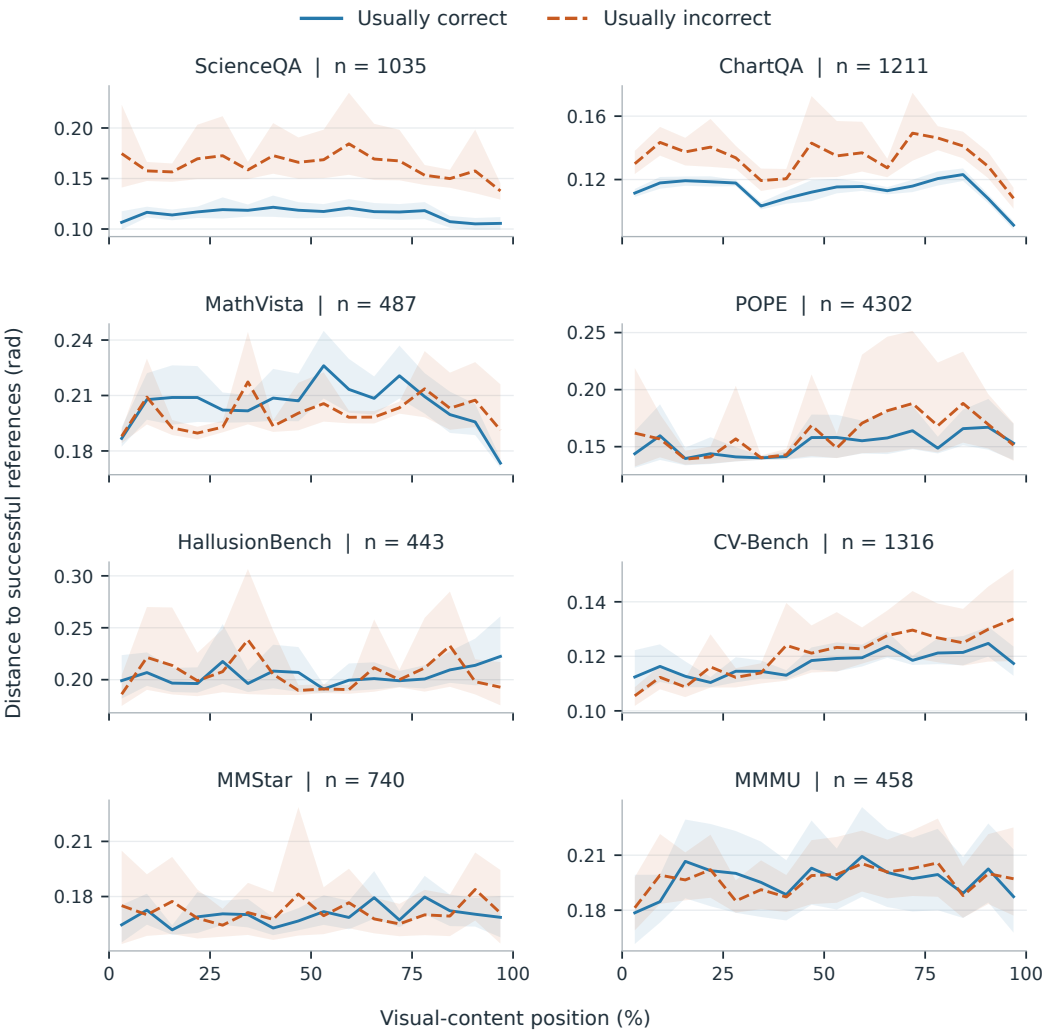

Figure 10: BAGEL's usually incorrect inputs have greater mean distance at every visual bin on ChartQA and ScienceQA. Other datasets show changes in the ordering of group means. Blue solid and orange dashed curves give mean angular kNN distances for usually correct and usually incorrect inputs, respectively, after averaging across available sampled layers. Shading gives pointwise 95% image-group intervals.

B.3 Outcome-group ordering can change across visual positions

To show the distances underlying Hedges' g, we plot each correctness group's mean angular distance across visual positions. At each bin, we first average each input's available distances across sampled decoder layers, then average within its correctness group.

The plotted points mark the centres of the 16 visual-content bins, and each panel's title gives the number of contributing inputs or its range when exclusions change that number across positions. Each panel has its own vertical scale to keep changes in distance visible while including all plotted means and confidence intervals.

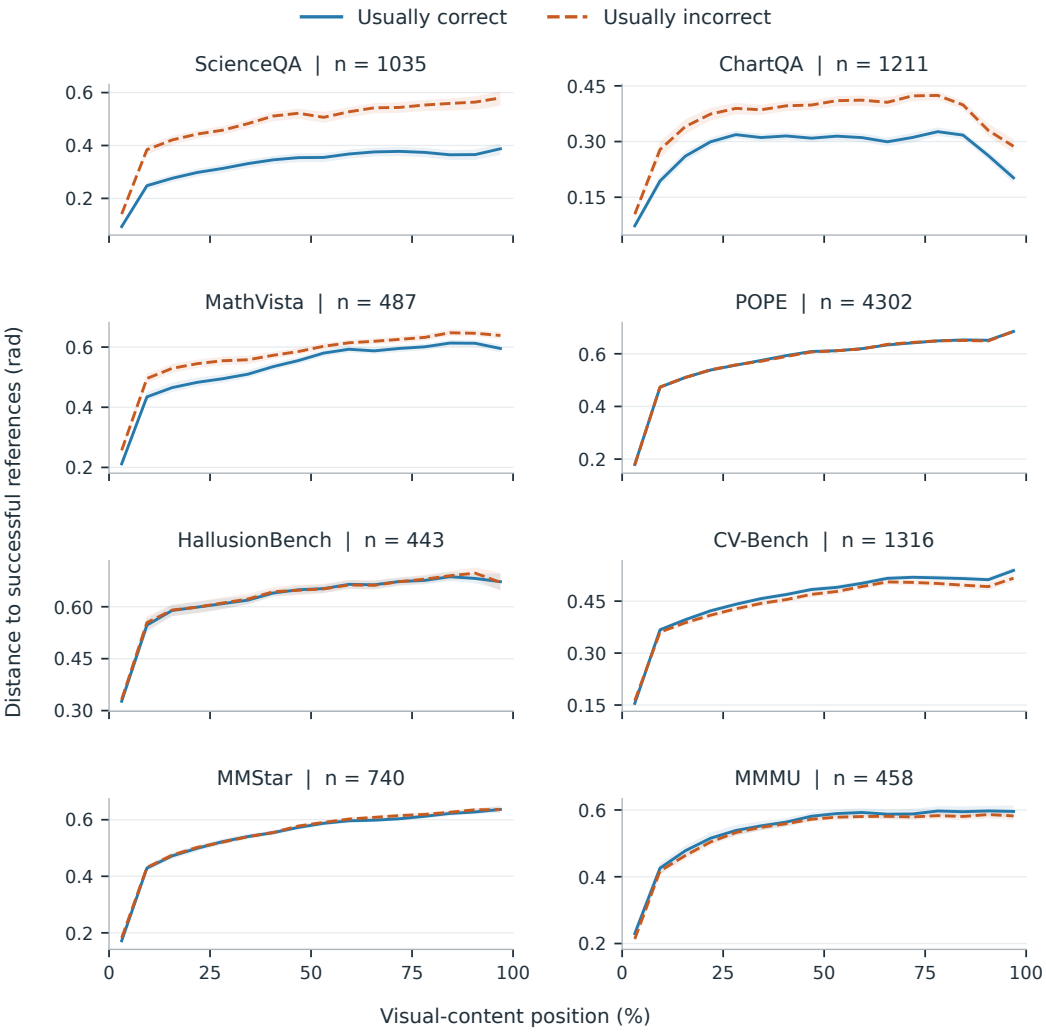

Figure 11: Emu3's usually incorrect inputs have greater mean distance at every visual bin on ChartQA, MathVista, and ScienceQA. On CV-Bench, they instead have lower mean distance after the first bin. Blue solid curves show usually correct inputs and orange dashed curves show usually incorrect inputs. For each input, distances are averaged across available sampled layers before computing the group mean. Shading gives pointwise 95% image-group intervals.

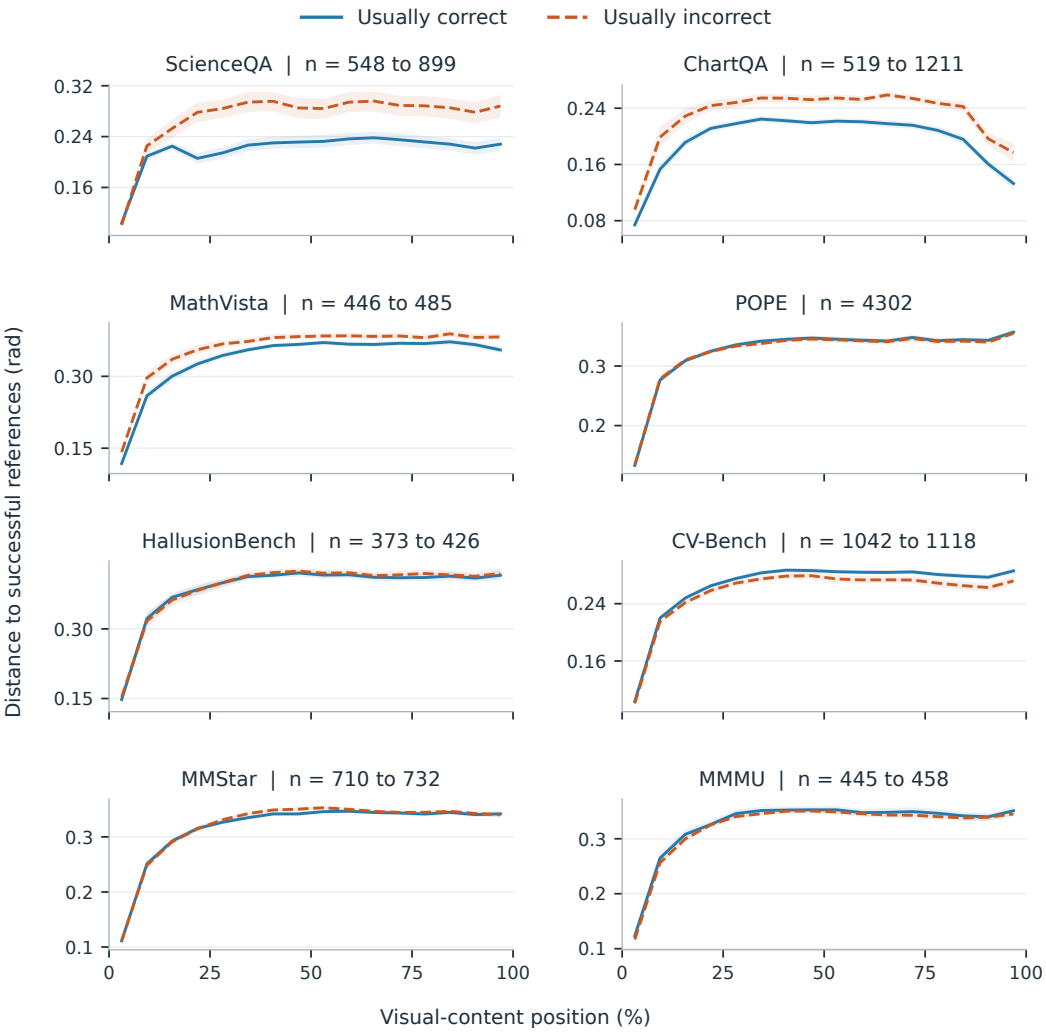

Figure 12: EVEv2's usually incorrect inputs have greater mean distance at every visual bin on ChartQA and MathVista. On seven of the eight datasets, exclusions of duplicate states change which inputs contribute at each position. For each input, we average distances across available sampled layers before computing the group mean. Solid blue curves show usually correct inputs and dashed orange curves show usually incorrect inputs. Shading gives pointwise 95% image-group intervals. Appendix B.5 examines how the exclusions affect the ScienceQA comparison.

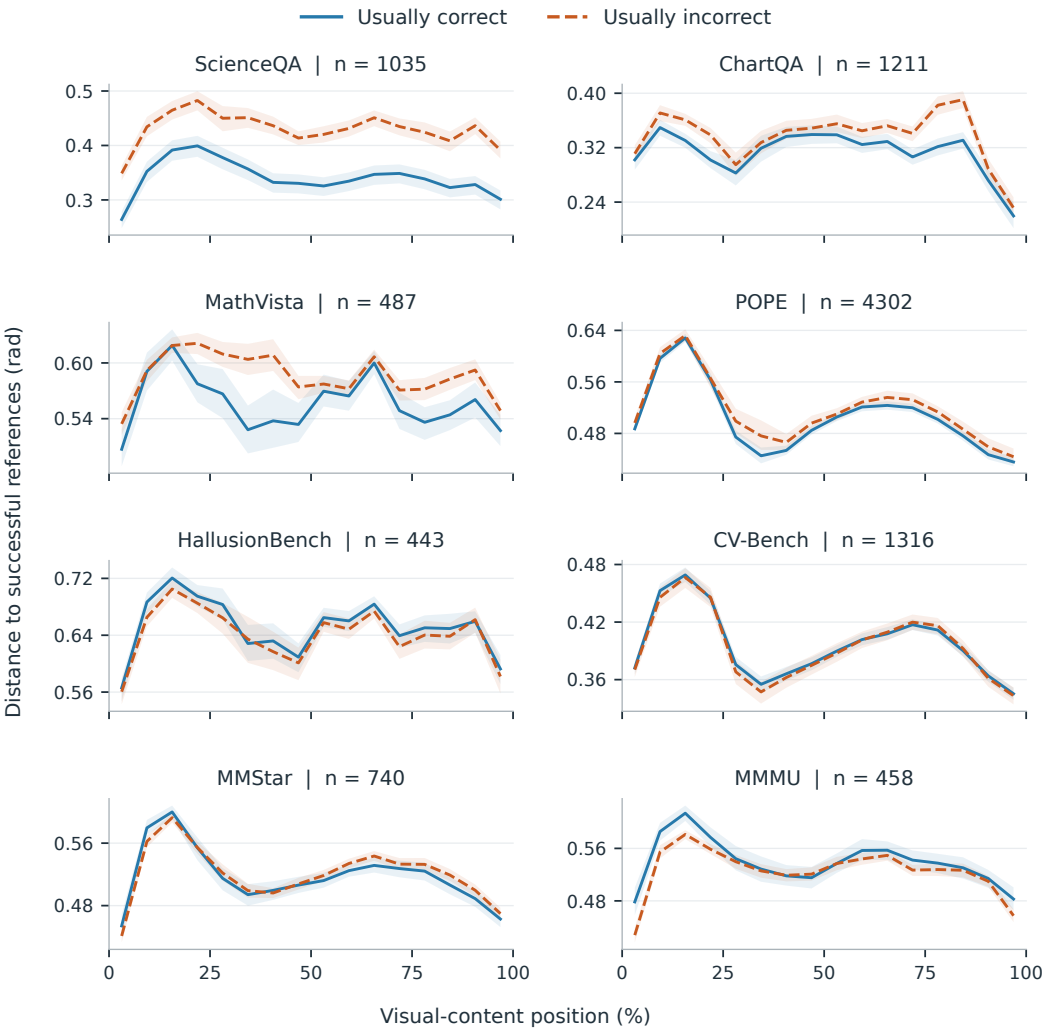

Figure 13: LLaVA's usually incorrect inputs have greater mean distance at every visual bin on ChartQA and ScienceQA. On MMMU, the ordering is reversed at most positions. For each input, we average distances across available sampled layers before computing the group mean. Solid blue curves show usually correct inputs and dashed orange curves show usually incorrect inputs. Shading gives pointwise 95% image-group intervals.

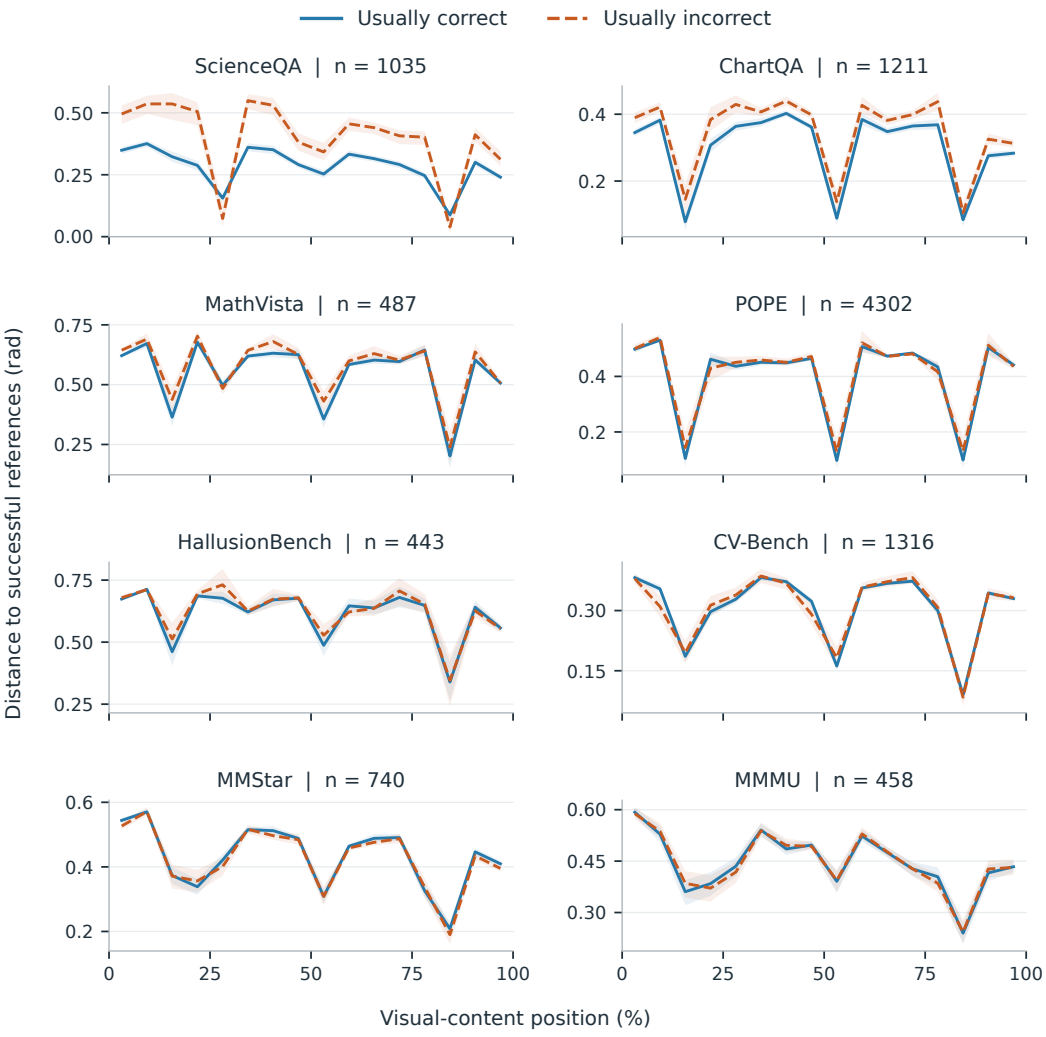

Figure 14: Molmo2's usually incorrect inputs have greater mean distance at every ChartQA bin and at 14 of the 16 ScienceQA bins. Distances are averaged across available sampled layers. Blue solid curves show usually correct inputs and orange dashed curves show usually incorrect inputs, with pointwise 95% image-group intervals in the shaded bands.

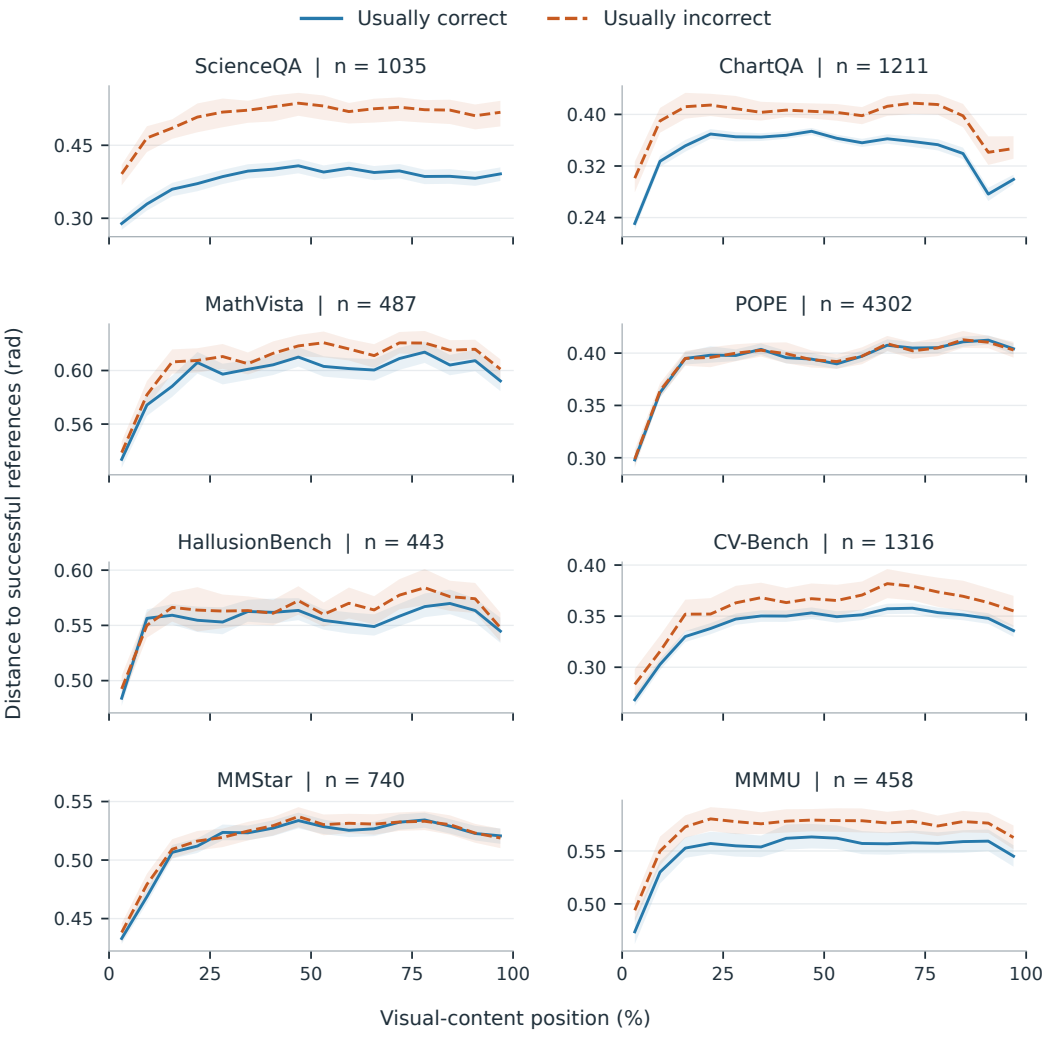

Figure 15: Qwen3-VL's usually incorrect inputs have greater mean distance at every bin on five datasets, including ChartQA and ScienceQA. All eight datasets are shown. Curves average distances across available sampled layers, with usually correct inputs in solid blue and usually incorrect inputs in dashed orange. Shading gives pointwise 95% image-group intervals.

B.4 LATE-REGION SEPARATION DIFFERS ACROSS TASK DISTRIBUTIONS

To examine variation hidden by the group means, we plot cumulative distance distributions for usually correct and usually incorrect inputs in the late visual region. Each input contributes its average distance over the last four visual bins and available sampled layers.

At a given horizontal coordinate, a curve gives the fraction of its group whose distance is at or below that value, so a rightward shift of the usually incorrect curve indicates greater distances for that group. Comparing the shapes of the curves also reveals whether their difference is spread across the distribution or concentrated in part of it. Alongside each pair, Hedges' g summarizes the difference in mean distance relative to within-group variation.

Because distance ranges differ across models and datasets, we choose the horizontal limits separately for each panel, marking any clipped tail with right-edge arrows and percentages. All inputs remain in the calculations of cumulative fractions and Hedges' g, including those beyond the displayed range. Counts and tail percentages list usually correct inputs first and usually incorrect inputs second.

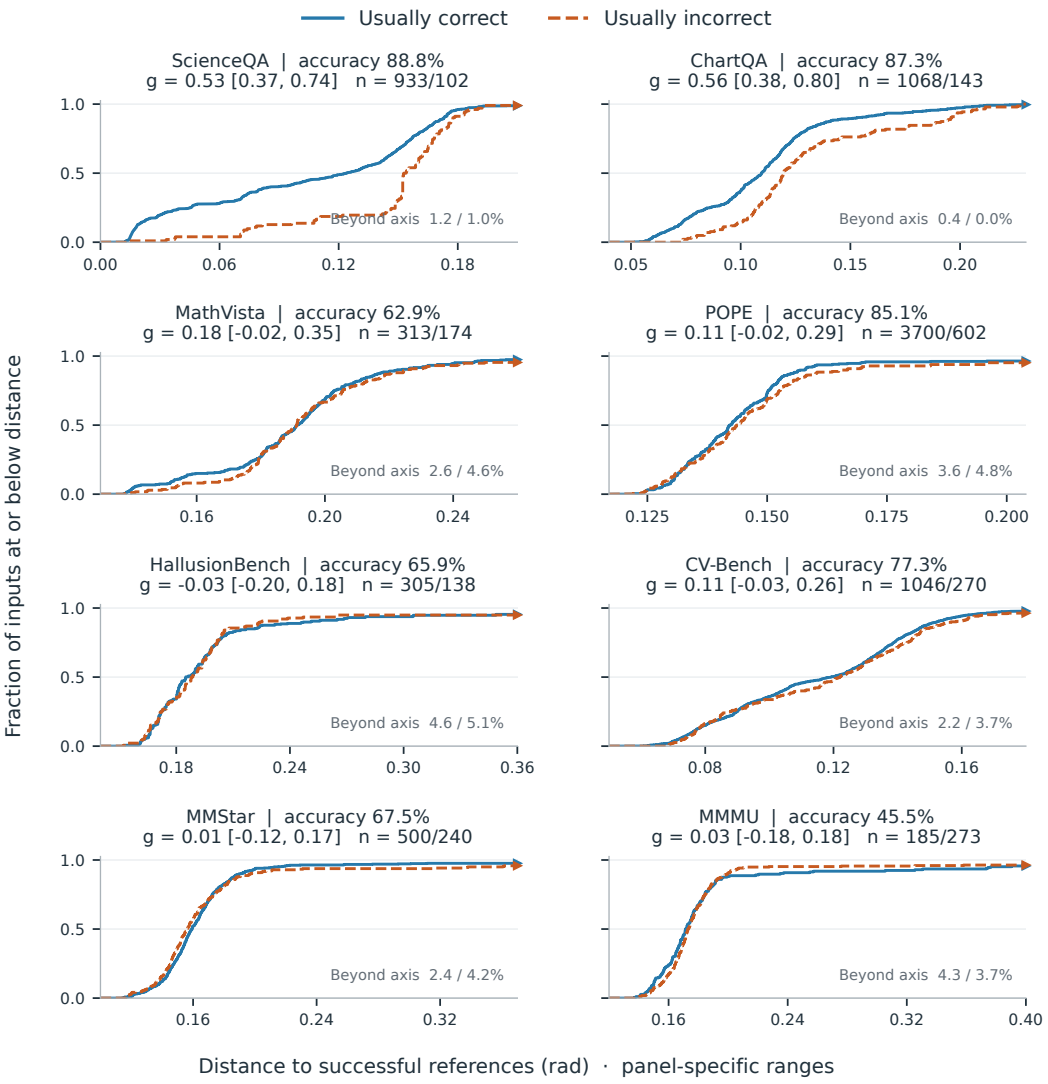

Figure 16: BAGEL's strongest late-region separation occurs on ChartQA and ScienceQA. Cumulative distributions show usually correct inputs in solid blue and usually incorrect inputs in dashed orange. Titles report mean response accuracy, Hedges' g with its pointwise 95% image-group interval, and input counts in correct/incorrect order. Tail percentages follow the same order.

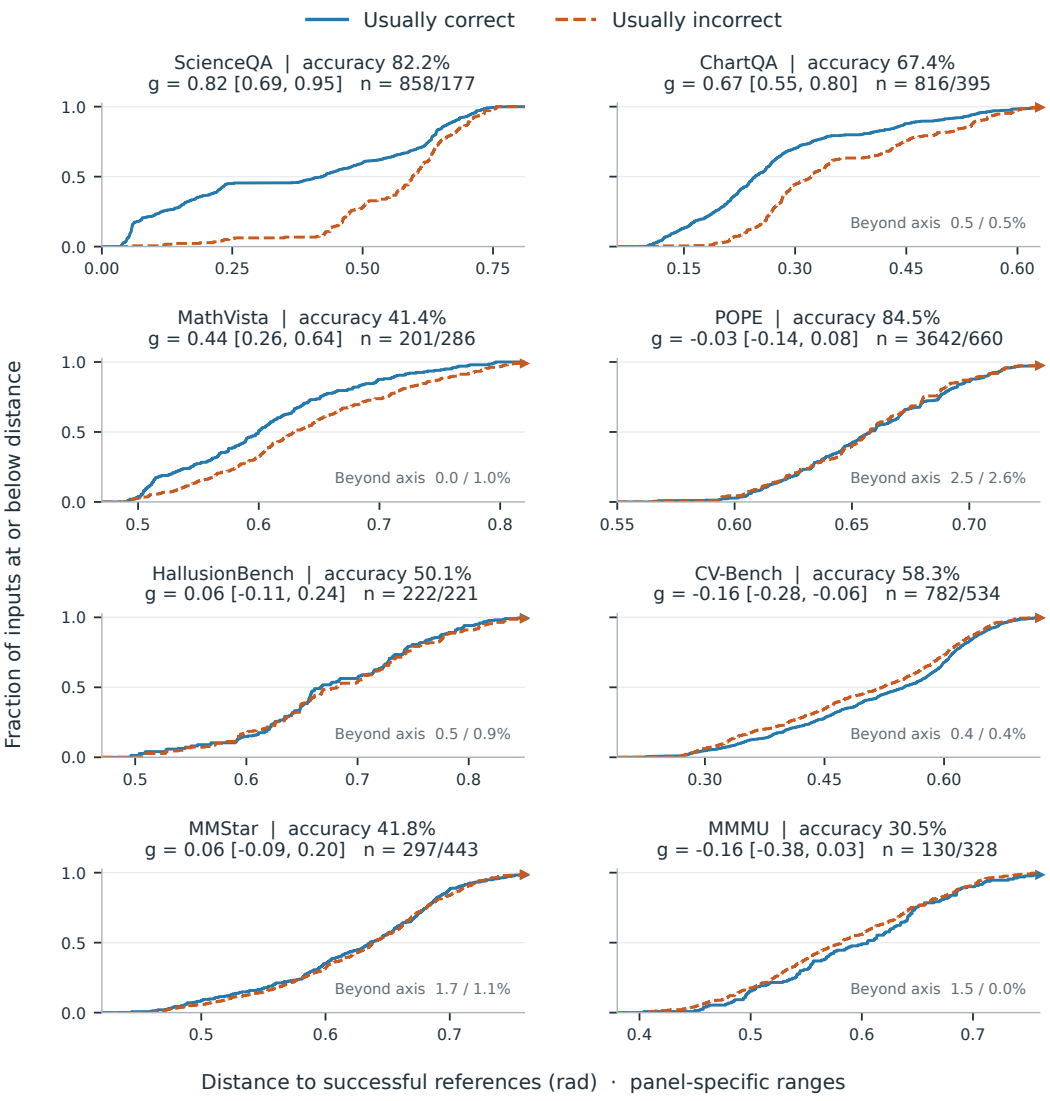

Figure 17: Emu3 separates most strongly on ScienceQA and ChartQA, while CV-Bench has a negative late-region estimate. Cumulative distributions compare usually correct inputs in solid blue with usually incorrect inputs in dashed orange. Titles give mean response accuracy, Hedges' g and its pointwise 95% image-group interval, and correct/incorrect counts. Tail percentages use the same ordering.

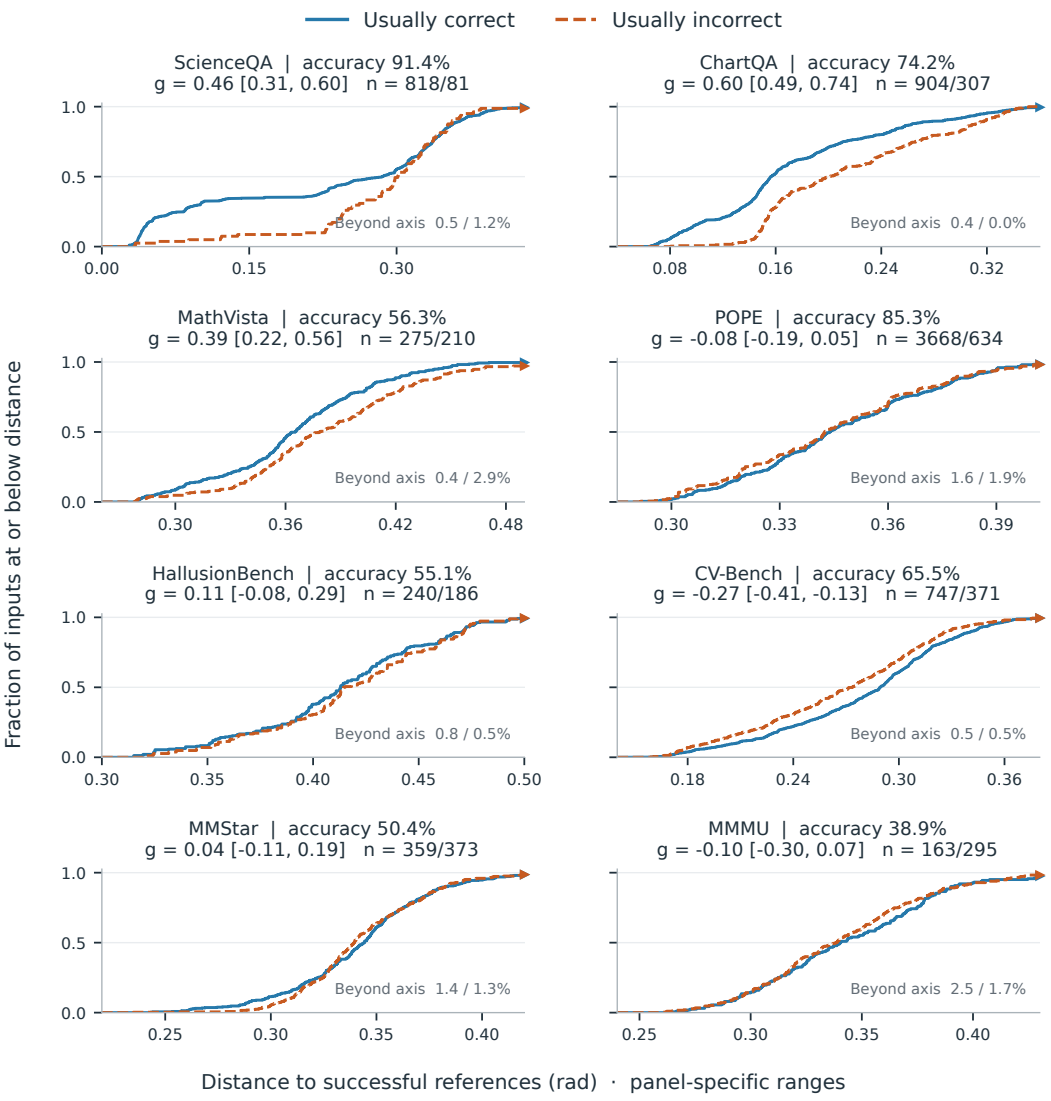

Figure 18: EVEv2 has positive late-region separation on ChartQA, ScienceQA, and MathVista, and negative separation on CV-Bench, when using each input's available measurements. Blue solid and orange dashed curves show usually correct and usually incorrect inputs. Titles report mean response accuracy, Hedges' g with its pointwise 95% image-group interval, and correct/incorrect counts. Tail percentages follow the same order.

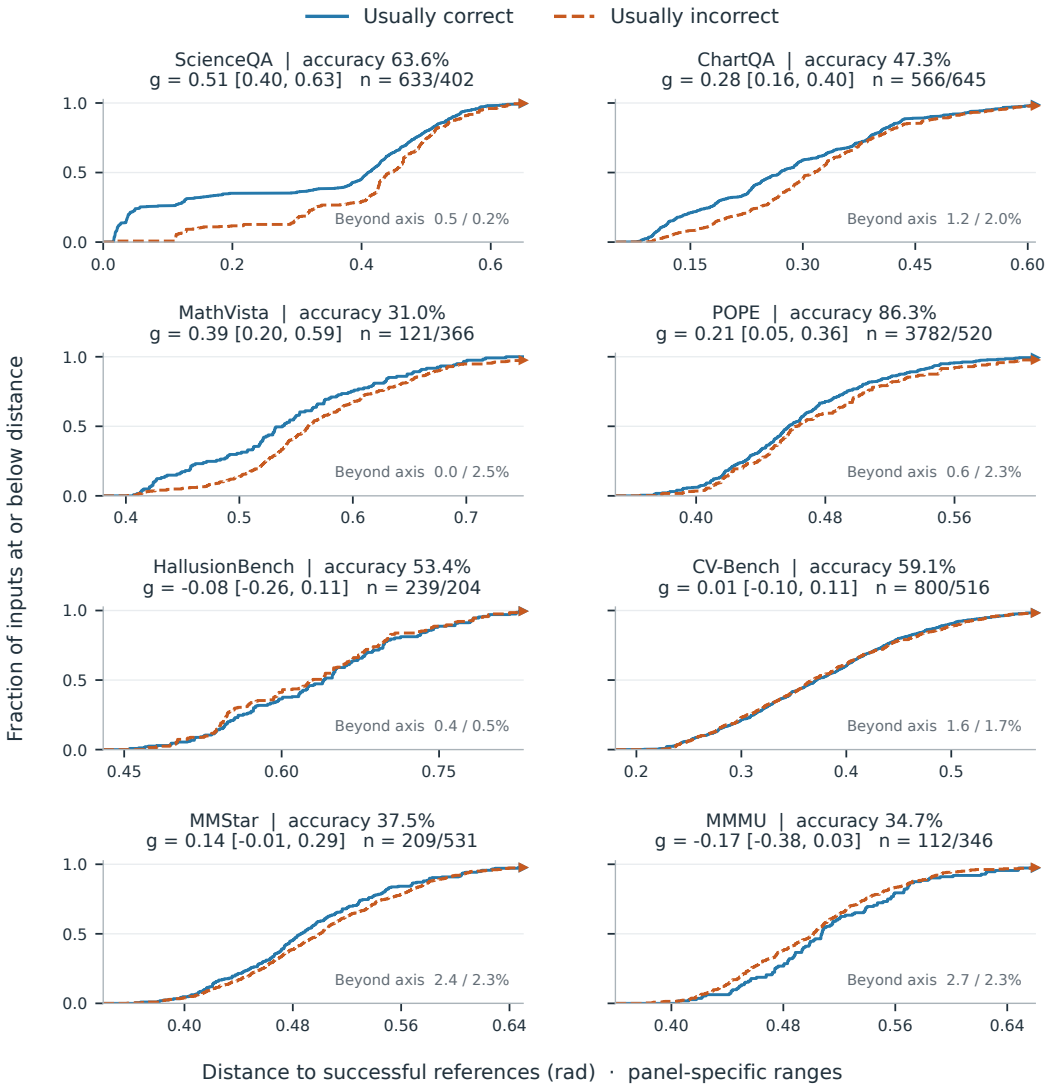

Figure 19: LLaVA separates most strongly on ScienceQA, followed by MathVista and ChartQA. Cumulative distributions show usually correct inputs in solid blue and usually incorrect inputs in dashed orange. Titles give mean response accuracy, Hedges' g and its pointwise 95% image-group interval, and input counts. Counts and tail percentages list correct inputs first.

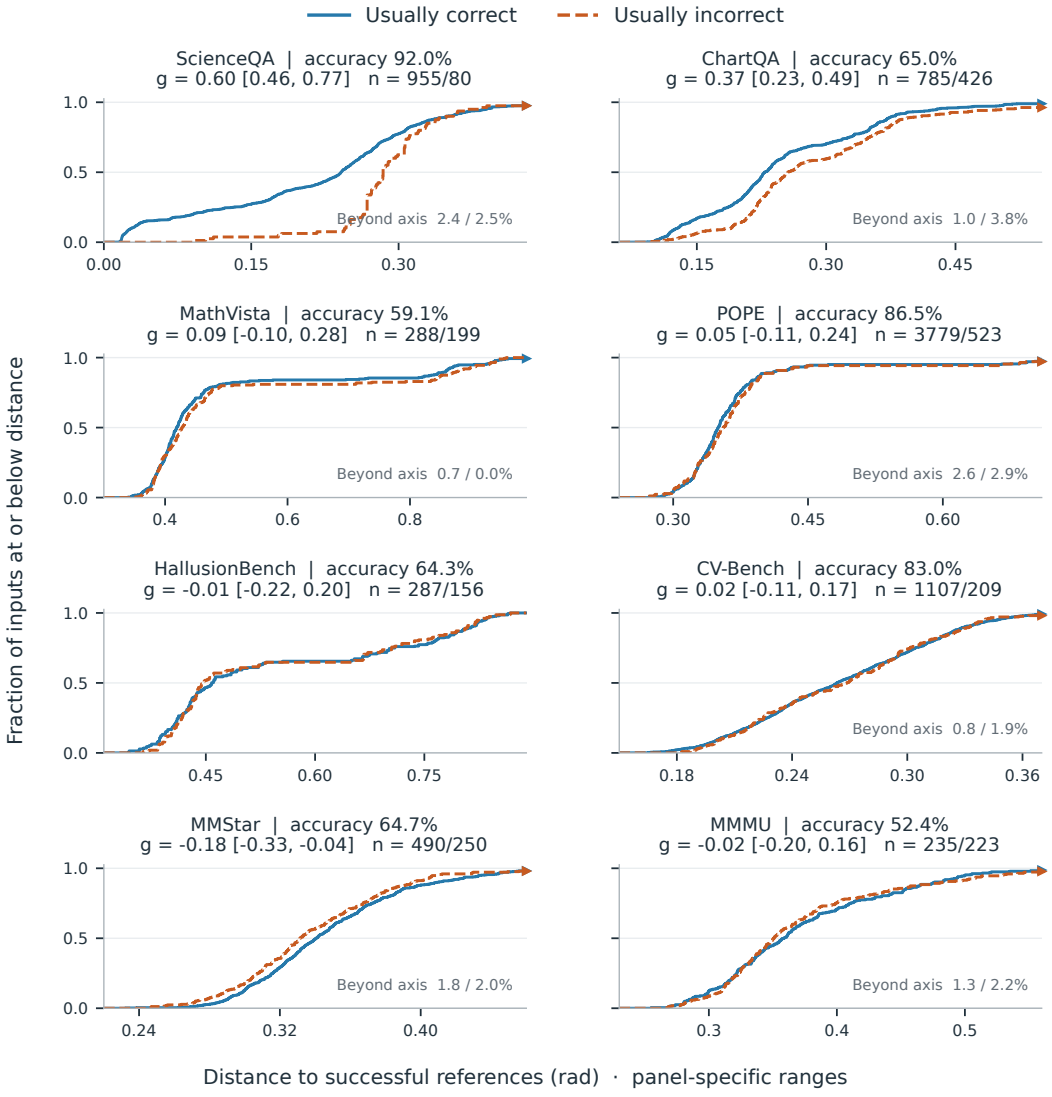

Figure 20: Molmo2 has positive late-region separation on ScienceQA and ChartQA and negative separation on MMStar. Solid blue curves show usually correct inputs and dashed orange curves show usually incorrect inputs. Titles report mean response accuracy, Hedges' g with its pointwise 95% image-group interval, and correct/incorrect counts. Tail percentages use the same ordering.

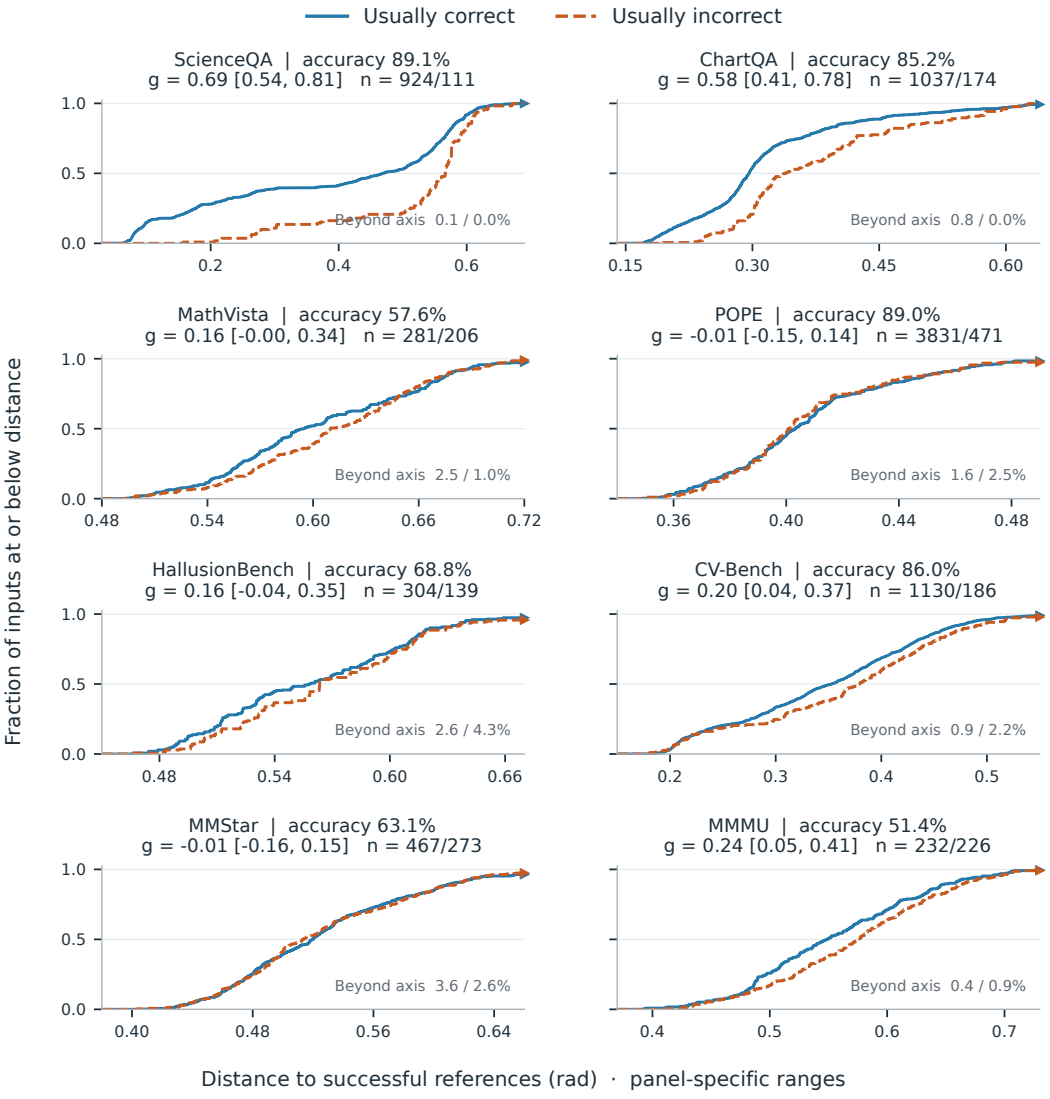

Figure 21: Qwen3-VL separates most strongly on ScienceQA and ChartQA, while MMStar and POPE have estimates near zero. Cumulative distributions show usually correct inputs in solid blue and usually incorrect inputs in dashed orange. Titles give mean response accuracy, Hedges' g and its pointwise 95% image-group interval, and correct/incorrect counts. Tail percentages follow the same order.

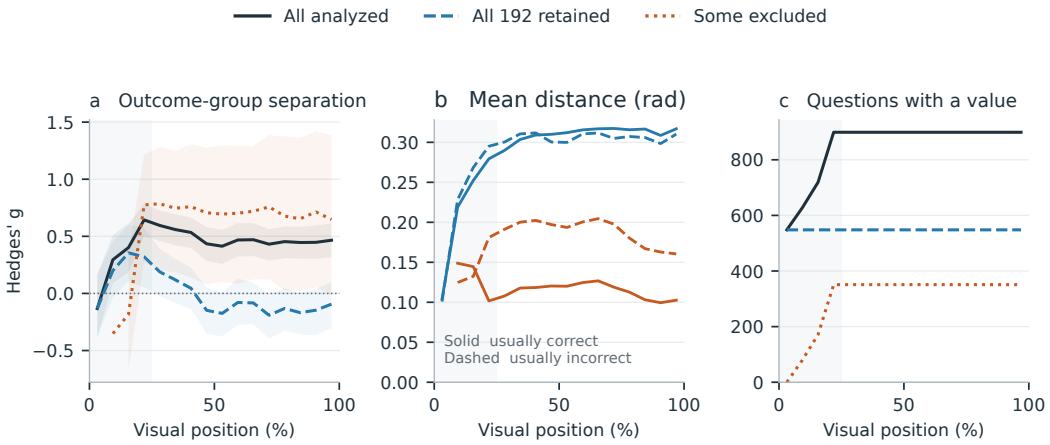

Figure 22: EVEv2's ScienceQA separation changes when we restrict which questions enter the comparison. Black curves use the 899-question analysed set, blue curves the 548 questions retaining all 192 measurements, and orange curves the remaining 351. Each bin uses the available questions from its set. **(a)** Hedges' g with pointwise 95% image-group intervals. **(b)** Mean distances, with solid lines for usually correct inputs and dashed lines for usually incorrect inputs. **(c)** Contributing question counts. The subset with exclusions has no contributing questions in the first bin and only two usually incorrect questions in the second. We report an interval only when at least 475 of the 500 bootstrap resamples give a defined effect size. At the second bin, only 433 resamples do so, and the interval is therefore omitted.

B.5 EVEv2 estimates depend on which measurements survive exclusion

As described in Appendix A, we exclude measurements whose evaluation vectors also occur in the reference partition at the same layer and position. These exclusions can change which questions contribute along the visual sequence, so we examine their effect on EVEv2's ScienceQA separation.

The 1,035 held-out questions each have 192 measurements across 12 decoder depths and 16 visual-position bins, giving 198,720 measurements, of which 19,416 are excluded by this rule. All raw states and eligible distances are present, so these exclusions reflect the analysis procedure rather than incomplete collection.

After these exclusions, 899 questions retain a distance in both the first and last quarters and enter the main analysis, while the other 136 lack a distance in one or both regions. Among the 899 analysed questions, 548 retain all 192 measurements and 351 have at least one excluded measurement. Figure 22 compares this full analysed population with each of its two subsets at every visual bin. For the early- and late-region estimates below, we compare the full population with the 548 questions retaining every measurement, using the same distance-averaging rule for both sets.

Under this restriction, early-region Hedges' g falls from 0.611 to 0.257, while the late-region estimate changes from 0.457 to -0.139, with a 95% interval of $[-0.343, 0.038]$ that includes zero.

The balance of outcomes also changes, since 69 of the 548 questions retaining every measurement are usually incorrect, compared with only 12 of the 351 with some exclusions. EVEv2's estimated separation is therefore sensitive to the questions selected by the measurement requirements.

C MEASUREMENT CHECKS AND ERROR PREDICTION

C.1 LOCAL ANGULAR MEASURES PRESERVE THE TASK ORDERING MOST CONSISTENTLY

In the main analysis, ScienceQA and ChartQA show greater separation than POPE when distance is measured to nearby successful states. To examine whether this ordering depends on the distance definition, we compare angular kNN with whitened angular kNN, centroid angle, and diagonal Mahalanobis displacement, using the same inputs and resampled reference clouds for all four measures.

These choices emphasize different geometric properties (Park et al., 2024). Centroid angle replaces local neighbours with the references' overall mean direction, while whitened angular kNN and diagonal Mahalanobis displacement adjust for variation along each coordinate.

For all four measures, we begin by normalizing the query and reference states to unit length, denoting the normalized query by u and the mean of the normalized references by μ. Centroid angle then compares u with the direction of this mean, $\mu/\|\mu\|_2$, leaving the measurement undefined if the mean is zero and therefore has no direction.

To adjust for reference variation, we subtract the reference mean from each coordinate and divide by its regularized reference standard deviation, giving greater weight to coordinates with less variation. Denoting the population variance of the references in coordinate a by σ_a^2, we use the transformation

$$w_a(u) = \frac{u_a - \mu_a}{\sqrt{\sigma_a^2 + \lambda}}, \qquad \lambda = \max\left(10^{-3}\frac{1}{p}\sum_a \sigma_a^2, \epsilon_{32}\right),$$

where p is the hidden-state width and ϵ_{32} is float32 machine epsilon. For each reference resample, we calculate its mean and variances separately and derive the ridge term λ from those variances, keeping the scale positive even when a coordinate has zero variance.

Whitened angular kNN normalizes the transformed query and reference vectors to unit length and averages the angles to the 15 nearest references. Diagonal Mahalanobis displacement instead measures distance from the reference mean through the square root of the mean squared transformed coordinate, $\sqrt{p^{-1}\sum_a w_a(u)^2}$.

For each of the six models, we compare Hedges' g on ScienceQA and on ChartQA with the corresponding estimate on POPE, giving 12 comparisons of the task ordering. Whitened angular kNN preserves greater separation on ScienceQA and ChartQA in all 12 comparisons, while centroid angle and diagonal Mahalanobis displacement each preserve 11.

Both exceptions concern ScienceQA in Molmo2, where centroid-angle Hedges' g is -0.231, compared with 0.133 on POPE. The corresponding diagonal Mahalanobis estimates are -0.062 and 0.104, with the full late-region results for all models given in Tables 5–7. These comparisons use point estimates without confidence intervals for the alternative measures.

Table 5: Whitened angular kNN preserves greater separation on ScienceQA and ChartQA than on POPE in all six models. Entries give late-region Hedges' g on the matched evaluation inputs.

Dataset	BAGEL	Emu3	EVEv2	LLaVA	Molmo2	Qwen
ScienceQA	0.512	0.843	0.535	0.476	0.788	0.664
ChartQA	0.507	0.734	0.647	0.386	0.411	0.523
MathVista	0.428	0.574	0.525	0.388	0.105	0.495
POPE	0.173	0.019	0.049	0.111	-0.039	0.152
HallusionBench	0.041	0.018	-0.025	-0.100	0.008	0.234
CV-Bench	0.011	-0.132	-0.209	-0.028	0.014	0.161
MMStar	-0.088	0.133	0.094	-0.220	-0.039	-0.088
MMMU	0.058	-0.145	-0.181	-0.529	-0.071	0.010

Table 6: Centroid angle preserves 11 of the 12 ScienceQA–POPE and ChartQA–POPE orderings. The exception is Molmo2 on ScienceQA. Entries give late-region Hedges' g on the same matched inputs.

Dataset	BAGEL	Emu3	EVEv2	LLaVA	Molmo2	Qwen
ScienceQA	0.355	0.638	0.210	0.578	-0.231	0.589
ChartQA	0.489	0.596	0.504	0.250	0.185	0.516
MathVista	0.132	0.324	0.294	0.319	0.091	-0.010
POPE	0.106	-0.051	-0.115	0.188	0.133	-0.044
HallusionBench	-0.030	0.103	0.118	-0.054	0.001	0.162
CV-Bench	0.118	-0.158	-0.237	0.031	-0.023	0.163
MMStar	0.023	-0.002	-0.072	0.267	-0.122	-0.007
MMMU	0.023	-0.108	-0.040	0.094	0.069	0.228

Table 7: Diagonal Mahalanobis displacement also preserves 11 of the 12 task orderings, with Molmo2 on ScienceQA as the exception. Entries give late-region Hedges' g on the same matched inputs.

Dataset	BAGEL	Emu3	EVEv2	LLaVA	Molmo2	Qwen
ScienceQA	0.545	0.706	0.300	0.567	-0.062	0.697
ChartQA	0.460	0.580	0.457	0.237	0.237	0.508
MathVista	0.130	0.402	0.282	0.333	0.092	0.053
POPE	0.093	-0.048	-0.083	0.172	0.104	-0.053
HallusionBench	-0.018	0.121	0.136	-0.074	-0.004	0.200
CV-Bench	0.132	-0.158	-0.183	0.025	0.002	0.118
MMStar	-0.014	0.030	-0.159	0.234	-0.172	-0.028
MMMU	0.030	-0.093	-0.082	0.012	0.048	0.217

C.2 Angular distance is usually more predictive than magnitude

To test whether direction and magnitude carry distinct predictive information, we fit three logistic predictors using angular distance alone, magnitude alone, or both on the same inputs for each model and dataset.

We obtain the magnitude feature by averaging token states within each position bin and taking the norm of that mean. Because states pointing in different directions can partly cancel, this norm depends on both their lengths and how their directions combine. We then average these norms across sampled layers and positions in the early or late region, just as we average angular distances over that region.

To fit and evaluate the predictors, we subdivide the original evaluation population while leaving the reference clouds fixed. An image-group hash assigns 60% to training, 20% to validation, and 20% to test, keeping all questions about an image together. All three predictors use only inputs with both an angle and a magnitude measurement at every sampled layer and position in the region being compared.

Without consulting correctness labels, we exclude inputs that share either a full array of available angle and magnitude measurements or a stored state capture with an input in another subdivision. The analysis covers 47 model–dataset pairs, excluding Emu3/MMMU because it preceded the replacement of that pair's incompatible collection.

For each predictor, we standardize the features using training observations alone. We then fit ridge-regularized logistic candidates with coefficient penalties of $0.001, 0.01, 0.1, 1$, leaving the intercept unpenalized, and retain the candidate with the lowest validation Brier score. This score measures the mean squared error of predicted error probabilities against binary labels that code usually incorrect inputs as one.

We evaluate the selected predictors by AUROC, measuring how often a usually incorrect input is ranked as more error-prone than a usually correct input. Chance ranking gives 0.5, and perfect ranking gives 1. To estimate uncertainty in the difference between two predictors' AUROCs, we resample whole test image groups 500 times and evaluate both predictors on each resample with their fitted parameters fixed. Each interval applies to its individual comparison, without correction for multiple comparisons.

Angular distance alone gives a higher AUROC point estimate than magnitude alone in 34 of the 47 early-region comparisons and 33 of the 47 late-region comparisons, as shown in Figure 23. Compared with the magnitude-only predictor, the predictor using both features has an AUROC improvement whose interval lies entirely above zero in seven early-region and ten late-region comparisons. Against the angular-distance-only predictor, its improvement has an interval entirely above zero in zero early-region and three late-region comparisons.

The relative usefulness of the two features nevertheless varies across models, even on the same task. On ScienceQA, Emu3's late-region AUROC is 0.763 using angular distance, 0.419 using magnitude, and 0.768 using both. For Qwen3-VL on the same task, magnitude alone instead slightly exceeds angular distance alone, with AUROCs of 0.681 and 0.675, respectively.

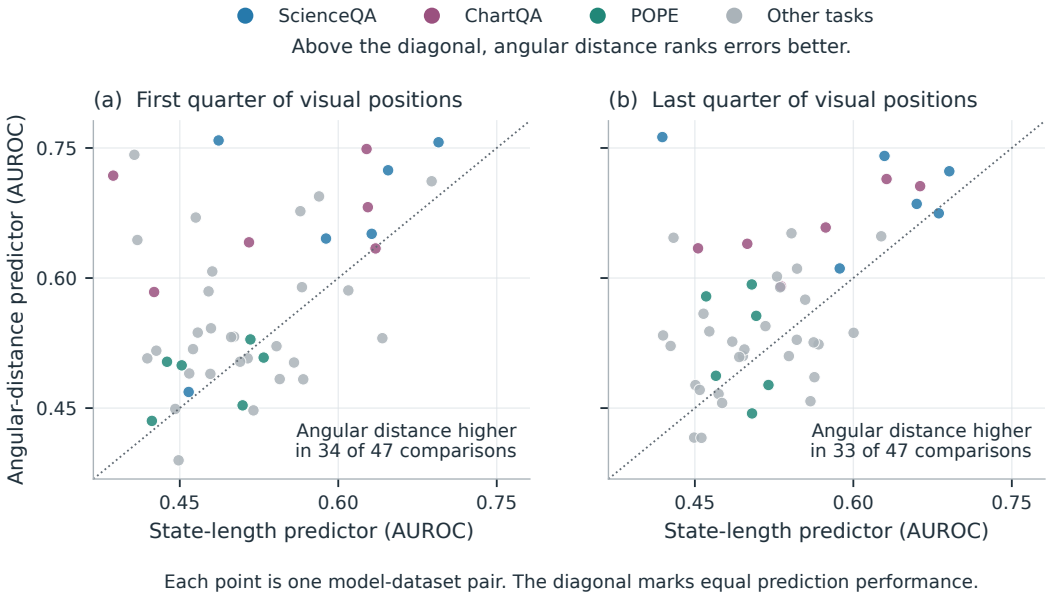

Figure 23: Angular distance predicts errors better than magnitude alone in most tested model–dataset pairs. Each dot compares two logistic predictors on the same held-out inputs. The horizontal coordinate is the AUROC of the predictor using mean state magnitude. The vertical coordinate is the AUROC of the predictor using angular distance. Points above the diagonal therefore favour angular distance, including 34 of 47 early-region and 33 of 47 late-region comparisons. These counts compare point estimates. Colours distinguish ScienceQA, ChartQA, and POPE, with the other five datasets in grey.

C.3 Prediction protocol and layer selection

In Section 5, we compare the geometry MLP's five reference-distance features with HALP's full hidden-state input on Qwen3-VL using ScienceQA and ChartQA. Both methods use the same evaluation inputs and majority-correctness labels obtained from 15 sampled responses.

We leave the reference partition unchanged and subdivide the original evaluation partition by image group into 60% training, 20% validation, and 20% test sets. A deterministic hash assigns groups to these sets, keeping all questions about an image together and giving both methods the same subdivisions.

To enter the comparison, an input must have all the features required by both predictors. We also check for identical full hidden-state vectors across subdivisions, excluding the corresponding inputs from every affected subdivision without consulting their labels. This duplicate check excludes no inputs on either task, leaving the training, validation, and test populations reported in Table 8.

For both methods, we repeat training with seeds 42, 43, and 44, then average the three separately fitted predictors' probabilities for each input before calculating AUROC.

Table 8: Training, validation, and test sets contain disjoint image groups. Error questions are usually incorrect inputs, and error groups are image groups containing at least one such question.

Task	Split	Questions	Groups	Error questions	Error groups
ScienceQA	Train	608	538	70	58
	Validation	207	189	18	16
	Test	220	199	23	22
ChartQA	Train	743	452	110	98
	Validation	243	150	32	29
	Test	225	138	32	28

Geometry predictor. The geometry MLP receives one angular distance from each of depth groups 0, 2, 5, 8, and 11 among the 12 groups described in Appendix A. All five distances come from the final visual-position bin and are standardized using the training observations alone before entering the predictor.

The MLP has three hidden layers of widths 512, 256, and 128 with ReLU activations, followed by a linear layer producing a single output logit, giving 167,425 trainable parameters. We train for 50 epochs with Adam, a learning rate of 0.001, weight decay of 0.0001, and batches of 32. Since some images have more questions than others, we give each image group equal total weight in the training loss. For each seed, we retain the checkpoint with the lowest binary cross-entropy on the validation set.

HALP predictor. For prediction directly from hidden states, we use HALP's released Qwen probe architecture, fitting a separate probe at decoder blocks 0, 8, 17, 26, and 35, counted from zero (Kogilathota et al., 2026). Each probe receives the 4,096-dimensional state at its block's image-end delimiter, converted from the stored precision to float32 while retaining the original feature scales. These states include Qwen3-VL's DeepStack additions where applicable and, at the last depth, the model's final RMS normalization.

Each probe has hidden layers of widths 512, 256, and 128, with ReLU activations, batch normalization, and dropout of 0.3 after each hidden activation, giving 2,263,809 trainable parameters. We train with Adam at a learning rate of 0.001, zero weight decay, and batches of 32, weighting each observation equally in the training loss.

For every layer and seed, we train for 50 epochs and use the final checkpoint, averaging the three seeds' predicted probabilities within each layer before calculating validation AUROC. We then select the layer with the highest validation AUROC, which is block 26 for ScienceQA and block 0 for ChartQA. Table 9 reports the results for all five layers.

Evaluation and supervision. After selecting the HALP layer, we compare the two methods on the test set using their averaged predictions. To estimate uncertainty, we form 2,000 bootstrap samples of the test image groups and record the difference in AUROC after evaluating both methods on the same groups in each sample. The paired intervals in Table 1 summarize these differences with the fitted models, selected HALP layer, and reference clouds held fixed. Only 22 ScienceQA image groups and 28 ChartQA image groups in the test sets contain usually incorrect questions, limiting the precision with which we can estimate a difference between the methods.

Beyond the classifier's training labels, the geometry predictor uses correctness labels to select successful references. Each reference resample contains 373 ScienceQA groups or 152 ChartQA groups at each sampled depth, drawn from a larger population whose labels must also be inspected. The

Table 9: The best validation layer differs between ScienceQA and ChartQA. Entries give AUROC after averaging each input's predicted probabilities over three training seeds. Asterisks mark the layers selected by validation AUROC for the main comparison. Decoder blocks are numbered from zero.

	ScienceQA		ChartQA	
Block	Validation	Test	Validation	Test
0	0.824	0.753	0.646*	0.680*
8	0.721	0.731	0.572	0.683
17	0.826	0.771	0.558	0.616
26	0.838*	0.789*	0.608	0.666
35	0.806	0.797	0.608	0.645

cloud sizes therefore undercount the labeled examples used by the procedure. Likewise, the reported classifier parameter counts exclude the cost of constructing and storing these reference clouds.